

ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA

ARCES - ADVANCED RESEARCH CENTER ON ELECTRONIC SYSTEMS  
FOR INFORMATION AND COMMUNICATION TECHNOLOGIES E. DE CASTRO

EDITH - EUROPEAN DOCTORATE ON INFORMATION TECHNOLOGY

CYCLE XXV - ING-INF/01

**EFFICIENT ULTRASONIC SIGNAL  
PROCESSING TECHNIQUES FOR AIDED  
MEDICAL DIAGNOSTICS**

Ph.D. Thesis of

**Francesca Galluzzo**

**Ph.D. Coordinator:**  
**Prof. Claudio Fiegna**

**Supervisors:**  
**Prof. Guido Masetti**  
**Ing. Nicolò Speciale**

January 2010- December 2012



---

# Contents

---

<b>Contents</b>	<b>i</b>
<b>Abstract</b>	<b>v</b>
<b>Sommario</b>	<b>vii</b>
<b>Introduction</b>	<b>ix</b>
<b>1 Technological Background</b>	<b>1</b>
1.1 US physics . . . . .	1
1.2 Ultrasound imaging systems architecture . . . . .	6
1.2.1 B-Mode imaging . . . . .	7
1.2.2 3D ultrasound imaging . . . . .	10
1.3 Ultrasound vs. other imaging modalities . . . . .	12
1.4 Computer aided diagnosis . . . . .	14
1.4.1 Ultrasound tissue characterization . . . . .	16
1.4.2 Ultrasound image segmentation . . . . .	17
1.5 GPU computing in medical imaging and CAD . . . . .	17
<b>2 UTC for Prostate Cancer Diagnosis</b>	<b>23</b>
2.1 Prostate anatomy and pathologies . . . . .	24
2.2 Prostate cancer diagnosis procedure . . . . .	26
2.3 CAD tools for PCa detection . . . . .	30
2.3.1 The binary classification problem . . . . .	30
2.3.2 UTC based CAD systems processing flow . . . . .	33
2.4 PCa CAD literature review . . . . .	43
2.5 Proposed approach for prostate biopsy protocol improvement: rtCAB . . . . .	44
2.5.1 Database collection . . . . .	46
2.5.2 rtCAB processing flow . . . . .	50
2.5.3 Parallel implementation of the rtCAB classification system . . . . .	51
2.5.4 rtCAB learning scheme . . . . .	56

2.5.5	rtCAB results . . . . .	57
2.6	Conclusions . . . . .	62
<b>3</b>	<b>Improving Prostate Biopsy Protocol with a CAD Tool based on SSL</b>	<b>63</b>
3.1	Semi-Supervised Learning . . . . .	65
3.2	Self-training and Co-training . . . . .	67
3.3	The proposed SSL algorithm: SelfCo3 . . . . .	69
3.3.1	The SelfCo3 algorithm . . . . .	70
3.3.2	Experimental Results . . . . .	71
3.4	A new Learning Procedure for computer aided PCa detection in ultra- sound images . . . . .	77
3.4.1	SelfCo3Core . . . . .	78
3.4.2	Supervised Model Selection . . . . .	80
3.4.3	Semi-Supervised Model Selection . . . . .	81
3.4.4	Experimental Results . . . . .	84
3.5	Conclusions . . . . .	89
<b>4</b>	<b>Ultrasound segmentation for real-time 3D echocardiography</b>	<b>91</b>
4.1	Heart anatomy and functioning . . . . .	92
4.2	Cardiovascular diseases and their diagnosis . . . . .	94
4.3	3D LV segmentation . . . . .	99
4.3.1	Statistical analysis techniques for comparing LV segmentation methods . . . . .	100
4.3.2	Literature review of 3D LV segmentation methods . . . . .	102
4.4	Active Contours and Level-Set Segmentation . . . . .	106
4.4.1	Discretization of the level-set equation (4.4) . . . . .	110
4.4.2	Choice of the involved implicit function and reinitialization . . . . .	110
4.4.3	Level-set evolution strategy . . . . .	110
4.5	Proposed approach: a rigorous parallel implementation of the Sparse Field algorithm . . . . .	116
4.5.1	GPU level-set methods . . . . .	117
4.5.2	Parallel implementation of the sparse field algorithm . . . . .	118
4.6	Experimental results . . . . .	124
4.6.1	Simulated experiments . . . . .	125
4.6.2	Detection of vessels with aneurysm in 3D rotational angiogra- phy images . . . . .	129
4.6.3	Left-ventricle segmentation in Real-Time 3D echocardiography . . . . .	129
4.7	Conclusions . . . . .	137
	<b>Conclusions</b>	<b>139</b>
	<b>Publications</b>	<b>143</b>

*CONTENTS*

iii

**Bibliography** 145

**Abbreviations** 161

**List of Figures** 163

**List of Tables** 167



---

# Abstract

---

Ultrasound imaging is nowadays a well established imaging technique for medical diagnostics. In fact, ultrasound is the fastest, least invasive, and least expensive screening modality for imaging organs and therefore it is commonly used in clinical practice for assessing possible abnormalities in several parts of the human body. However, the ultrasound images are difficult to be interpreted. Clinical image quality suffers from characteristics typical of ultrasound signals reducing the diagnostic reliability of ultrasound images.

In this scenario, Computer Aided Detection (CAD) systems can be used to support physicians during diagnosis providing them a second opinion. This thesis discusses efficient ultrasound processing techniques for computer aided medical diagnostics. In particular the thesis focuses on two major topics: (i) the Ultrasound Tissue Characterization (UTC) whose aim is to characterize and differentiate between healthy and diseased tissue; (ii) the ultrasound image segmentation whose aim is to detect the boundary of anatomical structures to automatically measure organ dimensions, and/or compute clinically relevant functional indices.

The research conducted in the context of UTC is aimed at developing a CAD tool for Prostate Cancer (PCa) detection in order to improve the current prostate biopsy protocol. In particular, this thesis contributes with: (i) the development of CAD tool based on a robust classification system and a large and consistent Trans-Rectal Ultrasound (TRUS) images database, to achieve clinically significant accuracy, and that exploits parallel computing techniques on modern Graphic Processing Units (GPU) to achieve real-time performance; (ii) the introduction of both an innovative Semi-Supervised Learning algorithm and a novel supervised/semi-supervised learning scheme for CAD systems training that allow to improve the CAD system performance reducing the data collection effort and avoiding collected data wasting. The resulting PCa CAD tool working on TRUS images is aimed at improving the current biopsy protocol by allowing physician to perform a lesion-directed biopsy. This is feasible providing physicians a binary risk map highlighting suspect tissue areas. An extensive clinical validation was performed to assess the system validity as a diagnostic support tool and its effectiveness at reducing the number of biopsy cores *w.r.t.* the standard biopsy protocol without negative impinging

on diagnostic accuracy.

Concerning the ultrasound segmentation, the final aim of the conducted research is the development of a heart disease diagnostic tool based on Real-Time 3D Echocardiography (RT3DE). The contributions of the thesis to this second application are: (i) the development of an automated level-set based segmentation framework for 3D images on GPU; (ii) the application of the above mentioned level-set segmentation framework to the myocardium segmentation in RT3DE. Experimental results on simulated and real images of different nature show the high efficiency and flexibility of the proposed framework. Again, an extensive clinical validation was performed to demonstrate the effectiveness of the proposed segmentation framework as a tool for quantitative analysis of 3D cardiac morphology and function in clinical applications.



---

# Sommario

---

L'ecografia è una tecnica di diagnostica per immagini molto utilizzata nella pratica clinica poichè non invasiva, poco costosa e che permette l'analisi in tempo reale di diverse parti del corpo umano. Tuttavia, la qualità delle immagini ecografiche risulta considerevolmente inferiore rispetto a quella delle immagini ottenute con altre tecniche di *imaging*, quali raggi X, risonanza magnetica e tomografia computerizzata. La bassa qualità delle immagini ecografiche rende l'interpretazione delle stesse complessa e soggettiva, e sovente ne limita la validità diagnostica.

In questo contesto è possibile utilizzare i cosiddetti sistemi di Computer Aided Detection (CAD) per supportare i medici nell'operazione di diagnosi fornendo loro una seconda opinione. In questa tesi vengono discusse tecniche efficienti di elaborazione dei segnali ultrasonici per diagnostica medica supportata da computer. In particolare la tesi tratta due tematiche fondamentali: (i) la caratterizzazione tissutale basata su ultrasuoni, il cui scopo è quello di caratterizzare i tessuti esaminati tramite *imaging* ultrasonico e distinguere tra tessuti sani e tessuti malati; (ii) la segmentazione di immagini ecografiche, il cui scopo è quello di individuare il contorno delle strutture anatomiche in esame al fine di misurare in modo automatico le dimensioni degli organi, e/o calcolare indici di funzionalità clinicamente rilevanti.

La ricerca condotta nel contesto della caratterizzazione tissutale basata su ultrasuoni è finalizzata allo sviluppo di un sistema di CAD per la diagnosi del tumore alla prostata allo scopo di migliorare l'attuale protocollo biptico utilizzato per la diagnosi di tale patologia. In particolare i contributi della tesi in oggetto sono: lo sviluppo di un sistema di CAD che utilizza un robusto sistema di classificazione e un database di ecografie transrettali (Trans-Rectal Ultrasound: TRUS) ampio e consistente al fine di raggiungere livelli di accuratezza clinicamente significativi, e che inoltre si avvale dell'implementazione parallela su moderni processori grafici (Graphic Processing Unit: GPU) al fine di ottenere prestazioni in tempo reale; (ii) l'introduzione di un nuovo algoritmo di apprendimento semi-supervisionato e di una nuova procedura di addestramento del CAD che consente di migliorarne le prestazioni permettendo di utilizzare l'intero ammontare di dati raccolti durante le sedute biptiche e riducendo la quantità di dati necessaria per addestrare il sistema in modo affidabile. Lo scopo del sistema di CAD sviluppato è di

guidare il medico durante la biopsia, indirizzando il campionamento dei tessuti verso le zone potenzialmente patologiche, migliorando così l'efficacia del protocollo bioptico attualmente utilizzato. Una esauriente validazione con dati clinici reali ha dimostrato la validità del sistema come strumento di supporto alla diagnosi: l'adozione del sistema consentirebbe infatti una riduzione del numero di campioni di tessuto estratti durante ciascuna seduta bioptica, evitando campionamenti non necessari ma mantenendo lo stesso potere diagnostico della procedura standard.

Per quanto riguarda la segmentazione di immagini ultrasoniche la ricerca ha riguardato lo sviluppo di uno strumento per la diagnosi dei malfunzionamenti cardiaci basato sull'utilizzo di ecocardiografia 3D *real-time*. I contributi della tesi per questa seconda applicazione sono: (i) lo sviluppo di un *framework* di segmentazione automatica per immagini 3D basato sulla tecnica di segmentazione *level-set* e operante su GPU; (ii) l'applicazione del suddetto *framework* di segmentazione ad immagini ecocardiografiche tridimensionali del miocardio. Risultati sperimentali effettuati sia su immagini reali che simulate dimostrano l'elevata efficienza e flessibilità di utilizzo del metodo proposto. Anche in questo caso, come nel precedente, è stata condotta un'approfondita validazione su dati clinici reali che dimostra l'efficacia del *framework* di segmentazione proposto come strumento di supporto al medico per effettuare un'analisi quantitativa della morfologia e della funzionalità cardiaca.

---

# Introduction

---

Ultrasound imaging is based on receiving and interpreting the backscattered echo of pulsed acoustic pressure fields. Diagnostic ultrasound is used in almost all medical fields and has already become the preferred imaging modality in a wide range of clinical applications. In fact, medical ultrasound has several advantages over other popular imaging techniques such as X-ray, Magnetic Resonance Imaging (MRI) or Computed Tomography (CT). First of all, ultrasound is a non-ionizing radiation modality providing real-time images and, in this respect, it undoubtedly exceeds any other imaging modality. Moreover, ultrasound equipments are appreciably cheaper and smaller than other imaging equipments, meaning that they are the tools of choice in emergency units and ambulatories. However, behind these attractive characteristics ultrasound imaging presents several drawbacks, such as low signal resolution, and orientation dependent acquisition, that make the diagnosis hard and subjective.

To overcome these drawbacks, many innovations have been proposed focusing on both the ultrasound image generation and processing. While the development of novel transducers or scanheads, the advancements in the beamforming schemes, the reduction of analog electronics, the advent of contrast agents, as well as the improvements and steps forward in 3D and 4D imaging, produced a considerable enhancement of the image generation modality, from the other side research efforts have been focused on the development of automatic image analysis tool to support the interpretation of biomedical images traditionally performed by physicians. This process is called Computer Aided Detection (CAD) and it is aimed at improving the diagnostic reliability of a classical *human made* image analysis. In ultrasound imaging, most than in any other imaging context, the CAD system development is a research topic. In fact, while some CAD tools working on imaging modalities such as CT, or MRI are already clinically used, the realization of ultrasound based diagnosis support tools matching the clinical standard and working in real-time is still a challenge.

In this scenario, the research activity described in this thesis focused on efficient ultrasound signal processing techniques for medical diagnostics. In particular, it addressed two major topics: the Ultrasound Tissue Characterization (UTC) aimed at characterizing and differentiating between healthy and diseased tissue, and the ultrasound image

segmentation aimed at detecting the boundary of structures as a step towards automated measurement of clinically interesting indices and metrics, in particular for heart functioning assessment. Although the developed characterization and segmentation techniques are applicable to the investigation of any biological tissue, the conducted UTC research is aimed at developing a CAD tool for Prostate Cancer (PCa) detection, while the ultrasound image segmentation has been investigated in the context of left-ventricle myocardium segmentation for assessing heart functioning.

CAD systems based on UTC combine information extracted from the input ultrasound echo signal and machine learning techniques to classify tissue. In the considered application, the CAD scheme is basically a binary classification system aimed at discriminating between normal and pathological tissues. A large and consistent ground truth images database and a reliable efficient training procedure are fundamental to guarantee a significant accuracy of the developed system. Performing UTC in real-time is another fundamental characteristic to use the CAD system output as a biopsy guidance, which is the final goal of our approach in the context of PCa detection. All these aspects will be treated in this work, showing author's contributions focused on: (i) the design and development of a robust classification system and new learning procedures for CAD system training to achieve clinically significant accuracy; (ii) the implementation of the CAD processing flow steps on highly parallel architecture, instead of conventional CPUs, to achieve real-time performance.

In the first part of this thesis a real-time Computer Aided Biopsy tool will be presented. The tool enhances Trans-Rectal Ultrasound (TRUS) video stream with a false color overlay image, representing a binary cancer risk map that allows physician to perform a lesion-directed biopsy. The CAD processing flow is based on a non-linear classification approach working on features of different nature derived from the Radio-Frequency (RF) echo signal. A large and consistent TRUS video sequences database was collected, in collaboration with the Department of Urology of the S. Orsola Hospital in Bologna. The database associates RF signals acquired during biopsies with the corresponding histopathological outcome, providing the ground truth for CAD training. The tool is trained using a specifically designed learning procedure based on standard supervised-learning techniques. The CAD processing flow was implemented exploiting Graphic Processing Unit (GPU) acceleration, so that the computer aided image analysis is processed at the same time as the visualization of standard TRUS video sequence. Experimental results show that the proposed CAD scheme is able to improve the standard biopsy protocol reducing the number of biopsy cores from 8-12 to 7 without negative impinging on diagnosis accuracy. However, since in the learning process only the image portion corresponding to the biopsied area can be considered, and due to the unknown distribution of tumor in each bioptic core, it is difficult to transform the physical labels of the extracted tissue into pixel labels. As a consequence, there is an intrinsic uncertainty in the ground truth and not all the collected diagnosed sample can be used for CAD system training. One of the main contributions of this research consists in exploiting

Semi-Supervised Learning (SSL) techniques to overcome this problem. In particular, a novel Semi-Supervised Learning algorithm named SelfCo3 was designed. Combining two popular SSL paradigms, namely Self-Training and Co-Training, the algorithm enlarges the original labeled training set and select the final classification model. Experiments both on benchmark dataset and on ground truth images from biopsy findings demonstrated both the effectiveness of SelfCo3 as general purpose SSL procedure, and its ability to enhance computer aided detection of PCa without requiring any additional labeled data and making full use of the whole ground truth dataset. This new SSL algorithm was also specified to deal with the PCa detection in ultrasound images exploiting all the information provided by the ground truth (*i.e.* the global pathological status of each biopsy core) to further improve prediction performance. By using the new version of the algorithm, named SelfCo3Core, an innovative mixed supervised/semi-supervised learning procedure was designed, and a new detection tool based on this procedure was developed. This new PCa CAD tool is based on the collected ground truth database and the expressly designed supervised/semi-supervised learning scheme. The proposed CAD scheme is able to improve the Positive Predictive Value (PPV) of the standard biopsy protocol of 55% consequently reducing the number of biospy cores, requested for an accurate diagnosis, from 8-12 to 6 without losing diagnostic power. Furthermore, exploiting parallel computing on GPU, the CAD is able to deliver a frame rate of about 40fps which confirms its effectiveness as real-time biopsy guide.

Concerning the application of ultrasound segmentation in the context of heart functioning assessment, the conducted research focuses on the segmentation of the left-ventricle (LV) myocardium in Real-Time 3D Echocardiography (RT3DE). Among the great number of tools available to assess myocardial function in patients with known or suspect diseases, RT3DE, providing truly volumetric images of the heart, is the preferred imaging modality. In clinical practice, a quantitative analysis of cardiac morphology and function consists in the manual contouring of the LV and the extraction of clinically relevant functional indices. Unfortunately, manual delineation of LV boundaries in RT3DE is tedious, time-consuming and suffers from a subjective bias due to the low signal to noise ratio of the underlying images. Therefore, a fast and automatic LV segmentation is essential to enable an efficient quantitative analysis of the RT3DE data in clinical applications. Nonetheless, despite the efforts made by medical vendors and researchers, 3D LV segmentation remains a time consuming and challenging task.

In this context, the conducted research is aimed at providing physicians with an accurate, fast, and automated segmentation tool for cardiac morphology and function assessment. The approach proposed to achieve this goal exploits the accuracy of the popular level-set segmentation methods and the parallel computing on GPU. In particular, a GPU level-set solver was developed. At the best of author's knowledge, it is the first rigorous GPU implementation of the level-set sparse field algorithm. Thanks to its high computational efficiency, and flexibility in the interface evolution, the proposed GPU level-set solver is able to handle different segmentation tasks, with high segmen-

tation accuracy and reduced computational costs *w.r.t.* its sequential counterpart and comparable performance *w.r.t.* existing GPU level-set solvers, as confirmed by experimental results on both simulated and real images of different nature. By exploiting a localized region-based segmentation approach the designed solver was also made able to deal with strongly inhomogeneous images as it is the case of RT3DE images. Thus, combing the segmentation method with an automated initialization procedure, a complete fast, accurate and automated segmentation framework for 3D echocardiography has been developed. Results of an extensive clinical validation, that compares a manual segmentation procedure performed by expert physicians with an automated segmentation procedure performed using the proposed tool, demonstrated the effectiveness of the developed framework at achieving accurate 3D LV segmentation in less than 1s for each echocardiographic frame, meaning that the method could be used for a quantitative analysis of 3D cardiac morphology and function in clinical applications.

The thesis is organized as follows. In the first chapter the technological background of the work is presented. Basic concepts on ultrasound physics and brief comparison between ultrasound and other popular imaging modalities is given to highlight advantages and drawbacks of using ultrasound. Furthermore, a description of the hardware architectures of both the ultrasound imaging systems and the GPUs is provided. The concept of Computed Aided Diagnosis and the fundamentals of the ultrasound tissue characterization (UTC) and ultrasound segmentation are also outlined in this chapter.

The second chapter concerns the development of a CAD tool for PCa detection. In particular, fundamental information about prostate anatomy, pathologies and diagnosis procedures are reviewed. Then the main architecture of an UTC based CAD scheme is described, and each step of such CAD processing flow is detailed with particular attention to the approaches used to implement each of these steps. Finally, after a review of the existing approaches for UTC based computer aided diagnosis of PCa, the rtCAB tool is presented and its performance evaluated.

Chapter 3 focuses on the use of SSL techniques to improve the CAD system performance. After a brief overview on the general SSL concepts, and SSL paradigms exploited to develop the SelfCo3 algorithm, the proposed algorithm and its specialization for the target application are presented and their performance evaluated. The chapter concludes presenting the new CAD tool for PCa detection based on the developed SSL techniques.

Chapter 4 deals with the application of the ultrasound segmentation to the LV myocardium segmentation in RT3DE. Some information about the heart anatomy and pathology, as well as diagnosis procedures are reviewed. An overview of the existing method for LV segmentation in 3D echocardiography is also provided. Then, the basics of the active contour and level-set segmentation are given, detailing the evolution strategy adopted in this research. Finally, after a brief review of the existing GPU level-set methods, the developed segmentation tool and its specialization for the target application are presented and their performance evaluated.

Conclusions summarize the main contributions of this thesis and provide some clues about future research in the considered applications.





# Chapter 1

---

## Technological Background

---

In this chapter the technological background of the work is presented. Basic concepts of ultrasound physics will be briefly recalled and the architecture of a typical ultrasound imaging system will be described focusing on the two modalities used in this work which are respectively the B-Mode and 3D imaging. A comparison between ultrasound and other popular imaging modalities is also provided to highlight advantages and drawbacks of using ultrasound. Then, the concept of Computer Aided Diagnosis and the fundamentals of the ultrasound tissue characterization and ultrasound segmentation are outlined. Finally, since one major author's contribution consists in the use of GPU computing for enhancing ultrasound segmentation, as well as several steps in UTC, an overview of the concept of GPU computing and its applications, as well as an overview of the GPU architecture are given.

### 1.1 Ultrasound physics

Sound is a mechanical pressure wave that propagates through a medium according to the model of elastic waves. Ultrasound is a cyclic sound pressure wave with a frequency higher than the upper limit of the human hearing range, which is about 20kHz. In medical imaging context ultrasound have usually frequencies on the order of MHz to focus waves beam along a specific direction.

Sound is transmitted through solids as both longitudinal and transverse waves. Longitudinal sound waves are waves of alternating pressure deviations from the equilibrium pressure, causing local regions of compression and rarefaction, while transverse waves are waves of alternating shear stress at right angle to the direction of propagation. Through gases, plasma, and liquids sound propagates only as longitudinal waves since there is no mechanism for driving motion perpendicular to the wave propagation. Since all biological tissues can be assimilated to liquid, the sound waves used to form medical

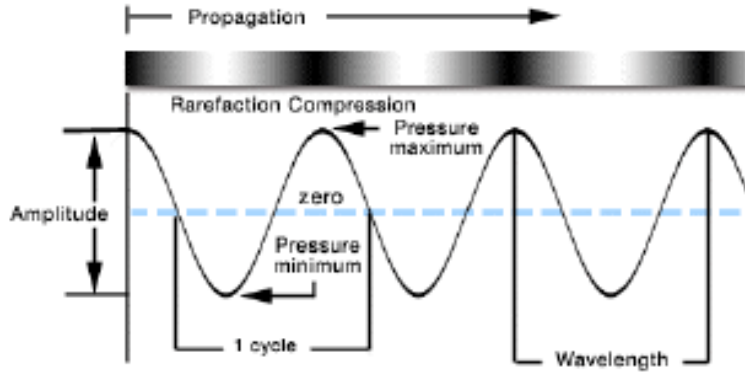


Figure 1.1: Longitudinal pressure wave.

images are longitudinal waves, and, as a consequence, we will consider a fluid model for the propagation of ultrasound waves.

A longitudinal wave (Figure 1.1) creates a sinusoidal back-and-forth motion of particles as it travels along in its direction of propagation. The particles are displaced from their original equilibrium state by a distance or displacement amplitude  $u(\mathbf{r}, t)$  and at a rate or particle velocity  $v(\mathbf{r}, t)$  as the wave disturbance passes through the medium. This change also corresponds to a local pressure disturbance  $p(\mathbf{r}, t)$ . Making the assumption of a linear, lossless, homogeneous, non-dispersive propagation medium, *i.e.* an idealized incompressible fluid, the particle velocity is related to the displacement as:

$$v(\mathbf{r}, t) = \frac{\partial u(\mathbf{r}, t)}{\partial t} \quad (1.1)$$

that can be expressed in terms of *velocity potential*  $\Phi(\mathbf{r}, t)$  as:

$$v(\mathbf{r}, t) = \nabla \Phi(\mathbf{r}, t). \quad (1.2)$$

The velocity potential allows to express the pressure field as:

$$p(\mathbf{r}, t) = -\rho_0 \frac{\partial \Phi(\mathbf{r}, t)}{\partial t}. \quad (1.3)$$

where  $\rho_0$  is the constant density of the material. This parameter has the physical dimension of a mechanical resistance and its unit of measurement is called *rayl* ( $\frac{\text{kg}}{\text{sm}^2}$ ). That being so, the propagation of the velocity potential is driven by the wave equation:

$$\nabla^2 \Phi = \frac{1}{c^2} \frac{\partial^2 \Phi}{\partial t^2} \quad (1.4)$$

where  $c$  is the longitudinal speed of sound, that in soft tissue is usually equal to  $1540 \frac{\text{m}}{\text{s}}$ . The ratio of a forward traveling pressure wave to the particle velocity of the fluid is called *acoustic impedance* and it is defined as:

$$Z = \frac{p}{v} = \rho_0 c \quad (1.5)$$

Another important characteristic of the sound waves is the *acoustic intensity*, which is the product of the sound pressure and the particle velocity. The instantaneous intensity expressed in terms of *acoustic impedance* is:

$$I = \frac{p p^*}{Z} \quad (1.6)$$

and it is measured in watts per square meter ( $\frac{W}{m^2}$ ). In three dimension, the plane wave solution of (1.4) is:

$$\Phi(\mathbf{r}, t) = \Phi_0 \left( e^{j(\omega t - \mathbf{k} \cdot \mathbf{r})} + e^{j(\omega t + \mathbf{k} \cdot \mathbf{r})} \right) \quad (1.7)$$

where  $\vec{k} = k_x \hat{x} + k_y \hat{y} + k_z \hat{z}$  is the wave vector. In (1.7) the first term represents a positive-going wave and the second represents a negative-going wave. In a practical situation, the actual variable would be the real part of the exponential, and only the positive-going wave would be considered. Note that, non trivial solutions of (1.4) (plane waves) are admitted if:

$$k = k_x^2 + k_y^2 + k_z^2 = \frac{\omega^2}{c^2}$$

where  $k_i$  is the component along direction  $i$  of the wave vector and  $\omega$  the angular frequency.

The solutions (1.7) are not valid anymore in a real propagation medium, where discontinuities, *i.e.* interfaces between regions with different acoustic impedance, are undoubtedly present. These discontinuities are cause of reflection, refraction, and diffusion phenomena. It is just the interaction of the ultrasound waves with the medium inhomogeneities the basis for the diagnostic application of ultrasound, as well as the reason behind the major limitations of this technique. In fact, from a physical point of view we can consider the echographic image as the graphical representation of the distribution of acoustic impedance variation in the medium under examination. On the other hand, these phenomena are also the cause of the loss of energy of the acoustic wave during its propagation, as well as of the presence of the characteristic noisy texture of the echographic images, which are the major limitations of this technique. Therefore, a deep knowledge of these phenomena and of the acoustic waves behavior, depending on the propagation medium characteristics, is mandatory to better understand the basis principle of the ultrasound imaging.

In the following subsections, a brief overview of the reflection, refraction, diffusion and absorption phenomena, and their implication in the ultrasound imaging formation will be given.

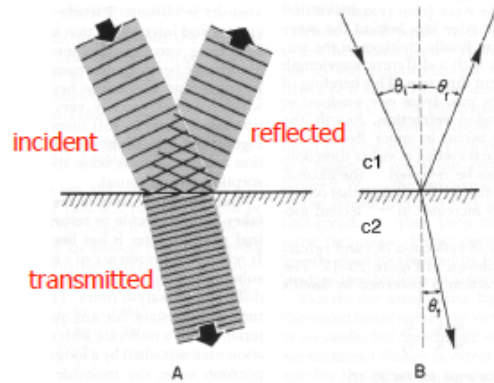


Figure 1.2: Reflection and refraction of plane waves in Snell's law.

### Reflection and Refraction

In presence of a discontinuity in the propagation medium larger than the wavelength the ultrasound wave is partly reflected and partly refracted due to the different acoustic impedance ( $Z$ ) between the two regions. The incidence, reflection, and transmission angles  $\theta_i$ ,  $\theta_r$  and  $\theta_t$ , are related each other according to the Snell's law:

$$\frac{\sin \theta_i}{\sin \theta_t} = \frac{c_1}{c_2} \quad (1.8)$$

The amplitude of the reflected and refracted waves, in case of incident wave normal to the medium discontinuity, are defined by the reflection ( $R$ ) and refraction ( $T$ ) coefficients:

$$R = \frac{P_r}{P_i} = \frac{Z_2 - Z_1}{Z_2 + Z_1} \quad T = \frac{P_t}{P_i} = \frac{2Z_2}{Z_2 + Z_1} \quad (1.9)$$

where  $P_r$ ,  $P_t$  and  $P_i$  are the pressure amplitudes of respectively the reflected, transmitted and incident wave. Among them, the signal that contributes to the ultrasound image generation is the reflected wave that comes back to the ultrasound source which usually acts as both transmitter and receiver. Since the distance covered by the ultrasound wave is proportional to its traveling time, evaluating the time interval between the wave generation and the echo reception, it is possible to construct a map of the tissue in the direction of investigation. Table 1.1 reports values of reflection coefficient for several common plane boundaries between two media in medical field.

### Diffusion and Scattering

Diffusion can be considered as the reflection of the incident wave in all directions. There are basically two types of diffusion phenomena. The first one, which is called *Rayleigh scattering*, is caused by reflecting surfaces having dimension lower or comparable with the wavelength of the incident field. In this case, the incident wave will be diffused towards different directions according to a radiation pattern that is function of the field

Interface	Reflection Coefficient
soft tissue - air	99.9%
soft tissue - lung	52.0%
soft tissue - bone	43.0%
vitreous humor - eye lens	1.1%
fat - liver	0.79%
soft tissue - fat	0.69%
soft tissue - muscle	0.04%
water - lucite	13%
oil - soft tissue	0.43%

Table 1.1: Reflection coefficient for common interfaces in medical field.

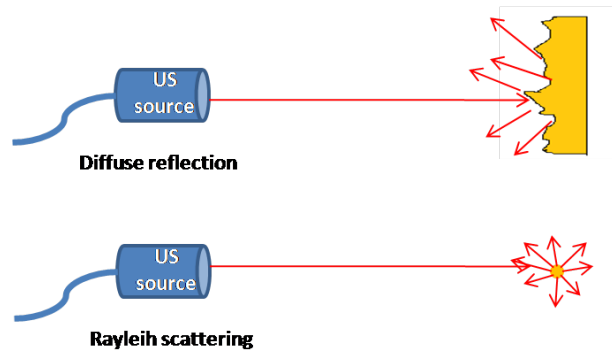


Figure 1.3: Diffuse reflection and Rayleigh scattering.

wavelength and of the obstacle geometry. In particular, when the dimensions of the non homogeneous region are negligible *w.r.t* the field wavelength, the obstacle can be modeled as a point scatter, and the incident field is diffused uniformly along all the directions in the form of spherical waves, as if the obstacle was a new ultrasound source. The second type of diffusion, is the so called *diffuse reflection* which is generated when the acoustic wave encounters rough interfaces between different media. In this case, although the interfaces have dimensions greater than wavelength, the presence of irregularities, with dimensions comparable or lower than the wavelength, causes a diffusion of the ultrasound beam along multiple reflection angles. This loss of coherence of the ultrasound beam produces a reduction of the backscattered echo intensity. The dependency of the scattering to the incident wave frequency allows to obtain information about the type of tissue that compose the physiological system under examination. In fact, the micro-structures concurring to the scattering generation are different depending on the incident wave frequency because of their different dimensions. In a physiological system such micro-structures can be micro-vessels, cells, and any tissue viscosity, elasticity or density variation. However, when the ultrasound wave encounters different small reflecting structures, the generated diffused echoes cause both constructive and destructive interference. The constructive interference enhances the measured backscattered echo, while the destructive interference generates the so called *speckle noise*. Therefore, the

obtained echographic image is not a faithful representation of the medium structure layout, but rather a representation of the obstacle hit by the ultrasound beam. Note that, although the speckle noise is dependent on the acquisition system properties, it also contains important informations about the imaged area, intuitively linked to the texture appearance in the ultrasound images.

### Attenuation

As for the propagation medium homogeneity assumption, also the lossless assumption considered so far cannot be maintained for a real propagation medium. In fact, in real materials, as well as biological tissues, the partial conversion of acoustic energy into heat (*absorption*), causes an amplitude reduction of the propagation wave. Absorption can be modeled as an exponential reduction of acoustic wave amplitude as function of the distance

$$A_s(\bar{r}) = A_0 e^{-\alpha|\bar{r}|} \quad (1.10)$$

where  $A_0$  is the wave amplitude and  $\alpha$  is the absorption coefficient depending on the frequency of the incident field. As reviewed before, also the reflection and refraction phenomena cause a reduction of the acoustic field during its propagation. The overall effect of reflection, diffusion and absorption causes an attenuation that can be modeled as an exponential law:

$$A(\bar{r}) = A_0 e^{-a|\bar{r}|} \quad a = \alpha + \alpha_s \quad (1.11)$$

where  $\alpha_s$  is the attenuation coefficient due to field diffusion. A practical consequence of the attenuation is the reduction of the visualization depth of the body that progressively decreases with the frequency increase. Table 1.2 reports values of attenuation for different materials and biological tissues expressed in terms of penetration depth for different frequencies, *i.e.* the depth at which the acoustic intensity is attenuated of 3 dB. In medical context, the low penetration depth of the high frequency beam is usually compensated using invasive transducers (trans-rectal, trans-vaginal, esophageal).

## 1.2 Ultrasound imaging systems architecture

The function of an ultrasound imaging system can be summarized as follows: the ultrasound probe generates a pulsed acoustic pressure field. The field propagates in the physiological system where, due to the presence of different macro- and micro-structures, multiple Radio-Frequency (RF) signals are generated. These raw signals are received by the same probe that works as both transmitter and receiver. Then, in the core of the imaging system, these signals are processed to generate the final output which is usually a Brightness Mode (B-Mode) image. In B-Mode imaging the returned echoes are displayed in a 2D image which intensity is proportional to echoes amplitude. Other imaging modalities exist such as the A-Mode, the M-Mode, the Real-Time B-Mode, the Doppler and Color Doppler Mode [1], some that are fallen into disuse and other that are

Material	f = 1 MHz	f = 5 MHz
Water	1360	54
Air	0.25	0.01
Brain	3.5	1
Liver	3	0.5
Fat	5	1
Muscle	1.5	0.3
Oil	3	0.12
Bone	0.2	0.04
Perspex	1.5	0.3
Polyethylene	0.6	0.12
Kidney	3	0.5
Blood	17	3
Vitreous humor	-	6

Table 1.2: Penetration depth at -3 dB for common materials and tissues.

used in specific diagnostic applications. Among them B-Mode is the most general purpose and therefore the most widely used modality. Recently 3D ultrasound imaging has received increasing attention since it allows to overcome some limitations and pitfalls of the conventional 2D ultrasound imaging such as the exam subjectivity, the impossibility to obtain the optimal *acoustic window* in some situations, and the impossibility of performing quantitative measurements without assuming geometrical models of the imaged structures.

In this manuscript we will focus on two of the mentioned imaging modalities: the classical B-Mode and the 3D ultrasound imaging, respectively applied to the prostate and heart imaging.

### 1.2.1 B-Mode imaging

The fundamental units of a B-Mode ultrasound imaging system are shown in the block diagram illustrated in Figure 1.4. The system architecture can be substantially divided in two sections: the transmit section, which drives the ultrasound transducer to generate the ultrasound beam, and the receive section which is devoted to the signal acquisition and processing, and the final image generation. Transmit and receive sections are adequately synchronized to accurately reconstruct the B-Mode image.

#### Ultrasound Probes

The ultrasound probes are all made by piezoelectric transducers that exploit the piezoelectric effect to generate acoustic waves by converting electrical energy into mechanical energy. The main distinction in ultrasound probes is between mono-element and multi-element transducers. Mono-element transducers have the advantage of requiring simpler control, but also have the drawback of requiring mechanical focusing and, as a consequence, a fixed focal length. Therefore, multi-element probes, that take advan-

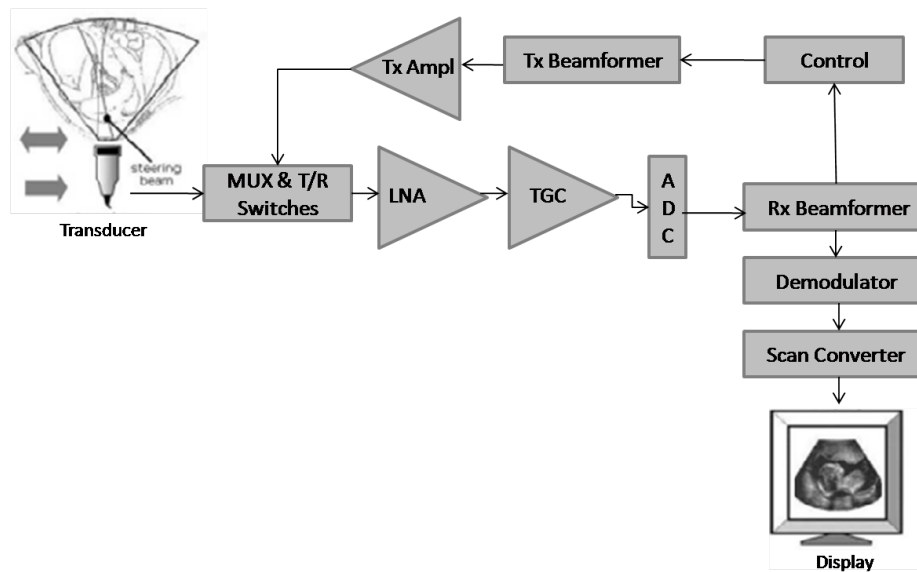


Figure 1.4: Block diagram of an ultrasound imaging system.

tage of the electronic focusing, are preferred in medical imaging applications, where the possibility of changing focus length, depending on the clinical requirements and employment condition, is mandatory. Multi-element transducers (arrays) can be further classified according to the array-element configuration: linear sequential, curvilinear, linear phased, 1.5D and 2D (matrix) arrays. The shape of the scanned region is determined by the array-element configuration. For a 2D image the scanning plane is the azimuth dimension, while to generate a volumetric image the beam needs to swap three spatial directions. This is done manually or mechanically if a 1D array is used, or electronically if a 2D array is used. Figure 1.5 shows typical array-configurations and the region consequently scanned by the acoustic beam. Driven by many clinical needs, a wide range of medical imaging probes, that are different not only in transducer geometry, but also in size and form, is available (see Figure 1.6).

### Transmit section

This section aims at generating the voltage pulses used to drive the transducer piezoelectric elements, so that a pulsed echo signal can be generated. Therefore, its structure complexity depends on the used transducer: the more piezoelectric elements must be driven the more the transmit section structure will be complex. Modern imaging systems are usually equipped with multi-element phased array probes with different forms and sizes according to diagnostic field (see Figure 1.6). Multi-element probes allow to implement focusing and image scan electronically. This is performed by the beamforming stage, which drives all the probe elements with a pattern of delayed pulses in order to focus the ultrasound beam and move it in the desired position, reducing the secondary



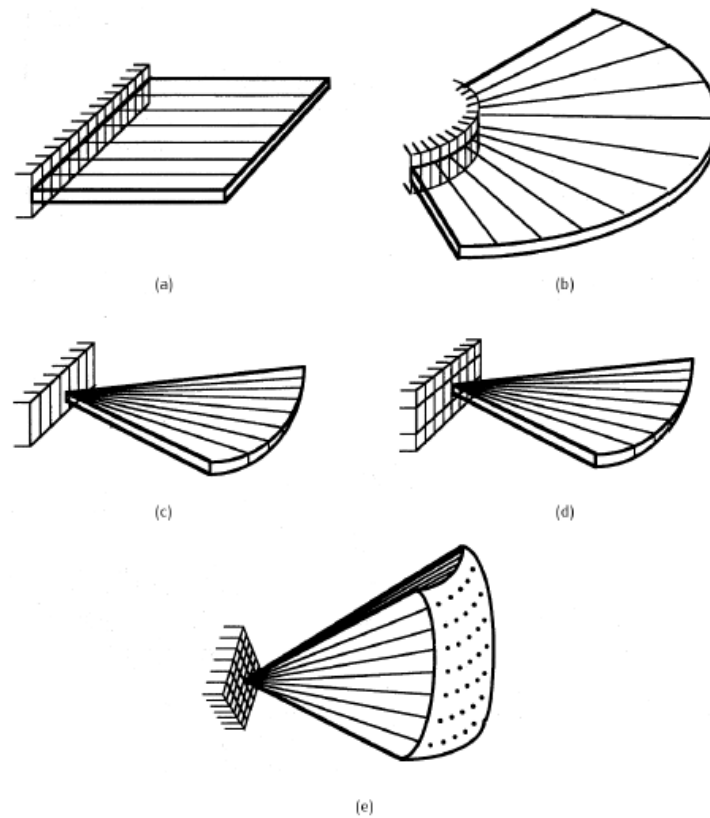


Figure 1.5: Configuration of multi-element probes and possible region scanned by the acoustic beam: sequential linear array (a), curvilinear array (b), linear phased array (c), 1.5D array (d) and 2D array (e).

lobes of the radiation pattern.

### Receive section

This section is equipped for acquiring, processing, storing and displaying the signal. The main elements of this section are: a low noise pre-amplifier, a time gain compensation amplifier, an analog to digital converter (ADC), a focusing and beamforming stage, a demodulation stage, and a scan-converter. The pre-amplifier is located immediately downstream of the transducer and it amplifies the echo signal to overcome the high attenuation that it encounters during the propagation. This amplification stage is usually characterized by low noise and high dynamical range. After that, in order to compensate the effect of the depth dependent attenuation, and therefore have a uniform representation of tissue reflectivity, a time dependent amplification, usually called time gain compensation (TGC) is performed. TGC is also necessary to reduce the signal dynamic before the analog to digital conversion. The ADC can be located upstream or downstream of the beamforming, and both its resolution and sampling frequency depend on



Figure 1.6: Examples of medical ultrasound probes (picture from [2]).

its position in the imaging system architecture, as well as on its characteristics. As in the transmit chain, in reception, a beamforming block provides for focusing and image scan. After the amplification and focusing, the echo signal must be processed to extract the desired information. Therefore, at this stage, the RF signal is demodulated by the Hilbert Transform providing the so called in phase-quadrature signal (IQ). Then, the envelope is computed as the absolute value of the IQ signal and logarithmic compression is applied. The compressed signal, known as baseband or B-Mode signal is the standard output of the ultrasound acquisition system. In fact, the enveloped signal is the most useful and intuitive output to interpret echoes because its amplitude is related to the tissue echogenicity. However, recently, some manufacturers have started to provide access to the raw RF signal in their equipments. The goal of this choice is twofold: on the one hand, using the phase information of the raw RF echo more information about the imaged tissue can be inferred, on the other hand, the study of this signal provides a way to reduce system dependent effects. As claimed by different authors [3, 4] the RF signal is the key to perform Ultrasound Tissue Characterization, which is one of the main topics of this thesis.

At this point a number of RF scan lines are collected into a memory matrix. In order to make a television or PC-style rectangular image, this information must be then spatially remapped by a process called *scan-conversion*. As each scan line is the result of an angular rotation of the ultrasound beam, pixel coordinates in the matrix are related to the true position in the body by a polar to Cartesian scan-conversion, allowing the typical B-Mode sector image to be displayed on screen. Finally, a gray-scale mapping is applied. It associates a darker and a brighter appearance to the weak and strong echoes, respectively.

### 1.2.2 3D ultrasound imaging

The coming of 3D visualization in medical imaging go back to the early 1970s, when the introduction of CT revolutionized diagnostic radiology by providing, for the first time, true 3D anatomical information to the physician. At the beginning this information was

presented as a series of contiguous 2D image slices of the body. However, the advent of 3D images from CT and (later) MRI provided a major impetus to the field of 3D visualization. Later on, in the early 1990s the advances of 3D visualization along with the advances in ultrasound imaging modality has lead to the development of 3D ultrasound imaging. It was a way to address the disadvantages of conventional ultrasound imaging. First, the subjectivity of the conventional ultrasound exam, where a series of 2D ultrasound images are mentally combined by the operator to form subjective impression of the 3D anatomy and pathology. In fact, this makes the success of the diagnostic or interventional procedure largely dependent on the skill and the experience of the operator. Second, diagnostic and therapeutic decisions often require accurate estimation of organ (or tumor) volume. Moreover, since the orientation and location of 2D ultrasound images are determined by the transducer, some views are impossible to achieve due to restrictions imposed by the patient's anatomy or position. In 3D ultrasound examination, an objective image is created combining the 2D images by a computer. Then, the obtained image can be viewed, manipulated and measured, by the physician, directly in 3D, and even a 2D cross-sectional image can be generated in any orientation at any anatomical site.

The 3D ultrasound imaging system architecture can be simply described by three blocks: *data acquisition*, *reconstruction* and *visualization*. The diversity of the 3D system architectures that were developed is mostly related to the data acquisition step. Three main different acquisition modalities have been proposed: free-hand acquisition, mechanical localizer acquisition, and acquisition with 2D arrays [5, 6]. The first two modalities use 1D array transducers. The free-hand scanning acquires planar views at arbitrary spatial position and orientation controlled by the operator as in conventional 2D echographic examinations. In mechanical scanning planar views are acquired with a 1D transducers moved at regular interval along a specified path. The third approach, instead, uses matrix phased-array transducers to scan a true three-dimensional volume at each pulse. This technology introduced by von Ramm [7], for cardiac applications, was fundamentally different from the former generation of 3D systems as the entire cardiac volume was acquired in real-time (one scan for one cardiac cycle). It is also important to note that acquiring data with 1D transducers usually requires the employment of breathing and ECG gating to avoid the artifacts and distortions caused by the breathing and organs movement. This is avoided using the 2D array visualization since the data acquisition is fast enough to avoid artifacts. The reconstruction stage consists in generating a 3D representation of the structures from the acquired set of 2D slices using knowledge of their position and orientation. Therefore, this stage is avoided if 2D arrays acquisition modality is used since the acquired data are already 3D data. As a consequence, the computational cost of the image acquisition is reduced. Finally, the visualization stage consists in the volume rendering of the 3D structure on the system display. Although the quality of 3D images critically depends on the acquisition and the fidelity of the image reconstruction, the viewing techniques to display the 3D image is crucial to

Modality	Ultrasound	X-ray	CT	MRI
What is imaged	Mechanical properties	Mean tissue absorption	Tissue absorption	Biochemistry ( $T_1$ and $T_2$ )
Access	Small windows adequate	2 sides needed	Circumferential around body	Circumferential around body
Spatial Resolution	Frequency and axially dependent	0.3-3 mm	$\sim 1$ mm	$\sim 1$ mm
Penetration	Frequency dependent	Excellent	Excellent	Excellent
Safety	Very Good	Ionizing radiation	Ionizing radiation	Very good
Speed	100 frames/sec	Minutes	1/2 minute to minutes	10 frames/sec
Cost	Low	Low	High	Very High
Portability	Excellent	Good	Poor	Poor

Table 1.3: Comparison of imaging modalities [8].

determine the information transmitted to the operator. More details on architecture of 3D ultrasound imaging systems can be found in [6, 5, 8]. While the first 3D ultrasound system were based on serial acquisition of 2D images, due to the difficulties of ECG and respiratory gating, and thanks to the technological maturation of 2D matrix array transducers, the development of real-time 3D imaging has received increasing attention, and, nowadays, is implemented in commercial high-end echo equipment, and is inserted in daily practice. In particular, real-time 3D imaging is widely used in some clinical application such as the fetal scanning, 3D intra-vascular ultrasound (3D IVUS), and, above all, Echocardiography. Real-Time 3D Echocardiography (RT3DE) is today a well established technique for imaging the heart since it allows truly volumetric visualization of this organ avoiding some of the problems intrinsically associated with 2D imaging (foreshortening, the need for geometrical assumption when computing volumes, out of plane motion artifacts, *etc.*). As stated in the introduction, the RT3DE is the application of interest in this thesis.

### 1.3 Ultrasound vs. other imaging modalities

As stated above, ultrasound, because of its efficacy and low cost, is often the preferred imaging modality. To better understand the reason behind this preference let's see how ultrasound compares to other imaging modalities. Results of this comparison are summarized in Table 1.3. The terms of comparison among medical imaging modalities can be roughly divided into two categories: *technical properties* and *practical features*. The physical principle, the access way to the visualization, the spatial resolution and the penetration, belong to the former category, while, safety, speed, cost, portability, and the dependence on the operator skills and experience of both the screening and related diagnosis operation, belong to the latter one.

Ultrasound images reflects the mechanical properties of the system under examination. In particular, ultrasound images can be viewed as maps of the mechanical structures of the body (according to their acoustic properties), that depend on density and stiffness or elasticity. The dynamic motion of organs such as the heart can be revealed by ultrasound operating up to hundreds of frames per second. Ultrasound imaging has a spatially variant resolution that depends on the size of the active aperture of the transducer, as well

as on its center frequency and bandwidth, and on the selected transmit focal depth. This lateral spatial resolution is depth-variant principally due to the beamforming and it usually varies in the range of 0.3-3mm for typical frequencies in the range of 1-15 MHz.

Another factor in determining resolution is attenuation which limits penetration, and increases with higher center frequencies and depth. Therefore, penetration decreases correspondingly so that fine resolution is difficult to achieve at deeper depths. By using specialized probes, that provide access to regions inside the body, this limitation can be partially overcome. Otherwise, access to the body is made externally through many possible acoustic windows, where a transducer is coupled to the body with a water-based gel. Except for regions containing bones, air, or gas, which are opaque to imaging transducers, even small windows can be enough to visualize large interior regions. Conventional X-ray and CT, instead, are based on the tissue absorption.

X-ray imaging is more straightforward than ultrasound imaging because X-rays travel at the speed of light in a straight ray path without diffraction effects. As a result of the ray paths, highly accurate images are obtained. As the X-rays pass through the body, they are absorbed by tissue so that an overall mean attenuation image results along the ray path. The depth information of structures imaged is lost because three-dimensional structures of the body are superimposed as a two-dimensional projection onto film or a digital sensor array. Spatial resolution is determined by focal spot size of the X-ray tube and scatter from tissue and it is about 1mm. X-rays cannot differentiate among soft tissues but can detect air (as in lungs) and bones (as in fractures).

CT scanning also involves X-rays. An x-ray fan-beam source on a large ring radiating through the subject's body to an array of detectors working in parallel on the opposite side of the ring. The ring is rotated mechanically in increments until complete coverage is obtained. Rapid reconstruction algorithms create the final image of a cross-section of the body. The resolution of CT images is typically 1 mm. CT scanning is particularly suited for imaging the brain, bone, lung, and soft tissue, so it is complementary to ultrasound.

MRI technique exploits the high water content of the body. When the human body is placed in a strong magnetic field, the water hydrogen atoms are excited. Then, the applied field is shut off and the original magnetic moment decays to equilibrium and emits a signal. This voltage signal is detected by coils, and two relaxation constants are sensed. The longitudinal ( $T_1$ ) and the transversal ( $T_2$ ) magnetization relaxation constants. These constants are used to discriminate among different types of tissue and for image formation. The resolution is mainly determined by the gradient or shape of the magnetic field, and it is typically 1 mm. Images are calculated by reconstruction algorithms based on the sensed voltages proportional to the relaxation times. Tomographic images of cross-sectional slices of the body are computed. MRI is used to visualize internal structures of the body in detail (especially brain).

Concerning the so called *practical use features* it must be said that diagnostic ultrasound is noninvasive, safe, and does not have any cumulative biological side effect.

The same consideration can be made for the MRI imaging. Conversely, both X-ray and CT are forms of ionizing radiation, that are harmful for patients. The low portability of X-ray equipments is also related to patient's safety. In fact, these systems tend to be stationary so that safety precautions can be taken more easily. Also, CT and MRI imaging systems are stationary in order to fit a person inside. As a consequence, all these equipments are more (MRI) or less (X-ray) expensive. On the contrary, ultrasound equipments are relatively low cost and portable. Finally, a comparison in term of operator skills dependency should be done. Taking X-ray or CT and MRI images is straightforward after adequate training, but, however, in all these cases making diagnosis from the acquired images requires considerable experience. Concerning ultrasound imaging, the operator dependency is one of its major drawbacks: an high skill level is needed to obtain good images with ultrasound. This expertise is necessary because of the number of access windows, the difference in anatomy and the many possible plane views. Nevertheless, a great deal of experience is required to make diagnosis from ultrasound images due to their challenging characteristics (poor image quality, missing boundaries, *etc.*).

In the last decades several attempts have been made to improve ultrasound images quality and enhance reliability of ultrasound based diagnostic examination. These attempts, involving several engineering and technological innovations, are aimed at directly enhancing the ultrasound image quality based on the principle that: if the image quality is enhanced and its dependency on the system effects is reduced, the difficulty in performing image interpretation and consequent diagnosis operation is reduced.

A different approach, that have also great evolved in last years, is the development of methods for Computer Aided Diagnosis of pathologies. The basic idea of these methods is not to replace physicians but rather provide them a second opinion to enhance the diagnostic reliability.

## 1.4 Computer aided diagnosis

The interpretation of biomedical images, traditionally performed by physicians, can be supported by automatic image analysis tools in order to improve the accuracy and consistency of diagnoses and also reducing the image reading time.

Early studies on quantitative analysis of medical images by computers were based on the idea that computers could completely replace physicians in detecting abnormalities. This concept, named computer diagnosis or automated diagnosis, was early replaced with an approach assuming that computer output could be utilized by physicians, but not replace them. This approach is known as Computer Aided Diagnosis, or Computer Aided Detection (CAD). The main difference between automated diagnosis and CAD is that with CAD the physicians use the computer output as a second opinion, but are the physicians themselves that make the final decision. Therefore, although the higher the system performance, the better the result of the overall diagnosis, the performance level of the CAD system does not have to be equal or higher than that of radiologists.

Conversely, with automated diagnosis, the system performance is required to be very high, higher than the average sensitivity of physicians, otherwise it would not be accepted by patients [9]. That being so, CAD systems have become quite popular in recent years as support decision tools aimed at overcoming current limits of diagnostic imaging and improving the reliability of physician judgment by reducing subjectivity, inter- and intra-observer variability and potential human errors.

Basically, CAD systems combine elements of artificial intelligence and digital image processing with radiological image processing. In the last decades, CAD has become one of the major research subjects in medical imaging and diagnostic radiology. Several CAD schemes have been proposed to detect different abnormalities and assess the functioning of different organs (breast, chest, colon, brain, liver, kidney, vascular and skeletal system, and prostate), working on different imaging modalities (X-ray, CT, MRI, Ultrasound). The final goal of a CAD system depends on its application: they can be used, for example, to just detect cancer lesions or both to detect and classify or diagnose them, as well as to extract quantitative measures such as organ or tumor volume or even guiding treatment planning and surgical operations.

There already exist some CAD tools commercially available, such as CAD tools of iCAD [10] or Hologic [11], and software packages for automated and semi-automated segmentation and volume quantification of the heart (3DQ advanced software suite by Philips Healthcare [12], 4DAutoLVQ by GE Healthcare [13]). However, despite the existence of these commercial tools, and the huge research effort of both academic institution and industry, they are rarely used in day-to-day clinical practice. For instance, some hospitals use CAD to support preventive medical check-ups in mammography (diagnosis of breast cancer), the detection of polyps in the colon, the detection of lung cancer, but this is not a well established, widespread practice. This is mainly due to the fact that, usually, the existing CAD schemes achieve high sensitivities but tend to have very low specificity, meaning that most of the visual marks that they generate are false positive. If a huge number of this false-positive marks are misinterpreted by the radiologists as true-positive, the number of patient subjected to follow up tests could increase. Moreover, very often, CAD schemes do not satisfy the time requirement of the target application. Similarly, the use of quantitative tools for assessing treatment and therapies for heart failure management is not widely common in clinical practice, since the accuracy of the available commercial tools is still heavily user-dependent and the performed analysis is too much time consuming. It is also important to note that the CAD tools that are already clinically used work on imaging modalities such as CT and MRI, but not ultrasound. This is due to the challenging characteristics of this imaging modality. It is thus clear that the realization of effective CAD systems, matching the clinical standard and working in real-time, is still a challenge, especially in the context of ultrasound imaging. As stated in the introduction, this motivates the research described in this thesis.

### 1.4.1 Ultrasound tissue characterization

The ultimate goal of Tissue Characterization is to provide a *parametric image* (spatial map) of tissue properties. UTC is a challenging task due to the interactions between biological tissue, therefore, it is not always strictly quantitative but it is used to discriminate tissue [14].

CAD systems based on UTC combine information extracted from the input ultrasound echo signal and machine learning techniques to classify tissue, providing its characterization. More practically, the analysis consists in segmenting the image into several regions of interest (ROIs), computing some characteristics, features or measures related to each ROI, and classifying them into one of few known categories. Any category represents a particular state of the tissue outlined in a ROI; the simplest case is binary classification, aimed at discriminating a normal and pathological tissue, neglecting the presence of benign abnormalities.

A large and consistent ground truth images database and a reliable efficient training procedure are fundamental to guarantee a clinical significant accuracy of the developed system. This is not obvious for two main reasons. First, very often, and especially in the context of PCa detection, there do not exist public large and consistent databases of ultrasound images. Second, creating a reliable ground truth dataset is really difficult. Indeed, even histopatologists, who have specimens of the actual tissue in question for microscopic examination, often have difficulty in making diagnoses, and this could introduce a certain degree of uncertainty in the ground truth. Finally, collecting such ground truth is difficult, expensive and time consuming, as it requires the effort of experienced personnel. In this context a specifically designed learning procedure is mandatory for a reliable and efficient CAD system training. All these aspects will be treated in this work, showing author contributions.

Performing UTC completely in real-time is another fundamental and difficult characteristic that a reliable CAD system should have to be used in clinical environment. In fact, this is necessary to preserve the real-time nature of the echographic examination. In order to achieve this goal, the approach followed in this work is to implement the most computationally demanding steps of the analysis on highly parallel architecture instead of run them on conventional CPUs, so that the entire process can be speed up. One possible choice could be to use reconfigurable hardware such as Field Programmable Gate Arrays (FPGAs). However, considerations regarding cost, flexibility, and developing time, make this approach to be not the best solution. On the contrary, use a modern Graphic Processing Unit (GPU) to satisfy the system time performance requirements could be more effective, efficient and more suited to exploit the intrinsic application parallelism. The exploitation of the computational power of GPUs for non graphic applications is a recent but well established trend known as *GPU computing*. For the sake of completeness a brief overview on GPU computing and its application to medical imaging, will be given hereinafter.



### 1.4.2 Ultrasound image segmentation

In some respects, image segmentation is a simpler problem *w.r.t.* the UTC, since the objective is to delineate organs and structures, and this, in principle, does not depend on actual tissue identification. Rather, it depends on the identifications of characteristics which distinguish the ultrasonic data from different anatomical regions. However, ultrasound image segmentation presents some characteristic artifacts that make the segmentation task complicated, and almost impossible by means of standard image segmentation techniques. In fact, while in other areas of medical imaging (CT, MRI) the application of standard image processing methods is sufficient to obtain good segmentation, in ultrasound satisfactory results can only be achieved by specific methods.

Concerning the ultrasound segmentation, the aim of the conducted research is the development of an heart disease diagnostic tool based on RT3DE. In fact, RT3DE has been recently proven to be a powerful technique for imaging the heart, and the development of tools performing an automated quantitative analysis of cardiac morphology and function has received increasing attention [15, 16, 17, 18, 13, 12]. In clinical practice, such quantitative analysis, aimed at early detecting cardiovascular diseases, relies on visual inspection of the images by expert cardiologists and the consequent extraction of functional indices such as stroke volume, ejection fraction, cardiac output and strain. This process involves both the segmentation and tracking of the myocardium muscle throughout the cardiac cycle, which is usually performed manually by physicians. A manual contouring of the myocardium is tedious and time-consuming, and suffers from a subjective bias due to the low signal-to noise ratio of ultrasounds scans. For this reason a fast automated segmentation procedure is highly desirable in order to reduce the inter- and intra-observer variability in the myocardial borders detection and to speed up the segmentation process. To achieve this goal, in this thesis, we exploit again the computational power of the general purpose GPUs and the accuracy of popular level-set segmentation techniques, to address the myocardium segmentation in RT3DE near real-time.

## 1.5 GPU computing in medical imaging and computer aided diagnosis

The GPU has become an integral part of today's mainstream computing systems. The modern GPU is not only a powerful graphics engine but the most pervasive massively parallel processing platform ever available, as well as the most cost effective. The GPU's rapid increase in both architecture capabilities and programmability has pushed the trend of using GPUs as general purpose processors for enhancing complex, computational demanding application belonging to many scientific areas. This effort in general purpose computing on the GPU is known as GPU computing [19]. Today GPU computing enables application that were previously thought unfeasible because of long execution times. As we enter the era of GPU computing, demanding applications with substantial

parallelism increasingly use the massively parallel computing capabilities of GPUs to achieve superior performance and efficiency. The GPU is designed for a particular class of applications with the following characteristics: large computational requirements, intrinsic parallelism, and throughput more important than latency. Over the past years, a growing community has identified other applications with similar characteristics and has successfully mapped them onto the GPU. The most common approach followed in GPU computing is, however, using heterogeneous CPU+GPU coprocessing, since CPU and GPU have complementary attributes that allow application to perform better by using both types of processors. Many applications consist of a mixture of fundamentally serial control logic and inherently parallel computations, and have been therefore accelerated using this approach. Such applications are, for example, quantum chemistry, molecular dynamics, computational fluid dynamics, quantum physics, bioinformatics, astrophysics, finite element analysis, video transcoding, speech recognition, computational modeling and medical imaging. In particular, in the latter domain, GPU computing has been applied in image reconstruction [20], real-time denoising, registration, deconvolution, segmentation and visualization. A comprehensive survey on the application of GPU computing in medical imaging can be found in [21].

As stated above, we are interested in using GPU computing to accelerate CAD systems processing flow, in order to preserve the real-time nature of the echographic examination and make them usable in clinical environment. Our approach for integrating GPU computing in the CAD systems development is based on the following principle: expose the application parallelism and, based on an intimate knowledge of the architecture parallelism, perform the application mapping, linking software and hardware parallelism. To better understand our approach, both the hardware architecture and programming model of modern GPUs will be described referring to the NVIDIA GPUs, since these are the architectures used in the presented research.

### **GPU evolution and architecture**

The first GPU, developed by NVIDIA in 1999, was expressly designed to implement the traditional graphic pipeline [19]. The first GPU architecture was based on a fixed function pipeline, meaning that once the programmer sent graphics data into the GPU's pipeline, the data could not be modified [22]. Fueled by the insatiable desire for life-like real-time graphics, many changes have been introduced in such an architecture both in terms of hardware architecture and programmable model. Quickly, the pipeline became programmable and the first attempts to general purpose GPU computing appeared since 2003. But, despite remarkable results were obtained, these early attempts faced several drawbacks, such as the necessity of an intimate knowledge of graphic Application Programming Interface (API) such as DirectX, OpenGL, and Cg [23], and the necessity of expressing problems in terms of vertex coordinates, textures and shader programs (*i.e.* in terms of graphic pipeline elements). It was the introduction of a unified graphic and computing architecture [24, 25], known as (*unified shader architecture*) that really made

the traditional graphics pipeline model purely a software abstraction, and the GPU a massively parallel fully programmable stream processor. NVIDIA combined this technology with another key technology: CUDA (Compute Unified Device Architecture), a software and hardware architecture that enabled GPUs to be programmed with a variety of high level programming languages (C, C++, Fortran, OpenCL, etc.). Together, these knew technologies were the real bases of GPU computing.

### **GPU hardware architecture**

GPUs are basically arrays of scalable processors. The fundamental GPU computational units are indeed scalar processors called streaming processors (SPs), or CUDA cores for NVIDIA GPUs. The SPs are organized into streaming multiprocessors (SMs), that usually also contain Shared Memory, a Register File, Floating-point units, Texture Units, and eventually other modules such as a Warp Scheduler, depending on the GPU generation. The number of SPs varies not only among different GPU generations, but also among different graphic cards. In fact, to address the different market segments, the GPU architecture scales the number of processor cores and memories to implement different products for each segment while using the same scalable architecture and software. NVIDIA scalable GPU computing architecture varies the number of SMs to scale computing performance, and the number of DRAM memories to scale memory bandwidth and capacity. In CUDA programming model *threads* are the base parallelism units. They are concurrent code fragments with associate state executed by the SPs. Each multithreaded SM provides sufficient threads, processor cores, and shared memory to execute one or more threads or group of threads. The parallel SPs within a SM execute instructions for parallel threads. Multiple SMs provide coarse-grained scalable data and task parallelism to execute multiple coarsegrained thread blocks in parallel. Multithreading and parallel-pipelined processor cores within each SM implement fine-grained data and thread-level parallelism to execute hundreds of fine-grained threads in parallel. Application programs using the CUDA model thus scale transparently to small and large GPUs with different numbers of SM and processor cores [25]. An example of GPU architecture referring to the third-generation NVIDIA Fermi computing architecture is shown in Figure 1.7. In this work two different NVIDIA GPUs architecture have been used: the Fermi architecture, and the previous GT200. Interested readers can find more details on these architecture in [26, 27].

### **Efficient Multithreading and SIMT Execution model**

The streaming multiprocessor is the lowest independent level of the processors hierarchy and implements multithreading and thread scheduling for up to 1536 concurrent threads (in NVIDIA Fermi architecture). To efficiently manage and execute these many individual threads, the multiprocessors employs the so called Single Instruction Multiple Threads (SIMT) execution model. This model is not a speculative execution model, and it is basically a variation of the classical single Instruction Multiple data (SIMD)

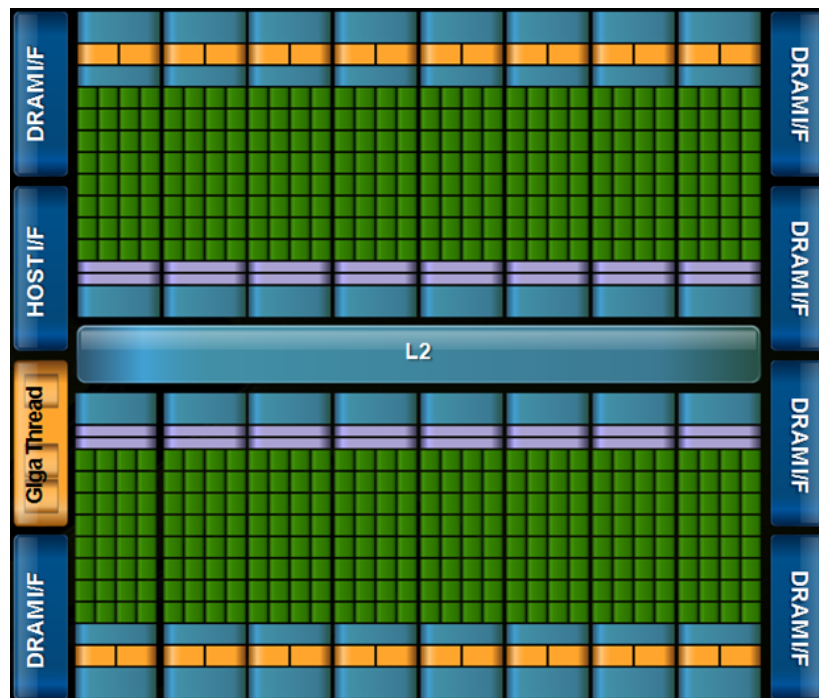


Figure 1.7: NVIDIA's Fermi Architecture (picture from [26]).

[24] that moves the parallelism unit from the data to the execution flow (thread). The SIMT instruction logic creates, manages, schedules, and executes concurrent threads in groups of 32 parallel threads called warps. Threads in the same warp belong to the same type and start execution in the same moment but are free to make branches and execute independently. The SIMT instructions control the execution of an individual thread, including arithmetic, memory access, and branching and control flow instructions. For efficiency, the SIMT multiprocessor issues an instruction to a warp of 32 independent parallel threads. The streaming multiprocessor realizes full efficiency and performance when all threads of a warp take the same execution path. If threads of a warp diverge at a data-dependent conditional branch, execution serializes for each branch path taken, and when all paths complete, the threads converge to the same execution path.

### CUDA scalable parallel architecture

CUDA is a hardware and software coprocessing architecture for parallel computing that enable NVIDIA GPUs to execute programs written in common high level language such as C. CUDA is based on three key extensions: a hierarchy of group of thread, shared memories, and synchronization barriers that allows programmers to better exploit the hardware parallelism by means a minimal set of C language extensions. In CUDA the GPU (device) is seen as a co-processor to which the CPU (host) demands operations with high parallelism. Therefore, a CUDA program is organized into a host program,

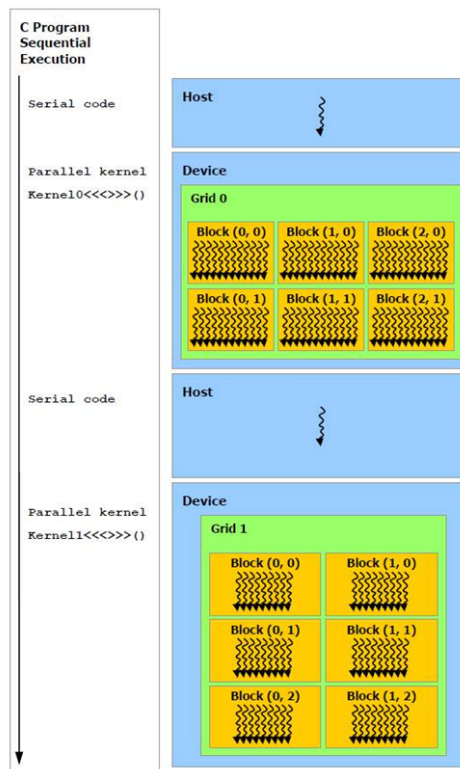


Figure 1.8: CUDA threads hierarchy and example of the execution flow of a CUDA program (picture from [28]).

consisting of one or more sequential threads running on a host CPU, and one or more parallel kernels suitable for execution on parallel computing GPU. A kernel executes a sequential program on a set of lightweight parallel threads. Threads are organized into blocks, that run on the same multiprocessor, can synchronize via barriers and communicate via a high-speed, per-block shared memory. The thread block is therefore the multiprocessor abstraction. Thread blocks are then organized into grid that executes a single CUDA program logically in parallel. Threads from different blocks in the same grid can synchronize via atomic operations and communicate via a global memory space shared by all threads. Sequentially dependent kernel grids can synchronize via global barriers and coordinate via global memory. CUDA requires that thread blocks be independent, which provides scalability to GPUs with different numbers of processor cores and threads. Thread blocks implement coarse-grained scalable data parallelism, while the lightweight threads comprising each thread block provide fine-grained data parallelism. Thread blocks executing different kernels implement coarse-grained task parallelism, while thread executing different paths implement fine-grained thread-level parallelism. Figure 1.8 shows the CUDA threads hierarchy, as well as a code mapping example that clarifies the execution flow of a CUDA program. Details on the CUDA programming model can be found in the CUDA programming guide [28].



## Chapter 2

---

# Ultrasound Tissue Characterization for Prostate Cancer Diagnosis

---

Prostate cancer (PCa) is one of the most frequently diagnosed neoplasia. Currently, the Trans Rectal Ultrasound (TRUS)-guided biopsy is the only diagnostic tool able to confirm the presence of PCa. Due to the limited diagnostic ability of TRUS the biopsy consists in a systematic sampling of those areas where cancer incidence is higher: accuracy grows with core number along with patient discomforts and risks probability. Computer Aided Detection (CAD) systems can be used to improve the current biopsy protocol providing radiologists a second opinion and allowing them to reduce the number of unnecessary biopsy cores without negative impinging on diagnosis accuracy.

The development of such tools requires a deep knowledge about anatomy and biology and medical procedures involved. Ultrasound Tissue Characterization (UTC), aimed at characterizing and differentiating between healthy and diseased tissue, is the more investigated methodology for implementing CAD tools for PCa detection. Since the problem of discriminating among different types of tissue can be viewed as a pattern recognition problem, all the UTC based CAD systems proposed in literature are based on machine learning techniques and their core is an automated classification system. Although encouraging, the results published so far are often not proven to be clinically significant and not always comparable, mainly due to the absence of a public images database and standard learning procedure. Furthermore these tools are usually not real-time and this prevents their applicability as support decision tools to perform lesion directed biopsy.

In this chapter some fundamentals information about prostate anatomy, pathologies and diagnosis procedures are reviewed first. Then, the problem of binary classification is briefly recalled and the main architecture of an UTC based CAD scheme is described. Each step of such a CAD processing flow is detailed, focusing on the approaches used

in this research to implement it. Finally, after a review of the existing methods for UTC-based computer aided diagnosis of PCa, a tool for real-time Computer Aided Biopsy (rtCAB) is presented. The rtCAB tool enhances TRUS video stream with a false color overlay image, representing a binary cancer risk map that suggests physician where to sample, thus reducing the number of biopsy cores. The CAD processing flow was implemented exploiting GPU acceleration, in order to perform the computer aided image analysis at the same time as the visualization of standard TRUS video sequence (at 25-50 fps).

A large and consistent TRUS video sequences database was collected to train and test the proposed tool. The tool was trained using a specifically designed learning procedure based on standard supervised-learning techniques. However, since in the learning process only the image portion corresponding to the biopsied area can be considered, and it is difficult to transform the physical labels in the extracted tissue into pixel labels, the classification of prostate tissues is an imperfect supervised problem. As a consequence, there is an intrinsic uncertainty in the ground truth and not all the collected diagnosed samples can be used for CAD system training. This problem will be discussed in the next chapter, where a novel mixed supervised/semi-supervised learning procedure, expressly designed to train the CAD system using the whole amount of collected data, is described.

## 2.1 Prostate anatomy and pathologies

The prostate is a compound tubuloalveolar exocrine gland of the male reproductive system in most mammals [29]. The main function of the prostate is to store and secrete a clear, slightly alkaline (pH 7.29) fluid that constitutes 10-30% of the volume of the seminal fluid that, along with spermatozoa, constitutes semen. The rest of the seminal fluid is produced by the two seminal vesicles [29].

A healthy adult human prostate is a chestnut shaped gland enveloped in a fibrous capsule. Its base is attached below the urinary bladder neck, and the apex is fixed to the urogenital urinary diaphragm. It borders with the rectum and with the fibromuscular stroma, and it is attached to the seminal vesicles. The excretory ducts of seminal vesicles open into the vas deferens, as they enter the prostate gland, and they are lined with the epithelium of the transition zone. Within the prostate, the urethra coming from the bladder is called the prostatic urethra and merges with the two ejaculatory ducts. The prostate is finally sheathed in the muscles of the pelvic floor (see Figure 2.1).

The prostate gland is divided in three different zones: the transition zone (TZ), the central zone (CZ), and the peripheral zone (PZ) (see Figure 2.2). The transition zone surrounds the urethra and extends from the ejaculatory ducts proximally. The peripheral zone encompasses the urethra from the base to the apex. The central zone is composed of tissue immediately surrounding the ejaculatory ducts and it expands inferiorly. The significance of this architecture is based upon the relationship of these three zones to



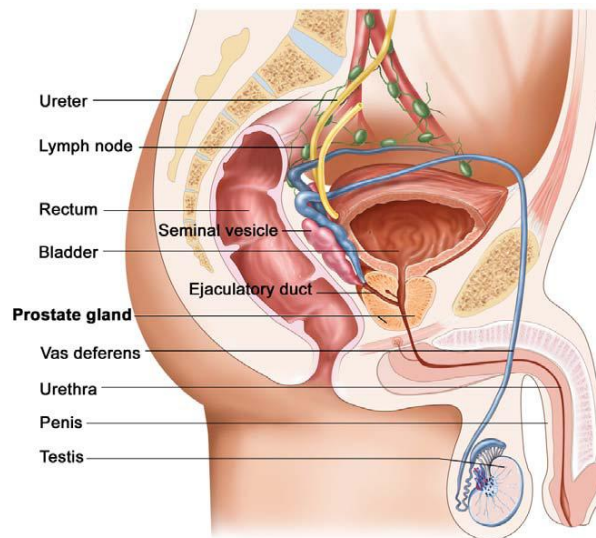


Figure 2.1: Prostate anatomy and its location in the body (picture from [30]).

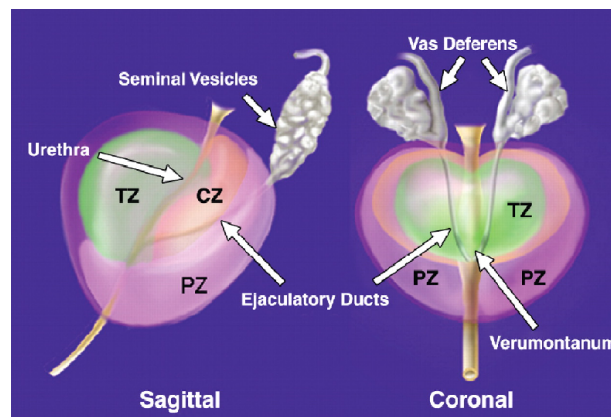


Figure 2.2: Prostate schematic illustration in sagittal and coronal view. PZ = peripheral zone, TZ = transition zone, CZ = central zone. (picture from [31]).

prostatic diseases [32]. In the young male the peripheral zone comprises 75% of prostate volume, the transition zone 20% and the central zone 5%. These ratios change with the age and after 40 years, the transition zone may enlarge and occupies most of the gland because of benign hyperplasia.

The prostate is the male organ most often smitten by diseases. Prostate diseases can be distinguished in three main categories: prostatitis, benign hyperplasia and prostate cancer. Prostatitis is an inflammation of prostate gland. It is a benign pathology and in the acute case is mainly treated with antibiotics. Benign hyperplasia (BPH) consists in prostate enlargement. It is fairly common among aging men and it occurs mainly in the transition zone. BPH can be treated with medication, a minimally invasive procedure or,

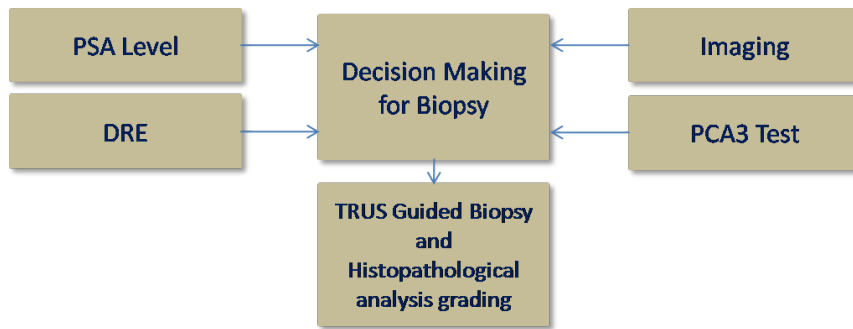


Figure 2.3: Diagnostic tools currently used in PCa detection.

in extreme cases, surgery that removes the prostate. Prostate cancer, is a malign pathology and is characterized by an abnormal and uncontrolled cells mutation and replication. If not detected early and in more aggressive forms, the disease can advance to stages characterized by local invasion of the surrounding tissues (seminal vesicles, bones, rectum), usually resulting in lethality. The heterogeneous and multifocal nature of PCa lesions make their detection difficult. In fact, prostate cancer tissue typically reveals a juxtaposition of benign cells, preneoplastic lesions and neoplastic lesions of varying severity [32], and multiple cancer lesions may emerge and evolve independently, posing significant difficulties in their detection. Typically 70% of prostate carcinoma arise in peripheral zone, 25% in the transition zone and 5% in the central zone [33].

PCa is the second most frequently diagnosed cancer and the sixth leading cause of cancer death in males worldwide [34]. As for other types of cancer, screening (*i.e.* testing to find a disease in people who do not have symptoms) can help to find PCa at an early stage when it is most easily curable in the hope that it can be treated more effectively. There is no doubt that screening can help spot many prostate cancers early, but there are limits to the PCa screening tests used today.

## 2.2 Prostate cancer diagnosis procedure

The current clinical procedure used to detect PCa is based on four diagnostic tools: Prostate Specific Antigen (PSA) dosimetry, Prostate CAncer gene 3 (PCA3) test, Digital Rectal Examination (DRE), and Trans-Rectal Ultrasound (TRUS) image analysis (see Figure 2.3). DRE is an internal examination of the rectum consisting in palpation of the organs adjacent to the rectum wall. Historically, it has been the principal method used in PCa screening but its reliability is limited by the fact that it allows to detect only large superficial lesions and it is strongly operator dependent. As a consequence, this test is usually combined with the measurement of the amount of PSA in the blood. PSA is a glycoprotein mainly produced in the epithelium of prostate gland. However, although high levels of PSA in the blood can be symptom of presence of PCa, also BPH, prostate infection, and urethral irritation can cause an increase in the PSA value. Therefore, this

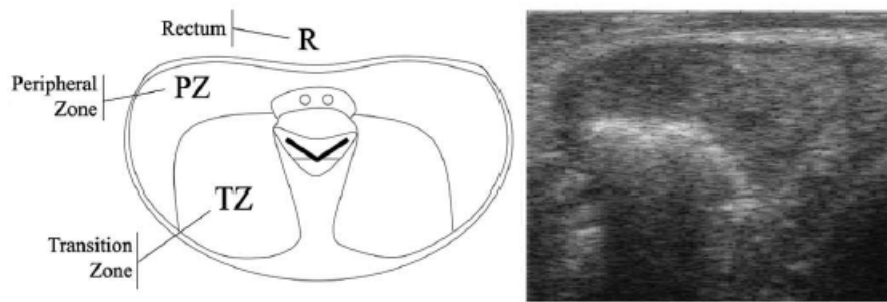


Figure 2.4: Schematic representation of prostate axial section with orientation inverted *w.r.t.* its real anatomy (left) and correspondent TRUS image (right).

measurement is not enough to diagnose the presence of PCa, and patients with abnormal DRE and high PSA values should be further evaluated for presence of PCa via prostate imaging.

Several types of imaging modalities can be used for PCa diagnosis and management but, as explained above, ultrasound imaging is usually the preferred modality. In particular, to image the prostate TRUS imaging is used. TRUS allows physicians to examine the prostate gland by inserting a trans-rectal ultrasound probe into the rectum and illuminating the gland with the ultrasound beam from the posterior side. Scanning usually begins within the axial plane at the base of the prostate. Seminal vesicles are visualized first, then prostate is imaged starting from the base to the apex. The normal prostate gland has a homogeneous, uniform echo pattern at TRUS. In fact, the anatomical separation between peripheral and central zones is generally not visible in echographic images and their normal echo pattern is used as reference for defining the other structures as hypoechoic or hyperechoic.

Standard TRUS images show the axial or lateral section of the prostatic gland. The gland anatomical shape is inverted in the image due to the acquisition setting as shown in Figure 2.4, where the superior hyperechoic boundary marks the interface between the rectum and the peripheral zone of the prostate. The transition zone is visible as a darker region below the peripheral zone. The gland contours are generally well defined and visible in the images. The parameters that drive the sonographic appearance of PCa are still uncertain and the appearance of carcinoma on ultrasound is variable, ranging from hypoechoic to hyperechoic. Although hyperechoic cancers are rare, at its early stages a tumor can appear anechoic, hypoechoic or isoechoic *w.r.t.* the surrounding normal tissues. Moreover, potential hypoechoic regions could also include BPH, or even normal biological structures, therefore TRUS diagnostic ability is low. Nevertheless TRUS allows an accurate measurement of prostate size, which is a useful information in determining the PSA density. As a consequence TRUS is better suited to guide prostate biopsy than for cancer staging.

Prostate biopsy is the only diagnostic tool able to confirm the presence of PCa. Uniquely among image-guided biopsies, prostate biopsy is not always lesion-directed but rather based on a systematic sampling of those areas where cancer incidence is higher. This is mainly due to the weak diagnostic significance of TRUS and the multifocality nature of the cancer, but also to the inherent inter- and intra-observer variability. All these facts concur to the necessity of performing many sampling to obtain reliable results. The number of tissue samples removed during a biopsy examination varies according to prostate volume, patient compliance, selected biopsy protocol, and in case of biopsy or re-biopsy [35]. From 6 (sextant protocol) [36] to 24 (saturation protocol) cores are removed. A pathologist then checks for cancer cells. The histopathological analysis of biopsy cores is the standard for cancer detection confirmation. The dominant and most reliable method for prostate carcinoma diagnosis and aggressiveness assessment in research as well as in clinical practice is the *Gleason grading*. This method consists in the categorization of glandular differentiation and growth pattern of the tumor in five basic grades patterns [37] at relatively low magnification (x10-x40). Due to the heterogeneous nature of prostate cancer lesions, histopathological inspection of prostate tissues often reveals an ensemble of benign lesions, preneoplastic lesions and neoplastic lesions with different aggressiveness. To take into account this heterogeneity, the five basic grade patterns are used to generate an histological score, which can range from 2 to 10, by adding the primary grade pattern to the secondary one. The primary pattern is the one that is predominant in area by simple visual inspection. The secondary is the second most common pattern. The Gleason grading score is therefore an indicator of cancer stage: the higher is the grade, the more advanced is the cancer. Typically, cancers with Gleason scores lower than 6 are considered well differentiated and associated with a good prognosis. Those with a Gleason score of 8-10 have the worst prognosis and the highest risk of recurrence. Gleason score is often combined with PSA level and clinical stage for risk assessment. The primary goal of staging is to distinguish patients with organ-confined, locally invasive, or metastatic diseases.

Sometimes, after a negative biopsy, doubts can remain on the patient healthy status. PCA3 gene test is used to predict the presence of malignancy in men undergoing repeated biopsies. In fact PCA3 is a prostate specific gene that expresses a non coding RNA and it is highly over-expressed in prostate cancer. It has been shown that it could be useful as a tumor marker for patients for whom DRE and PSA suggest possible PCa but the first prostate biopsy returns a normal result. This occurs in approximately 60% of cases and, on repeated testing, 20-40% have an abnormal biopsy result. PCA3 however does not provide a definitive answer as to whether a man has cancer or not. A positive biopsy remains the gold standard in diagnosing PCa. Therefore after a negative biopsy patients are usually subjected to repeated biopsies so that the likelihood of missing cancer diminishes with each sampling.

Prostate biopsy is not completely safe for the patient, since it carries the risk of bleeding infection or other side effects [38]. The complications encountered performing

biopsy increase in case of repeated or systematic sampling protocols involving the extraction of many cores. Furthermore, the reliability of the sampling process is still matter of debate. The first and the most widely used sampling protocol is the *sextant protocol* but it is known that it can miss some tumors because it undersamples the peripheral zone [35]. Therefore, several variations to the sextant protocol have been proposed to increase its detection rate. First attempts were aimed at modifying the biopsies trajectories to better sampling the peripheral zone, while later approaches converged to extend the sextant protocol extracting more cores (8,10,12). Although extended protocols can improve diagnosis accuracy, it is still matter of debate whether they are substantially better of the modified sextant protocol, and how many cores should be extracted. As a trade-off between the achievable diagnosis accuracy and the probability of adverse events, standard biopsy protocol currently consists in the random sampling of 8-12 cores depending on the gland volume.

Research efforts put into improving prostate biopsy protocol regarded both the image processing techniques and the clinical procedure. These efforts were specifically aimed at improving the so called Positive Predictive Value (PPV). The PPV is the probability that a malign classified sample is actually pathological. Thus PPV is the performance indicator used in clinical environment to express the prediction capability of a method. In particular, the PPV of the standard biopsy protocol is the probability of sampling a pathological core given a pathological patient. It is usually quite low, between 20% and 25%. Let  $p_0$  be the standard biopsy protocol PPV, the probability of correctly diagnosing PCa with  $N_0$  cores can be expressed as:

$$\left[ 1 - (1 - p_0)^{N_0} \right] \quad (2.1)$$

therefore the detection rate of the protocol, considering the extraction of 8 cores, is about 89%. A detection rate lower than 100% means that the protocol is quite inefficient. This is mainly due to the fact that most of the sampled tissue is benign and thus represent unnecessary biopsy. Therefore, to improve the efficiency of the standard biopsy protocol, it is mandatory to increase the PPV, so that the number of sampled cores can be reduced while maintaining the same detection rate. In fact, if  $(p_1 > p_0)$  is the PPV of a new detection tool, using this tool the number of cores can be reduced according to:

$$N_1 = N_0 \left[ \frac{\log(1 - p_0)}{\log(1 - p_1)} \right]. \quad (2.2)$$

In this scenario CAD systems can be used to improve standard biopsy protocol efficiency. In particular CAD can provide physicians a risk map to guide the biopsy needle insertion into the suspicious regions so that a lesion directed biopsy can be performed and the number of unnecessary biopsy cores can be reduced.

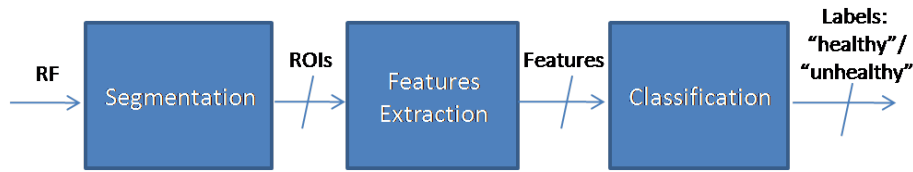


Figure 2.5: Common scheme of UTC-based CAD systems architecture.

## 2.3 CAD tools for PCa detection

The shortcomings of the current clinical methods have stimulated a great deal of studies aimed at improving the detection rate of prostate cancer by computer aided techniques. Ultrasound Tissue Characterization (UTC) is the more investigated methodology for implementing CAD tools for PCa detection. As stated above, the ultimate goal of UTC is to provide a *parametric image* (spatial map) of tissue properties. In fact, the ultrasound echo signal contains valuable information about the physical properties of the tissue that, in general, can be estimated analyzing both the RF signal before the envelope detection and the B-Mode image. CAD systems based on UTC combine information extracted from the input ultrasound echo signal, before or after the envelope detection, and machine learning techniques to classify tissue, providing its characterization. The results of this characterization can be displayed to the physician, together with the conventional B-Mode image, to be used as a biopsy guidance. Usually a binary classification of tissue is considered, *i.e.* imaged tissue are classified as normal (benign, not-suspicious) and pathological (malign, suspicious). In the next subsections the basic concepts of binary classification will be recalled to better understand how UTC based CAD systems work and how they can be implemented. These systems are based on a common scheme shown in Figure 2.5. Each block of this architecture will be described and its function explained.

### 2.3.1 The binary classification problem

Classification is an example of the more general problem of pattern recognition which is the assignment of some sort of output value to a given input value. We can define the problem of binary classification as the problem of assigning unknown objects (patterns) to one of two known categories. The first step is to identify measurable quantities that make these patterns distinct from each other. These measurements are known as *features*. In more general cases  $n$  features  $x_{i,i=1,2,\dots,n}$  are used and they form the feature vector  $x = [x_1, x_2, \dots, x_n]$ . Each of the feature vectors identifies uniquely a single pattern. The vector space associated with these vectors is often called *features space*. A classifier can be defined as the *separating hyperplane* that partitions the features space into regions that correspond to one of the two classes. An example, is shown in Figure 2.6. The straight line is known as *decision line*, and it constitutes the classifier.

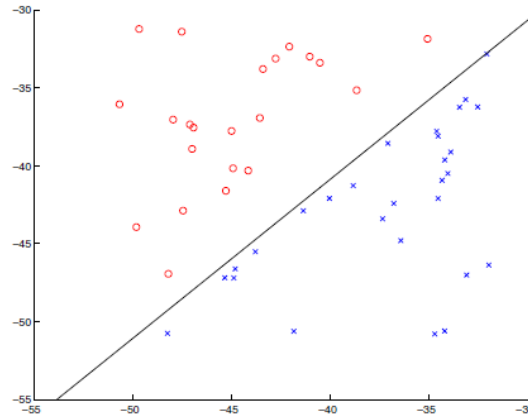


Figure 2.6: An example of separating hyperplane in the two-dimensional case.

This approach is known as *geometrical approach*, which is the most frequently used in many classification tasks as well as in this thesis: well known classifiers such as Support Vector Machines (SVM) [39, 40], Fisher Linear Discriminant (FLD) [39, 40], Kernel Fisher Discriminant (KFD) [41], Neural-Networks [40] are based on this approach. Other approaches to the classifier design are the *probability approach* where the optimal Bayes decision rule assigns patterns to the class with the maximum a posteriori probability (Naive Bayes Classifier [40]), and the *similarity based approach* according to which similar patterns belong to the same class (Template Matching [40]).

In order to be able to correctly classify unknown patterns a classifier training is required. Usually, in classification problems some patterns whose true classification is known are available and are used to train the classifier. These patterns (data) are known as *training patterns (data)*. The training data are constituted by couples  $(x_i, y_i)$  with  $i = 1, 2, \dots, l$  where  $x_i$  is the feature vector,  $y$  is the label identifying the class to which a datum belong to, and  $l$  is the number of samples in the training set. Basically, the classifier training consists in finding the best separating hyperplane of the input data which is defined by the equation:

$$\langle \mathbf{w} \cdot \mathbf{x} \rangle + b = 0 \quad (2.3)$$

where  $\mathbf{w}$  is the weight vector, defining a direction normal to the separating hyperplane,  $\mathbf{x}$  is the feature vector and  $b$  is the offset (also called threshold or bias). Mathematically, this corresponds to find the *discriminant function*  $f : X \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$  :

$$f(\mathbf{x}) = \langle \mathbf{w} \cdot \mathbf{x} \rangle + b. \quad (2.4)$$

This function characterizes a linear classifier. By using the *dual representation* the linear discriminant function (2.4) can also be written as:

$$f(\mathbf{x}) = \sum_{i=1}^l \alpha_i y_i \langle \mathbf{x}_i, \mathbf{x} \rangle + b. \quad (2.5)$$

where  $\mathbf{w} = \sum_{i=1}^l \alpha_i y_i \mathbf{x}_i$ , *i.e.*  $\mathbf{w}$  is considered a linear combination of the training data [42]. The advantage of this representation is that hypothesis can be expressed as a linear combination of the training points so that the decision rule can be evaluated using just inner products between test points and training points.

In real applications problems for which a linear classifier can result in acceptable performance are not the rule. In general the surfaces dividing the space in various class regions are non linear. What type of nonlinearity must be adopted and what type of optimizing criterion must be used in order to locate a surface in the right place of the feature space are all classifier design stage matters. Furthermore, we assumed that a training set was available and the classifier was designed by exploiting this *a priori* known information. This is known as a *supervised pattern recognition*, and the approach used to train the classifier is known as *supervised learning*. Other approaches are the *unsupervised learning* and the *semi-supervised learning*. They are used when class labels for the training data are respectively not available, or partially available.

Other problems encountered in the design of a real classification system include: which kind of features must be used to highlight the data membership to a specific class, how to generate them, and find the best set of features to use. These problems respectively concern the so called *features generation* and *features selection* stage. The goal of the features generation stage is to generate features that exhibit high information-packing properties from the class separability point of view. The goal of the feature selection stage, instead, is to find the best number of features to use. In practice, in many applications, a larger than necessary numbers of feature candidates is generated and the best of them should be adopted. The best feature set is that one maximizing the predictive ability. This procedure is aimed at avoiding the so called *curse of dimensionality* phenomenon according to which, with a fixed number of training samples, the predictive power reduces as the dimensionality increases. In fact, not all the features extracted have the same relevance and can bring redundant information. Furthermore, high dimensionality of the feature set causes more complex classification models which tend to overfit the training data. For this reason it is essential to select few significant and non-redundant features. Feature selection algorithms [43] focus on the realization of the optimal feature subset for a certain problem and can be roughly divided into three categories: filters, wrappers and hybrid methods. Filters methods sort all the available features based on their ability to represent a specific class. The *ranking* criterion can measure the distance, dependency, or consistence between features extracted from each pattern and the respective class. Wrappers sort attributes from a features set according to the performance of a classifier trained on that feature set. Hybrid methods take advantage both of filter and wrapper methods. The filter measure is used to decide the best subset for a given cardinality, while the wrapper mining algorithm selects the best



cardinality, *i.e.* the final best subset among the subsets selected for each cardinality. Forward and backward sequential search or random search techniques are used instead of a complete search procedure to reduce the computational cost of filtering procedure.

### 2.3.2 UTC based CAD systems processing flow

As stated above, typical CAD systems for PCa detection combine information from the acquired RF echo signal and machine learning techniques to provide tissue characterization. The characterization results can be displayed to physicians with conventional B-Mode image to guide the biopsy procedure. All the methods presented in literature are based on a common scheme which comprises: (i) the image partitioning into regions of interest (ROIs); (ii) the extraction of some features from each ROI; (iii) the classification of each ROI as healthy or unhealthy.

#### Segmentation

In this context segmentation consists in dividing the image into several regions of interest. ROIs can be defined as irregularly shaped areas in the image, or rectangular regions covering the whole image with or without overlapping. Segmentation into irregular ROIs is usually performed following some anatomical characteristics of the imaged tissue. As a consequence it has the advantage to give informations about tissue characteristics. On the other hand it is difficult to perform it automatically and usually requires a large amount of user supervision. The image subdivision into regular ROIs, instead, is easier to perform and it is usually a complete automatic procedure. Neither human intervention, nor complex segmentation algorithms are required to perform this kind of segmentation. However, it does not provide morphological tissue informations. The ROI consists of a two-dimensional arrangement of recorded echo signals, which are used to compute features useful to classify the set of outlined tissue. Considering each ROI as a point in an  $n$ -dimensional feature space identified by the feature vector  $\mathbf{x} = [x_1, \dots, x_n]$ , it can be assumed that ROIs with similar characteristics will form clusters in the feature space. In order to perform a completely automated image analysis, the segmentation into regular ROIs has been adopted in this thesis.

#### Feature Estimation

This step consists in the selection of significant features which are able to capture the characteristics of normal and pathological tissues, ideally providing a representation of different tissues in the feature space as separate clusters. In UTC useful features can be estimated from the RF signal both before and after the envelope detection. A large number of features have been proposed to extract interesting information from the ultrasound echo signal. According to the nature of the measured quantities *w.r.t.* the tissue properties they can be divided into three categories:

- **Spectral features** describe variations of physical properties making a backscattering signal that provides information about the spectral behavior of the RF signal. Example of these features are the wavelet coefficients of the RF signal, their polynomial fitting [44], the estimate of the mean central frequency of the RF signal, and the coherent and diffuse component obtained by signal decomposition [45].
- **Statistical features** are represented by the parameters of the statistical distributions modeling the RF signal and its envelope which provide information about scatterers density, regularity and amplitude. Examples of statistical distributions used for the envelope signal modeling are the Nakagami distribution [46] and the Rayleigh distribution [47], while the Generalized Gaussian distribution [48] is used for the RF signal modeling.
- **Textural features** give information about the arrangement of gray levels in the B-Mode image, that corresponds to the image texture. Their usefulness for tissue characterization purpose is related to the close correlation between image texture, tissue microstructures, and related phenomena. Different kinds of textural parameters have been proposed in literature: Haralick [49] and Unser [50] features are both based on the gray levels distribution statistics, while Fractal features [51] rely on modeling and computation of the fractal dimension.

Two kinds of feature were computed the envelope signal to characterize different tissue aspects in this thesis. Shape and scale parameters of the Nakagami distribution and Unser textural features.

**Nakagami features:** this group of features is derived from a statistical model of the envelope signal based on the Nakagami distribution [46]. Nakagami probability density function (pdf) writes as:

$$\rho(\rho) = \frac{2m^m \rho^{2m-1}}{\Gamma(m)\Omega^m} \exp\left(-\frac{m}{\Omega}\rho^2\right) \quad (2.6)$$

where  $\rho$  is the real envelope of the RF signal. The shape parameter  $m$  and the scaling factor  $\Omega$  are estimated on the signal envelope. The Nakagami distribution can give a good description of signal envelope statistics and is able to well capture different scattering structures. As discussed in [52], the most discriminant feature is the shape parameter, in fact, different values of  $m$  corresponds to different scattering types. Also the scaling factor is important: by conveying information about the local scatter energy, its logarithm is strongly related to the log-compressed image usually visualized by physicians. The Nakagami parameter  $m$  can be obtained from the moments of the envelope and can be expressed as:

$$m = \frac{[\mathbb{E}(\rho^2)]^2}{\mathbb{E}[\rho^2 - \mathbb{E}(\rho^2)]^2} \quad (2.7)$$

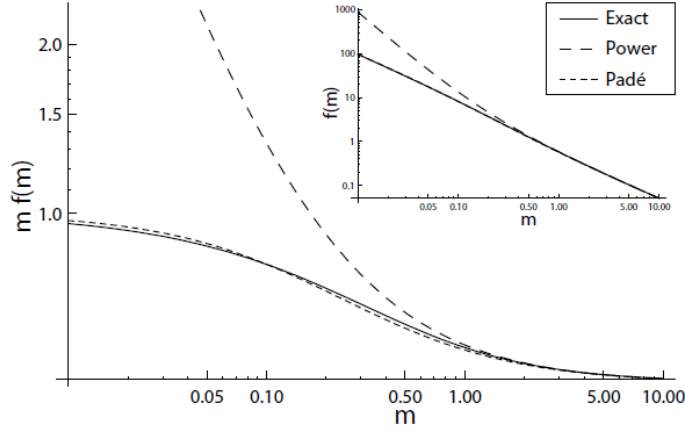


Figure 2.7: Comparison of the power series approximation in the ML Nakagami estimator and the corresponding Padé Approximant.

where  $\mathbb{E}()$  stands for expectation. The scaling factor  $\Omega$  can be obtained as:

$$\Omega = \mathbb{E}(\rho^2) \quad (2.8)$$

It is important to note that while estimating  $\Omega$  is not computational intensive, the *Maximum Likelihood* estimation of  $m$  requires solving the nonlinear equation

$$f(m) = \phi(m) - \ln(m) = \Delta \quad (2.9)$$

where  $\phi(m)$  is the digamma function and  $\Delta$  is a quantity depending on data. As no closed form solution is known for this equation, authors in [53] suggest to asymptotically expand  $f(m)$  by means of a power series with strictly negative exponents. However a much better approximation can be obtained by means of Padé Approximants [54]. In fact, substituting  $f(m)$  with its approximant of order (1,2), a rational function in the form

$$\hat{f}(m) = \frac{p_0 + p_1 m}{q_0 + q_1 m + q_2 m^2} \quad (2.10)$$

can be used to obtain a closed ML estimator [55]. Parameters  $p_i$  and  $q_i$  were obtained by applying numerical optimization techniques. As shown in Figure 2.7, the proposed solution is much more accurate, especially for low values of  $m$ .

**Unser features:** these features are statistical attributes generated from histograms of the sum and difference of gray levels in the ROIs. The texture analysis method proposed by Unser [50] is an alternative to the classical Spatial Grey Level Dependence Method (SGDLM) based on co-occurrence matrix and it allows to reduce computational and memory requirements of this method. Basically, the Unser's method replaces the co-occurrence matrix with estimates of the first order probability functions along its principal axes, namely the sum and difference histograms. In particular, the method reckons that sum and difference define the principal axes of the second-order probability

function of a stationary process, and as a consequence, the co-occurrence matrix used for texture description can be replaced by its associated sum and difference histograms which can be directly estimated from the image.

Let  $Y(k, l)$  be a discrete texture image defined on a  $K \times L$  rectangular grid and considered as a realization of a bidimensional stationary and ergodic process. Let  $D$  be a subset of indices specifying a texture region to be analyzed. The sum and difference histogram are defined as [50]:

$$h_s(i, d_1, d_2) = h_s(i) = \text{Card} \{(k, l) \in D, s_{k,l} = i\}, i = \{2, \dots, 2N_g\} \quad (2.11)$$

$$h_d(j, d_1, d_2) = h_d(i) = \text{Card} \{(k, l) \in D, d_{k,l} = j\}, j = \{-N_{g+1}, \dots, N_{g-1}\} \quad (2.12)$$

where  $N_g$  is the number of quantized image gray levels and  $s_{k,l} = (y_{k,l} + y_{k+d_1, l+d_2})$  and  $d_{k,l} = (y_{k,l} - y_{k+d_1, l+d_2})$  are, respectively, the non-normalized sum and difference between two pixels distant one each other  $d_1$  in the horizontal direction, and  $d_2$  in the vertical direction. The relative displacement  $(d_1, d_2)$  can also be characterized by a distance  $d$  in radial units, and an angle  $\Theta$  w.r.t. the horizontal direction [50]. The normalized sum and differences histograms are estimates of the sum and difference probability functions. As discussed in [50], nine statistical parameters can be computed from sum and difference histogram: mean, variance, contrast, homogeneity, cluster shade, cluster prominence, energy, correlation and entropy. These measures can be replicated on all the four angular directions ( $0^\circ, 45^\circ, 90^\circ, 135^\circ$ ), giving a total of 36 attributes.

Using these features, two different classification approaches will be shown. In the first approach UTC is based only on the extraction of Nakagami distribution parameters, while the second approach we propose is a multifeature approach, where both Nakagami parameters and Unser features are estimated from the echo signal. In fact, recent studies [56] have demonstrated that combining features extracted from the RF signal and textural parameters enhances classification performance. Combining these features, a total of 38 parameters is available. In this case, to reduce the burden of features estimation while maintaining a high significance of the chosen features set, a hybrid feature selection algorithm was applied. Hybrid feature selection algorithms combine filter and wrapper models to select the best features set. As in filter models an independent measure is used to decide the best subset for a given cardinality, then, as in wrapper methods, a classification algorithm is used to select the overall best subset among the best subsets across different cardinalities. In particular, in this thesis, a Mutual Information Hybrid Feature Selection (MIHFS) algorithm [57] was used to rank and prune the whole available feature set. In this algorithm, the chosen ranking measure is the min-Redundant Max-Relevance (mRMR) criterion proposed in [58]. The mRMR measure is based on mutual information between the current feature set and class corrected with the averaged mutual information between features in the feature set. Maximizing this measure yields to define a feature set  $S$  with maximum relevance with respect to a class  $c$

$$D(S, c) = \frac{1}{|S|} \sum_{Y_i \in S} I(Y_i, c) \quad (2.13)$$

Index	Feature Name
01	Unser homogeneity 90°
02	Unser homogeneity 0°
03	Unser contrast 45°
04	Nakagami m
05	Unser energy 135°
06	Unser contrast 90°
07	Unser cluster shade 135°
08	Unser entropy 0°
09	Unser entropy 90°

Table 2.1: Selected and ranked feature set obtained applying the MIHFS algorithm to the whole Nakagami-Unser feature set.

and minimum redundancy

$$R(S) = \frac{1}{|S|^2} \sum_{Y_i, Y_j \in S} I(Y_i, Y_j) \quad (2.14)$$

where  $Y_i$  is a feature and  $I(X, Y)$  is the mutual information of two discrete random variables  $X, Y$ , representing two features, or a feature and a class. In practice, the first step of the algorithm consists in finding the subset  $S$  that maximizes the mRMR measure defined as:

$$\Phi(D, R) = D - R \quad (2.15)$$

The MIHFS works into two steps. In the first step (filter) all features are ranked according to the mRMR measure following a sequential forward selection as search technique. In the second step (wrapper) MIHFS, exploiting a FLD classifier, evaluates the mining performance of the ranked features set for increasing set size. At different cardinalities, the subset which maximizes the mRMR measure is selected and the performances of FLD trained on this subset are computed. The best cardinality and consequently the best subset is chosen as the one performing the minimum FLD misclassification error. Typically the best cardinality is smaller than the maximum number of features because not all of them have the same discriminant power. For the selected subset the MIHFS algorithm produces a ranked list of features which highlights their predictive skill.

The selected and ranked set obtained applying the MIHFS algorithm to the whole Nakagami-Unser feature set is constituted of 10 attributes and is shown in Table 2.1.

### Classification

This step consists in classifying selected ROIs into one of few known categories representing a particular state of the tissue. As stated above, only two classes, respectively suspicious and not-suspicious, were considered in this research.

In general, such a classification problem is a supervised problem. In fact, usually a data set  $(x_k, y_k)$  of ROIs feature vector and label pairs is available, and is used to

determine the predictive model representing the relationship  $y = f(x)$ . The found predictive model is then used to separate in the feature space suspicious and not-suspicious ROIs. Several studies reported that features of different nature hardly are linearly inter-correlated, and that a non linear classification model can be able to extract valuable information from a mixed feature set and to reach higher level of accuracy [56]. On the other hand, complex non linear classification models fail in preserving physical significance of features and their true dependence from the pathology. Moreover, linear classifiers work very well when the speed of classification is an issue, and the number of features is large.

An efficient technique to obtain non linear classification model maintaining the advantages of the linear classifiers, is the so called *kernel trick* [59]. This technique is based on the application of a non linear function  $\Phi$  mapping the data into a different higher dimensional feature space. Such higher dimension in the novel feature space produces an increased separability. The kernel trick recognizes that the classification in this new space can be obtained without actually performing the transformation, because in the predictive function  $\Phi$  only appears in the form of an inner product. Therefore non linear approaches can be performed efficiently by using kernel functions  $K(\cdot, \cdot)$  which act like dot products in the re-mapped feature space [59]. It is not necessary to compute  $\Phi$  or to define it explicitly. Instead it is sufficient to define the kernel function  $K(\cdot, \cdot)$ . It can be shown that any symmetric positive semidefinite function suffices. To summarize a kernel function is a function that for each  $x, z \in X$ :

$$K(x, z) = \langle \Phi(x) \cdot \Phi(z) \rangle \quad (2.16)$$

By using the dual representation (2.5), when the kernel function is known, the discriminant function for a training data set of size  $l$  can be expressed as:

$$f(x) = \sum_{i=1}^l \alpha_i y_i K(x_i, x) + b \quad (2.17)$$

Commonly used kernel functions are:

- polynomial function kernel:  $K(x, z) = (x^T z + c)^d$ ;
- radial basis function (RBF) or Gaussian kernel:  $\exp(-\frac{\|x-z\|^2}{2\sigma^2})$ ;
- sigmoidal function kernel:  $K(x, z) = \tanh(a_1 x^T z + a_2)$ ;

Usually kernel functions have several parameters that have to be tuned to optimize the classification model performance: cross-validation on a validation data set is usually employed for this parameter tuning. Cross-validation is one of the techniques for statistical resampling used to evaluate performance and improve robustness of machine learning models and to estimate statistical significance levels. It allows to obtain reliable

classification models and efficient parameter tuning but it is usually very time consuming.

In this thesis four different non linear classification models obtained using the kernel trick technique were employed: a Fisher Linear Discriminant(FLD) [39, 40] combined with a quadratic data mapping (QFLD), a FLD combined with a non linear feature extraction stage based on Generalized Discriminant Analysis (GDA+FLD), and a Support Vector Machine (SVM) [39, 40] and a Kernel Fisher Discriminant (KFD) [41] both based on Gaussian kernels.

**SVM classifier:** its basic principle is finding the best separating hyperplane as the one that has the largest distance to nearest training data points of any class. This distance is called the function margin and, in general, the largest the margin is, the lower the general error of the classifier is. Finding the best hyperplane with the SVM means solving a constrained minimization problem which can be solved with the Lagrangian multipliers. The discriminative function in terms of support vectors,  $sv$ , is expressed as follows:

$$f(x) = \sum_{i \in sv} \alpha_i c_i \langle x_i, x \rangle + b \quad (2.18)$$

where the summation includes only the training examples  $x_i$  that are support vectors, and  $\alpha_i$  are coefficients determined as Lagrange multipliers in the optimization procedure. In its original formulation SVM was a linear classifier. In 1992, Boser, Guyon and Vapnik suggested a way to create non-linear classifiers by applying the kernel trick [60] to maximum-margin hyperplanes. The resulting algorithm is formally similar, except that every dot product is replaced by a non-linear kernel. An advantage of the SVM classifier is that it concentrates directly on samples  $sv$  and, as a consequence, the hyperplane can be derived from a number of data smaller than the total number of training data. In addition this approach is shown to balance training classification error and model complexity, thereby avoiding overfitting, a situation in which the model is so finely tuned to the training samples that fails to perform well on new data. This classification algorithm is time consuming.

**FLD and related non linear classification models: GDA+FLD and QFLD** The FLD is one of the most widely used and simple linear classification algorithms. Let  $C_i$  be a class of  $N_i$  points  $x_{n,n=1,\dots,N}$  with  $N$  the total number of points to be classified. Basically, the FLD is given by the vector  $w$  which maximizes:

$$J(w) = \frac{w^T S_B w}{w^T S_w w} \quad (2.19)$$

where:

$$S_B = (m_1 - m_2)(m_1 - m_2)^T \quad (2.20)$$

and

$$S_W = \sum_{i=1,2} \sum_{n \in C_i} (x^n - m_i)(x^n - m_i)^T \quad (2.21)$$

are the between and within class scatter matrices respectively and  $m_i$  is defined by  $m_i = \frac{1}{N_i} \sum_{n \in C_i} x(n)$ . The intuition behind maximizing  $J(w)$  is to find a direction which maximizes the projected class means (the numerator) while minimizing the classes variance in this direction (the denominator). The bias  $b$  (also called threshold) of the linear rule (2.3) must be determined based on another principle: for example  $b$  can be chosen as the one satisfying the equality:

$$\langle w \cdot m_1 \rangle + b = -(\langle w \cdot m_2 \rangle + b) \quad (2.22)$$

Combining this simple linear classifier with a non-linear feature extraction stage, an efficient non-linear classification model can be obtained.

Feature extraction techniques consist in realizing linear or non-linear combination of original data to enhance the class separability. In other words, the feature extraction step consists in mapping the input vector of observations  $x \in \mathbb{R}^n$  onto a new feature description  $z \in \mathbb{R}^m$  which is more suitable for given task.

The first feature extraction technique we propose to use is a quadratic data mapping that projects the input vector  $x \in \mathbb{R}^n$  onto the feature space  $\mathbb{R}^m$  where  $m = n \frac{(n+3)}{2}$ . In the case of the quadratic data mapping used as the feature extraction step, the discriminant function can be written in the following form:

$$f(x) = \langle x \cdot Ax \rangle + \langle x \cdot b \rangle + c. \quad (2.23)$$

The classification algorithm defined by equation (2.23) is known as QFLD.

The second feature extraction method used in this thesis is the Generalized Discriminant Analysis (GDA) [61]. It applies kernel transformations to map the input data into a new features space, then the resulting GDA features are projected in the direction which maximizes the between-class distance and minimizes the within-class distance of samples of the mapped training set. In this case the discriminant function can be written as:

$$f(x) = \langle \alpha \cdot k(x) \rangle + b. \quad (2.24)$$

where  $k(x)$  is a vector of kernel evaluations. Note that threshold  $b$  in both (2.23) and (2.24) can be chosen as it is done for the classical FLD, for example using (2.22).

### **KFD classifier**

The Kernel Fisher Discriminant is the non-linear extension of the linear FLD. The input training vectors are assumed to be mapped into a new feature space  $F$  by a mapping function  $\Phi$  by means of the kernel trick, and the ordinary FLD is applied on the mapped



training data. The aim is to find a direction  $\psi = \sum_{i=1}^l \alpha_i \phi(x_i)$  in the feature space  $F$  given by the weights  $\alpha = [\alpha_1, \dots, \alpha_l]$  such that the criterion:

$$\frac{\langle \alpha \cdot M\alpha \rangle}{\langle \alpha \cdot (N + \mu E)\alpha \rangle} = \frac{\langle \alpha \cdot m \rangle^2}{\langle \alpha \cdot (N + \mu E)\alpha \rangle} \quad (2.25)$$

is maximized. This criterion is the feature space counterpart of the criterion (2.19) defined in the input space. The vector  $m$ , matrices  $N$ , and  $M$  are defined as:

- $m = m_1 - m_2$ ;
- $M = (m_1 - m_2)(m_1 - m_2)^T$ ;
- $m_y = \frac{1}{|I_y|} K 1_y, y \in Y$ ;
- $N = K K^T \sum_{y \in Y} |I_y| m_y m_y^T$ ;

where the vector  $1_y$  has entries  $i \in I_y$  equal to 1 and remaining entries zeros. The kernel matrix  $K[l \times l]$  contains kernel evaluations  $K_{ij} = K(x_i, x_j)$ . The diagonal matrix  $\mu E$  is a regularization term. The discriminant function  $f(x)$  is given once found a vector  $\alpha$  as:

$$f(x) = \langle \alpha \cdot k(x) \rangle + b \quad (2.26)$$

where the  $k(x) = [k(x_1, x), \dots, k(x_l, x)]^T$  is a vector of kernel evaluations. The discriminant function (2.26) is known up to the bias  $b$  which is determined by the linear SVM with  $L_1$ -soft margin applied on the projected data obtained using the kernel [62].

Beside the selection of a well suited classification algorithm a ground truth and an appropriate learning methods are essential for designing an efficient and reliable classification model. Furthermore, appropriate metrics are required to asses CAD diagnostic reliability, based on both statistical and medical evaluation criteria.

Practical software implementations of these classifiers were obtained in Matlab<sup>TM</sup> environment using the Statistical Pattern Recognition Toolbox [62].

### Ground Truth and Learning Methods

A ground truth dataset is necessary for designing and validating a classification algorithm. The ground truth data for a UTC based CAD system is constituted by ultrasound images from biological tissues and information about them coming from medical examination. Ground truth data are divided into a training set and a testing set. The training set consists of a set of ROI-label pairs, already classified by a chosen gold standard like histopathological analysis, which are used to perform the classifier training. The testing set, instead, is used to validate the classification algorithm, and evaluate the CAD system performance. Test ROIs labels are used only to compute the performance criteria and do not provide any further information to the classifier. In some cases also a validation set is extracted from the ground truth in order to tune some parameters of the classification

algorithm.

Therefore, training CAD systems achieving good performance requires large amount of samples collected during medical examinations along with the corresponding diagnoses made by expert physicians. Collecting large amounts of labeled data is difficult, expensive and time consuming, as it requires the effort of experienced personnel. As a consequence, even for very common and highly diagnosed diseases as prostate carcinoma or breast tumor there do not exist public standard databases. Furthermore, for prostate cancer, not all the collected diagnosed samples can be used to extract labeled ground data. In fact, the histopathological analysis of tissue sample extracted during biopsy does not provide the distribution of tumor in each core. Therefore there is an intrinsic uncertainty in the ground truth dataset that can be obtained from these samples. These problems are strong limitations for the development of reliable CAD tools. Two main contributions of this research are aimed at overcoming these limitations. One is the realization of a large and consistent TRUS video sequences database that constitute the ground truth for developing a diagnosis support tool for PCa detection. This database was collected in collaboration with the department of Urology of S.Orsola Hospital in Bologna. The second contribution is the design of Semi-Supervised Learning (SSL) algorithms for CAD system training. In fact, using a SSL approach in place of supervised one it is possible to train the CAD system using both labeled and unlabeled data, consequently reducing the number of collected diagnosed samples necessary to design a reliable support tool and using the whole amount of collected data. While the database realization is described in this chapter, the development of an innovative SSL algorithm and of a new learning procedure for CAD system training will be described and discussed in the next chapter.

### Performance Evaluation Metrics

To assess the CAD diagnostic reliability it is possible to use the classical statistical measures used to evaluate the performances of a binary classification test. Such measures are the well known Sensitivity (SE) and Specificity (SP) and the consequently derived Accuracy (Acc) and Positive Predictive Value (PPV). SE measures the portion of actual positives which are correctly identified as such. SP measures the proportion of negatives which are correctly identified. Acc is a measure of how well a binary classification test correctly identifies or excludes a condition. On the other hand, PPV is defined as the portion of the true positives against all the positive results (both true positives and false positives). They are defined as:

$$SE = \frac{TP}{TP+FN} \quad (2.27)$$

$$SP = \frac{TN}{TN+FP} \quad (2.28)$$

$$Acc = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.29)$$

$$PPV = \frac{TP}{TP+FP} \quad (2.30)$$

where TP, FP, TN and FN are, respectively, the number of True Positive, False Positive, True Negative and False Negative samples.

In a binary UTC problem, where the two categories to be classified are suspicious and not-suspicious tissue, it is common to define TP the pathological samples classified as suspicious, FP the misclassified normal regions, TN the normal samples classified as not-suspicious, and FN the misclassified pathological regions. According to these definitions, in medical diagnostics, SE and SP provide informations about the diagnostic tool ability in recognizing pathological and normal tissue respectively. The diagnostic ability of the method is also expressed by the probability that a malign classified sample is actually pathological, provided by the PPV, and the probability that both pathological and healthy cases are correctly classified. These performance evaluation metrics can be defined as ROI-based, pixel-based, core-based, or patient based, considering as a sample of the investigated dataset respectively a ROI, a pixel, a biopsy core, or a patient.

### **Real-Time Performance**

Achieving real-time performances is mandatory in order to use a CAD tool in clinical environment. Non real-time detection tools are interesting for research purposes but not appropriate for clinical employment. Based on the CAD tool architecture described above, it is easy to understand that to achieve real-time performance, the computational cost of processing flow steps must be reduced, focusing on the image segmentation and features estimation steps that are the most onerous steps. In this thesis, we exploit the application intrinsic parallelism to implement segmentation, features extraction, and the classification steps on highly parallel architectures. In particular, considerations about cost, portability and flexibility, make the modern GPU the better choice. Tissue characterization presents three level of parallelism: ROI-level, feature-level, and pixel-level. In fact, being ROIs independent from each other, feature extraction and classification can be performed in parallel for each ROI. Furthermore, for each ROI, several features can be computed in parallel and usually each of them involves computations on each individual ROI pixel.

## **2.4 PCa CAD literature review**

From the above discussion it is easy to understand that the different methods for CAD PCa detection based on UTC can be classified according to the type of data required (RF signal, B mode image) and the implementation of each step of the CAD processing flow. A review of the published methods for PCa detection can be found in [56]. In this chapter that review is integrated with some works not included in it. UTC based on the acoustic parameters such as attenuation and backscattering coefficients extracted from RF echo signals has been studied since the late 1960s [63]. A number of groups theoretically analyzed the capability of RF-spectrum parameters to discriminate tissue types in prostate and many methods based on the extraction of RF features have been proposed. These

methods usually focus on important backscattering tissue parameters such as slope, axis intercept and midband value ([63], [3], [4]). Besides RF-spectrum analysis, many researchers have used textures extracted from ultrasound B-scan images for prostate tissue characterization. The basic idea of this approach is that the speckle pattern in the ultrasonic image can reveal structural information about tissues. First works based on this principle consisted on a biopsy ground truth and analysis of rectangular regions around needle insertion points. Basset *et al.* [64], Huynen *et al.* [65], Houston *et al.* [66], and Llobet [67] in their studies used first and second order statistic textural parameters and simple decision trees to perform classification. Further approaches converged on a multi-feature approach based on the extraction of parameters of different nature from both the RF signal and B-Mode images [3, 4, 68], as well as clinical data like PSA value [69] and morphological feature [70], to enhance classification performances. Also, different ground truths and classifiers have been used in later approaches. In particular, in [69, 4] authors used ground truth coming from prostatectomy and histological analysis of prostate slices, and Mohamed *et al.* in [71] used a gold standard based on radiology visual inspection. Concerning the classifiers both classical approaches like Support Vector Machine (SVM) [71] and complex approaches like Artificial Neural Networks (ANNs) or Self Organizing Kohonen map [3, 4] have been used. Different segmentation approaches have also been proposed. They can be roughly divided into two categories: the automatic segmentation of the whole prostate into regular ROIs, and the automatic and adaptive segmentation of the gland into ROIs of irregular size based on the analysis of the acquired signal.

A comparison of the mentioned CAD schemes for PCa detection is reported in Table 2.2. From this table we can observe that: (i) most of the works are based on the use of rectangular equally sized ROIs; (ii) combining features from B-Mode and RF signal results in more effective classification; (iii) since features of different nature are hardly intercorrelated, only nonlinear classification models seems to be suitable for classification. A fair comparison of these methods is difficult especially due to the lack of publicly available database. Furthermore, clinical evaluation of the methods is missed, and no information about computational cost is available for these approaches.

According to the above mentioned requirements, a tool for real-time Computer Aided Biopsy (rtCAB) has been developed and it is described in the following section. Method performances have been compared with performance of the standard biopsy protocol in order to assess its effectiveness as biopsy guide and useful to improve the PCa diagnosis procedure efficiency.

## 2.5 Proposed approach for prostate biopsy protocol improvement: rtCAB

The CAD tool we propose is aimed at improving the current prostate biopsy protocol by marking the potential pathological tissues regions and discarding the healthy ones.

WORK	Data		ROI		TECHINQUE		RESULTS		
	Type	Cases(Malign)	Size	Number	Features	Classifier	SE	SP	ACC
Basset [64]	B-Mode	16(6)	64 x 64	37	Textural	Decision Tree	83	71	-
Houston [66]	B-Mode	25(11)	121 x 10	25	Textural	Decision Tree	73	86	80
Mohamed [71]	B-Mode	20(20)	Irregular	-	Textural	SVM	83.3	100	93.75
Llobet [67]	B-Mode	289	384 x 288	4539	Textural	SVM	68.4	53.3	61.6
Schmitz [3]	RF	33(33)	128 x 16	3400	Multi.	Kohonen Map	82	88	-
Scheipers[4]	RF	100(100)	128 x 16	170484	Multi.	Neuro-Fuzzy	-	-	75
Feleppa [69]	RF	200(110)	64 x 64	1019	Multi.	Neural N.	-	-	80
Mohamed [68]	RF	20(20)	Irregular	108	Multi.	SVM	83.3	100	94.4
Huynen [65]	B-Mode	51(46)	Rectangular	-	Textural	Decision Tree	80.6	88.20	-
Han [70]	RF	51(51)	25x25	200	Multi.	SVM	92	95.9	-

Table 2.2: Comparison of the the published methods for prostate tissue characterization with ultrasounds.

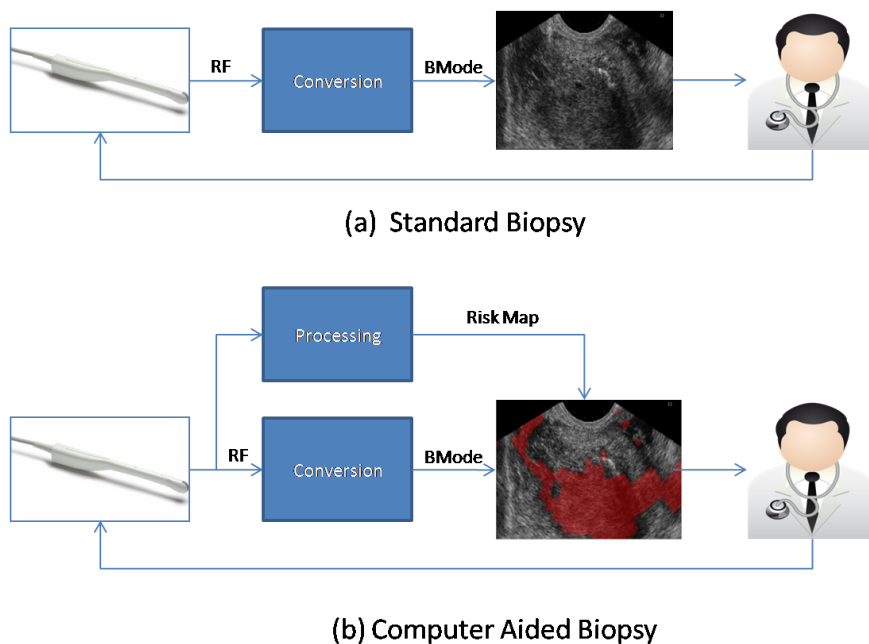


Figure 2.8: Ultrasound signal processing chain in standard (a) and computer aided (b) biopsy.

Providing physician with a binary risk map superimposed to the conventional B-Mode image it is possible to perform lesion-directed biopsies, and, as a consequence, to reduce the number of unnecessary sampled cores.

Figure 2.8 shows the basic principle that underpins our approach: in the standard TRUS-guided biopsy the RF echo signal generated by tissues is acquired and it is converted into a B-Mode image whose echogenicity guides the biopsy; in computer aided biopsy a second branch is added to this loop. In this branch the RF signal is processed in order to highlight tissue characteristics useful to discriminate between suspicious and not-suspicious regions. Based on these information the CAD generates a risk map that, superimposed to the original B-Mode image, guides physicians to the highest risk regions.

To develop a support tool for performing lesion-directed biopsy the computer aided analysis of TRUS images should be performed at the same time as the standard TRUS video sequence visualization, usually performed at 25-50fps, so that the real-time nature of the echographic examination could be preserved and a computer aided biopsy session could be as short as the the standard one. Furthermore, to exploit current medical imaging technology and avoid additional costs due to the necessity of particular equipment, physicians training, or longer biopsy sessions, the CAD tool must be fast, user-friendly, and easily integrable in standard ultrasound imaging equipment. To achieve these goals we propose to exploit GPU computing, meaning that the RF echo signal processing steps are implemented exploiting parallel programming techniques on commercial graphic cards.

Basically, the proposed CAD tool relies on the common UTC based CAD architecture described above, although specific tricks were applied to achieve real-time performance. The classification system, constituted by the blocks performing preprocessing, segmentation, feature extraction and ROIs classification, is the core of the CAD scheme. Beside it, processing blocks dedicated to the diagnostic information rendering are also present.

The rtCAB processing flow, the parallel implementation of the classification system, as well as the procedure used for its training on medical ground truth, are detailed in the following sections. Specific attention is also devoted to the realization of the ground truth database.

### 2.5.1 Database collection

The main limitations preventing the realization of an automatic detection tool in ultrasound imaging are the non standard orientation on the images and the absence of a public database for algorithm testing and validation. In the context of PCa detection in ultrasound images, the problem of free ultrasound imaging orientation is overcome thanks to the fact that the current TRUS guided biopsy protocol exploits a standard visualization of the gland which appears in a fixed region of the image with a fixed position *w.r.t.* the biopsy needle. This provides robust clinical references for image interpretation.

However, the absence of an accurate mathematical model for the ultrasound signal traveling in prostate tissue leads to the necessity of a large and consistent TRUS images database. Such a database, made up of the RF-signals acquired during biopsies along with the corresponding tissue histopathological outcomes, was collected in collaboration with the department of Urology of the S. Orsola Hospital in Bologna. Up to now, it consists of 1323 video sequences belonging to 129 patient (79 healthy, 50 unhealthy) \*. To collect the database, each patient was submitted to a routinely systematic prostate gland biopsy (8-12 cores) for PCa suspect. Biopsies were performed by the same operator. During each examination the same TGC profile, impulse bandwidth, acquisition gain, aperture angle maximum depth and focal depth were set in order to standardize

---

\*Currently, the TRUS video sequences database is not public.

acquisition conditions. For each patient the following parameters were collected before biopsy:

- age
- PSA levels
- presence/absence of palpable lesions at DRE
- prostate gland volume
- presence/absence of focal lesions at TRUS
- clinical history

The TRUS images of prostate sagittal section were acquired by a TECHNOS (Esaote s.p.a.) ultrasonographer equipped with a trans-rectal probe EC123 employed with ultrasound maximum frequency of 7.5MHz. The incoming RF ultrasound raw signals were extracted from the ultrasonographer, just after the analog to digital conversion, through a 1 Gbit/s optical fiber link and recorded on disk by means of FEMMINA [72], an hardware and software platform dedicated to ultrasonic signal and image processing. RF data were acquired before, during and after each bioptic core extraction. A unique alphanumeric code was assigned to each core and its corresponding recording. Each core was then examined by expert pathologists and the following parameters were collected:

- position relative to prostate gland
- histological classification
- presence/absence of neoplastic lesions
- total volume of neoplastic lesions
- gleason score
- core length

An example of the form used during biopsy session to collect the above mentioned data is shown in Figure 2.9.

**Patient Code:** \_\_\_\_\_  
(institution number – patient number 1,0 01)

**Patient Initials:** \_\_\_\_\_  
(name 1 letter- surname 2 letters)

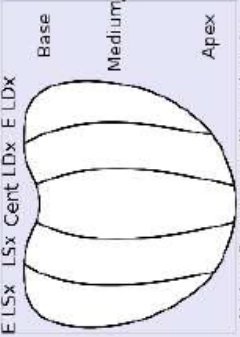
**Session Data:**

<b>Measure Setting:</b> _____	<b>Frequency:</b> _____	<b>Depth:</b> _____
<b>Gain:</b> _____	<b>Focus:</b> _____	

**NOTES:**

**BIOPSY PROTOCOL SCHEME**

E L Sx L Sx Cent L D x E L D x



Mark on figure core position and number

**HISTOLOGICAL EXAMINATION**

LEFT SIDE				RIGHT SIDE					
FILE number	Core number	Gleason Score	% pathological tissue	Pathology	FILE number	Core number	Gleason Score	% pathological tissue	Pathology

Figure 2.9: Example of form used during biopsy sessions to collect information about patient history, clinical information and results of DRE and TRUS examination performed before and during biopsy.



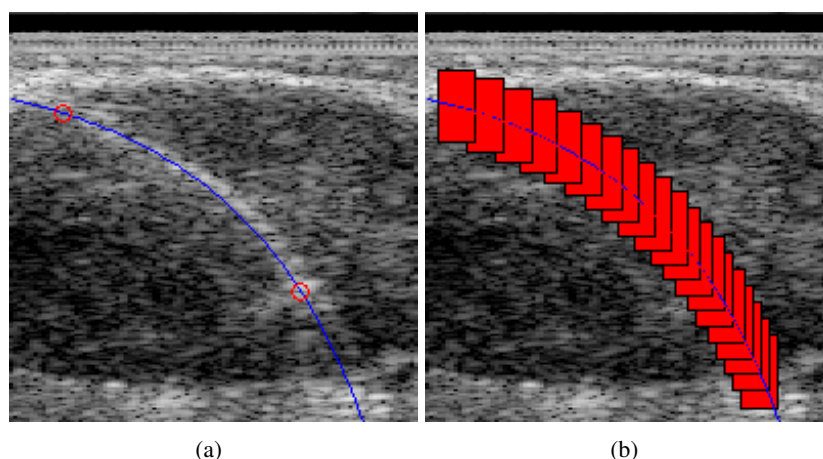


Figure 2.10: Needle segmentation procedure: in figure (a) the penetration track identification in the RF space is shown. The ROIs automatically generated on the needle track are shown in figure (b).

Acquired data were then processed providing that pathological information is confined within the bioptic core. To reliably associate RF signal to histology, only the image portion corresponding to the needle was used, as shown in Figure 2.10. Needle track is manually detected within the core extraction frame, exploiting its high reflectivity and almost fixed location. Segmentation proceeds automatically in the frame preceding needle insertion once the maximum penetration depth and prostate gland insertion point are identified. Each core is further divided into 20 equispaced and partially overlapped ROIs ( Figure. 2.10 (b)).

Table 2.3 shows the structure of the resulting database. A univocal code identifying the pathological status is attributed to each core after histopathological evaluation, and each ROI is associated to a label. ROIs labeling reckons that tumor distribution within each core is unknown, therefore only ROI belonging to malign cores with pathology covering more than 70% of the sample were labeled as suspicious. ROI belonging to the remaining malign cores were considered unlabeled as they cannot be accurately labeled a priori due to PCa imprecise localization. In addition, cores diagnosed with precancerous lesions were considered unlabeled as they clearly are neither benign nor equal to PCa. Finally, all ROIs belonging to benign cores were labeled as not suspicious. ROIs, along with their labels, constitute the ground truth.

As shown by numbers reported in Table 2.3, a large part of the collected data would be wasted if a traditional supervised learning approach is used to train the classifier. This observation highlights the necessity of a complex learning scheme able to deal with unlabeled data in order to use the whole amount of collected data. This goal can be achieved using semi-supervised learning techniques as it will be explained in chapter 3.

Table 2.3: Ground Truth Database.

Code	Description	#Cores	#ROI	Labels
M	PCa Percentage $\geq 70\%$	39	780	Malign
UM	PCa Percentage $<70\%$	128	2560	Unlabeled
U	Precancerous Lesions	123	2460	Unlabeled
BB	Benign Prostate Tissue from Benign Glands	557	11140	Benign
BM	Benign Prostate Tissue from Malign Glands	234	4680	Benign
BB*	Other Tissue from Benign Glands	180	3600	Benign
BM*	Other Tissue from Malign Glands	62	1240	Benign
Tot	129 patient (79 healthy, 50 unhealthy)	1323	26460	

### 2.5.2 rtCAB processing flow

In order to achieve real-time performance, rtCAB makes use of a twofold signal processing datapath, a schematic representation of which is depicted in Figure 2.11. Along the first branch, the incoming RF ultrasound signal undergoes envelope detection and scan conversion. In order to provide physician a real-time video feedback of the organ he is sampling, a fast digital envelope detection algorithm [73] and an high speed digital scan converter [74] based on bilinear interpolation were chosen. Algorithms were optimized and combined to seamlessly perform detection, logarithmic compression and conversion in one step. As the chosen envelope detection algorithm operates memoryless only on a few input samples, parallel processing techniques were also employed to further speed up this step. This step has been implemented in parallel on GPU using CUDA [28] and ran on a Nvidia GTX280 graphic card with 240 CUDA cores and 1GB of memory. A processing time around 1 ms was obtained during the conversion of a 2792 x 250 RF image to a 300 x 300 pixel BMode image.

The second branch is devoted to the classification task. First, an improved version of the envelope detection algorithm used in the first branch is applied to the incoming data. Comparing the output of this stage to the output of a gold standard, Hilbert Transform based, envelope detector, an accuracy of -49.2dB nRMSE was measured on standard acquired ultrasound data. Note that, depending on the type of features that must be computed, the envelope detection step is followed by a logarithmic compression step. In fact, while statistical features such as the Nakagami parameters are directly computed on the signal envelope, textural features must be computed after the envelope detection and logarithmic compression of the RF signal. We will refer to the combination of these steps as preprocessing step. Then, the obtained signal envelope, and/or its log-compressed version, is automatically divided into fixed size rectangular regions of interest. Following segmentation, appropriate features are estimated over each ROI. In particular using rtCAB two different types of features can be extracted from each ROI, namely the Nakagami distribution parameters and the Unser textural features, as explained in subsection 2.3.2. The next stage is responsible for the classification of each ROI as suspicious or

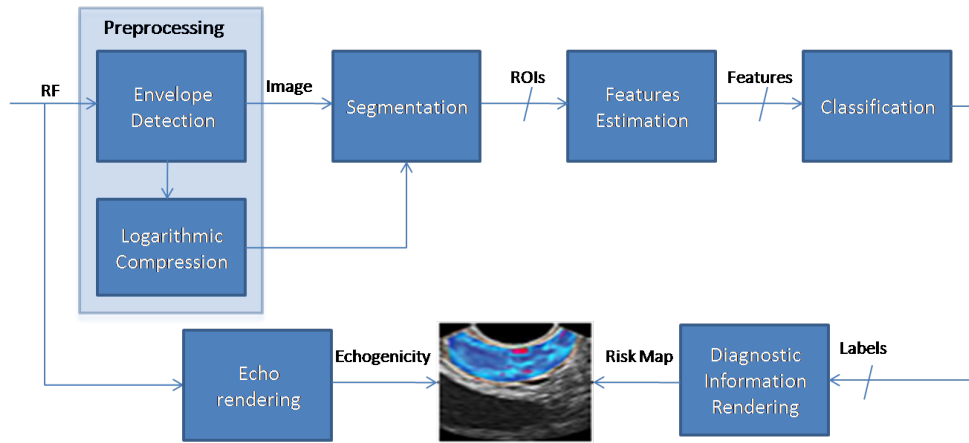


Figure 2.11: Signal processing flow graph of rtCAB: from left to right the incoming RF signal undergoes envelope preprocessing, segmentation, feature estimation and classification. Preprocessing step comprises envelope detection and/or logarithmic compression depending on the kind of features that must be computed. Results are scan converted and superimposed to the original B-Mode image.

not-suspicious based on the extracted features. Classification is performed by means of a non linear classifier based on the kernel trick principle. In particular, the used classification system is a supervisedly trained non linear classifier combining a GDA feature extraction step and a FLD classifier with an optimized shift of the decisional threshold (see subsection 2.3.2). The classification result, consisting in a set of labels for the classified ROIs, is then used to generate a false color overlay image that, superimposed to the scan converted image, forms a risk map which guides urologists to sample the highest cancer risk region.

To achieve real-time performance all these steps were implemented in parallel on GPU. Each step is implemented into one or more CUDA kernels in order to make the classification framework as much as possible flexible *w.r.t.* the introduction of new features and the use of different classification algorithms, and to efficiently manage data dependencies between different steps. In the following paragraph, we will provide some details on the parallel implementation of the rtCAB processing flow focusing on the classification system.

### 2.5.3 Parallel implementation of the rtCAB classification system

The input of our parallel implementation procedure are the RF signal acquired during the echographic examination and a group of parameters set by the user. These parameters define (i) the number and the type of features that must be computed; (ii) the number of the desired ROIs as well as their dimensions, and the distance between centers of two consecutive ROIs that allows to define ROIs overlap; (iii) the classifier parameters. The RF input signal consists in a  $N_{scan} \times N_{dp}$  matrix, where  $N_{scan}$  is the number of scan

lines and  $N_{dp}$  is proportional to the depth of the investigated tissue region. This matrix is stored in the GPU global memory in a 1D buffer. Both the signal envelope and its log-compressed version are stored in the same way. The size of each buffer is equal to the RF signal matrix size and the used data type is 16-bit integers.

### Preprocessing

Two different CUDA kernels are devoted to the preprocessing step. The first one computes only the signal envelope, while the second one computes both the signal envelope and its logarithmic compression. The logarithmic compression is required only to compute textural features. For this reason, which kernel is executed is defined runtime based on a feature's flag set by the user. The implemented envelope detection algorithm [73] takes on that under the assumptions of linear propagation, narrow ultrasound beam and pulse bandwidth, and negligible signal attenuation, a continuous time combined amplitude and phase modulated (AM/PM) signal can be adopted to model the RF echo signal [75]. By using this model the echo signal  $x(t)$  is written as:

$$x(t) = A(t) \sin(2\pi f_c t + \phi(t)) \quad (2.31)$$

where  $A(t)$  is the signal amplitude (corresponding to the tissue echogenicity),  $f_c$  is the central frequency of the ultrasound pulse and  $\phi(t)$  is the phase term. After sampling with period  $T_s = \frac{1}{f_s}$  the echo signal can be written as:

$$x_k = A_k \sin(\omega_0 k + \phi_k) \quad (2.32)$$

where  $\omega_0$  is the normalized ultrasound pulse central frequency and  $A_k$  and  $\phi_k$  are the discretized versions of  $A(t)$  and  $\phi(t)$  respectively. As discussed in [73], if the amplitude is assumed to be slowly varying function of time, an approximate value for a sample of the envelope can be obtained from two consecutive samples of the signal (knowing the relationship between the carrier and the sampling frequency):

$$A = \frac{1}{\sin(\omega_0)} \cdot \sqrt{x_k^2 + x_{k+1}^2 - 2x_k x_{k+1} \cos(\omega_0)} \quad (2.33)$$

A better estimation can be obtained by exploiting a slightly more complex model which takes into account three adjacent samples.

Basically, the envelope detection kernel computes  $A$  using three samples, implementing a slightly modified version of equation (2.33). For these operations the parallelism has the granularity of the individual signal sample, meaning that each thread is associated to one signal sample  $x_k$ . Therefore each thread takes the samples  $x_k, x_{k-1}, x_{k+1}$ , computes the corresponding amplitude value and stores it into the envelope buffer. The envelope detection kernel is launched with two dimensional thread blocks for efficiency purposes. These blocks are then organized into bi-dimensional grid whose size is consequently computed to traverse the entire signal domain. In particular the x-dimension of the block is associated to the signal matrix rows, while the y-dimension is associated

to its columns. By appropriately combining thread and block indexes (*i.e.* vectors used to identify threads and blocks [28]) each 2D coordinate is converted into a 1D memory address.

The second preprocessing kernel is slightly different from the envelope detection one: before storing the envelope value in global memory, each thread applies it the logarithm mapping function that, being  $p(i, j)$  a pixel of an input image  $l$  is defined as:

$$Q(i, j) = \frac{\log(|p(i, j)|)255}{\log(1 + |R|)} \quad (2.34)$$

where  $R$  is the value with the maximum magnitude in  $l$ . Note that, once the signal envelope and/or its log-compressed version are created, the RF signal buffer is deallocated to save GPU memory space. For the sake of simplicity, in the following we will refer to the output of this stage as the *input image* of the segmentation step.

### Segmentation

As stated above, the segmentation step consists in partitioning the signal envelope (or its log-compressed version) into fixed size rectangular ROIs partially overlapped according to the user defined parameters. Again, for these operations, the parallelism has the granularity of the individual pixel. The segmentation kernel is launched with one-dimensional thread blocks of fixed size for efficiency purposes. These blocks are then organized into bi-dimensional grid whose size is consequently computed to traverse the entire image domain: the x-dimension of the block is computed to traverse each entire ROI in parallel, while the y-dimension is equal to the number of ROIs to process all the ROIs in parallel. Starting from the first pixel on the top-left corner of the image, in parallel, groups of pixels belonging to the different ROIs are identified and copied into another memory buffer, named *ROIs buffer*, which is a matrix whose columns are the ROIs. This matrix is also treated as a 1D memory buffer and resides in global memory. Basically, each thread computes the index to extract a pixel from the image and to copy it in the correct position of the ROIs buffer. This process is illustrated in Figure 2.12. Once the ROIs are created the signal buffer is deallocated to save GPU memory space. Also for the ROI buffer the data type is 16-bit integers.

### Features Extraction

All the computed features are stored into a  $N_{feat} \times N_{roi}$  matrix stored in global memory as a 1D buffer. In this case the data type is 32 bit floating point. Globally, the features extraction stage is implemented into 16 CUDA kernels, one devoted to the Nakagami parameters computation and the remaining 15 devoted to the Unser features estimation.

**Nakagami:** as stated above, these features are the shape parameter  $m$  and the scaling factor  $\Omega$  of the Nakagami distribution (2.6), computed according to equation (2.7) and (2.8) respectively. As explained in subsection (2.3.2), for the computation of the shape parameter the Padé Approximants were used. The CUDA kernel implementing

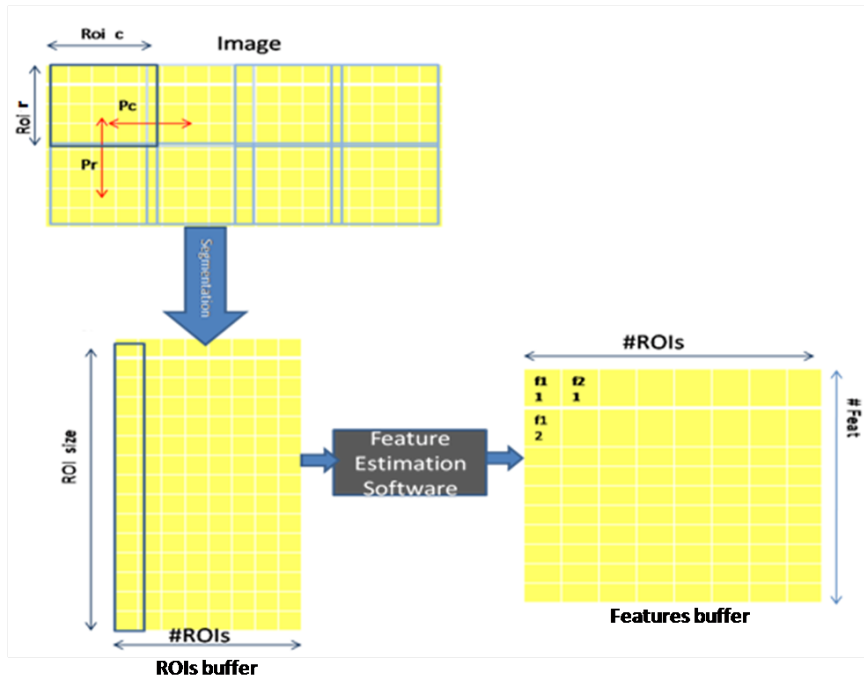


Figure 2.12: Schematic representation of the segmentation and features estimation procedure: from the input image the ROIs are identified based on the user defined parameters and stored per-column in the ROIs buffer; the ROIs buffer is the input of the features estimation software that computes  $N_{feat} \times N_{roi}$  features and store the, in the features matrix.

these computations is launched with one-dimensional blocks of fixed size for efficiency purpose, organized into one-dimensional grid whose size is consequently computed in order to process all the ROIs. Operations, are performed in parallel for each ROI and sequentially within each ROIs.

**User:** as stated above, these features are statistical attributes generated from histograms of the sum and difference of gray level in the ROIs. Four different kernels are devoted to the histograms computation one for each angular direction ( $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ ,  $135^\circ$ ). In each of these kernels both the sum and difference histograms for a given angular direction are computed. The histogram computation kernels are launched with mono-dimensional blocks of fixed size for efficiency purposes, organized into one-dimensional grid whose size is consequently computed to process all the ROIs. Two histograms for each ROI must be computed. All the histograms are collected into a  $(N_{roi} \cdot N_{dir}) \times N_g$  matrix, where  $N_{dir}$  refers to the number of angular directions on which features must be computed, and  $N_g$  is the number of quantized image gray levels. The histograms matrix resides in global memory as 1D buffer. To compute the histograms one should consider each couple of pixels distant  $d$  to each other in the selected direction (see subsection 2.3.2). Pixels belonging to each couple are added (or subtracted) and the resulting sum

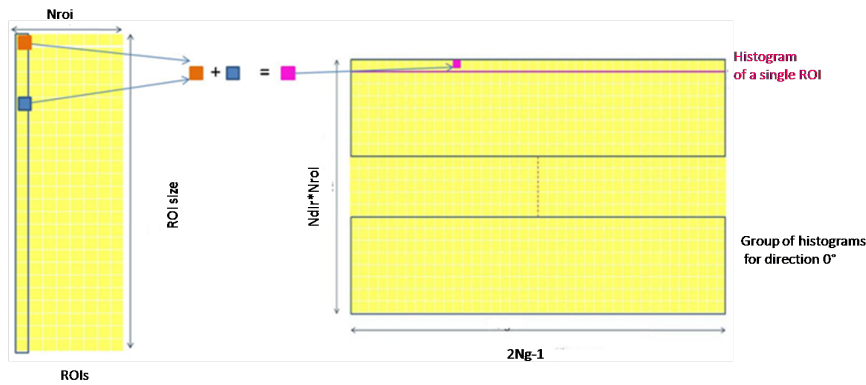


Figure 2.13: An example of the histogram computation procedure for  $D = 1$  in the angular direction  $0^\circ$ : two pixels are extracted from the ROI buffer and their sum is used as array index to update the appropriate histogram location. Note that in the ROIs buffer each ROI is stored per-column, therefore, into an ROI of size  $N_r \times N_c$  pixels consecutive in the horizontal direction are distant  $N_r$  to each other in memory.

(or difference) is used as array index to access the histogram location that must be incremented. Each histogram element is also normalized to the total number of occurrences previously computed by the CPU. Each thread is associated to one histogram element. Therefore, in parallel for each ROI, each thread computes the indices to extract pixels belonging to a couple from the ROIs buffer, sums or subtracts them and updates the corresponding histograms positions. An example of the histogram computation for the direction  $0^\circ$  and distance between pixel ( $d$ ) equal to 1 is reported in Figure 2.13. Note that only kernels for the directions necessary to compute the required features are launched.

Once histograms are computed, required features should be computed as well. Nine kernels are used to compute the features according to the equations in Table II of [50]. In addition, two auxiliary kernels are used to compute mean and contrast if they are not explicitly required but they are involved in the computation of other required features. As for the histograms computation, this operations subdivision makes the implementation extremely flexible: each kernel is launched only if the correspondent feature is needed. All features kernels are launched with the same execution configuration, *i.e.* the same dimension of thread blocks and grid, except that for one grid dimension that is different for each feature since it corresponds to the number of directions for which that feature is required. The other grid dimension is equal to the number of ROIs, while the threads blocks are one-dimensional block of size  $2Ng$ . By doing so, each thread is associated to one histogram element. Note that since all the features involve linear combination of histogram elements [50], all these kernels implement similar operations. They are all based on parallel reduction techniques [76] and take advantage of the GPU shared memory to efficiently compute the elements sum or subtraction involved. In each kernel, once the feature value is computed, it is stored in the feature buffer (see

Figure 2.12) that resides in global memory as a one-dimensional buffer. This buffer is accessed appropriately combining thread and block indices.

### Classification

Two different CUDA kernels are used to implement the classification step. The first one is devoted to the computation of the GDA Gaussian kernel (GDA CUDA kernel), while the second one (FLD CUDA kernel) computes the final discriminant function according to (2.5). The FLD CUDA kernel computes the decision function and performs the classification in parallel for each ROIs and sequentially on the training data. It means that each thread computes the value of the decision function for a given ROI and classifies it comparing the decision function value with the numerical values associated to the classes. The classification result is then stored in GPU global memory, again as a 1D buffer (of 16-bit integers), to be used to create the false color risk map. This kernel is launched with mono-dimensional block of fixed size for efficiency purposes, organized into mono-dimensional grid whose size is consequently computed to process all the ROIs in parallel.

#### 2.5.4 rtCAB learning scheme

The classification model selection is based on a two-step learning scheme. The first step concerns the selection of the training set and the variance  $\sigma$  of the GDA kernel. The second step optimizes the linear classifier threshold in order to provide the classification system with maximum PPV. The performance statistics considered in the first step are ROI-based. ROI-based accuracy, sensitivity, specificity, or PPV are computed as model selection criteria. The selected metric is the one providing the best core-based classification performances in the second step.

The data set for this step includes only a fraction of the database shown in Table 2.3. In particular, it contains the not-suspicious ROIs coming from healthy patients (143 BB cores, 2860 ROIs) and the suspicious ROIs belonging to the cores with tumor volume percentage larger than 70% (18 M cores, 360 ROIs). In fact, as stated above, these are the only ROIs that can be a priori labeled as suspicious with sufficient accuracy (due to the unknown distribution of tumor in each core).

In the first step of the model selection 500 random data sets of 360 unhealthy and 360 healthy ROIs are generated and separated in training set (600 ROIs) and validation set (120 ROIs). For each data set a cross-validation over 30 values of  $\sigma$ , equally distributed between 0.1 and 6, is performed. The output of the model selection step is constituted by the training set  $[\hat{X}_{trn}, \hat{Y}_{trn}]$  and the GDA kernel variance  $\hat{\sigma}$  corresponding to the maximum sensitivity in the classification of the validation set. The process for the first part of the learning scheme is shown in Algorithm 1. As stated above, the choice of the sensitivity as ROI-based criterion to select the model is arbitrary: other metrics may be used as well.



---

**Algorithm 1:** Training set and GDA parameter selection

---

**Input:**  $[X_{trn}, Y_{trn}]$ : training set data matrix and labels  
 $[X_{val}, Y_{val}]$ : validation set data matrix and labels

**for**  $iter=1$  to 500 **do**

- Select random  $[X_{trn}, Y_{trn}]$  and  $[X_{val}, Y_{val}]$ ;
- for**  $\sigma=0.1$  to 6 **do**
  - Train on  $[X_{trn}, Y_{trn}]$  using  $\sigma$  as GDA kernel variance;
  - Test on  $[X_{val}, Y_{val}]$ ;
  - Compute ROI-based Sensitivity:  $SE([X_{trn}, Y_{trn}], \sigma)$ ;

**Output:**  $([\hat{X}_{trn}, \hat{Y}_{trn}], \hat{\sigma}) = \arg \max SE([X_{trn}, Y_{trn}], \sigma)$ 

---

In the second step of the learning process, described in Algorithm 2, the FLD threshold is tuned according to core-based statistics in order to maximize the PPV of the classification model over the whole dataset. The training set for this step is composed of all the cores (benign, malign and unknown) of all the unhealthy patients for a total of 204 cores. Using the classification model selected in the previous step, a cross-validation is performed on the values of the threshold varying between the minimum and the maximum value of the FLD discriminant function. The number of thresholds considered is 1000, ranging from a model with 100% ROI-based specificity and a model with 100% ROI-based sensitivity. For each value of the threshold, the classification model is applied to the 204 cores and some core-based statistics are computed for each patient.

In this case a core is considered unhealthy if at least one of its ROIs is classified as suspicious. As a consequence a core-based true positives value (TP) is the number of malign cores correctly classified, while the core-based false positives value (FP) is the number of misclassified benign cores. Thus the core-based PPV represents the probability of PCa positiveness of a pathological gland according to the detection tool.

The final value of the threshold is selected as the one providing the largest median PPV over all pathological patients. Whether there are several threshold values corresponding to the maximum PPV, the highest value is selected in order to maximize the sensitivity constrained to PPV maximization.

### 2.5.5 rtCAB results

The rtCAB performances were tested considering two cases: in the first one only the shape and scale parameter of the Nakagami distribution were estimated on the signal envelope for each ROI, while in the second case, Unser textural features were also computed for each ROI: the resulting feature set is reported in Table 2.1. CAD performance were evaluated in terms of PPV in order to make a comparison with the performance of the standard biopsy protocol performed by physician. This allows to assess the CAD performance as biopsy guide and its effectiveness at improving the efficiency of the PCa diagnosis procedure.

**Algorithm 2:** Threshold Selection

---

**Input:**  $th=0$ , initial threshold at zero  
 $C$ : classification model defined by  $([\hat{X}_{trn}, \hat{Y}_{trn}], \sigma)$   
 $DFC$ : FLD discriminant function

**for**  $th=\min(DFC)$  to  $th=\max(DFC)$  **do**

- for**  $patient = 1$  to  $22$  **do**
  - Classify by  $C$  any core of this patient;
  - Compute core-based PPV ( $patient, th$ );
  - Compute core-based SE ( $patient, th$ ).

$PPV_m(th) = \text{med}(PPV(patient, th))$  median over patient;

Set  $\tilde{th} = \arg \max PPV_m(th)$ ;

**if**  $th$  not unique **then**

- $\hat{th} = \max(\tilde{th})$  corresponding to the largest SE;

**else**

- $\hat{th} = \tilde{th} \rightarrow SE$  constrained to PPV;

**Output:**  $\hat{th}$

---

Figure 2.14 depicts the core based PPV and SE for these classification tasks. Each horizontal bar represents a patient from the database, identified by a three letter code: the color in each bar indicates how the respective statistics changes as a function of the threshold over the given patient. The vertical dashed line is the best threshold  $Th_{best}$  selected by the learning scheme. At  $Th_{best}$  rtCAB in both cases provides a median PPV of 35% with a median SE of 100%. As a comparison, on the same dataset, the standard biopsy protocol features a PPV of 24%. Therefore, the proposed method improves PPV by 46% without losing diagnostic power. Consequently, according to the results of this study, it could be possible to achieve the same diagnostic value of a standard biopsy (8-12 cores), with no more than 7 cores in a rtCAB guided biopsy (see equation (2.2)).

To further verify the effectiveness of the proposed detection tool as prostate biopsy guide the CAD was let to classify a video sequence of images and classification results were compared with the histopathological outcomes. The same evaluation was carried out for both the classification tasks. Examples of classified images and their corresponding B-Mode with superimposed needle trace are shown for a healthy patient (Figure 2.15 (a) and 2.15 (b) and for an unhealthy one (Figure 2.15 (c) and 2.15 (d)). The frames corresponding to biopsy sampling (top) and the previous one with the superimposed binary risk map (bottom) elaborated by rtCAB using only Nakagami parameters (case 1) are shown on the left side; results of rtCAB using both Nakagami parameters and Unser textural features (case 2) are on the right side.

Note that these images have been completely generated by the tool, meaning that both the B-Mode image and the overlay image with the risk map were obtained from the RF signal acquired with standard ultrasound equipment. The tool provides radiologist with a user-friendly graphical interface shown in Figure 2.16. From this interface

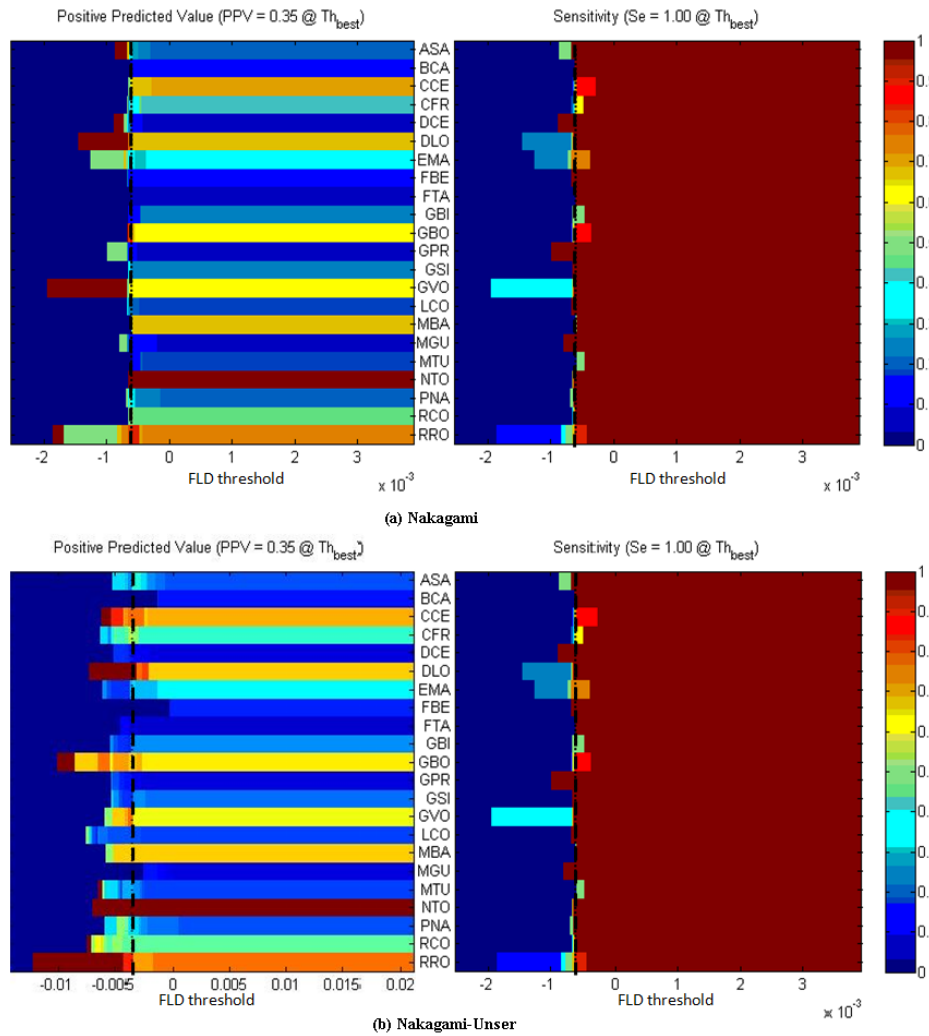


Figure 2.14: The core based PPV and SE of both the Nakagami (a) and Nakagami-Unser (b) classification tasks. The x-axis represents the 1000 threshold values. At right the color bar shows the color coding for the SE and PPV value. Each horizontal bar represents a patient from the database identified by a three letter code: the color in each bar indicates how the respective statistics changes as a function of the classifier threshold over the given patient. The vertical dashed line is the best threshold  $Th_{best}$  selected by the learning scheme.

the user can choose if perform or not the classification for each frame, and select some visualization and classification parameters, such as the classifier threshold.

Patients were diagnosed as healthy or unhealthy as a result of histological analysis. For the unhealthy patient, the considered tissue sample histology indicates a tumor volume equal to 90%, while tumor volume estimated considering the image portion marked by the CAD is equal to 77% for case 1 and to 87% for case 2. For the healthy patient, both the classification tasks assigned to a benign core a tumor volume equal to 5%.

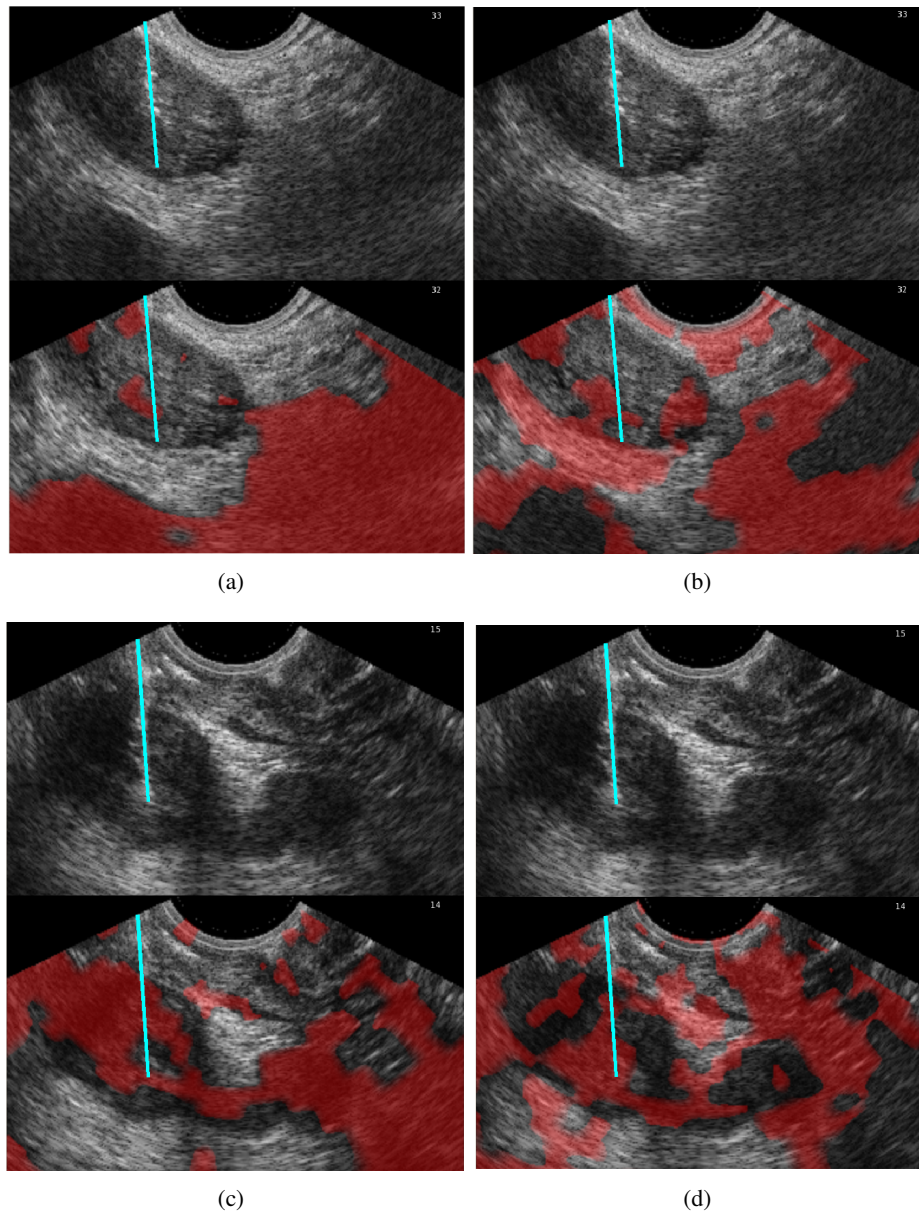


Figure 2.15: Examples of B-Mode images with superimposed needle trace and the corresponding image classified by the CAD for a healthy (a-b) and a unhealthy (c-d) patient. Images (a-c) refer to the Nakagami classification task while images (b-d) refer to the Nakagami-Unser one. For each image the frame on the top is the one corresponding to the biopsy sampling, while the previous frame with superimposed the binary risk map generated by the CAD is shown on the bottom.

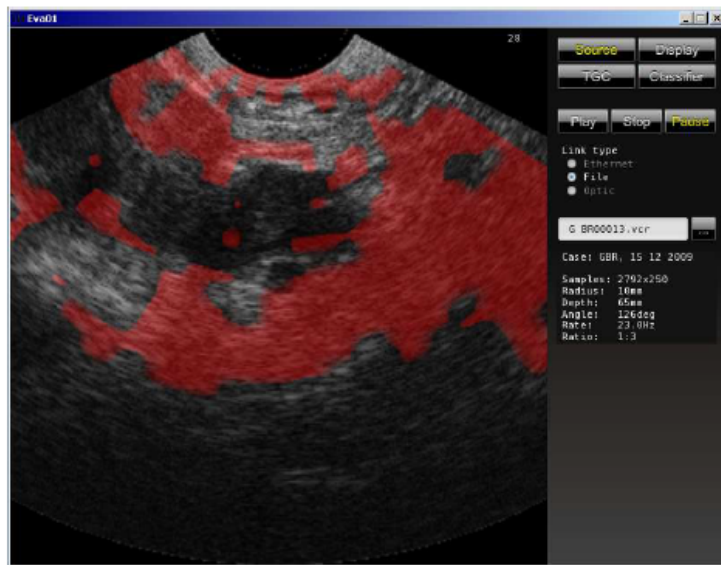


Figure 2.16: rtCAB tool interface.

From these results it can be observed that the proposed CAD scheme detects the presence of PCa with sufficient accuracy. However, although the number of cores required for an accurate diagnosis could be reduced using the proposed tool, some FP are still present, as confirmed by the healthy core identified by the tool as a core with a tumor volume equal to 5%. This could be explained by the intrinsic ground truth uncertainty. In fact, due to the unknown tumor distribution in each core, only malign cores with tumor covering the 100% of the sample should be used to train the classifier to ensure that ROIs labeled as suspicious effectively correspond to unhealthy tissue. However, core diagnosed with 100% tumor volume are rare, and therefore all the cores with tumor volume equal or greater than 70% were used for training. It means that the training set contains also suspicious ROIs that could correspond to healthy tissue.

The effect of this error to the classification performance can be reduced extending the size of the ground truth. Obviously, a larger ground truth can be obtained collecting more diagnosed samples, but it is difficult and expensive. An alternative way to achieve this goal consists in using learning techniques able to deal with unlabeled data so that all the collected samples can be used to extend the ground truth and enhance classification performance. This approach were adopted in this thesis and it will be described in the following chapter.

All the images in Figure 2.15 show noisy classification results: in fact, also regions outside the prostate boundaries are marked by the CAD. Higher classification accuracy is required to reduce this effect, achievable with a more complex specifically designed learning procedure based on multifeature approach.

Using the Nakagami parameters the classification of a 2792x250 image segmented into 100x10 pixels ROIs requires 8.3ms, while 14.8ms are necessary in case of segmen-

tation into 200x20 ROIs. Classification times using the features set in Table 2.1 are respectively 24.3ms for smaller ROIs and 69.9ms for bigger ones. In both cases the rendering time is about 1ms. Therefore, considering a 2792x250 image segmented into 100x10 pixels ROI, our detection tool is able to deliver a minimum frame rate of 38fps, confirming its effectiveness for prostate biopsy guidance.

## 2.6 Conclusions

Prostate cancer is a widespread and often lethal pathology and early stage detection is fundamental to treat it more effectively. The current clinical procedure used to detect PCa is based on four diagnostic tools: PSA evaluation, PCA3 test, DRE, and TRUS image analysis. Results are combined in decision making for performing or repeating a prostate biopsy, as the TRUS-guided biopsy is the only diagnostic tool able to confirm the presence of PCa. In fact, PCa patterns high variability limits TRUS diagnostic ability. As a consequence, biopsy consists in a systematic sampling of higher cancer risk regions with consequent increase of cost and patient risks.

In this scenario, CAD systems can be used to improve prostate biopsy protocol efficiency by providing physicians a second opinion thus allowing a reduction of the number of unnecessary biopsy cores without negative impinging on diagnosis accuracy.

The main goal of the research presented in this chapter was the development of CAD tool for PCa detection based on UTC. As highlighted by an accurate literature review and the analysis of the basic architecture of UTC based CAD systems, combining different features from RF signal and/or B-Mode images and using a non linear classification model is a valuable approach for prostate tissue characterization. A reliable medical ground truth and an appropriate learning scheme are essential for an effective CAD system training. The lack of a publicly available ground truth database and a standard learning procedure are the main limitations in the development of clinically valuable CAD tool for PCa. Furthermore, to be used as biopsy guide in clinical environment, such a CAD tool must be able to perform its image analysis in real-time. Therefore the detection tool we designed is a real-time Computer Aided Biopsy tool using a non linear classification process which can take advantage of both Nakagami and Unser textural features. A custom designed learning procedure based on CV is used to train the classifier on a ground truth database, collected in collaboration with the department of Urology at S.Orsola Hospital in Bologna. The CAD processing flow was implemented exploiting CUDA parallel processing acceleration and it is characterized by a minimum frame rate of 38fps suitable for clinical applications. The proposed CAD tool is able to reduce the number of biopsy cores from 8-12 to 7, preserving the same diagnostic power of the standard biopsy protocol. However some false positive are still present. The intrinsic ground truth uncertainty suggest the use of more complex learning techniques able to deal with unlabeled data to enhance classification performance, as it will be explained in the following chapter.

## Chapter 3

---

# Improving Prostate Biopsy Protocol with a Computer Aided Detection Tool based on Semi-Supervised Learning

---

The realization of an automatic CAD tool for PCa detection in ultrasound imaging is mainly hampered by the absence of an accurate mathematical model for the ultrasound signal traveling in prostate tissue which leads to the necessity of a large and consistent TRUS images database. Such a database, made up of the RF-signals acquired during biopsies along with the corresponding tissue histopathological outcomes, was realized and its structure was described in chapter 2. However, due to the unknown distribution of tumor in each core there is an intrinsic ground truth uncertainty that makes the problem of prostate tissue characterization an imperfect supervised problem. As a consequence, not all collected samples could be labeled *a priori* and employed to train the CAD system with standard Supervised Learning (SL) techniques. Moreover, even discarding a large amount of the collected data some uncertainty still remains (see section 2.5). Its effect to the classification performance can be reduced extending the size of the ground truth. Nonetheless, collecting large amounts of diagnosed samples data is difficult, expensive and time consuming, as it requires the effort of experienced personnel. This problem is much more critical, if only a part of the collected diagnosed samples can be actually used to create the ground truth.

Exploiting Semi-Supervised Learning techniques (SSL) instead of the commonly used SL is an effective way to overcome these problems. In fact, being SSL methods able to deal with unlabeled data, it is possible to train a CAD system by using both labeled and unlabeled data. This avoids collected data wasting and reduces diagnosed

samples collection effort and patient discomforts connected to the diagnosis procedure.

All previous observations motivate the research described in this chapter aimed at developing a new PCa CAD tool working on TRUS images, representing an extension of the one presented in the previous chapter. This new CAD tool is based on the ground truth database detailed in section 2.5 and a novel, expressly designed, supervised/semi-supervised learning scheme. At its core there is the innovative SSL algorithm, SelfCo3Core which was designed and developed in this research. This chapter will demonstrate that by using SSL for CAD system training allows to use the whole amount of collected data thus relaxing the requirements on the diagnosed samples collection effort. Note that, as rtCAB, the new CAD tool here proposed is aimed at improving the current biopsy protocol by providing physicians a binary risk map, so that a lesion-directed biopsy can be performed. Therefore, it will be demonstrated that the proposed CAD scheme is effective at improving the efficiency of the biopsy protocol by increasing the positive predicted value, and reducing the number of cores requested for an accurate diagnosis. By comparing the proposed CAD and a state-of-art detection tool based on SL it will be also demonstrated the effectiveness of the SSL at enhancing CAD performance without requiring any additional labeled data and making use of the whole ground truth dataset.

SSL is a wide and complex machine learning field. Several SSL techniques have been proposed in literature, but which method one should use strongly depends on the application, since, being the number of labeled data usually small, SSL makes strong model assumptions. A wrong SSL method for a given application not necessarily improves performance over supervised learning but could even degrade them.

In this chapter, after a brief overview of the general SSL concepts, the characteristics that a SSL method suitable for the target application should have are outlined. Two common SSL paradigms were identified as suitable approaches for the target application, and they were combined in a new SSL algorithm named SelfCo3. After a detailed description of the algorithm, both its performances as general purpose SSL algorithm and its ability at enhancing CAD system performances are evaluated. To further enhance CAD performance, SelfCo3 was improved introducing a kind of prior data knowledge that makes it specifically designed for training a CAD tool for PCa detection in TRUS images. This improved algorithm version is named SelfCo3Core, and it is the core of a mixed supervised/semi-supervised learning procedure also proposed to enhance CAD system training. Both SelfCo3Core and the complete learning procedure are accurately described. Then, performances of the new CAD scheme based on the designed learning procedure are extensively evaluated, demonstrating its effectiveness at improving standard prostate biopsy protocol.



### 3.1 Semi-Supervised Learning

Traditionally, there have been two fundamentally different types of tasks in machine learning. The first task is *Unsupervised Learning* (UL). Given a set of learning sample  $X$  the goal of unsupervised learning is to find interesting structures in  $X$  unraveling data underlying similarities and cluster similar data together. The second task is *Supervised Learning* (SL). The goal is to learn a mapping from  $x$  to  $y$ , given a training set made of pairs  $(x_i, y_i)$ , where  $x_i \in X$  are the samples and  $y_i \in Y$  their labels. The task is well defined, since a mapping can be evaluated through its predictive performance on test samples. The case where  $y$  takes values in a finite set (discrete labels) is called classification.

*Semi-supervised learning* (SSL) is halfway between supervised and unsupervised learning. In addition to unlabeled data, a SSL algorithm is provided with some supervision informations but not necessarily for all samples. Often, these informations will be the targets associated with some of the samples. In this case, the dataset  $X$  can be divided into two parts: the points  $X_l = (x_1, \dots, x_l)$ , for which labels  $Y_l = (y_1, \dots, y_l)$  are provided, and the points  $X_u = (x_{l+1}, \dots, x_{l+u})$ , the labels of which are not known [77]. The goal of SSL is to understand how combining labeled and unlabeled data may change the learning behavior and design algorithms that take advantage of this combination.

Most SSL strategies are based on extending either unsupervised or supervised learning to include additional information typical of the other learning paradigm. Specifically, semi-supervised learning encompasses several different settings, including [78]:

- **Semi-supervised classification.** Also known as classification with labeled and unlabeled data (or partially labeled data), is an extension to the supervised classification problem. The training data consists of both  $l$  labeled instances and  $u$  unlabeled instances. The goal of semi-supervised classification is to train a classifier  $f$  from both the labeled and unlabeled data, such that it is better than the supervised classifier trained only on the labeled data.
- **Constrained clustering.** This is an extension to unsupervised clustering. The training data consists of  $n$  unlabeled instances as well as some supervised information about the clusters. The goal of constrained clustering is to obtain better clustering than the clustering from only unlabeled data.

In this context, we will focus on semi-supervised classification, so basically we are interested in methods that use unlabeled data to either modify or re-prioritize hypotheses obtained only from labeled data.

In supervised classification, the training set is fully labeled, so one is always interested in the performance on future test data. In semi-supervised classification, instead, the training set contains some unlabeled data. Therefore, there are two distinct goals. One is to predict the labels on future test data, the other is to predict the labels on the unlabeled instances in the training set. The inductive SSL satisfies the former goal, while

the transductive learning satisfies the latter. Since the aim of this research is to exploit SSL to enhance performances of a classification system *w.r.t.* the ones obtainable with a standard SL approach, we will focus of the inductive SSL.

Several SSL techniques have been proposed in literature. Some often-used methods include: EM with Generative Mixture Models, Self-Training, Co-Training, Transductive Support Vector Machines (TSVMs), Graph-Based Methods, and so on. A comprehensive review of the SSL techniques can be found in [78, 79, 77].

The method to be used strongly depends on the application. In fact, because the number of labeled data is small, SSL makes strong model assumptions: one should use the method whose assumptions best fit the problem structure. Theory about SSL identify three main assumptions, namely the *smoothness assumption*, the *cluster assumption* and the *manifold assumption* [77]. However, in practice, associating a given classification problem to one of these assumptions could be difficult, or not sufficient to select the appropriate SSL method. As a consequence, other problem requirements, such as the necessity of fast learning algorithms or the flexibility in using different classification algorithms, have to be considered. It is worth noting that blindly selecting a semi-supervised learning method for a specific task will not necessarily improve performance over supervised learning. In fact unlabeled data can lead to worse performance with the wrong assumptions.

In practical applications SSL is particularly useful when there are more unlabeled data than labeled one. This is likely to occur if obtaining data points is cheap but obtaining the labels costs a lot of time, effort, and/or money. This is the case in many application areas of machine learning, such as *speech recognition*, *web page classification*, or *classification of protein sequences*. Beside these common applications, the development of a reliable CAD system is also a machine learning task where SSL can be extremely valuable. In fact, achieving good classification performance requires a large amount of training samples. Also, data must be collected during medical examinations, along with the corresponding diagnoses made by expert physicians. Collecting large amounts of diagnosed samples data is difficult, expensive and time consuming, as it requires the effort of experienced personnel. Conversely, collecting unlabeled data is easier and less expensive, as no such interaction is required. In particular, in the context of computer aided PCa detection, reducing the number of diagnosed samples necessary to achieve good performance is mandatory to contain both costs and discomfort for the patients (see section 2.2). In order to do this, it is possible to train the CAD system by using both diagnosed samples (labeled data) and undiagnosed samples (unlabeled data) by exploiting SSL techniques in place of supervised ones. The development of specifically designed SSL algorithms and learning procedures for CAD systems training is the major goal of the research presented in this chapter.

From that discussed above and in the previous chapter, a SSL method suitable for our target application has to be:

- *fast*: a huge amount of unlabeled data implies the need for fast and efficient SSL

algorithms;

- *flexible*: to be easily and efficiently inserted in existing classification framework the choice of the classification algorithm should be left as open as possible;
- *features assumptions free*: due to the heterogeneous and complex nature of biological tissue, the assumption of a specific statistical model describing the classification problem, or the presence of natural features splitting cannot be made.

For these reasons we focused on two popular SSL paradigms in this research: Self-Training and Co-Training, and we combined them to develop SelfCo3, an innovative SSL algorithm.

SelfCo3 is a very general algorithm that can be used in different data mining scenarios. Based on encouraging results obtained using this algorithm for CAD system training it was improved to better exploit all data and information provided by the available ground truth so that prediction performance could be further enhanced. This improved algorithm version is called SelfCo3Core. In addition, a new mixed supervised/semi-supervised learning procedure including SelfCo3Core has been expressly designed to improve CAD performance. Both the two versions of the learning algorithm and the learning procedure will be detailed next.

### 3.2 Self-training and Co-training

Self-Training is a wrapper-algorithm that repeatedly uses a supervised learning method. It is probably the earliest idea about using unlabeled data in classification and is a commonly used technique for SSL. In the Self-Training approach, a classifier is trained with a small amount of labeled data and then it is used to classify the unlabeled data [78]. For each prediction, a subset  $S$  of unlabeled data, along with their predicted labels, is added to the original training set to enlarge it. We will refer to these data and their predicted labels as *soft-labeled data*. Typically a confidence value is measured to decide which unlabeled data should belong to  $S$ . The classifier is then re-trained on the enlarged training set and the procedure is repeated. The Self-Training method is described by the pseudo-code in Algorithm 3. Note that it is also possible for  $S$  to be the whole unlabeled data set. In this case,  $L$  and  $U$  remain the whole training set, but the assigned labels on unlabeled instances might vary from iteration to iteration.

Self-Training has been applied to several natural language processing tasks, as well as to parsing and machine translation. Yarowsky [80] used self-training for word sense disambiguation. Riloff *et al* [81] used it to identify subjective nouns. Maeireizo *et al* [82] classified dialogs as *emotional* or *non-emotional* with a procedure involving two classifiers. Rosenberg *et al.* [83] applied self-training to object detection systems from images showing that the semi-supervised technique compares favorably with a state-of-the-art detector.

**Algorithm 3:** Pseudo-code describing Self-Training

---

**Input:** classifier  $f$ , labeled data  $\{(x_i, y_i)\}_{i=1}^l$ , unlabeled data  $\{x_j\}_{j=l+1}^{l+u}$

Initially, let  $L = \{(x_i, y_i)\}_{i=1}^l$  and  $U = \{x_j\}_{j=l+1}^{l+u}$

Repeat:

Train  $f$  from  $L$  using supervised learning.

Apply  $f$  to the unlabeled instances in  $U$ .

Remove a subset  $S$  from  $U$ ; add  $(x, f(x))_{x \in S}$  to  $L$ .

---

The Self-Training assumption is that its own predictions, at least the high confidence ones, tend to be correct. This is likely to be the case when the classes form well-separated clusters. The major advantages of Self-Training are its simplicity and the fact that it is a wrapper method. In fact it means that the choice of the classification algorithm used in the procedure is left completely open. However, this method has two drawbacks: first, a training step can lead to a worse classification model, if a misclassified test sample is added to the training set; second, the confidence level measurement could be difficult to perform, as it usually depends on the selected classification approach and sometimes could also be time consuming (for example if 10-fold cross-validation is applied to find the confident example to label in every iteration of training algorithm [84]).

While in Self-Training a classifier teaches himself, in Co-Training two classifiers teach each other. In its original formulation the Co-Training approach assumes that data can be described by two independent, sufficient and redundant features sets, also called views [85, 86]. Under this assumption, two classifiers are initially trained with labeled data from the two views respectively. Each classifier then processes unlabeled data and selects some of them to be used as soft-labeled data to teach the other. Data are selected on confidence basis. Each classifier is retrained with the newly labeled data given by the other classifier, and the procedure is repeated (see Algorithm 4).

As for Self-Training the main advantage of Co-Training is that it is a wrapper method, and therefore it is widely applicable to many tasks. The main drawbacks of this method are, again, the necessity of confidence measurements and the request of two different views, which is usually regarded as too strong [87, 88] since, for a general task, the features may not naturally split into two views. To overcome the latter drawback several solutions were adopted such as the random splitting of features in two artificial views and the use of different supervised learning algorithms.

Nigam and Ghani [87] performed empirical experiments to compare Co-Training with Generative Mixture Models and EM. Their results also show that it is better to probabilistically label the entire  $U$ , instead of a few most confident data points. They name this paradigm co-EM. Finally, when there is no natural feature splitting, the authors suggest to create artificial splitting by randomly break the feature set in two subsets. They showed that Co-Training with artificial feature splitting still helps. Collins and Singer [89], Jones [90] used Co-Training, co-EM and other related methods for information ex-

**Algorithm 4:** Pseudo-code describing Co-Training

---

**Input:** labeled data  $\{(x_i, y_i)\}_{i=1}^l$ , unlabeled data  $\{x_j\}_{j=l+1}^{l+u}$ , a learning speed  $k$ .

Each instance has two views  $x_i = [x_i^{(1)}, x_i^{(2)}]$ .

Initially, let the training sample be  $L_1 = L_2 = \{(x_1, y_1), \dots, (x_l, y_l)\}_{i=1}^l$ .

Repeat until unlabeled data is used up:

Train a *view-1* classifier  $f^{(1)}$  from  $L_1$ , and a *view-2* classifier  $f^{(2)}$  from  $L_2$ .

Classify the remaining unlabeled data with  $f^{(1)}$  and  $f^{(2)}$  separately.

Add  $f^{(1)}$ 's top  $k$  most-confident predictions  $(x, f^{(1)}(x))$  to  $L_2$ .

Add  $f^{(2)}$ 's top  $k$  most-confident predictions  $(x, f^{(2)}(x))$  to  $L_1$ .

Remove these from the unlabeled data.

---

traction from text. Zhou *et al.* [91] proposed a Co-Training algorithm using Canonical Correlation Analysis which needs only one labeled point. Also, other approaches have been proposed to relax the Co-Training assumptions on the splitting of features. Goldman and Zhou [84] used two learners of different type but both takes the whole feature set, and essentially use one learner's high confidence data points, identified with a set of statistical tests, in  $U$  to teach the other learning and viceversa. Later, the same authors [92] propose a single-view multiple-learner Democratic Co-learning algorithm. An ensemble of learners with different inductive bias are trained separately on the complete feature of the labeled data. They then make predictions on the unlabeled data. If a majority of learners confidently agree on the class of an unlabeled point  $x_u$ , that classification is used as the label of  $x_u$ . Then  $x_u$ , with its label, is added to the training data. All learners are retrained on the updated training set. The final prediction is made with a variant of a weighted majority vote among all the learners. Similarly Zhou and Li [88] proposed Tri-training. In Tri-training two classifiers label data for a third exploiting the prediction agreement: if two of them agree on the classification of an unlabeled point, the classification is used to teach the third classifier. Classifiers are made different by manipulating the original labeled data set via bootstrap sampling, and the final hypothesis is generated via majority voting. Finally, Johnson and Zhang [93] proposed a two-view model that relaxes the conditional independence assumption.

Within this scenario, the proposed algorithm, named SelfCo3, combines Self-Training and Co-Training to overcome the drawbacks of each approach and improve classification performance and method applicability.

### 3.3 The proposed SSL algorithm: SelfCo3

The SSL algorithm designed in this research leverages ideas from both the Self-Training and Co-Training algorithms. In fact the developed algorithm, according to the Co-Training style, uses more than one classifier. As the same copy of the soft-labeled data is used to refine all involved classifiers, it could also be assimilated to Self-Training.

**Algorithm 5:** Pseudo-code describing the SelfCo3 algorithm

---

**Input:**  $L$ =labeled training set;  $L_v$ =labeled validation set;  $U$ =unlabeled training set;  
Classification models  $C_1, C_2, C_3$  (previously selected classification models)

**while** not ( $U$  is empty or there are not data for which classifiers unanimously agree on labeling ) **do**

**for**  $i=1$  to 3 **do**

Train classifier (i) with  $L$ ;

Test classifier (i) on  $U$  and label  $U$  according to its predictions;

Test classifier (i) on  $L_v$  and compute the validation error;

Perform majority voting on predicting labels;

Selects a subset  $S$  from  $U$  based on majority voting: only data for which classifiers unanimously agree on labeling are selected;

Eventually shrink  $S$  balancing classes;

Add  $S$  to  $L$ ;

Select the final classification model as the one corresponding to the minimum validation error;

**Output:** [Enlarged training set, best classification model]

---

Three different classifiers are trained separately on the complete feature set of the labeled data. Classifiers then label unlabeled data and both unanimous agreement on labeling and class balancing are used as conditions to decide which soft-labeled data could be used for training set enlargement.

### 3.3.1 The SelfCo3 algorithm

Let  $L$  and  $U$  denote the labeled and the unlabeled set respectively, drawn independently from the same data set  $D$ . Let  $L_v$  be a labeled validation set drawn from  $D$  with null intersection with  $L$ . Let  $C_1, C_2$ , and  $C_3$  be three suitable selected classification models. Based on these assumptions, our SSL algorithm repeats the same sequence of steps described in Algorithm 5 until  $U$  is empty or there are no data for which classifiers unanimously agree on labeling. More in details, at the beginning of each iteration, all classifiers are trained on  $L$ , and are let classify  $U$ : three predictions for each datum  $u_i \in U$  are obtained. All the three classifiers are also used to classify  $L_v$  and the validation error  $VE$  is computed for each classifier. Collected informations are used as follows:

- the best classifier is chosen as the one corresponding to the minimum  $VE$ ;
- soft-labeled data  $S$  are selected as a subset of  $U$  based on classifiers unanimous agreement on the predicted labels;
- the selected subset  $S$  is eventually shrunk according to class balancing;
- selected soft-labeled data are added to  $L$ .

At the end of each iteration, classifiers are refined by the same copy of the soft-labeled data. At the end of all iterations, the so enlarged training set and the best classifier se-

Table 3.1: Benchmark reference datasets from UCI machine learning repository [94].

Dataset	Area	Attributes	Size	pos/neg
Australian	Financial	14	690	55.5%/44.5%
Bupa	Life	7	345	42.0%/58.0%
Colic	Life	27	368	63.0%/37.0%
Diabetes	Life	8	768	65.1%/34.9%
German	Financial	20	1000	70.0%/30.0%
Ionosphere	Physical	34	351	35.9%/64.1%
Kr-vs-kp	Game	36	3196	52.2%/47.8%
Tic-tac-toe	Game	9	958	65.3%/34.7%
Wdbc	Life	32	569	37.3%/62.7%

lected at the latest iteration are returned as components of the final classification model.

The proposed algorithm presents some key advantages: first of all, since diversity among classifiers is guaranteed by using three different classification algorithms, neither different views, nor any labeled data splitting is required. Therefore, the algorithm can be applied to common data-mining scenarios, and it can start with stronger classifiers (*i.e.* classifiers able to achieve good prediction performance also if they are trained with standard SL on the initial labeled data set). Moreover, together with the requirement of unanimous labeling agreement, classifiers strong enough to give a good prediction avoid the necessity of an explicit labeling confidence measurement. In fact, it could be reasonably assumed that if classifiers are different enough, the probability that all of them will predict a wrong label for a given unlabeled data is lower than the one achievable with standard majority voting. Avoiding an explicit confidence level measurement also makes the algorithm computing time saving. Finally, class balance preservation, avoiding bias of training steps, contributes to obtain stronger classification models at each algorithm iteration. Since the SelfCo3 algorithm is a wrapper method, it does not put any constraints on the classification algorithms selection. In this study, three different classification algorithms are used: a non linear Support Vector Machine (SVM) with Gaussian kernel, a non linear Kernel Fisher Discriminant (KFD) with Gaussian kernel, and a non linear classifier obtained combining a quadratic data mapping (working as a non linear feature extraction step) with a Fisher Linear Discriminant (QLFD) (see section 2.3.2).

It is noteworthy that, as for every Self-Training and Co-Training algorithm, the learning procedure performance are the result of a trade-off between the advantage introduced by the training set size increasing, and the disadvantage due to the possibility that classification noise increases during the learning procedure.

### 3.3.2 Experimental Results

In order to demonstrate the effectiveness of the proposed approach, in the following subsections we present and discuss results from two very different types of experiment:

benchmark numerical examples from the well known University of California-Irvine (UCI) Machine learning repository [94] and biomedical data from our TRUS video sequences database described in section 2.5. It means that we test both performance of SelfCo3 core as *general purpose* SSL algorithm and its ability at enhancing CAD system performance, *i.e.* its usefulness for our target application.

### Validation On Benchmark Datasets

The first set of experiments was carried out by using data from the UCI Machine learning repository [94]. Nine different UCI data sets were used. Some information on these data sets are tabulated in Table 3.1, where “Area” represents the data type, “Attribute” the number of attribute to be classified, “Size” the number of instances, and “pos/neg” shows the interclass data distribution. For each data set, about 25% of the available data were kept as test samples, while the rest was used as the pool of training samples, preserving the class balance.

Training set was split into a labeled  $L$  and an unlabeled  $U$  training set under four different label rates, namely 20%, 40%, 60% and 80%. Note that the label rate is defined as the percentage of labeled data *w.r.t.* the total amount of data. For instance, if the whole training set contains 100 data and the label rate is 20%, then 20 samples are labeled and 80 are unlabeled. Using this data partition, under each label rate, three independent runs with different random selection of  $L$  and  $U$  were performed. Classification error averaged over different runs was selected as performance metric.

SelfCo3 performance have been compared with performance of a slightly modified version of itself called SelfCo2 where only two classifiers are used: SVM with Gaussian kernel and SVM with linear kernel. SelfCo3 was also compared with two SSL algorithms proposed by Zhou [88], both based on the same guiding principles as SelfCo3: Tri-training and self-training2. In Tri-training two classifiers label data for a third. Classifiers are made different by manipulating  $L$  and the final hypothesis is generated via majority voting. Self-training2 uses the same three initial classifiers as those used by Tri-training. Each classifier is refined by the same copy of the newly labeled data and unlabeled samples are labeled via classifiers majority voting. The enlarged training set is then employed to train a single classifier which is used in prediction [88]. Note that neither Tri-training nor self-training2 use class balance as condition to enlarge the training set. Among classifiers used by Zhou in [88] we selected BP neural networks because it features the best mean performance improvement when used in Tri-training.

Tables 3.2(a) to 3.2 (d) sum up performance evaluation results for each UCI dataset under different label rates. In particular, for all the considered algorithms, the tables show the average error of the final model generated by the SSL algorithm (column “final”), the average error of the same classifier trained using only data belonging to  $L$  (column “init”), and the relative improvement. In fact each SSL procedure starts with a supervised learning step where each classifier is trained using only the original labeled data.



Table 3.2: Classification errors of the initial and final classification models and the corresponding improvements of SelfCo3, SelfCo2, Tri-training and self-training2 under different label rates.

(a) 20% label rate

Dataset	SelfCo3			SelfCo2			Tri-training BP			self-training2 BP		
	Init	Final	Impr%	Init	Final	Impr%	Init	Final	Impr%	Init	Final	Impr%
Australian	.214	.188	<b>12.0</b>	.177	.173	2.41	.200	.181	9.60	.149	.151	-1.30
Bupa	.422	.344	18.4	.422	.339	<b>19.7</b>	.337	.295	12.5	.348	.348	0.00
Colic	.206	.206	0.00	.194	.217	-11.4	.254	.232	8.60	.283	.243	<b>14.1</b>
Diabets	.382	.356	6.71	.369	.349	5.56	.286	.257	<b>10.3</b>	.264	.278	-5.30
German	.271	.252	<b>7.08</b>	.282	.278	1.51	.301	.287	4.90	.315	.311	1.30
Ionosphere	.156	.140	10.3	.161	.145	10.0	.201	.178	<b>11.3</b>	.167	.175	-4.50
Kr-vs-Kp	.090	.047	<b>48.5</b>	.057	.036	35.5	.034	.027	22.2	.030	.030	-1.40
Tic-tac-toe	.251	.049	<b>80.6</b>	.273	.375	-37.0	.071	.036	49.0	.015	.015	0.00
Wdbc	.071	.046	<b>35.0</b>	.071	.067	5.00	.042	.033	22.2	.032	.030	7.10
Average			<b>24.3</b>			<b>3.47</b>			<b>16.7</b>			<b>1.11</b>

(b) 40% label rate

Dataset	SelfCo3			SelfCo2			Tri-training BP			self-training2 BP		
	Init	Final	Impr%	Init	Final	Impr%	Init	Final	Impr%	Init	Final	Impr%
Australian	.158	.145	8.11	.158	.147	6.76	.152	.135	<b>11.4</b>	.155	.151	2.50
Bupa	.367	.311	<b>15.1</b>	.328	.344	-5.08	.353	.302	14.4	.340	.344	-1.10
Colic	.256	.228	10.9	.250	.217	<b>13.3</b>	.203	.192	5.40	.225	.232	-3.20
Diabets	.364	.336	7.74	.379	.338	<b>10.8</b>	.245	.233	5.00	.243	.257	-5.70
German	.273	.254	<b>7.03</b>	.282	.276	2.27	.296	.281	5.00	.282	.297	-5.20
Ionosphere	.113	.107	4.76	.113	.118	-4.76	.182	.159	<b>12.5</b>	.155	.163	-4.90
Kr-vs-Kp	.026	.016	<b>37.9</b>	.026	.019	27.6	.020	.015	25.0	.015	.020	-37.1
Tic-tac-toe	.107	.036	<b>66.7</b>	.131	.172	-31.4	.025	.021	16.7	.015	.014	9.10
Wdbc	.053	.043	20.0	.043	.053	-25.0	.040	.031	<b>23.5</b>	.052	.052	0.00
Average			<b>19.8</b>			<b>-0.61</b>			<b>13.2</b>			<b>-5.07</b>

(c) 60% label rate

Dataset	SelfCo3			SelfCo2			Tri-training BP			self-training2 BP		
	Init	Final	Impr%	Init	Final	Impr%	Init	Final	Impr%	Init	Final	Impr%
Australian	.186	.167	<b>10.3</b>	.147	.143	2.90	.148	.137	7.80	.134	.155	-15.9
Bupa	.339	.311	8.20	.322	.333	-3.45	.318	.283	<b>11.0</b>	.348	.383	-10.0
Colic	.206	.189	<b>8.11</b>	.194	.200	2.78	.192	.178	7.50	.240	.225	6.10
Diabets	.328	.300	<b>8.59</b>	.305	.310	-1.68	.248	.238	4.20	.249	.247	0.70
German	.261	.256	1.64	.252	.241	4.24	.303	.285	<b>5.70</b>	.307	.295	3.90
Ionosphere	.102	.102	0.00	.102	.091	10.5	.140	.285	<b>16.2</b>	.091	.095	-4.20
Kr-vs-Kp	.016	.014	16.7	.016	.014	16.7	.016	.013	<b>21.1</b>	.012	.011	6.90
Tic-tac-toe	.047	.032	<b>32.0</b>	.089	.124	-40.4	.022	.018	18.7	.026	.022	15.8
Wdbc	.050	.046	7.14	.060	.057	5.88	.040	.031	<b>23.5</b>	.031	.033	-7.70
Average			<b>10.3</b>			<b>-0.28</b>			<b>12.8</b>			<b>-0.49</b>

(d) 80% label rate

Dataset	SelfCo3			SelfCo3			Tri-training BP			self-training2 BP		
	Init	Final	Impr%	Init	Final	Impr%	Init	Final	Impr%	Init	Final	Impr%
Australian	.150	.143	4.29	.141	.141	0.00	.160	.135	<b>15.7</b>	.137	.145	-5.60
Bupa	.300	.283	5.56	.311	.317	-1.79	.345	.318	7.80	.356	.313	<b>12.0</b>
Colic	.233	.206	<b>11.9</b>	.189	.194	2.86	.199	.181	9.10	.200	.185	7.30
Diabets	.349	.318	<b>8.28</b>	.341	.331	3.01	.238	.220	7.30	.265	.286	-7.80
German	.250	.244	2.56	.256	.250	2.50	.288	.252	<b>12.5</b>	.287	.299	-4.20
Ionosphere	.081	.064	20.0	.081	.069	13.3	.117	.087	<b>25.8</b>	.095	.095	0.00
Kr-vs-Kp	.011	.010	4.17	.011	.011	-4.17	.013	.009	26.7	.010	.006	<b>42.3</b>
Tic-tac-toe	.037	.028	<b>25.0</b>	.037	.054	-45.0	.024	.019	17.6	.018	.025	-38.5
Wdbc	.036	.025	<b>30.0</b>	.046	.035	23.1	.045	.038	15.8	.051	.049	4.50
Average			<b>12.4</b>			<b>-0.69</b>			<b>15.4</b>			<b>1.11</b>

Results show that SelfCo3 has better average classification performance compared to the other algorithms under all label rates. In fact, averaging improvements across all the datasets and all label rate results into an improvement of about 16.7% for SelfCo3, 14.5% for Tri-training and -0.84% for self-training<sup>2</sup>. Also SelfCo2 obtains a positive average improvement, although limited to 0.47%. Performance improvements across data sets and label rates are also less stable than those achieved with SelfCo3. Moreover, SelfCo3 presents at each label rate the highest improvement for almost four data sets out of nine. Even if SelfCo3 and Tri-training have comparable performance considering this evaluation metric, nevertheless SelfCo3 has the best performance and features the biggest average improvement for label rates 20% and 40%, which are highly representative of the data distribution for our target application. The above considerations confirm that SelfCo3 algorithm is effective at achieving better classification performance *w.r.t.* standard SL and comparable performance *w.r.t.* existing SSL procedures.

### Application to PCa detection in TRUS images

As stated above, the purpose of the research presented in this chapter is exploiting SSL techniques to train CAD systems by using both diagnosed and undiagnosed samples reducing data collection costs and avoiding collected data wasting. In particular, the proposed SSL algorithm is aimed at improving classification performance of a CAD tool for PCa detection in ultrasound images. As already said, in such a scenario reducing the number of diagnosed samples necessary to achieve good performance is mandatory to reduce both costs and discomfort for the patients (see section 2.2).

For this reason, the performance of the proposed approach were also evaluated on experimental data from US images.

The CAD system used in this study has basically the same structure as rtCAB (see section 2.5). Therefore, the CAD processing flow consists of three main steps after preprocessing: (i) Segmentation; (ii) Features Extraction; (iii) ROIs classification. The main difference is constituted by the classification algorithm: while the reference CAD scheme uses a non linear classifier obtained combining a feature extraction stage based on GDA with Gaussian Kernel and a FLD (GDA+FLD) trained in supervised manner, the detection tool here described performs classification by means of a non linear classifier based on the *kernel trick* principle and selected by SelfCo3. The features set used in this study is the one composed by 10 features selected by an hybrid feature selection procedure among Nakagami's distribution parameters and Unser's textural features as explained in subsection 2.3.2. The detection tool was trained by using the TRUS video sequence database detailed in subsection 2.5.1. As stated above, from this ground truth database only ROIs belonging to cores with tumor covering more than 70% of the sample and ROIs from healthy cores can be used to train the CAD system with standard SL.

In this scenario, SSL procedure allows to enhance classifier performance by using the entire amount of collected data in the training. For this reason, in the application context, two kind of evaluations were performed: in the first case (Evaluation 1), the

dataset  $D$  is constituted by ROIs clearly identified as suspicious or not-suspicious.  $D$  is then randomly split in labeled ( $L$ ) and unlabeled ( $U$ ) datasets as previously done for UCI datasets. In the second case (Evaluation 2), ROIs belonging to malign and benign cores compose  $L$ , whereas ROIs coming from unclassifiable cores compose  $U$ , since, as stated above, they are truly unlabeled data. In both cases, SelfCo3 was compared with the reference supervised trained classification system based on GDA+FLD. Again SelfCo2 was used as a further term of comparison.

In both evaluations the data sets contained not-suspicious ROIs coming from healthy patients (162 BB cores, 3240 ROIs), suspicious ROIs coming from unhealthy patients (23 M cores, 460 ROIs), and unlabeled ROIs coming from unknown cores (107 U and UM cores, 2140 ROIs).

### Evaluation 1

During the first evaluation, training and testing sets were obtained by a random extraction of a balanced number of suspicious and not-suspicious ROIs.

Seven label rates ranging from 21% to 85% were considered. Under each label rate, five independent runs with different random selection of  $L$  and  $U$  sets were executed and performance of SelfCo3 were evaluated by means of average classification error.

For each label rate, Table 3.3 reports the average classification error of GDA+FLD, the average error of both the final model generated by SelfCo3 and its supervised trained counterpart, and the relative improvements for both SelfCo3 and SelfCo2 (columns SL3, SelfCo3, SL2 and SelfCo2). Results show that the final classification model generated by SelfCo3 outperforms its supervised-trained counterpart for all label rates obtaining an average improvement equal to 6.2%. Best improvements for small label rates confirm the algorithm effectiveness at improving learning performance when only a small amount of labeled data is available. It is worth to note that improvements are also significant for large label rates, indicating that SelfCo3 can effectively enhance learning performance with a larger amount of labeled data. The proposed classification model also outperforms GDA+FLD at all label rates, confirming that SelfCo3 is effectively useful to enhance CAD training performance. Performance of the final model selected by the SSL procedure trained in supervised manner are comparable to the ones featured by GDA+FLD, confirming the effectiveness of the selected classification algorithms for the specific CAD application. Table 3.3 also shows that SelfCo2 is suitable for the target application since the final model generated by SelfCo2 outperform both GDA+FLD at all label rates, and its supervised trained counterpart excepting at 67%. Nevertheless, SelfCo3 performs better than SelfCo2 in terms of performance enhancement, confirming that using three different classification algorithms increases the robustness of the SSL procedure.

Table 3.3: Evaluation 1: classification errors of compared classification models and relative improvements.

Label Rate	GDA+FLD	SL3	SelfCo3	SL2	SelfCo2	SelfCo3 vs. SL3	SelfCo3 vs. GDA+FLD	SelfCo2 vs. SL2	SelfCo2 vs. GDA+FLD
21%	.330	.349	.300	.327	.311	14.0%	8.98%	4.98%	5.58%
36%	.325	.317	.297	.320	.310	6.31%	8.62%	3.00%	4.43%
48%	.323	.319	.291	.312	.320	8.77%	9.90%	0.51%	3.96%
57%	.314	.308	.295	.316	.306	4.16%	6.12%	3.04%	2.54%
67%	.313	.303	.299	.302	.308	1.32%	4.35%	-2.12%	1.53%
74%	.322	.314	.296	.310	.306	5.61%	8.19%	1.29%	4.96%
85%	.322	.310	.300	.317	.309	3.10%	6.72%	2.52%	3.98%

## Evaluation 2

For the second evaluation, the whole available dataset was first divided into labeled ( $L$ ) and unlabeled ( $U$ ) data sets.  $U$  contains only ROIs belonging to U and UM cores, while  $L$  contains ROIs from BB and M cores (see Table 2.3). The 25% of the data in  $L$  were kept to constitute the testing set while the remaining part was used for training preserving class balance.

In this case, several label rates ranging from 24% to 79% were obtained by keeping different amount of unlabeled data while maintaining a fixed amount of labeled data. Under each label rate, five independent runs with different random selections of unlabeled ROIs were performed and performance of SelfCo3 were evaluated on the average classification error. Table 3.4 sums up the results of this evaluation, ordered as described for Table 3.3. Once again, the final classification model generated by SelfCo3 outperforms both its supervised-trained counterpart and GDA+FLD. It is worth noting that the achieved improvements are attributed to the addition of ROIs belonging to unknown cores, which were unusable without a semi-supervised approach to CAD training. It can therefore be stated that our SSL algorithm is effective at providing performance enhancement without requiring any additional labeled data and avoiding collected data wasting in PCa diagnosis. Concerning SelfCo2, as in the previous evaluation it can be stated that although this algorithm is also useful to enhance CAD training performance, its final classification model is outperformed by the SelfCo3 final model.

Case studies performed on ground truth images from biopsy findings show that SelfCo3 is able to both enhance performance achievable with a standard SL algorithm and to outperform a previously developed application-driven classification framework without requiring any additional labeled data and making full use of the whole ground truth dataset. The improvement obtained on average by the final model generated by SelfCo3 across all considered label rates, *w.r.t* its supervised trained counterpart and the state-of-art classification system is equal to 8.23% and 12.9% respectively. Starting from these encouraging results to further improve classification performance of the CAD tool we observed that: (i) if some informations about unlabeled data are available they can be

Table 3.4: Evaluation 2: classification errors of compared classification models and relative improvements.

Label Rate	GDA+FLD	SL3	SelfCo3	SL2	SelfCo2	SelfCo3 vs. SL3	SelfCo3 vs. GDA+FLD	SelfCo2 vs. SL2	SelfCo2 vs. GDA+FLD
24%	.305	.290	.259	.284	.280	10.9%	15.2%	1.54%	8.33%
32%	.305	.297	.257	.290	.286	13.6%	15.8%	1.51%	6.32%
42%	.296	.286	.261	.284	.283	8.59%	11.6%	0.31%	4.15%
52%	.305	.292	.260	.285	.290	10.8%	14.6%	-1.85%	4.88%
61%	.305	.278	.267	.280	.288	3.79%	12.3%	-2.82%	5.75%
68%	.310	.296	.274	.301	.293	7.69%	11.6%	2.62%	5.38%
79%	.296	.275	.269	.287	.289	2.23%	9.17%	-0.61%	2.66%

used to help the SSL algorithm to label the unlabeled data for training set enlargement; (ii) an optimized training of each classifier involved in the SSL procedure can be applied to enhance the algorithm performance, since, by doing so, learning procedure can start with stronger classifiers.

### 3.4 A new Learning Procedure for computer aided PCa detection in ultrasound images

#### Core knowledge introduction

Sometimes, although data labels are not available, additional information on unlabeled data class membership could be available. Such complementary information may be exploited to reduce SSL algorithms sensitivity to classification error propagation, *i.e.* the probability of adding to the training set data on which classifiers agree on a wrong labeling is lowered. In practical applications we may also have statistical information about a group of ROIs instead of precise ROIs labeling. The computer aided PCa detection is such an application. In fact, the tumor core volume represents a prior knowledge information on a group of ROIs. SelfCo3 performance can be enhanced by exploiting this *core-knowledge*. Basically, the core knowledge is used to impose an additional requirement to the iterative labeling procedure. SelfCo3, extended with the core-knowledge constitutes a new SSL algorithm, named SelfCo3Core, specifically designed for training a CAD tool for PCa detection in TRUS images.

#### Mixed Supervised/Semi-supervised Learning Procedure

As stated above, the proposed SSL algorithm starts training three different classifiers on the original labeled data: the stronger the initial classifiers are, the lower the probability of incorrect unknown data labeling is, and, as a consequence, performances of the SSL algorithm increase. That above justifies the introduction of an additional supervised learning step to optimize classifiers before involving them in the SSL procedure. Con-

cerning the SSL procedure itself, since its performances depend on both unlabeled data type (*i.e.* ROIs core membership) and label rate used for training, it is better to perform the SSL step for different label rates and for different unlabeled datasets in order to select the best final classification model. Based on these assumptions, we propose to use a mixed supervised/semi-supervised learning procedure and divide them in two parts: a Supervised Model Selection (SMS) concerning the optimized design of classifiers used by the SSL procedure, and a Semi-Supervised Model Selection (S-SMS) which exploits the proposed SSL algorithm to select the best final classification model.

### 3.4.1 SelfCo3Core

Let  $L$  and  $U_s$  be some labeled and unlabeled set respectively, each drawn independently from the same data set  $D$ : while  $L$  contains ROIs from M and BB cores (see Table 2.3),  $U_s$  is made up of ROIs from UM and U cores. Let  $L_v$  be a labeled validation set drawn from  $D$  in such a way that cores from a given patient will belong either to  $L$  or to  $L_v$ . Let  $U_{tot}$  be a data set made up of all available unlabeled ROI and the corresponding tumor core volume. Let  $C_1$ ,  $C_2$  and  $C_3$  be three suitably selected classification models.

On these bases, SelfCo3Core repeats a sequence of operations until  $U_s$  is empty or there are no data for which classifiers unanimously agree on labeling. At the beginning of each iteration  $C_1$ ,  $C_2$  and  $C_3$  are trained on  $L$ , and are let label  $U_s$ : three predictions are obtained for each  $u_i$  belonging to  $U_s$ . All the three classifiers are also tested over  $L_v$  and Validation Error ( $VE$ ) is computed for each of them. Finally, they are used to classify  $U_{tot}$  computing an estimated tumor volume for each core. Collected information are used as follows:

- The best classifier is chosen as the one corresponding to the minimum  $VE$ ;
- Soft-labeled data are selected as a subset of  $U_s$  based on classifiers unanimous agreement on the predicted labels;
- The selected subset is eventually shrunk according to tumor core volume estimation results: only data belonging to cores whose estimated core volume is within  $\pm 20\%$  of the real one for at least one classifier are considered;
- The selected subset is eventually shrunk according to class balancing;
- Selected soft-labeled data are added to  $L$ .

At the end of each iteration, classifiers are refined by the same copy of the soft-labeled data. At the end of all iterations, the so enlarged training set and the best classifier are returned as components of the final classification model. The flowchart in figure 3.1 describes the learning algorithm working.

Therefore the main difference between SelfCo3 and SelfCo3Core consists in using tumor core volume as an information about a group of unlabeled ROIs to strengthen

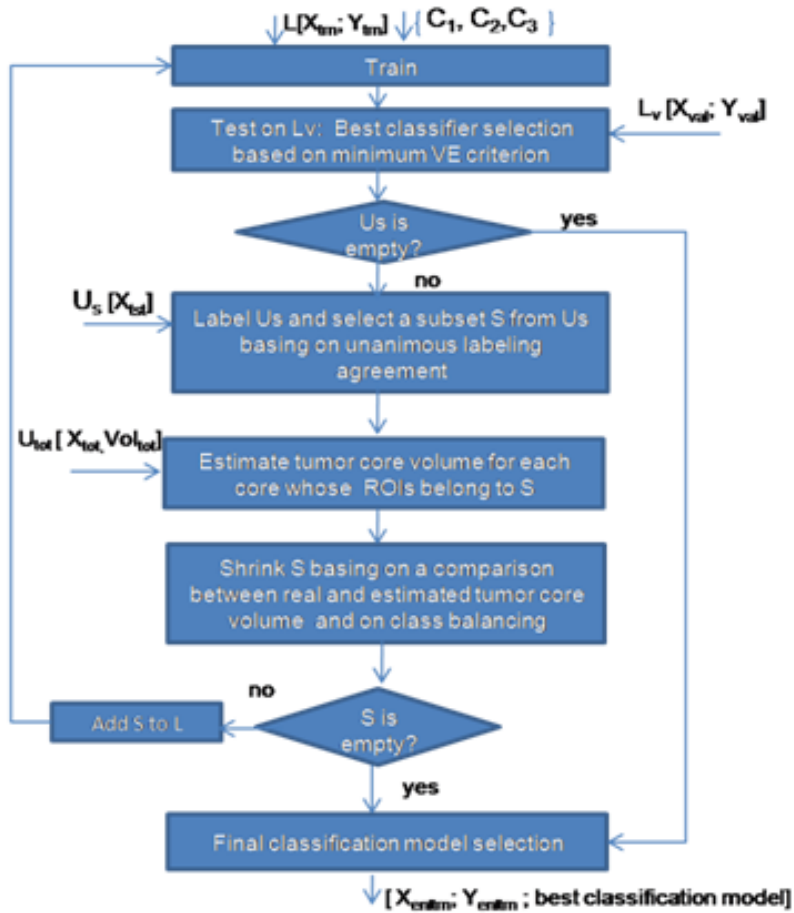


Figure 3.1: SelfCo3Core Flowchart

the algorithm. By comparing classifiers predictions with histology, only ROIs whose estimated core volume is in good agreement with the real one are used for training set enlargement. This comparison between tumor core volume estimation and its real value reduces the probability of ground truth uncertainty to negative imping on labeling accuracy.

Our learning procedure extends SelfCo3Core reckoning that classifier parameters can be conveniently tuned according to practical environment and that the initial labeled training set can be different for each classifier. If accurate parameters tuning and training set selection are performed, SelfCo3Core initial classifiers will be stronger and SSL performance will increase. As a consequence, we divided the learning procedure in two parts as shown in Figure 3.2 (gray boxes): a Supervised Model Selection (SMS) concerning the selection of the best model for each classifier kind involved into SSL, and a Semi-Supervised Model Selection (S-SMS) which uses SelfCo3Core to select the best final classification model.

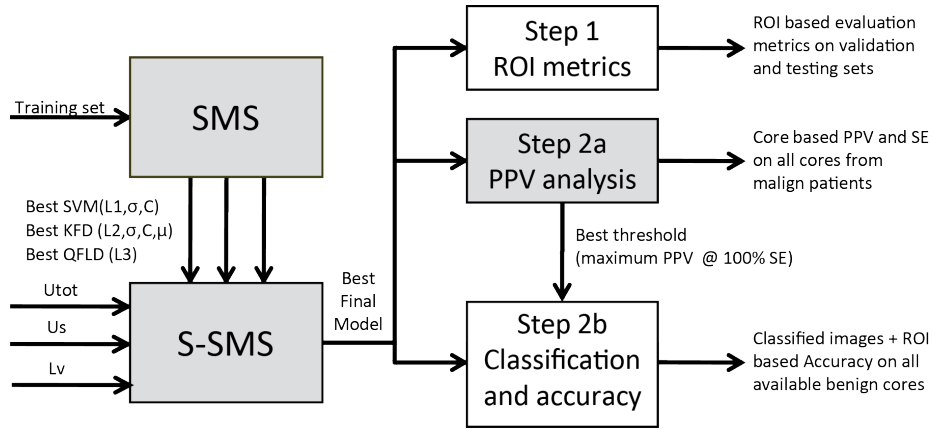


Figure 3.2: Training and validation flows: training steps are represented by gray boxes while validations by white ones.

### 3.4.2 Supervised Model Selection

SMS consists in the choice of the best training set for each involved classifier, the variance  $\sigma$  of the Gaussian kernel for both SVM and KFD, the best regularization constant  $C$  for SVM and for the 1D-SVM used for the KFD bias selection, and the best regularization constant  $\mu$  for KFD. This selection is performed by means of cross-validation: for several values of  $\sigma$ ,  $C$ , and  $\mu$ , a number of tests over different, randomly selected, training and validation sets is performed. Sizes of these sets differs for each classifier as they are chosen according to the number of features and model parameters following [95]. In particular, the size of the validation set is selected according to:

$$f = g \cdot \sqrt{\frac{C(\frac{N}{\alpha^2})}{h_m}} \quad (3.1)$$

where  $f$  is the fraction of dataset that should be reserved for validation;  $g$  is the fraction of dataset that should be reserved for training;  $C$  is the constant of Chernoff bound ( $C=1.5$ );  $h_m$  is the largest complexity of the families of models considered;  $N$  is the number of these families;  $\alpha$  is the risk of being wrong. The size of the training set is thus consequently selected reckoning that:  $f + g = 1$ .

The resulting ROI-based VE, Sensitivity (SE), Specificity (SP), Accuracy (ACC), and PPV averaged over different tests are analyzed for each model in order to select the best parameters as those corresponding to minimum mean VE. Given a set of parameters, the training sets are selected as those minimizing the VE. The choice of VE as ROI-based criterion is arbitrary: other metrics may be used as well. The output of SMS is constituted by the training set  $[X_{trn}, Y_{trn}]$  and the required parameters for each classifier.

The SMS procedure is described in Algorithm 6.



### 3.4.3 Semi-Supervised Model Selection

In this part of the learning procedure, the classification models chosen during SMS are employed into a SSL procedure based on SelfCo3Core, slightly modified to be combined with the previously described step. By using different training sets, selected by SMS, the SSL procedure starts with the best models according to SL. The labeled set for SelfCo3Core is therefore replaced by three sets  $L_1, L_2, L_3$ . Then, for a fixed number of times, unlabeled ROIs are mixed and a portion of them is used as unlabeled data set for SelfCo3Core. For each of these iterations, SelfCo3Core is applied for different label rate. In fact, since the SSL procedure performance depends on both the number of ROIs and their cores membership, it is better to take into account different pools of unlabeled data to select the best final classification model. Therefore, SelfCo3Core is applied  $N_{tests} \times N_p$  times, where  $N_{tests}$  is the number of different unlabeled data mix and  $N_p$  is the number of label rates considered. At the end, the  $N_{tests} \times N_p$  values of the previously mentioned ROI-based performance statistics computed over  $L_v$  are used to select the best classification model, made up by an algorithm whose parameters were chosen during SMS, and a new, enlarged, training set. The model is chosen by a two step selection procedure: first a model for each label rate is selected, then a specific label rate is fixed. Both these steps are based on VE minimization. The second step of the mixed SL/SSL procedure is described in Algorithm 7.

---

**Algorithm 6:** Pseudo-code describing the Supervised Model Selection (SMS) step of the learning procedure.

---

**Input:**  $L=[X_{trn}, Y_{trn}]$  training set: data matrix and labels;  $L_v=[X_{val}, Y_{val}]$  validation set: data matrix and labels;

**Select best SVM**

**for**  $iter=1$  to  $nTests$  **do**

select random  $[X_{trn}, Y_{trn}]$  and  $[X_{val}, Y_{val}]$ ;

**for**  $C=C_{min}$  to  $C_{max}$  **do**

**for**  $\sigma=\sigma_{min}$  to  $\sigma_{max}$  **do**

Train with L using  $\sigma$  as Gaussian kernel variance and C as SVM regularization constant;

Test on  $L_v$ ;

Compute ROI-based testing error: TSTerr on  $L_v$ ;

Compute others ROI-based evaluation metrics (ACC, SE, SP, PPV);

Average ROI-based TSTerr (meanTSTerr) and others evaluation metrics over different tests;

Select best parameters  $(\sigma_1, C_1) = \text{argmin}(\text{meanTSTerr}(\sigma, C))$ ;

Select best training set  $(L_1=[X_{trn1}, Y_{trn1}])=\text{argmin}(\text{TSTerr}([X_{trn}, Y_{trn}]; \sigma_1, C_1))$ ;

**Select best KFD**

**for**  $iter=1$  to  $nTests$  **do**

select random  $[X_{trn}, Y_{trn}]$  and  $[X_{val}, Y_{val}]$ ;

**for**  $\mu=\mu_{min}$  to  $\mu_{max}$  **do**

**for**  $C=C_{min}$  to  $C_{max}$  **do**

**for**  $\sigma=\sigma_{min}$  to  $\sigma_{max}$  **do**

Train with L using  $\sigma$  as Gaussian kernel variance, C as regularization constant of the 1D-SVM used to optimize the bias of KFD and  $\mu$  as the KFD regularization constant;

Test on  $L_v$ ;

Compute ROI-based testing error: TSTerr on  $L_v$ ;

Compute others ROI-based evaluation metrics (ACC, SE, SP, PPV);

Average ROI-based TSTerr (meanTSTerr) and others evaluation metrics over different tests;

Select best parameters  $(\sigma_2, C_2, \mu_1) = \text{argmin}(\text{meanTSTerr}(\sigma, C, \mu))$ ;

Select best training set  $(L_2=[X_{trn2}, Y_{trn2}])=\text{argmin}(\text{TSTerr}([X_{trn}, Y_{trn}]; \sigma_2, C_2, \mu_1))$ ;

**Select best QFLD**

**for**  $iter=1$  to  $nTests$  **do**

select random  $[X_{trn}, Y_{trn}]$  and  $[X_{val}, Y_{val}]$ ;

Train with L;

Test on  $L_v$ ;

Compute ROI-base testing error: TSTerr on  $L_v$ ;

Compute others ROI-based evaluation metrics (ACC, SE, SP, PPV);

Select best training set  $L_3=[X_{trn3}, Y_{trn3}] = \text{argmin}(\text{TSTerr}([X_{trn}, Y_{trn}]))$ ;

**Output:**  $[L_1, \sigma_1, C_1, L_2, \sigma_2, C_2, \mu_1, L_3]$

---

---

**Algorithm 7:** Pseudo-code describing the Semi-Supervised Model Selection (S-SMS) step of the learning procedure.

---

**Input:**  $L_1=[X_{trn1}, Y_{trn1}]$ ,  $L_2=[X_{trn2}, Y_{trn2}]$ ,  $L_3=[X_{trn3}, Y_{trn3}]$ , labeled training sets;  
 $L_v=[X_{val}, Y_{val}]$ , labeled validation set;  
 $U[X_{tst}]$  (Group of ROIs coming from unknown cores),  $U_{tot}=[X_{cores}, Vol_{cores}]$  (All the ROIs coming from unknown cores along with information about core volume)  
Classification models  $C_1, C_2, C_3$  (best classification models selected in the previous part of the learning process)

**for**  $i=1$  to  $nTests$  **do**

**for**  $i=1$  to  $nP$  **do**

        select random  $U[X_{tst}]$ ;

**while** not ( $U$  is empty or there are not data for which classifiers unanimously agree on labeling) **do**

**for**  $i=1$  to 3 **do**

                Train classifier (i) with  $L(i)$ ;

                Test classifier (i) on  $U$  and label  $U$  according to its predictions;

                Test classifier (i) on  $U_{tot}$  and label  $U_{tot}$  according to its predictions;

                Test classifier (i) on  $L_v$  and compute the validation error;

            Perform majority voting on predicting labels;

            Selects a subset  $S$  from  $U$  based on majority voting: only data for which classifiers unanimously agree on labeling are selected;

**for**  $i=1$  to 3 **do**

                Estimate tumor core volume for each core whose ROIs belong to  $S$   
                based on classifiers labeling of  $U_{tot}$ ;

            Eventually shrink  $S$  based on a comparison between real and estimated tumor core volume: only data belonging to a core whose estimated core volume is within  $\pm 20\%$  of the real one, for at least one classifier, are considered;

            Eventually shrink  $S$  balancing classes;

            Add  $S$  to  $L_1, L_2$  and  $L_3$ ;

        Select final classification model according to the minimum validation error at latest iteration;

        Test final classification error on  $L_v$  and compute TSTerr on  $L_v$ ;

        Compute others ROI-based evaluation metrics

Select the best unlabeled dataset for a given original unlabeled training set and (*i.e.* upstream its partitioning under different label rates) and a given label rate. Selection is performed minimizing TSTerr.

**Output:** [Enlarged training set, best classification model, best label rate]

---

### 3.4.4 Experimental Results

In order to demonstrate the effectiveness of the complete learning procedure and therefore evaluate the proposed CAD tool performance, two steps were considered as shown in Figure 3.2. The first assesses learning efficiency by comparing the obtained classification models in terms of classification error over specific validation and testing sets. The second verifies the effectiveness of the whole detection tool in clinical environment through the comparison to standard biopsy protocol.

#### CAD performance evaluation: Step 1

To evaluate CAD performance, a comparison to the state-of-art classification system based on GDA+FLD, described above (see section 2.5), was performed. For fair comparison SMS was extended to train also this classifier.

The data set for this study is made up of 292 cores from the collected database. It contains ROI from BB cores (162 cores, 3240 ROI), ROIs from M cores (23 cores, 460 ROIs), and ROIs from U and UM cores (107 cores, 2140 ROIs). The total amount of labeled data was divided into three parts: a training set (512 ROIs) and a validation set (124 ROIs) used respectively into the SMS and S-SMS, and a testing set (280 ROIs) used to evaluate the generalization ability of the obtained classification models. Class balancing was preserved within all these sets.

During SMS, 200 random data set of labeled ROI are generated and split into training and validation set. A Cross Validation (CV) over 30 values of  $\sigma$  equally distributed between 0.1 and 6, 3 values of  $C$  (10, 100, 1000), and 2 values of  $\mu$  ( $10^{-3}$ ,  $10^{-4}$ ) was performed on each. Then, during S-SMS, SSL is performed for 10 different original unlabeled set and 8 different label rates ranging from 15% to 91%. Label rates were obtained by keeping different amount of unlabeled data while maintaining a fixed amount of labeled data. The validation set used to select the final model is the same used within SelfCo3Core internal runs.

Table 3.5 reports the outcomes of these selection steps. VE of the GDA+FLD classifier, of the final model generated by SelfCo3Core, and of its supervised trained counterpart are shown for each label rate as well as their relative improvements. Values obtained for the label rate selected by the S-SMS have been boldfaced. The final column reports the classification algorithm chosen by S-SMS. Results show that the final model generated by SSL outperforms its SL counterpart, confirming that SelfCo3Core is effective at enhancing learning performance *w.r.t.* a standard SL algorithm. This is highly pleasing as improvements are due only to the introduction of unlabeled ROI, unusable without a SSL approach. SelfCo3Core is hence effective at providing a performance enhancement without requiring any additional labeled data. The final model generated by SSL outperforms GDA+FLD too, confirming its efficacy for the target application: an improvement in excess of 20% is obtained at the selected label rate of 53%.

To evaluate obtained classification models in terms of generalization ability, their performance over a completely unknown test set were evaluated in terms of classification

Table 3.5: Classification errors of compared classification models and relative improvements.

Label Rate	GDA+FLD	SL	SSL	SSL vs. SL	SSL vs. GDA	Class. Alg.
15%	.328	.328	.320	2.38%	2.38%	KFD
20%	.328	.320	.312	2.44%	4.76%	SVM
24%	.328	.336	.320	4.65%	2.38%	QFLD
29%	.328	.328	.305	7.14%	7.14%	KFD
38%	.328	.320	.281	12.2%	14.3%	SVM
<b>53%</b>	<b>.328</b>	<b>.320</b>	<b>.258</b>	<b>19.5%</b>	<b>21.4%</b>	<b>SVM</b>
73%	.328	.320	.297	7.31%	9.52%	SVM
91%	.328	.320	.312	2.44%	4.76%	SVM

error. On this testing set the classification error of GDA+FLD is equal to 43.2%, while the ones of the final model generated by the SSL procedure and its supervised-trained counterpart are respectively equal to 40.0%, and 39.2%. The final model generated by our learning procedure outperforms both its supervised trained counterpart and the state-of-art classification model also operating on an unknown testing set, showing relative improvements respectively equal to 2% and 9.25%. It can thus be stated that SSL is also effective at enhancing prediction performance maintaining a good generalization ability.

### CAD performance evaluation: Step 2

The standard biopsy protocol efficiency can be conveniently expressed in terms of PPV. Known the gland to be pathological, PPV represents the probability of a single core to be PCa positive according to the detection tool. This figure is low due to the high number of false positives, *i.e.* cores which were not useful for confirming the presence of PCa. To improve the efficiency of the biopsy protocol, it is mandatory to increase the PPV, reducing the number of cores while maintaining the same detection rate over patients. To verify the effectiveness of the proposed method at achieving this goal, a so called *PPV Analysis* is performed: the classification model generated by the procedure described in Step 1 is tested over a set of 26 malign patients (247 cores) to evaluate PPV improvement *w.r.t.* randomized biopsy cores. This analysis can also be considered a further learning step, as CV is performed on classification model threshold to select the one maximizing the median PPV over the whole dataset under a 100% median sensitivity constraint over patients. CV is performed by varying the threshold value between the minimum and the maximum of the discriminant function: this allows to choose between models spanning from 100% ROI-based SP to 100% ROI-based SE. For each threshold value, the classification model is applied to the 247 cores and some core-based statistics are computed for each patients. A core is considered unhealthy if at least one of its ROI is classified as malign. The final threshold value is selected as that one providing the largest median PPV over all pathological patients constrained to SE maximization. The proposed method was compared to the state-of-art CAD tool in order to evaluate the

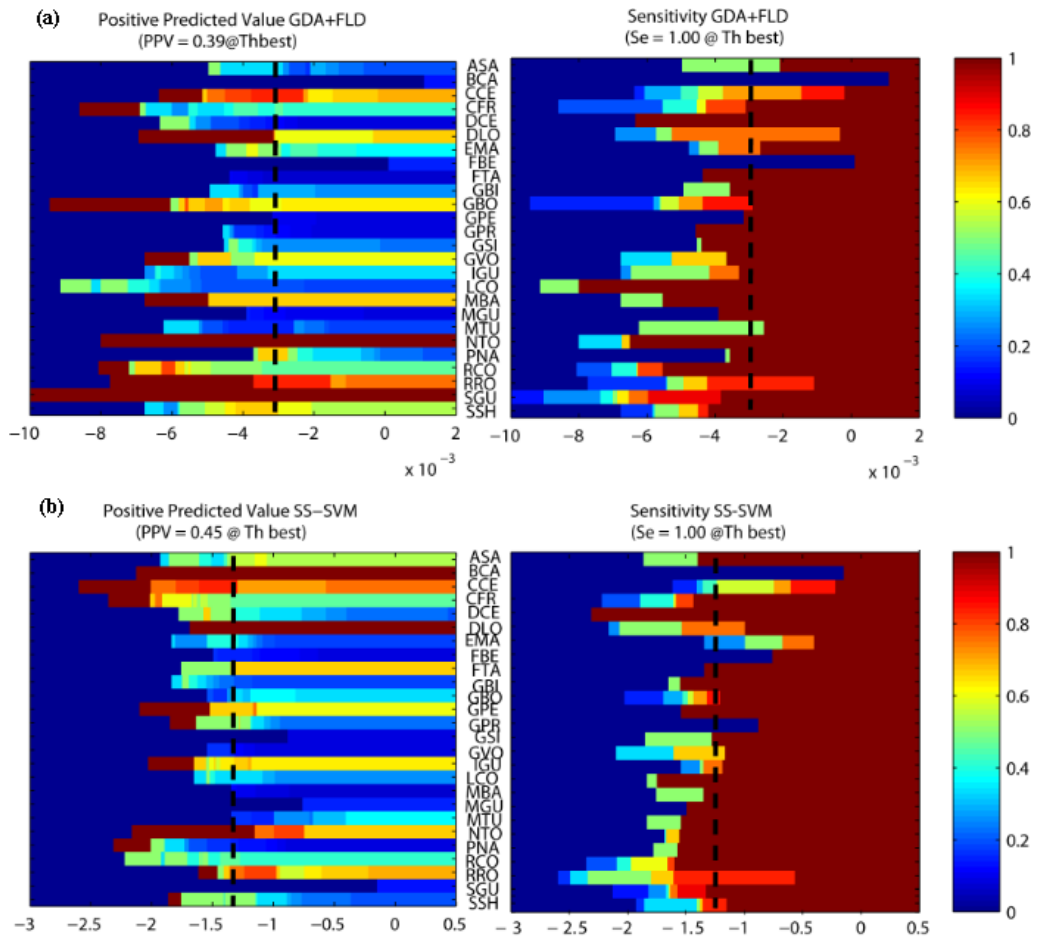


Figure 3.3: The core based PPV and SE of both the state-of-art (a) and the proposed detection tool (b). The x-axis represents the threshold values. At right the color bar shows the color coding for the SE and PPV value. Each horizontal bar represents a patient from the database identified by a three letter code: the color in each bar indicates how the respective statistics changes as a function of the classifier threshold over the given patient. The vertical dashed line is the best threshold  $Th_{best}$  selected by the learning scheme.

performance improvement.

Figure 3.3 depicts the core based PPV and SE for two different classification models: GDA+FLD trained by using only SMS on Figure 3.3(a) and SVM trained with the complete SMS/S-SMS learning procedure on Figure 3.3(b). Each horizontal bar represents a patient from the database, identified by a three letter code: the color in each bar indicates how the respective statistic changes as a function of the threshold over the given patient. The vertical dashed line is the best threshold  $Th_{best}$  selected by the system to discriminate between suspicious or not-suspicious ROIs. At  $Th_{best}$  semi-supervised trained SVM (SS-SVM) provides a median PPV of 45%, while supervised trained GDA+FLD scores 39%, both with a median SE of 100%. As a comparison, on the same dataset, the

standard biopsy protocol (8-12 cores) features a PPV of 29%. The proposed method improves PPV of 55% without losing diagnostic power. Moreover, as explained in section 2.2, if  $p_0$  is the standard prostate biopsy protocol PPV, and  $p_1 > p_0$  the detection tool PPV, the number of cores can be reduced according to  $N_1 = N_0 \lceil \frac{\log(1-p_0)}{\log(1-p_1)} \rceil$ . Therefore, our detection tool allows to reduce the number of cores from 8-12 to 6.

Note that, using this threshold selection, the CAD tool is tuned for reducing false positives by avoiding false negatives. To confirm this statement, both the classification systems were tested on all available benign cores (260 cores), and ROI-based accuracy was measured for the same threshold value considered in the PPV analysis. At  $Th_{best}$ , GDA+FLD shows an accuracy of 91.3%, while SS-SVM resulted into 94.4%, in accord with theory.

A further step toward the verification of the effectiveness of the proposed detection tool as for biopsy guide was taken by letting the CAD classify a video sequence of images and comparing classification results to the ground truth. The same evaluation was carried out on the reference tool: results were compared to assess performance enhancement. Examples of classified images and their corresponding B-Mode with superimposed needle trace are shown for an unhealthy patient (Figure 3.4(a) and 3.4(b)) and for an healthy one (Figure 3.4(c) and 3.4(d)). The frames corresponding to biopsy sampling are on the top, the same with superimposed the binary risk map are on the bottom, respectively, on the left the map elaborated with the reference CAD, on the right the result of the proposed CAD. Patient were diagnosed as healthy or unhealthy as a result of histological analysis. For the unhealthy patient, the considered tissue sample histology indicates a tumor volume equal to 90%, while tumor volume estimated considering the image portion marked by the CAD is equal to 70% for the new detection tool and to 84% for the state-of-art ones. For the healthy patient, the state-of-art classification system assigned to a benign core a tumor volume equal to 35%, while the proposed classification system does not highlight any presence of PCa in the tissue sample. Hence an unnecessary bioptic sampling would have been avoided for the healthy patient, while the presence of PCa would have been correctly diagnosed for the unhealthy one with an acceptable accuracy. These results confirm the effectiveness of the proposed classification framework at enhancing performance of both the state-of-art biopsy protocol and CAD.

To be used for biopsy guide, the CAD tool must be able to preserve the real-time nature of the echographic examination. Exploiting the application intrinsic parallelism, both features extraction and classification steps can be implemented making full use of CUDA parallel processing capabilities [28]. That being so, considering a  $2792 \times 250$  image segmented into  $100 \times 10$  pixels ROIs, our detection tool is capable of delivering a frame rate of about 40fps confirming its effectiveness for prostate biopsy guidance.

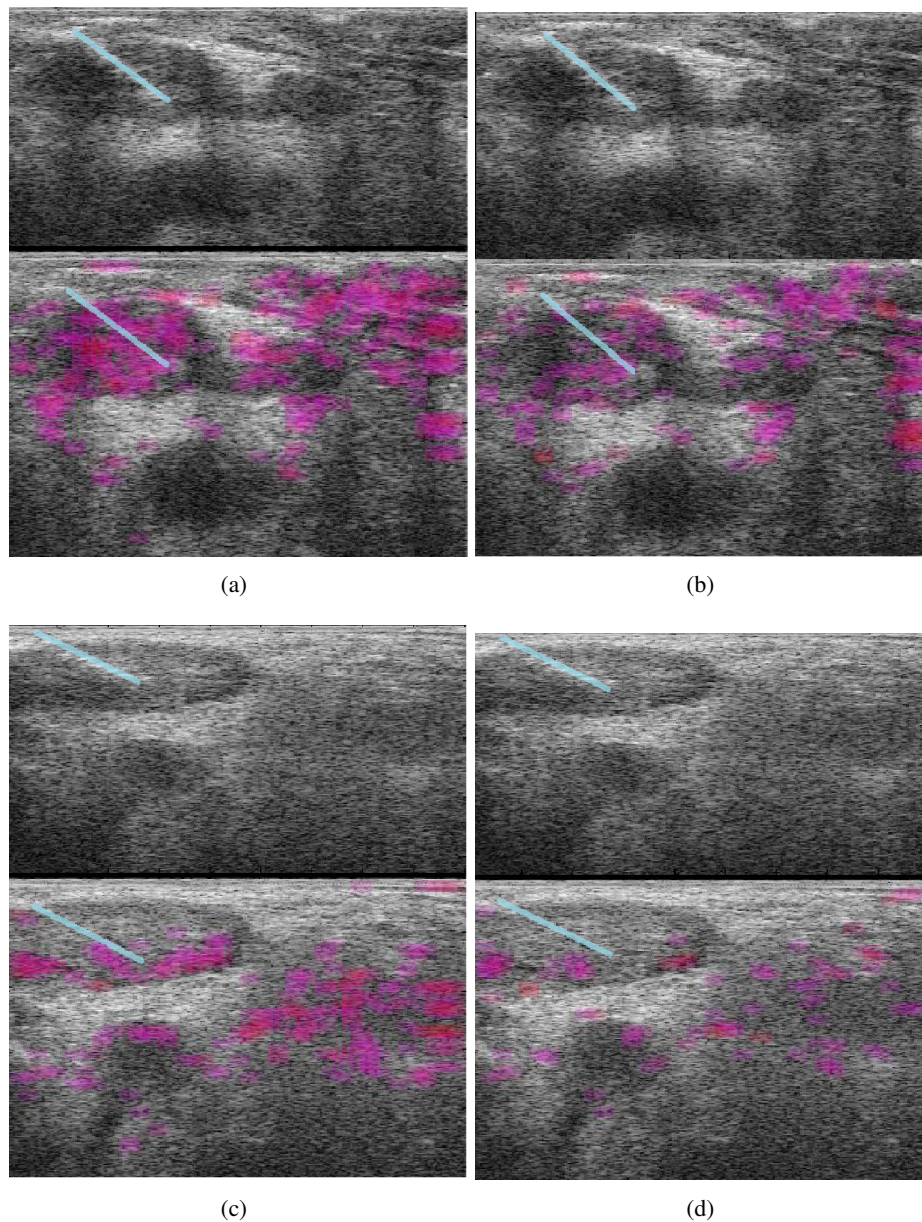


Figure 3.4: Example of B-Mode images with superimposed needle trace and the corresponding image classified by the CAD tools for a unhealthy (a-b) and healthy (c-d) patient. Images (a-c) refer to the state-of-art CAD tool while images (b-d) refer to the proposed one. For each image the frame on the top is the one corresponding to the biopsy sampling, while the previous frame with superimposed the binary risk map generated by the CAD is shown on the bottom.



## 3.5 Conclusions

Several common data mining scenarios are characterized by the presence of a small amount of labeled data and a huge amount of unlabeled ones. In many practical classification tasks obtaining data is cheap but obtaining the labels costs a lot of time, effort, and/or money. These are the situations in which SSL performs well since it allows to train the classification system using both the labeled and unlabeled data. The development of a reliable CAD system is also a machine learning task where SSL can be extremely valuable. In fact achieving good classification performance requires a large amount of training samples. Nonetheless data must be collected during medical examinations along with the corresponding diagnoses made by expert physicians. In particular, in the context of computer aided PCa detection, reducing the number of diagnosed samples necessary to achieve good performance is mandatory to contain both costs and discomfort for the patients, and to use the whole amount of collected data overcoming the intrinsic ground truth uncertainty due to the pathology and diagnosis nature themselves. To overcome these limitations a novel SSL algorithm, named SelfCo3, was proposed. SelfCo3 combines the Self-Training and Co-Training learning paradigms by using three classifiers which are trained separately on the total amount of labeled data and then refined by the same subset of the unlabeled data along with labels unanimously predicted by classifiers. In this data selection class balance is preserved as well. Diversity among classifiers is obtained by using different classification algorithms. Under these conditions neither any labeled data splitting, nor confidence measure is required, thus lowering the computational cost. Experiments on standard benchmark datasets verified the effectiveness of the proposed method as a general purpose SSL algorithm. Case studies performed on ground truth images from biopsy findings show that SelfCo3 is able to both enhance performance achievable with a standard SL algorithm and to outperform a previously developed application-driven classification framework without requiring any additional labeled data and making full use of the whole ground truth dataset.

The new SSL algorithm was also specified to deal with the PCa detection in ultrasound images exploiting all the available informations on the ground truth and thus further enhancing CAD performance. SelfCo3 was thus extended to SelfCo3Core, an algorithm version that exploits tumor core volume knowledge to further improve prediction performance. On these basis, a novel CAD tool for PCa detection in TRUS images was proposed. It is based on a TRUS video-sequences database realized in cooperation with physicians and a mixed supervised/semi-supervised learning procedure using SelfCo3Core. The database was realized ensuring the exact correspondence between the RF-signal and the tissue sampled by biopsy in order to guarantee ground truth reliability. A comparison with a state-of-art detection tool based on SL demonstrate the effectiveness of SSL at enhancing prediction performance with an improvement on classification error greater than 20% on a validation set and equal to 9.25% on an unknown testing set. The proposed CAD is able to improve the PPV of the standard biopsy protocol of 55%. Consequently, the number of biopsy cores could be reduced from 8-12 to 6 without los-

ing diagnostic power. Exploiting parallel computing the CAD is able to deliver a frame rate of about 40fps which confirms its effectiveness as real-time biopsy guide.

## Chapter 4

---

# Ultrasound segmentation for real-time 3D echocardiography

---

Cardiovascular disease, specifically myocardial infarction, is the main cause of long term disability and death throughout the industrial world. Modern technology provides a number of tools to assess myocardial function. Among them Real-Time 3D Echocardiography (RT3DE) has been recently proven to be a powerful imaging modality for cardiac morphology and function assessment compared with the current gold standard (MRI). Fast and automatic segmentation of the left ventricle (LV) is essential to enable an efficient quantitative analysis of the RT3DE data. Manual LV segmentation is tedious and time-consuming and suffers from a subjective bias due to the low signal to noise ratio of ultrasound scans. In this context, an automated segmentation procedure or one with minimal manual intervention is highly desirable to reduce the variability in the myocardial borders detection and to speed up the segmentation process. Nonetheless, despite the efforts from medical vendors and researchers 3D LV segmentation remains a time consuming task. Furthermore, segmentation of cardiac US remains a non trivial task due to the US data challenging characteristics. Thus, the 3D LV segmentation in RT3DE has received increasing attention and it is currently an active topic.

Level-set (LS) has been proven to be a powerful and flexible method for 3D image segmentation. Unfortunately, its inherent computational complexity has historically limited its clinical applicability.

In this context, the research presented in this chapter is aimed at exploiting the accuracy of level-set methods and the GPU computational power to address the LV myocardium segmentation in RT3DE near real-time. The final goal of this work is to provide physicians with an accurate, fast, and automated segmentation tool for cardiac morphology and function assessment.

To achieve this goal a GPU level-set solver has been developed. At the best of our

knowledge, it is the first rigorous GPU implementation of the popular level-set sparse field algorithm. Our solver combines high computational efficiency, and flexibility in the interface evolution, that make it able to handle different segmentation tasks. Thanks to its flexibility, by using a localized region-based segmentation approach we made our solver able to deal with strongly inhomogeneous images such as the myocardial wall in RT3DE. The research activity described in this chapter was carried out during an internship at CREATIS, Centre de Recherche et d' Application en Traitement de l'Image et du Signal, in Lyon, France.

The chapter is organized as follows: first, some information about the hearth anatomy and pathology, as well as diagnosis procedures are reviewed. Then, an overview of the existing method for LV segmentation in 3D echocardiography is provided. As soon as the motivation that underpin the choice of level-set as segmentation strategy will be clear, the basics of the active contour and level-set segmentation will be provided, detailing the evolution strategy adopted in this research. Then, after a brief review of the existing GPU level-set methods, the developed segmentation tool will be described in detail. Finally, performance of our approach will be evaluated in two steps. First, method performance as general purpose segmentation framework will be evaluated on both simulated and real datasets. As practical application particular attention has been paid to the detection of vessels in 3D Rotational Angiography (3DRA) images. Then, results of an extensive clinical validation demonstrate the applicability of the proposed segmentation framework in clinical environment for our target application.

## 4.1 Heart anatomy and functioning

The human heart is the hollow muscle that pumps blood throughout the blood vessels by repeated, rhythmic contractions. It is one of the most vital organs in the human body, and it is located between the lungs, in the middle of the chest, behind the sternum and above the diaphragm (see Figure 4.1).

It is enclosed in a double-walled protective sac called *pericardium*, which is attached to the central part of the diaphragm and the back of the breastbone. This sac protects the heart, anchors its surrounding structures, and prevents overfilling of the heart with blood. The inner layer also provides a smooth lubricated sliding surface within which the heart organ can move in response to its own contractions and to movement of adjacent structures such as the diaphragm and lungs. The pericardium is filled with *pericardial fluid*, which acts as a lubricant to allow normal heart movement within chest.

The outer wall of the human heart is composed of three layers. The outer layer is called the *epicardium*, it is mainly constituted by connective tissue, and works as protective layer. The middle layer of the heart is called the *myocardium* and is composed of an involuntary striated muscle tissue. The inner layer is called the *endocardium* and is in contact with the blood that the heart pumps. The endocardium is mainly constituted by epithelial and connective tissues and it merges with the inner lining (endothelium) of

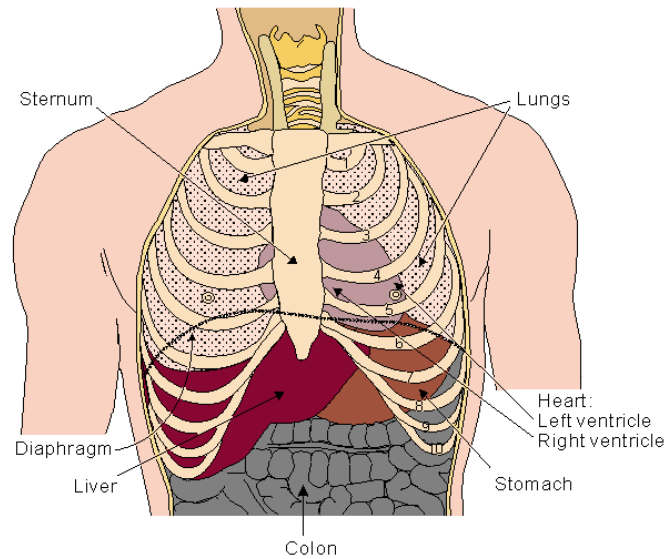


Figure 4.1: Heart position in the human body (picture from [96]).

blood vessels and covers heart valves.

The heart is divided into four main chambers: the two upper chambers are called the *left-* and *right- atria* and the two lower chambers are called the *left-* and *right-ventricles*. A thick wall of muscle, named *septum*, separates the right side and the left side of the heart [97]. Blood flows through the heart in one direction, from the atria to the ventricles, and out of the great arteries. Blood is prevented from flowing backwards by the tricuspid, mitral, aortic, and pulmonary valves. The mitral and tricuspid valves are atrioventricular valves since they are found between the atria and ventricles. The aortic and pulmonary semi-lunar valves separate the left- and right- ventricle from the pulmonary artery and the aorta respectively.

The heart acts as a double pump: blood is received by the atria, moved to the ventricles, and then returned to the circulatory system. Blood flow within the heart is regulated by coordinated contractions (*systole*) and relaxations (*diastole*) of the four chambers. In particular both atria contract simultaneously as the ventricles are relaxed and viceversa. This allows blood to flow from the atria into the ventricles, and then again into the circulatory system. Refluxes are prevented by the action of the valves. The function of the right side of the heart is to collect de-oxygenated blood and pump it into the lungs so that carbon dioxide can be dropped off and oxygen picked up. In particular, during the atrial diastole oxygen-depleted blood is received in the right atrium from the *superior vena cava*. Then right atrium contracts and blood flows through the tricuspid into the right ventricle. During ventricular systole the tricuspid closes and blood in the right ventricle is pumped into the pulmonary artery and then to the lung, where it is oxygenated. The left side of the heart collects oxygenated blood from the lungs into the left-atrium. From the left-atrium the blood moves to the left-ventricle which pumps it out to the body via

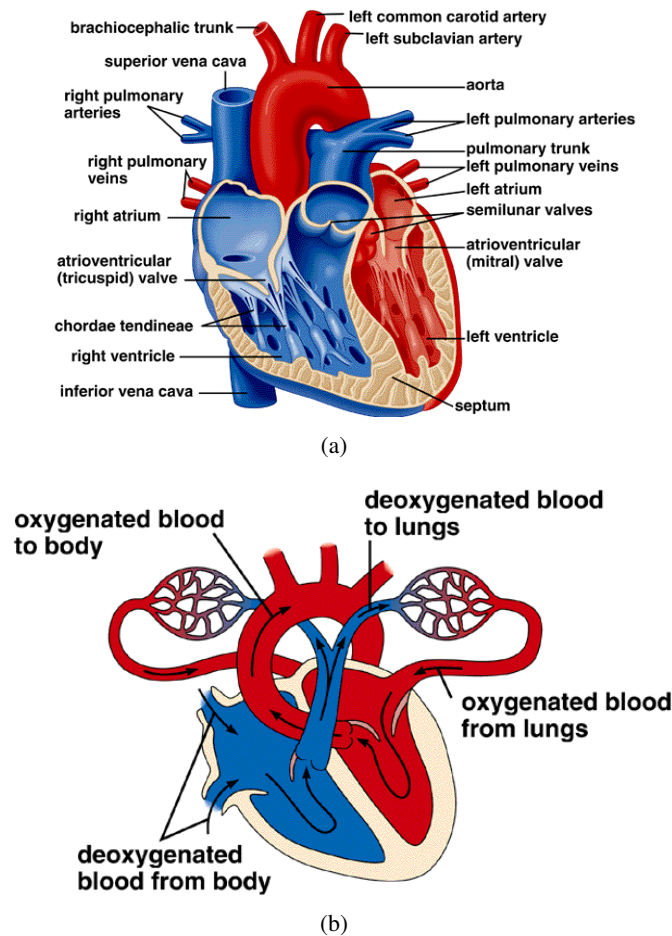


Figure 4.2: Internal view of the heart anatomy (a) and path of blood through heart (b) (picture from [98]).

the aorta. This operation is repeated cyclically and is referred as *cardiac cycle*. In an adult healthy heart, it repeats with a frequency of roughly 72 beats per minute (bpm). The sequence of contractions and relaxations is regulated by the propagation of electric pulses, released periodically by the sinoatrial node, located in the right atrium. Both an internal view of the heart anatomy and the path of blood through heart are reported in Figure 4.2.

## 4.2 Cardiovascular diseases and their diagnosis

Cardiovascular disease (CVD) is a class of diseases that involve the heart or blood vessels (arteries, capillaries and veins). Cardiovascular disease refers to any disease that affects the cardiovascular system, principally cardiac disease, vascular diseases of the brain and kidney, and peripheral arterial disease [99]. The most common causes of cardiovascular disease are atherosclerosis and hypertension. CVD remains the biggest

cause of death and morbidity for both genders within industrialized nations. Each year CVD causes over 4 million deaths in Europe and 1.9 million in the European Union [100]. Some common diseases that directly affect the heart (also known as heart diseases) are:

- **Ischemic heart disease:** is a disease characterized by reduced blood supply to the heart. The coronary arteries supply blood to the heart muscle and no alternative blood supply exists, so a blockage in the coronary arteries reduces the supply of blood to heart muscle. Most ischemic heart disease is caused by atherosclerosis.
- **Coronary heart disease:** is the end result of the accumulation of atheromatous plaques within the walls of the arteries that supply the myocardium. While the symptoms and signs of coronary heart disease are noted in the advanced state of disease, most individuals with coronary heart disease show no evidence of it for decades as the disease progresses before the first onset of symptoms arise (often a sudden heart attack). After decades of progression, some of these atheromatous plaques may rupture and start limiting blood flow to the heart muscle. This disease is the most common cause of sudden death.
- **Myocardial infarction:** commonly known as heart attack, is a disease that occurs when the blood supply to a part of the heart is interrupted, causing death of heart tissue. It is the leading cause of death for both men and women all over the world. The onset of symptoms in myocardial infarction is usually gradual, over several minutes, and rarely instantaneous. Chest pain and dyspnea are the most common symptoms of acute myocardial infarction, but at least one-fourth of all myocardial infarctions are silent, without any symptoms. Regional wall motion abnormalities or blood enzyme levels are suggestive of a silent myocardial infarction.
- **Cardiomyopathy:** is the measurable deterioration of the function of the myocardium for any reason, usually leading to heart failure. People with cardiomyopathy are often at risk of dangerous forms of irregular heart beat and sudden cardiac death. Cardiomyopathy symptoms and signs may mimic those of almost any form of heart disease, and can also be frequently asymptomatic. ECG abnormalities are often present, although the changes are frequently nonspecific, and a pattern characteristic of left-ventricular hypertrophy may be present.

Generally, cardiovascular diseases are diagnosed using an array of laboratory tests and imaging studies. The primary part of diagnosis consists in the analysis of medical and family histories of the patient, the study of risk factors, and the physical examination (including the check of blood pressure, heart rate and heartbeat). Then, these findings are combined with the results from several tests and procedures. Tests that can be performed include: simple blood tests (for checking cholesterol and other markers that may indicate heart disease), cardiac characterization, electrocardiogram (ECG), and cardiac imaging

tests. In *cardiac characterization*, a short tube (sheath) is inserted into a vein or artery in patient leg (groin) or arm and a tube (guide catheter) is then inserted into the sheath. Aided by X-ray imaging, physician threads the guide catheter through that artery until it reaches the heart. Heart pressures is then measured and dye is injected. Thus the doctor can see the blood flow through the heart, blood vessels and valves to check for abnormalities.

*Electrocardiography* monitors electrical activity of the heart, primarily as recorded from the skin surface. The ECG allows observation of the heart electrical activity by visualizing waveform beat origin (typically from the sinoatrial node) following down the bundle of HIS (a collection of heart muscle cells specialized for electrical conduction) and ultimately stimulating the ventricles to contract forcing blood through the body. Much can be learned by observing the QRS morphology (named for the respective portions of the polarization/repolarization waveform of the wave). Rhythm abnormalities can also be visualized as in slow heart rate bradycardia, or fast heart rate tachycardia.

The goal of *cardiac imaging* tests is studying the cardiac performance in a noninvasive, reproducible and inexpensive way. In fact, cardiac imaging allows to study both the cardiac morphology and physiology by enabling the measurement of heart capability to provide the human body with satisfactory quantities of blood. *Ejection fraction*, *stroke volume* and *cardiac output* are fundamental indices for cardiac function assessment.

Ejection fraction is the volumetric fraction of blood pumped out of the ventricle with each heart beat or cardiac cycle. It can be applied to either the right-ventricle, which ejects via the pulmonary valve into the pulmonary circulation, or the left-ventricle, which ejects via the aortic valve into the systemic circulation. By introducing the end-diastolic volume (EDV), defined as the volume of blood within a ventricle immediately before a contraction, and the end-systolic volume (ESV), defined as the volume of blood left in a ventricle at the end of contraction, the ejection fraction (EF) can be defined as:

$$EF = \frac{(EDV - ESV)}{EDV} = \frac{SV}{EDV} \quad (4.1)$$

where SV is the stroke volume, *i.e.* the difference between EDV and ESV, representing the volume of blood ejected with each beat. As a consequence, EF can be considered as the fraction of the EDV that is ejected with each beat. Damage to the muscle of the myocardium, such as that sustained during myocardial infarction or in cardiomyopathy, impairs the heart's ability to eject blood and therefore reduces EF. SV is intrinsically controlled by EDV (the degree to which the ventricles are stretched prior to contracting). An increase in the volume or speed of venous return will increase EDV and therefore SV. Decreased venous return has the opposite effect, causing a reduction in SV. Elevated ESV, instead, reduces SV. Though not usually affecting SV in healthy individuals, increased ESV will hinder the ventricles in ejecting blood, causing reduced SV. Increased ESV may be found in aortic stenosis and arterial hypertension.

Cardiac output (CO), is another important functional index, and it represents the volume of blood being pumped by the left- or right-ventricle in the time interval of one



Measure	Typical Value	Normal Range
EDV	120 ml	65 – 240 ml
ESV	50 ml	16 – 143 ml
SV	70 ml	55 – 100 ml
EF	58%	50 – 70 %
CO	5.25 L/min	4.0 – 8.0 L/min

Table 4.1: Functional indices of an healthy heart: typical values and normal ranges.

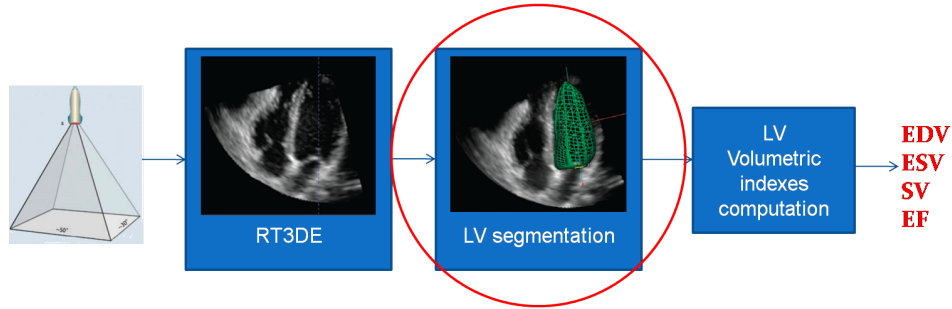


Figure 4.3: Workflow for cardiac functional indices computation considering the real-time 3D echocardiography as cardiac imaging modality.

minute. The CO can be defined as the product of the SV and the heart rate ( $CO = SV \times HR$ ). Diseases of the cardiovascular system are often associated with changes in CO, particularly the pandemic diseases of hypertension and heart failure. CVD can be associated with increased CO as occurs during infection and sepsis, or decreased CO, as in cardiomyopathy and heart failure. The ability to accurately measure CO is important in clinical medicine as it provides for improved diagnosis of abnormalities, and can be used to guide appropriate management. Typical values and normal ranges of the described indices for an adult healthy male of 75Kg are reported in Table 4.1.

The large majority of cardiopathies usually affect the left-ventricle (LV), because it is responsible for pumping the blood throughout the entire human body, and therefore is the heart chamber to which the most intense mechanical effort is demanded. For this reason, although all the cardiac functional indices can be measured for both the left- and the right- ventricle, usually, a quantitative analysis of cardiac morphology and functional assessment relies on the computation of these indices for the LV. All these functional quantities are basically measurements allowed by cardiac imaging. In fact, they can be evaluated from images of the heart by tracing the endocardial contours at end-diastole and end-systole, extrapolating estimates of EDV and ESV from these contours [101] and then computing the derived indices (see Figure 4.3).

Dedicated technologies such as Echocardiography, CT, MRI and various radionuclide techniques, such as Multigated Angiography (MUGA) scanning, have definitively allowed clinically relevant mathematics regarding ischemic and congenital heart diseases and heart failure. Among these imaging modalities, Echocardiography has evolved as

a well-established imaging technique for the noninvasive evaluation of myocardium abnormalities and it is one of the most widely used diagnostic tests in cardiology. Besides the previously mentioned aspects connected to safety, low cost, and portability, Echocardiography provides a wealth of helpful information, including the size and shape of the heart (internal chamber size quantification), pumping capacity, and the location and extent of any tissue damage. Moreover, Echocardiography alone offers the possibility of real-time observation of ventricular wall motion and deformation, which are fundamental for assessing the extent of possible myocardial ischemia and infarction.

Besides SV, EF and CO, left ventricular hypertrophy (LVH) is a further significant index. It consists in the muscle thickening and consequent increment of mass, and can be connected with pathological situations implying a muscle overload, which is the case of aortic stenosis, aortic insufficiency, and hypertension. LVH as well can be evaluated in Echocardiography by jointly tracing endo- and epicardial contours, and evaluating the distance between the two. Finally, recently, the echocardiographic measures of LV myocardial deformation, strain and strain rate, have been proposed as important novel indices of cardiac functionality [102, 103, 104].

The assessment of cardiac morphology and function by ultrasound imaging has made a significant step forward by the introduction of Real-Time 3D Echocardiography (RT3DE), as it allows a truly volumetric imaging of the heart avoiding some of the problems intrinsically associated with 2D imaging, such as foreshortening, out-of-plane motion artifacts, the need for geometrical assumption when computing volumes, *etc.* Furthermore, several studies have shown that clinical indices derived from the LV volume measurement can be obtained from 3D ultrasound acquisition with sufficient accuracy *w.r.t.* the current gold standard MRI [15, 16, 17, 18].

All the above mentioned measurements require an accurate segmentation and tracking of the myocardial muscle from ultrasound scans at the end-systolic and end-diastolic phase. Unfortunately, manual delineation of LV boundaries in RT3DE is tedious, time-consuming and suffers from a subjective bias due to the low signal to noise ratio of the underlying images. In this context, an automated segmentation procedure or one with minimal manual intervention, is highly desirable to reduce the variability in the myocardial borders detection and to speed up the segmentation process.

In this scenario, CAD systems can support physicians performing an automated quantitative analysis of the cardiac morphology and function, and allowing them to overlay on the echographic image information characteristic of the heart pathology to be detected. Segmenting the cardiac structures and tracking them along the complete cardiac cycle is a key step needed to achieve this goal. In order to be used in clinical environment such a diagnostic tool needs to work in real-time. This is mainly hampered by the computational requirements of the segmentation step. In fact, this step is also a bottleneck of the tracking procedure since it is needed to extract the target object that must be followed in time to maintain both the spatial coherence (segmentation), and the temporal coherence (tracking). Combining segmentation and motion estimation leads

to a complete technique to assess myocardial function. Fast, accurate and automated segmentation of the LV is thus essential to enable an efficient quantitative analysis of the RT3DE images. To date, there has been a considerable amount of research on the delineation and tracking of myocardium muscle from ultrasound data, but there is not yet a fully satisfactory solution. An overview of interesting approaches presented in literature will be given in the following section, focusing on the segmentation task. In fact, the main goal of the research presented in this chapter is the development of an automated segmentation framework for 3D Echocardiography. The adopted approach consists in exploiting the accuracy of the well known level-set segmentation methods and the GPU computational power to address the LV myocardium segmentation in RT3DE in real-time.

### 4.3 3D LV segmentation

As stated above, fast and automatic segmentation of the LV in 3D echocardiography is essential to enable an efficient quantitative analysis of cardiac morphology and function. Segmentation of cardiac ultrasound has received increasing attention in medical imaging community and is still an active research area. In fact, despite the efforts from medical vendors and researchers 3D LV segmentation remains a time consuming task and heavily relies on user interaction. Furthermore, segmentation of cardiac US remains a non trivial task due to the ultrasound data challenging characteristics. The main challenges for segmentation of ultrasound images lie in the poor quality of the signal due to the low spatial resolution of the hardware system, attenuation, and motion artifacts that lead to ill-defined myocardial borders and additional frequency-dependent speckle noise that corrupts the specificity of gray level values in characterizing an interface or a particular tissue type. The literature on methods for automatically or semi-automatically segmenting and tracking the LV in 3D Echocardiography is extensive. Most attention has been given to segmenting and tracking the motion of the endocardium (blood pool/tissue border) to allow for estimation of left-ventricular areas/volumes and derived measures that are used in diagnosis and assessment of CVD disease, but a few methods also discuss the epicardium segmentation [105, 106, 107]. In principle, most automated methods are suitable for both. However, the epicardium is harder to detect since it is usually less visible and has a varying appearance in different segments. Epicardial border detection is usually assisted by the endocardial border detection, *e.g.* by assuming typical distances between both borders. Moreover, beside the methods for segmentation in RT3DE several methods to be applied to the earlier 3D ultrasound imaging modalities have been proposed [108, 109, 110]. Comprehensive reviews of these methods can be found in [111, 112]. Here we focus on LV endocardium segmentation methods on RT3DE. We roughly divide these methods into three categories according to the used segmentation approach: statistical models, machine learning based methods, and geometrical models, and we report some examples for each category. However, it must be noted that a fair

and quantitative comparison among these methods cannot be done, due to the differences in the ground truth. The differences consist in the used reference methods, database patient populations, different data acquisition conditions and equipment and image quality. Additionally, even the evaluation metrics used to assess methods performance could be different. However, when the proposed method is aimed at performing a quantitative analysis of the cardiac morphology and function a widely applied procedure to assess its performance consist in comparing LV functional indices obtained using the proposed methods, with the ones obtained using a reference method (usually manual tracing from expert cardiologists or MRI) by means two popular statistical analysis: *Regression Analysis* and *Bland-Altman Analysis* [113]. As it will be shown next, this is also the procedure used in this thesis to evaluate clinical applicability of the developed GPU segmentation framework that will be described in the following sections. Therefore, we just briefly recall the concept of Regression and Bland-Altman analysis before to go on with 3DLV segmentation methods review.

### 4.3.1 Statistical analysis techniques for comparing LV segmentation methods

**Regression Analysis.** In statistics, linear regression is an approach to modeling the relationship between a scalar dependent variable  $y$  and one or more explanatory variables denoted  $x$ . In linear regression, data is modeled using linear predictor functions, and unknown model parameters are estimated from the data. Such models are called linear models. Most commonly, linear regression refers to a model in which the conditional mean of  $y$  given the value of  $x$  is an affine function of  $x$  (see Figure 4.4). When applied to method comparison data, it provides useful information about proportional, constant, and random error via, respectively, the slope, intercept, and standard deviation of the residuals. Usually, a good agreement between two methods according to linear regression analysis is indicated by an high correlation coefficient [114] and a regression fit close to the identity line. Note that, in the ordinary linear regression model (OLR) the regression line is calculated by minimizing the squared residuals in the  $y$  direction (least squares). OLR assumes an error-free  $x$  variable and a constant analytical imprecision of the  $y$  variable, both of which are seldom met in practice. For this reason also other statistical analysis are typically used.

**Bland-Altman analysis** is a statistical method which allows the clinician to compare two different measurement techniques. The Bland-Altman graph plots the difference between two techniques against their averages (see Figure 4.5). The resulting scatter diagram allows the clinician to determine:

- Bias: average difference, ideal bias = 0;
- Precision: standard deviation that describes range for 68% of comparison points;

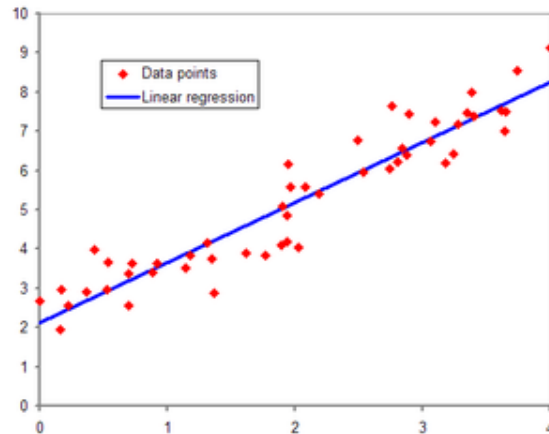


Figure 4.4: Linear regression plot.

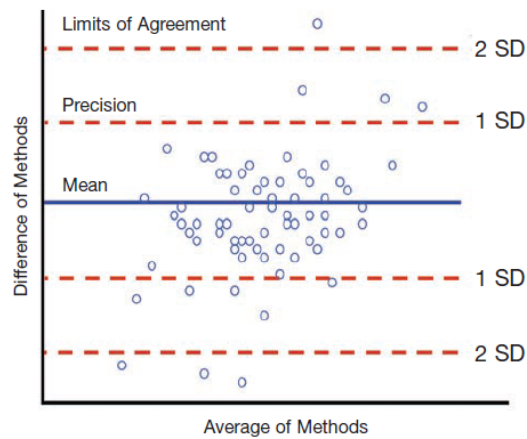


Figure 4.5: Bland-Altman analysis plot.

- Limits of Agreement: 0.96 standard deviation that describes the range for 95% of comparison points.

Bland-Altman analysis is typically used to compare measurement techniques against a reference value, especially when the reference value may not have a true gold standard. Bland and Altman suggest that when a new technology has bias and precision comparable with the previous technology, then it may be accepted in the clinical setting. Basically, the idea of this method is to see how does the difference between two methods varies within the range of agreements, to check if it is just a bias (*i.e.* coming from a systematic error) or if there is a real disagreement (given by the 95% interval of the error).

In this thesis, to evaluate performance of the developed GPU segmentation frame-

work, we will compare manual analysis performed by three expert physicians and the automated analysis performed using our framework, plotting the difference between the two methods against the average of the experts measurements, instead of using the average of the methods to test their agreement. In fact, since our ground truth is constituted by the average of measurements performed by three experts, we already know the reference method, thus we don't have the necessity of using the average of methods as reference. Note that this does not change neither the bias nor the limits of agreement, which are the relevant metrics that come out from the Bland-Altman analysis. To assess the statistical significance of the found biases a paired-t-test [115] is used. A significance level of 0.05 is usually employed, meaning that one is 95% sure that the bias is significant, *i.e.* that there is a statistically consistent difference between the measurements, and this is also our approach.

It is important to observe that, using the Bland-Altman analysis, instead of, or in addition to, the traditional linear regression analysis provides a more reliable analysis. In fact the correlation coefficient  $r$  only gives the best linear fit which doesn't have necessarily a linear slope (corresponding to a perfect agreement). Moreover, it is a requirement for linear regression analysis that the  $x$  values are fixed by the design of the experiment and are not subject to error, but this is an uncommon occurrence when biological methods of measurement are compared.

Concerning the regression analysis, in this thesis, the dependent variable  $y$  is the measurement obtained using the GPU segmentation framework, while the explanatory variable  $x$  is the average of the three experts measurements.

### 4.3.2 Literature review of 3D LV segmentation methods

**Statistical modeling** methods model the statistical variations in actual patient data from large sets of images with expert-drawn borders. Patient variability is condensed into a relatively simple mathematical model which has only a few parameters, but with very strong descriptive power. Among statistical methods, *Active Shape Models (ASM)* use mainly a shape model for border detection. First, the average shape is placed on the image, then the parameters of the shape model are found iteratively: the local image features drive the shape model to the actual borders under the statistical shape model geometrical constraint. One representative example of this methods category is the one proposed by Hansengard *et al.*. In [116] they presented a fully automatic real-time algorithm for LV segmentation based on 3D Active Shape Model and an extended Kalman filter to predict and update model parameters. The method was evaluated on 21 patients, and compared to manually verified segmentations from a custom-made semiautomatic segmentation algorithm. In all patients meshes corresponding to the endocardial border were detected. The comparison was based on results of regression and Bland-Altman analysis. Contours detected by the ASM show good overall agreement for the point-to-surface distance. Significant biases with relatively narrow limits of agreement were found in EDV, ESV and EF estimation ( $[\mu \pm 1.96\sigma]$  equal to  $-5.9 \pm 21$  for EDV,  $6.2 \pm 19$

for ESV, and  $-7.7 \pm 12$  for EF). Authors claim that a real-time segmentation can be performed at 25fps. Other interesting methods belonging to this category are [117, 118].

*Active appearance models (AAM)* use a somewhat different border detection strategy, by taking both texture and shape variability into account. An appearance model is obtained by applying principal component analysis on the combination of the shape and texture models. The model is then adapted to match the image iteratively: the difference between the model synthesized image and the real image determines the next best estimate of the appearance model. Examples of segmentation based on AAM are [119, 120, 121]. The major advantage of the population-based statistical model is that by explicitly learning variations from real examples, the method only find plausible results, even in presence of typical ultrasound artifacts. However, a large database is needed which is representative of the expected variations, and the accuracy directly depends on the quality and consistency of the expert-drawn borders.

**Machine learning methods** can be divided in classification and clustering methods. *Classification methods* use expert-created databases for automated grouping and recognition of many types of objects. Beside the application of classification in computer aided medical diagnosis seen above in this thesis, one application of the classification method is also the detection of the LV border in ultrasound images. Examples of segmentation frameworks based on machine learning techniques rely on the use of different types of classifiers, such as classification trees or Bayesian classifiers. An example of these methods is the work of Lempitsky *et al.* [122] that proposed a machine learning approach to treat the problem of automatic segmenting the myocardium in RT3DE as a two-class 3D patch classification task. Authors demonstrated that solving such a task using random forests, which are the discriminative classifiers, allows to obtain accurate delineations in a matter of seconds (from 2 to 22 depending on the tree size). Authors also claim the possibility of working in real-time by using efficient GPU implementations of the random forest algorithm, but neither details nor validation are provided. The method was evaluated on 3D exams from different healthy subjects with the myocardium for left and right ventricles delineated by an expert. Performance of the method have been expressed in terms of recall-precision and ROC-curves, while any clinical functional index is provided. The major drawbacks of the classification based segmentation method is the necessity of many consistently delineated example images to build an accurate classifier, considerably more than for statistical models.

*Clustering methods:* as in other applications, in segmentation, clustering techniques are used to categorize image pixels into distinct groups, based on image features. Some approaches [123] categorize each pixel into myocardial tissue or blood, based on its intensity, others use features based on phase [124, 125]. The segmented object may not necessarily be in one piece, thus allowing maximum freedom in shape and topology. However, due to its characteristics, in ultrasound imaging, it is often necessary to incorporate higher-level knowledge. Therefore, clustering is often integrated into modeling methods, such as geometrical models.

**Geometrical models** are the most common approaches for border detection. The border is represented as a curved surface which separates the lumen from the cardiac wall. This surface is influenced by geometrical constraints (similarity to a given shape, smoothness, *etc.*). To find the object boundaries, an initial guess of the surface is placed on the image, either automatically or interactively. This surface is then iteratively deformed to a new position, which is associated with the true border. Image features guide the deformation that goes on until the surface does not change significantly anymore. The well known approaches belonging to this category are deformable models, active contours, balloons and snakes, that are all based on an energy minimization to drive the surface evolution. A mathematical energy function is defined, which consists of an external and an internal component. The external component is determined by the image features, and the internal component limits the area and curvature of the surface to ensure smoothness. The total function is then optimized iteratively. These methods may differ in the mathematical representation of the contours, the type of image features, and the way of obtaining the initial guess of the surface. Deformable models are very efficient at segmenting high resolution 2D images with a simple mathematical framework [126, 108, 127], but limitations arise in applying these methods to 3D images. Deformable models based on elaborated mesh structures [128] are extremely difficult to implement and manipulate while ensuring, at the same time, stability and adaptivity of the mesh as it deforms. A solution to this problem is to embed the deformable model into a level-set framework with an implicit formulation that extends naturally to three and higher dimensions. An example of explicit deformable models is represented by the work proposed by Angelini *et al.* in [129]. In this work, first, the 3D+T data was pre-processed with a wavelet analysis (using brushlets) to reduce speckle. Then a 2D balloon deformable model, using an intensity gradient based force, was fitted to each slice and the volume derived from this. The method was tested on 6 clinical datasets. The gold standard used was cardiac MRI and comparison with manual segmentation on ED and ES frames were performed. The mean percentage errors for volumes and ejection fraction measures showed a significant improvement of quantitative accuracy with the deformable model when comparing to manual tracing. Later, Angelini *et al.*, extended this 2D parametric deformable model to 3D [130]. In order to overcome the difficulties encountered in a direct explicit extension, they exploited an implicit formulation of the method through level-set. Doing so they implemented a 3-D deformable model level-set segmentation method following the same approach used by Chan and Vese [131] for developing the active contours without edges. An arbitrary initialization of the object anywhere in the image is used but the initialization is not automated. The method was evaluated on 10 RT3DE exams, for both RV and LV measurements, in comparison with both MRI and manual tracing. For LV measurements, compared with manual tracing, they reported a confidence interval ( $\mu \pm 1.96\sigma$ ) equals to  $16.11 \pm 50.07$  for EDV,  $-6.64 \pm 34.37$  for ESV, and  $-0.59 \pm 22.17$  for EF, and correlation coefficients respectively equal to 0.63, 0.62, and 0.45. Execution time for this method is not provided. Other level-set based 3D LV segmentation meth-



ods have also been proposed [132, 133, 134, 18, 135, 136, 137]. Among them we cite methods that have been presented along with clinical validation on a sufficient number of patients. Corsi *et al.* [135, 136] proposed a gradient-based 3D level-set deformable model for RT3DE volume data that relies on edge information. The classical inflationary term in the speed function was removed in order to avoid leakage. Hence, and due to the strong inhomogeneities in the data corrupted by speckle noise, they need to manually initialize the model close to the endocardial borders. Two studies have been performed by the authors comparing LV volumes and EF measurements from MRI and 3-D level-set segmentation on RT3DE data. In the first study [135], for the EF measurement they reported a correlation coefficient equal to 0.98, between manual tracing on MRI and level-set on RT3DE (computed on a 'small' number of cases, according to the authors) from a data base of 21 healthy or diseased patients. In a second study [136], using the same method, authors reported correlation coefficient for LV EF measurement equal to 0.87 for a group of nine patients out of a database of 21 patients with ischemia or dilated cardiomyopathy. For these methods execution time is not provided.

Caiani *et al.* [18] proposed a volumetric analysis technique based on level-set. Their image segmentation technique is applied without any shape algorithm assumption resulting in the detection of a geodesic surface in a Riemann space derived from the 3D image dataset. The method was tested on 44 patients and a statistical analysis was used to assess method performance. In particular, regression and Bland-Altman analysis were performed. Results show high correlation when compared with MRI data ( $r=0.97$  for EDV and ESV,  $r=0.93$  for EF). Also low biases and relatively narrow limits of agreement were reported ( $(\mu \pm 1.96\sigma)$  equal to  $-4.1 \pm 29$  for EDV,  $-3.5 \pm 33$  for ESV, and  $-0.8 \pm 14$  for EF). Comparison with manual contouring is not provided in this work. This method needs a manual initialization of the endocardial surface. The time required to perform a complete quantitative analysis of the data was approximatively equal to 5 minutes.

Although level-set methods allow to obtain accurate LV segmentation in RT3DE, they are also characterized by long computational time that limits their applicability in clinical environment. To overcome this limitation, very recent approaches come back to the use of explicit deformable models to reduce the computational burden of the segmentation problem by reducing its dimensionality. Duan *et al.* [138] proposed a variational segmentation framework for cardiac image segmentation in which the deformable boundary is represented as a geometric function, where one of the coordinates of the points within the boundary is given explicitly as a function of the remaining coordinates. The major limitation of this method was that topological flexibility is not preserved and the framework had no explicit control on the interface smoothness. In order to overcome these limitations, Barbosa *et al.* [139, 140] proposed a B-spline formulation for [138] that allowed using local and global region-based energy terms. However, using explicit formulation some limitation to the topological flexibility still remains.

To exploit the flexibility and the accuracy of level-set techniques reducing the com-

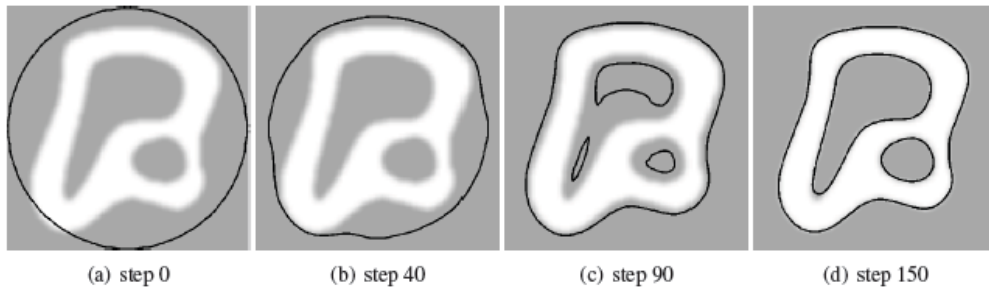


Figure 4.6: Example of segmentation using active contours (picture from [147]).

putational time, and therefore allowing to perform a quantitative analysis of 3D cardiac morphology and function in clinical applications, in this study we adopt the following strategy: we exploit level-set methods that have been proposed to accelerate the level-set, focusing on the Sparse field (SF) method [141] and, according to the intrinsic application parallelism, to further improve the level-set based segmentation efficiency we exploit parallel implementation on highly parallel architectures like the modern GPUs, which represent the best choice in terms of cost, development time and flexibility. Exploiting GPU computing techniques to accelerate the myocardium segmentation in echocardiography requires, from one side, designing a fast level-set solver, from the other side, specializing it for the target application.

Our approach to this task is based on active contours segmentation embedded in level-set formalism. In the following sections, basic concepts of active contour and level-set segmentations will be provided detailing the evolution strategy adopted in this research. Then, the developed framework will be described.

#### 4.4 Active Contours and Level-Set Segmentation

Since their first introduction in [142], *Active Contours* (also known as *deformable models* or *active surface* in 3D applications) have proven to be powerful tools in the image processing domain [143, 144, 145, 146]. Active contours segmentation consists in propagating an interface (curve in 2D, surface in 3D) into the image domain to detect a target object. Using this approach the segmentation problem is expressed as the minimization of an energy functional that reflects the properties of the object to be recovered.

The active contours principle used for segmentation purpose in a 2D image is shown in Figure 4.6. In this example, the object to be segmented (white region) has two holes and blurred contours. From a particular initialization (including the whole object), an active contour automatically evolves to properly detect the shape of the object of interest.

Let  $\Gamma(\mathbf{p}, \tau) : [0, 1] \times [0, T] \rightarrow \mathbb{R}^2$  represent a family of evolving curves, where  $\mathbf{p}$  parametrizes the curve geometry and  $\tau$  parametrizes the family of evolving closed curve, in active contours framework the general evolution equation of the contour is given by the following Partial Different Equation (PDE):

$$\begin{cases} \frac{\partial \Gamma(\tau)}{\partial \tau} = V \widehat{N} \\ \Gamma(\tau = 0) = \Gamma_0 \end{cases} \quad (4.2)$$

where  $\widehat{N}$  is the unit inward normal of the curve  $\Gamma$  and  $V$  is the normal velocity. If an implicit representation of the curve  $\Gamma$  is chosen, active contours are defined in an Eulerian framework and this implementation is known as *level-set method*. In the level-set formulation the evolving interface is implicitly represented as the zero level-set of a time-dependent, higher dimensional, smooth function, usually called *level-set function*. By doing so, the level-set representation is independent from the interface parameterization, and the interface evolution is controlled by the evolution of the corresponding implicit function.

Let  $I : \Omega \rightarrow \mathbb{R}^d$  be a given image where  $\Omega \subset \mathbb{R}^d$ . In the level-set formalism  $\Gamma \subset \Omega$  is represented as the zero level-set of a Lipschitz-continuous function  $\phi$  of dimension  $d + 1$ , satisfying:

$$\begin{cases} \phi(\mathbf{p}, \tau) > 0, & \text{for } \mathbf{p} \in \Omega_{in}(\tau) \\ \phi(\mathbf{p}, \tau) < 0, & \text{for } \mathbf{p} \in \Omega_{out}(\tau) \\ \phi(\mathbf{p}, \tau) = 0, & \text{for } \mathbf{p} \in \partial\Omega_{in}(\tau) = \Gamma(\tau) \end{cases} \quad (4.3)$$

where  $\Omega$  is an open region in  $\mathbb{R}^{d+1}$ ,  $\Omega_{in}$  is a region in  $\Omega$  bounded by  $\Gamma$ , and  $\Omega_{out} = \Omega \setminus \Omega_{in}$  (see Figure 4.7).

The problem of segmenting one object from the background is then handled by the evolution of one level-set function driven by a PDE; its steady state divides the image into two regions that delimit the boundary of the object of interest. This PDE is known as level-set equation and its general form can be written as:

$$\frac{\partial \phi}{\partial \tau} = F \cdot \|\nabla \phi\| \quad (4.4)$$

where  $F(\cdot)$  is a speed function defined on  $\Omega$  whose values can vary over time. According to (4.4)  $F(\cdot)$  makes the level-set evolve along a path normal to the interface (see Figure 4.8). The speed function  $F(\cdot)$  is generally derived from the minimization process of an energy criterion. Usually,  $F(\cdot)$  is composed by two terms: one data attachment term  $D(\cdot)$  which enables the surface to reach the boundaries of the object of interest in the image, and one curvature term  $C(\cdot)$  which ensures to keep some smooth properties of the surface over time.

A wide range of deformation can be implemented by defining appropriate speed functions. In literature, many different speed functions have already been proposed to solve specific segmentation problems. For example, intensity thresholding solutions, such as the one proposed by Lefohn *et al.* [148], have been successfully applied when the object of interest is well defined by its intensity values. However, such simple strategy usually failed in complex situations, leading to the creation of more sophisticated approaches. In this context, edge-based active contour models utilizing image gradients

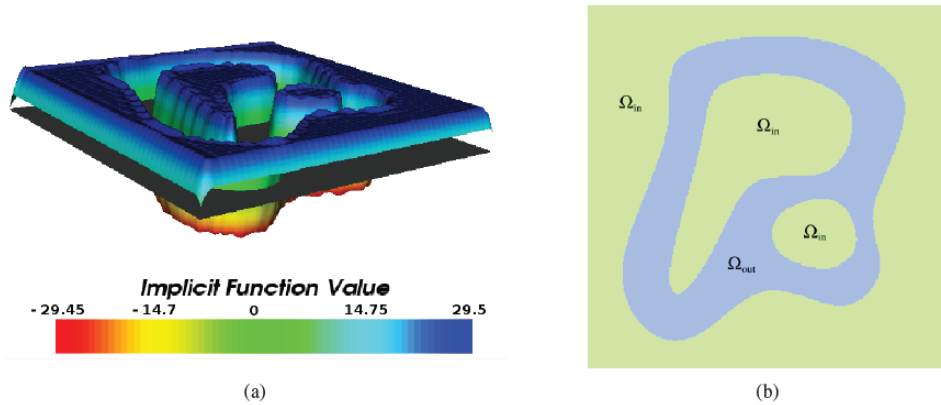


Figure 4.7: Level-set function for a two dimensional domain  $\Omega$  represented as an elevation map (a) and the corresponding domain partition (b) (picture from [147]).

have been proposed in order to identify object boundaries [149]. This type of highly localized image information is adequate in some situations, but has been found to be very sensitive to image noise.

More recently, work in active contours has been focused on region-based flows. Such approaches make the level-set evolve by taking into account the statistical properties of the foreground and background regions. Some of the most well-known and widely used region-based active contour models (for example the models proposed by Chan and Vese [131] or Yezzi [150]) assume the different image regions to be of constant intensity. More advanced techniques attempt to model regions by known intensity histograms [151], distributions [152], texture maps or motion [153]. Unfortunately, since these approaches use global statistics, they may fail when segmenting heterogeneous objects (object having spatially variant features). In this context, localized approaches such as the one proposed by Lankton *et al.* [154] have proven their effectiveness in detecting such objects. In practical situations, such as in medical image segmentation, the object of interest is sometimes partially occluded, leading to unsatisfactory segmentation results. In these cases, high level prior knowledge about the expected shape could be exploited in order to successfully segment the desired target. The introduction of general shape prior in the speed function has been widely studied in level-set literature [153, 155, 156, 157, 158]. In this study, we also exploit the high prior knowledge introduction to improve segmentation accuracy of our solver for the target application.

In practice, to perform level-set segmentation one has to take care to several practical issues: (i) the discretization of the level-set equation; (ii) the choice of the appropriate level-set function; (iii) the choice of the most appropriate level-set evolution strategy, that comprises the choice of a speed function well suited for the target application and the choice of a level-set method that fit the application requirements such as time complexity requirements.

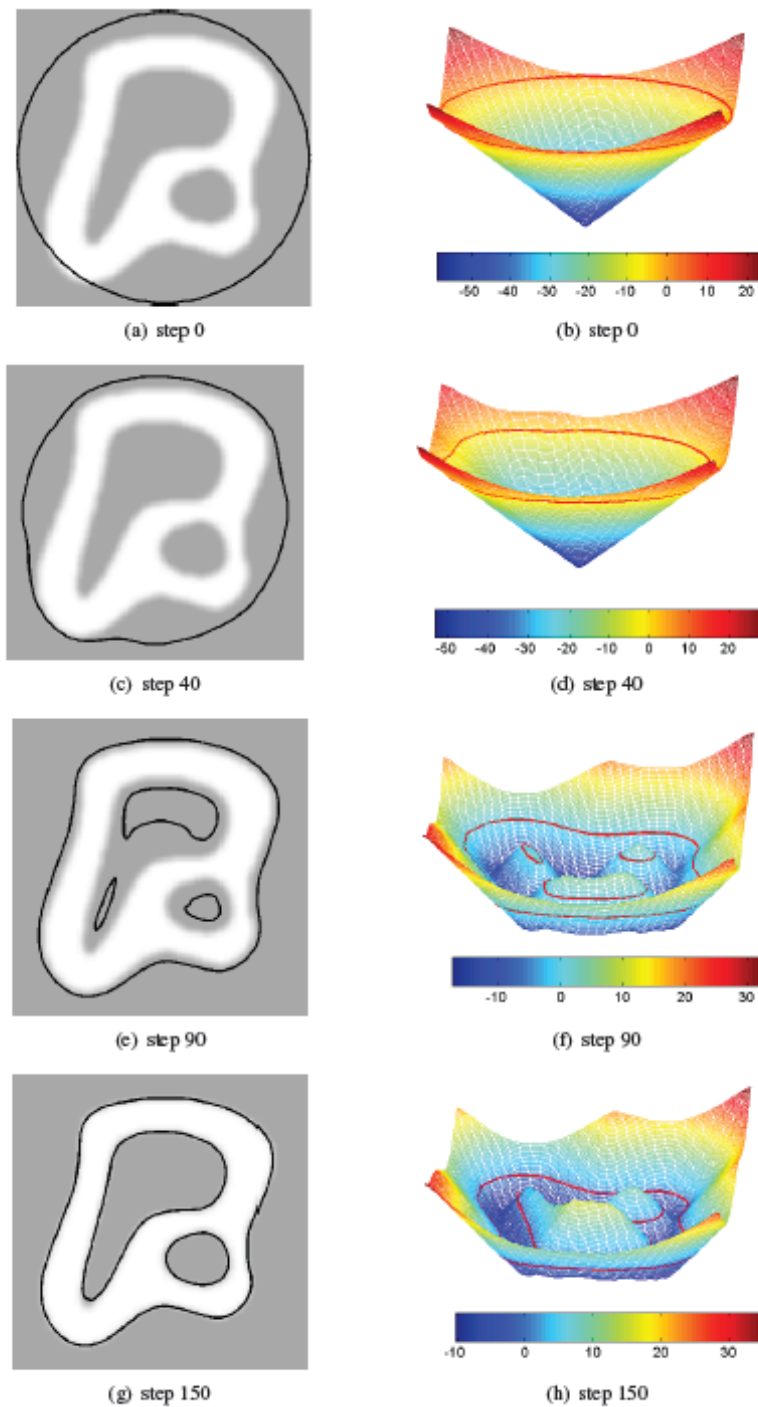


Figure 4.8: Evolution of the embedding function (b,d,f,h), along with the evolution of its zero level-set (red) while segmenting a synthetic image (a,c,e,g). The black curve on (a,c,e,g) corresponds to the projection of the zero level on the synthetic image (picture from [147]).

#### 4.4.1 Discretization of the level-set equation (4.4)

A lot of work has been dedicated to the discretization of equation (4.4) using finite difference methods. These numerical schemes have been introduced by Osher *et al.* [159]. They allow to obtain accurate and unique solutions, and involve upwind differencing, essentially non oscillatory schemes borrowed from the numerical solutions of conservation laws and Hamilton Jacobi equations [159].

The discrete version of (4.4) can be written as:

$$\phi(\mathbf{p}, \tau + \Delta\tau) = \phi(\mathbf{p}, \tau) + \Delta\tau \cdot F(\mathbf{p}, \tau) \cdot \|\nabla\phi(\mathbf{p}, \tau)\| \quad (4.5)$$

with

$$F(\mathbf{p}, \tau) = D(\mathbf{p}, \tau) + \alpha \nabla \cdot \frac{\nabla\phi(\mathbf{p}, \tau)}{\|\nabla\phi(\mathbf{p}, \tau)\|} \quad (4.6)$$

where the second term of (4.6) corresponds to the curvature term  $C(\mathbf{p}, \tau)$  and  $\alpha$  is the parameter that weights the influence of the the curvature term.

#### 4.4.2 Choice of the involved implicit function and reinitialization

In the course of its evolution, the level-set function can develop flat or steep regions leading to difficulties in both numerical approximation of derivatives and speed of convergence [149, 160, 159]. In order to overcome this difficulty, the following scheme is generally used: (i) initialize the level-set as a signed distance function (relative to the interface). This choice is generally motivated by the interesting numerical properties of such function (for example  $\|\nabla\phi\| = 1$ ); (ii) reshape the level-set function periodically in order to resurrect this distance function property. Let  $\gamma$  be the zero level-set and  $\mathbf{x}$  an image voxel, the level-set function defined using the signed distance function is written as:

$$\phi(\mathbf{x}) = \begin{cases} -dist(\mathbf{x}, \gamma) & \text{if } \mathbf{x} \in \gamma_{in} \\ 0 & \text{if } \mathbf{x} \in \gamma \\ dist(\mathbf{x}, \gamma) & \text{if } \mathbf{x} \in \gamma_{out} \end{cases} \quad (4.7)$$

Figure 4.9 shows an example of a level-set function  $\phi$  embedding a 2D contour and its discretization on a rectilinear grid of points where each point has a value of  $\phi$  corresponding to its distance from the zero level-set.

#### 4.4.3 Level-set evolution strategy

##### Speed functions

As stated above, the speed function  $F(\cdot)$  in (4.4) is usually the result of two contributes, a data attachment term  $D(\cdot)$  and a curvature term  $C(\cdot)$ . Let  $\mathbf{x}$  be an image voxel, a general speed function  $F(\cdot)$  can be written as:

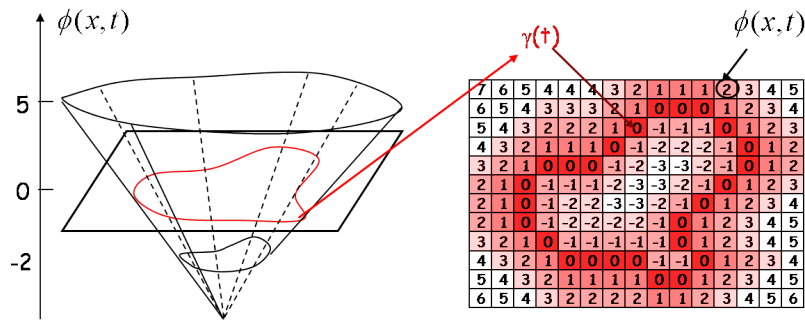


Figure 4.9: An example of a level-set function  $\phi$  embedding a 2D contour and its discretization on a rectilinear grid of points where each point has a value of  $\phi$  corresponding to its distance from the zero level-set.

$$F(\mathbf{x}) = D(\mathbf{x}) + C(\mathbf{x}) \quad (4.8)$$

In this research, in order to illustrate the ability of the proposed segmentation approach to deal with different speed functions and therefore handle different segmentation tasks, we will use four different speed functions of two types: one which is constant over time and one whose intrinsic parameters vary during the evolution process. Since the curvature term will be the same for all the used speed functions, to understand the purpose of each of them we will only refer to the data attachment term.

The first speed function we will use is the one proposed by Lefohn [148]. The corresponding data attachment term states as:

$$D(\mathbf{x}) = \epsilon - (|I(\mathbf{x})| - T) \quad (4.9)$$

where  $\epsilon$  and  $T$  are two parameters and  $I(\mathbf{x})$  corresponds to the image intensity value at point  $\mathbf{x}$ . This criterion can be interpreted as follows: if the value of  $I(\mathbf{x})$  is between  $T - \epsilon$  and  $T + \epsilon$  then  $D(\mathbf{x})$  will encourage surface growth, otherwise  $D(\mathbf{x})$  will encourage surface contraction. Since  $T$  and  $\epsilon$  are constant real values, the corresponding speed function is time constant. This speed function basically corresponds to a thresholding method, and it is not properly derived from an energy function minimization. We decided to use it for two reasons: first, despite its simplicity, it can be used in all that segmentation tasks where the object to be segmented is well defined by its intensity values, second, it is useful to perform a fair comparison with existing GPU level-set methods since it is usually the only speed function used by authors to validate their approach [161, 162, 163, 148].

The second speed function we will use corresponds to the region-based model proposed by Chan and Vese [131] and its data attachment term is written as:

$$D(\mathbf{x}) = (I(\mathbf{x}) - u)^2 - (I(\mathbf{x}) - v)^2 \quad (4.10)$$

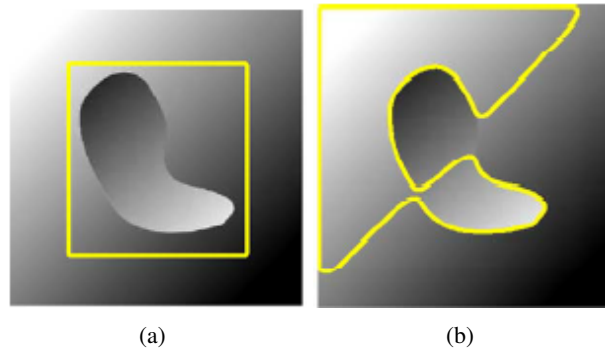


Figure 4.10: Synthetic image with heterogeneous intensity profile: Initial contour (a) and unsuccessful result of region-based segmentation (b).

where  $u$  and  $v$  correspond to the local mean inside and outside the surface computed at each iteration, respectively. This method aims at minimizing the variance inside and outside the surface during its evolution. The corresponding speed function is thus time dependent. Region-based active contours proceed by partitioning the image into regions which are homogeneous *w.r.t.* some defined criteria. The Chan-Vese method fairly represents the most commonly adopted level-set strategy, and it has several advantages *w.r.t.* thresholding or edge-based strategies: it is particularly useful to segment object in presence of noise and it is robust against initial surface placement. However, region-based techniques are derived on the base of a strong hypothesis, consisting in the fact that regions of interest can be differentiated in terms of their global statistics. In many practical situations, the object to be segmented may have heterogeneous intensity profiles, for instance due to spatial variations in the illumination. In these situations traditional region based algorithms may lead to wrong segmentation results (see Figure 4.10).

To accurately segment these objects, a new class of active contours energies has been presented in [154], which utilizes local information instead of global one. The basic principle that underpins this methods is that, if the homogeneity hypothesis is not globally satisfied, there is a greater probability for it to be satisfied within smaller sub-regions. As a consequence, the mathematical framework for localizing region-based energies proposed by Lankton [154] makes the assumption that the analysis of local regions leads to the construction of a family of local energies at each point along the curve. To optimize these local energies each point is considered separately and moves to minimize or maximize the energy computed in its own local region. To compute these local energies, local neighborhoods of each point are defined and split into local interior and local exterior by the evolving curve, and the energy optimization is then done by fitting a model to each local region.

The variational formulation consequently derived generalizes the region-based formalism including such a localizing principle. The data term of a localized region-based model states as:



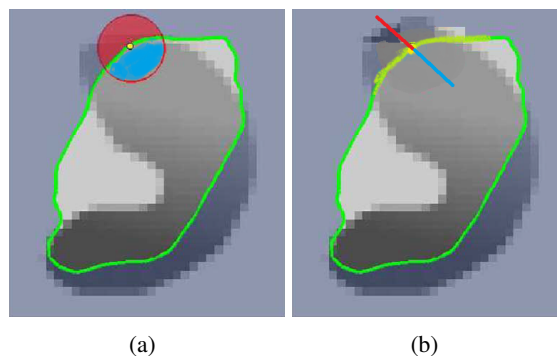


Figure 4.11: Local neighborhoods of contour points: circular (spherical) neighborhood (a) and neighborhood on the vector normal to the evolving surface (b). In each neighborhood local interior is indicated in blue, while local exterior is indicated in red.

$$D(\mathbf{x}) = \int_{\Omega} B(\mathbf{x}, \mathbf{y}) \cdot \nabla_{\phi(\mathbf{y})} G(I(\mathbf{y}), \phi(\mathbf{y})) d\mathbf{y} \quad (4.11)$$

where  $B(\cdot)$  is a binary mask function centered at point  $\mathbf{x}$  and  $G(\cdot)$  is a feature function depending on the application.  $B(\cdot)$  allows to select a neighborhood around  $\mathbf{x}$  where the local parameter driving the evolution are estimated. In [154], authors suggest the use of a radial mask around the interface points under evaluation. In this study, in order to maintain low computational cost, we define  $B(\cdot)$  as a set of points belonging to the normal direction to the interface at point  $\mathbf{x}$  and whose distance is smaller than a parameter  $\rho$ .  $B(\cdot)$  can therefore be defined as:

$$B(\mathbf{x}, \mathbf{y}) = \begin{cases} 1, & \text{if } \mathbf{y} = \mathbf{x} + k \times \hat{N}, k \in [-\rho, \rho] \\ 0, & \text{otherwise} \end{cases} \quad (4.12)$$

where  $\hat{N}$  is the normal vector to the interface at point  $\mathbf{x}$  and  $\rho$  the size of the local neighborhood. An example of the two different local neighborhoods is shown in Figure 4.11. Since the local statistics are computed at each iteration also this speed function is time-dependent.

The localized region-based approach is particularly useful to segment the endocardial borders in RT3DE images, since they appears as strongly inhomogeneous images.

To apply the localized region-based approach to the segmentation of the LV in RT3DE we will use the localized version of the global region-based Yezzi's energy [150]. This energy relies on the assumption that foreground and background regions should have maximally separate mean intensities. Optimizing the energy causes the curve to move so that interior and exterior means have the largest difference possible. The corresponding speed function data term states as:

$$D(\mathbf{x}) = (u - v)^2 \quad (4.13)$$

where  $u$  and  $v$  correspond to the local mean inside and outside the surface computed at each iteration, respectively.

In particular, for our target application, a slightly modified version of the Yezzi's energy is used [140]. This modified version takes into account the darker appearance of the blood *w.r.t.* the myocardial tissue. The data term of the corresponding speed function is:

$$D(\mathbf{x}) = (u - v) \quad (4.14)$$

The localized version of (4.13) is defined as:

$$D(\mathbf{x}) = \int_{\Omega} B(\mathbf{x}, \mathbf{y}) \delta_{\phi}(\mathbf{x}) \cdot \left( \frac{(I(\mathbf{y}) - u_x)}{A_u} + \frac{(I(\mathbf{y}) - v_x)}{A_v} \right) d\mathbf{y} \quad (4.15)$$

Basically, with this modification, whenever  $u > v$  this segmentation energy will have a positive value, which is penalized in the minimization strategy. Doing so, we are enforcing the mean outside to be greater than the one inside, and the energy minimization will converge to a local optimum corresponding to the interface maximizing the signed distance separation between the local means around the interface.

Even the localized region-based approaches can lead to unsatisfactory segmentation results when segmenting meaningful objects from images when they are occluded, when some part of them are in low contrast with regions or even missing. In these cases, high level prior knowledge about the expected shape can be exploited in order to improve the segmentation accuracy.

For this reason, in our segmentation framework, we also investigate the introduction of a shape prior term that constrains the interface to remain locally close to an ellipsoid over time. In fact, to successfully use shape prior knowledge, the reference shape model should be as much as possible similar to the target shape: ellipsoid, is the shape that better approximates the ventricle shape in 3D.

Inspired by the framework of Chen *et al.* [156], the corresponding expression is:

$$D(\mathbf{x}) = \left( \frac{\langle \nabla \psi(\mathbf{x}, \lambda), \nabla \phi(\mathbf{x}) \rangle}{\nabla \phi(\mathbf{x})} + \psi^2(\mathbf{x}, \lambda) \cdot \kappa \right) \quad (4.16)$$

where  $\psi(\cdot)$  evaluates the algebraic distance of a point  $\mathbf{x}$  to an ellipsoid defined by the set of parameters  $\lambda$ ,  $\kappa$  is the curvature of the evolving interface and  $\langle \cdot, \cdot \rangle$  denotes the scalar product.

In level set literature, the shape information is often taken into account by considering the linear combination, of the data attachment (and curvature) term (s), and a shape prior term [164, 156]. How the segmentation procedure is modified introducing the shape constraint, will be detailed in the following (see subsection 4.5.2).

### Fast level-set methods

Level-set method is very popular in imaging community since it intrinsically provides many advantages: it naturally deals with topology changes, it provides intrinsic geo-

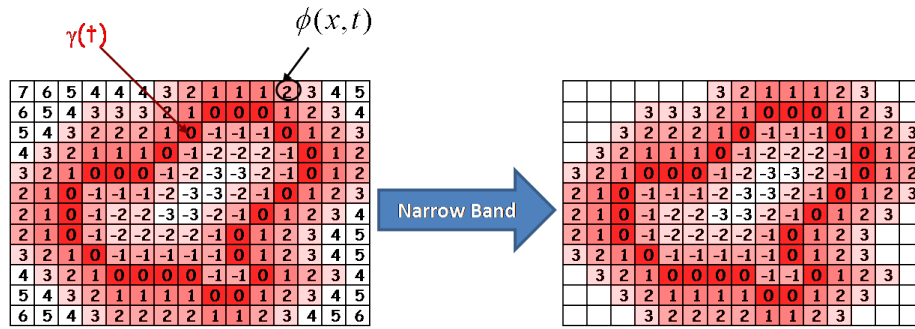


Figure 4.12: Computational domain restriction obtained using the narrow band approach.

metrical properties of the detected object such as normal and curvature information. Moreover the same formalism works in any dimension. Unfortunately, the flexibility of the level-set method results in long computational times and therefore has limited its attractiveness for several image processing domains such as medical imaging. Several strategies have been proposed to encounter this limitation. Among them, the most efficient algorithms make the level-set evolve only into a region close to the zero level-set, usually called narrow band. There exist principally two families of such models: the narrow band [165] and the sparse field methods [141].

Narrow band methods restrict the computation of the level-set evolution equation to a band of few voxels (usually 10 or 20) around the zero level-set. The embedding function  $\phi$  usually corresponds to a signed distance function which is initialized using a fast marching algorithm. Whenever the evolving interface reaches the boundaries of the local region defined by the band, the embedding  $\phi$  is reinitialized and the computational domain is updated.

Considering a discrete representation of the level-set function on a rectilinear grid of points and the use of the signed distance function as level-set function (see Figure 4.9), the basic principle of the narrow band method can be illustrated as in Figure 4.12. Thus, basically, the narrow band algorithm restricts the computation of the level-set PDE to a band of 10-20 grid points around the level-set of interest. Grid points outside the band are set to a constant value and therefore do not get updated, as shown in Figure 4.12. All grid points within the band undergo the same level-set update equation. Where appropriate, the velocity is extended from the position of the zero level-set (designated curve or surface) to the rest of the computational domain using nearest distance. Special care must be taken at the edge of the domain to ensure that the boundary conditions are appropriate. As the level set approaches the edge of the computational domain, the solver stops the Euler updates, computes the current position of the level-set (to subgrid accuracy), and rebuilds the embedding within a new computational domain centered around the current surface. A trade-off exists between the size of the band and the frequency at which the evolving surface requires a reinitialization of the narrow band.

The sparse field method takes the narrow band concept to the extreme by restricting the level-set function computation on a narrow domain that is only one voxel (and therefore one cell) wide. A narrow band, precisely of the width needed to compute the derivatives, is maintained, via a distance transform as the surface evolves. Thus, it is essentially a narrow band method in which the domain is reinitialized after every Euler update. This reinitialization is made computationally feasible via an approximation to the signed distance transform. Keeping track of this narrow band at each iteration requires maintaining lists of neighborhood points of the points belonging to the zero level-set. The sparse field method computes the velocity of the zero-level set using a first-order approximation to its position from the updated grid points. Thus, there is neither an extension of velocity required nor any boundary stability issues, unlike the narrow band method.

Despite the design of narrow band strategies, sequential level-set solvers (CPU level-set solvers) remain too much time consuming for effective (interactive) use in 3D applications. To overcome these problems, we developed a new GPU level-set segmentation framework for 3D images, in order to enhance the computational cost reduction provided by the narrow band methods exploiting the GPU computational power. In particular, our GPU level-set solver is a rigorous GPU implementation of the sparse field algorithm.

In recent years, the development of fast GPU level-set solvers has received increasing attention [162, 163, 148, 166, 167, 168, 169, 170, 161]. In this thesis, as a background of our work, we provide an overview of the existing GPU level-set methods focusing our attention on the current most efficient GPU level-set solvers that involve narrow band strategies. Therefore, in the following section, after a brief review of the existing GPU level-set solver the proposed approach will be described, and its performance as both a general purpose level-set segmentation framework for 3D images, and as support tool for automated quantitative analysis of cardiac morphology and function will be evaluated.

## **4.5 Proposed approach: a rigorous parallel implementation of the Sparse Field algorithm**

As stated above, the major goal of the research described in this chapter is the development of a GPU based level-set segmentation framework for 3D images and its application to the LV myocardium segmentation in RT3DE. The developed segmentation framework is, to the best of our knowledge, the first rigorous GPU implementation of the level-set sparse field algorithm. As a key property it avoids the doubly-linked-lists management inherent to the sparse field model which makes it poorly suited for parallel programming. Moreover, our GPU level-set solver presents an interesting compromise between very high computational efficiency and flexibility in speed function involvement. In the next sections, we show that the proposed GPU level-set solver is able to reach high computational efficiency with no reduction in segmentation accuracy compared to its sequential counterpart, and obtain comparable performance with existing

GPU level-set solvers. We also show that the flexibility of our approach in the interface evolution makes our level-set solver able to handle different segmentation tasks. Thanks to its flexibility, using a localized region-based segmentation approach, we make our GPU level-set solver able to deal with strongly inhomogeneous images such as the myocardial wall in RT3DE, achieving the final goal of this work, which is the development of a fast and accurate automated segmentation framework that acts as a support tool for cardiac morphology and function assessment. To handle this segmentation task, we also first introduce in a GPU based level-set segmentation framework a shape prior knowledge, focusing on the segmentation of shapes which may be approximated by an ellipsoid.

#### 4.5.1 GPU level-set methods

Narrow band strategy has been successfully ported on GPU: the corresponding classical GPU solvers are based on a close cooperation between GPU and CPU [162, 163, 148]. In the approach proposed by Lefohn [162, 163, 148] the CPU uses virtual memory paging schemes to partition the level-set field into tiles and maps each active tile (*i.e.* each tile containing elements in the narrow band) to the GPU memory. The GPU performs level-set computations and generates a down-sampled memory image of active tiles which is subsequently downloaded by the CPU at the end of each iteration. The CPU is responsible for traversing the down-sampled image, determining the narrow band for the next iteration, and communicating this result back to the GPU.

More recently, Jehong *et al.* [170] proposed an approach that reduces the interaction between CPU and GPU (avoiding the consequent inherent latency) by traversing the domain of active tiles in parallel on the GPU. When a thread determines that a new tile should become part of the narrow band, the thread appends that tile to a list of active tiles stored in GPU memory. The sparse level-set method proposed by Jalba *et al.* [171] parallelizes the Sorted Tile List (STL) method reducing the memory requirement of the classical narrow band method. The STL method divides the domain into fixed active tiles: tiles outside the narrow band are discarded and the remaining narrow band tiles form the active set which is a list of tiles lexicographically ordered by coordinates. The level-set is updated keeping track of a  $3^3$  pointers to the current and neighbouring active tiles. As the interface evolves new tiles are created and existing tiles that are no longer close to the zero level-set are removed. The corresponding GPU level-set method is similar to the one proposed by Lefohn since it uses tiles to represent the narrow band around the interface. Although the memory requirements are relaxed and the complex paging mechanism is avoided, a close cooperation between CPU and GPU is still present. Moreover performance of their method are investigated only for surface reconstruction from unorganized cloud of points.

The sparse field (SF) algorithm [141] has been proven to give the most efficient results with sequential level-set solver thanks to the use of a narrow band of minimal size. However, despite its attractiveness, to the best of our knowledge, there doesn't exist any

rigorous GPU implementation of such approach. The main reason is the difficulty to apprehend in parallel the linked-list management inherent to the sparse field method. In this context, Roberts *et al.* have recently proposed in [161] a GPU algorithm for level-set segmentation whose principle is both inspired from narrow band and SF algorithms. From the observation that computations near the evolving surface can be avoided in regions where the level-set field has locally converged, the authors proposed an algorithm based on the tracking of active computational domain according to both the temporal and spatial derivatives of the level-set field. By doing so, Roberts *et al.* show that their model reduces the total number of processed level-set field elements by 16 and the corresponding algorithm is 14-times faster than previous narrow band GPU level-set solvers with no reduction in segmentation accuracy for typical 3D medical images. Even if the temporally coherent sparse field algorithm sounds to be promising, this method was validated by the authors using only the time-constant speed function (4.9). In this case, the assumption that a part of the surface that has been converged should not move again is correct and thus the algorithm produces accurate results. However, when time varying speed functions such as (4.10) are considered, the same assumption could not be maintained because there is no guarantee that a converged part of the surface will not move again during the segmentation process.

#### 4.5.2 Parallel implementation of the sparse field algorithm

Sparse field algorithm efficiently manages the level-set evolution by restricting the width of the involved narrow band to its minimum size (1 voxel width). More particularly, the grid points inside the narrow band (active points) are initialized with a signed distance function computed from an initial interface (that corresponds to the zero level). Then, both the level-set values and the computational domain are updated at each iteration according to the level-set equation. Since it computes the updates on so few points, this approach requires maintaining a neighborhood around the active points in order to compute the derivatives that control the process with sufficient accuracy. This neighborhood is maintained in layers whose number should correspond to the width needed to calculate the derivatives using finite difference scheme. Since only first and second-order derivatives are considered, then five layers are used.

Let  $L_0$  be the active set, *i.e.* the set of voxels belonging to the zero level-set. The neighborhood of  $L_0$  is defined in layers  $L_i$  for  $i \in \{\pm 1, \pm 2\}$ , where  $i$  indicates the distance of a neighborhood point from the nearest active point (see Figure 4.13). More practically, the values of  $\phi$  for the points belonging to  $L_0$  varies between  $-0.5$  and  $0.5$ . The values of  $\phi$  in each neighborhood layer are kept 1 unit from the layer closest to the active set. The evolution of the level-set is then realized by updating the  $\phi$  values thanks to (4.4) and consequently moving the corresponding points in the appropriate layer. During the evolution further five status lists  $S_i$  are used in order to track points that change from a layer to another one [141].

Efficient sequential implementations of this algorithm use doubly-linked-lists to im-

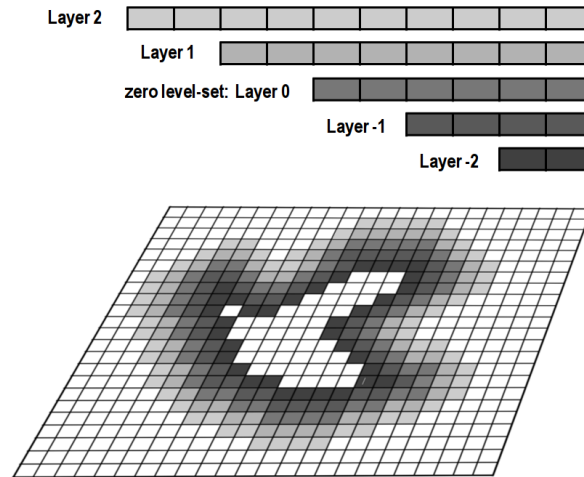


Figure 4.13: Illustration of the definition of the active set and its neighborhood in layers.

plement both  $L_i$  and  $S_i$ . Unfortunately, these structures are poorly suited to parallel implementation. Our approach overcomes this limitation by treating each list as a 1D buffer, whose size is equal to the support of  $\phi$ , and by using the voxel coordinates as array indices. Each list contains only values '1' or '0' to indicate the presence (or not) of a voxel in the list (see Figure 4.14). In addition we use two further arrays: one to maintain the level-set values ( $\phi$ ) and another to record the status of each point and provide a way to track which list a point belongs to (*label*). *Label* contains values in the range  $[-3, 3]$  and allows encoding each layer as well as regions inside and outside all layers. In this context, the parallelism of our implementation has the granularity of the individual voxel. It means that each thread manages an element of each buffer.

### Initialization procedure

From an input binary mask, we pre-initialize in parallel the *label* and  $\phi$  buffers to the maximum and minimum value of the label range. We then initialize the active set  $L_0$  by observing that, if a foreground voxel has at least one of its 6 neighbors in the background, it is a contour voxel and it should belong to  $L_0$ , so *label* and  $\phi$  must be updated. These operations are summarized in Algorithm 8. Then, in parallel for each point  $p$  in the domain, we check if it belongs to  $L_0$ . If so, we sequentially analyze its neighbor values  $\eta(p)$  in *label* and use this information to populate  $L_{\pm 1}$  with the neighbors of the active points and consequently update *label* and  $\phi$ . Similar operations are then performed for others layers.

This workflow is divided into different CUDA kernels to manage the data dependencies between different layers (when a thread  $r$  processes a point  $p_r$  in layer  $L_i$  one has to take care that its neighbors in layer  $L_{i\mp 1}$  have been completely processed). Therefore five different kernels are run in order to implement this procedure: one for the initializa-

---

**Algorithm 8:** Initialization procedure

---

**Input:**  $mask$   
**Output:**  $label, \phi, L_0, L_{-1}$  and  $L_1$

```

for all points  $p$  in  $\Omega$ , in parallel do
     $label(p) = 3; \phi(p) = 3;$ 
    if  $mask(p) == 1$  then
         $label(p) = -3; \phi(p) = -3;$ 
for all points  $p$  in  $\Omega$ , in parallel do
    if  $mask(p) == 1$  and at least one point in  $\eta(p) = 0$  then
         $L_0(p) = 1; label(p) = 0; \phi(p) = 0;$ 
for all points  $p$  in  $\Omega$ , in parallel do
    if  $L_0(p) == 1$  then
        for each point  $q$  in  $\eta(p)$  do
            if  $label(q) == -3$  then
                 $L_{-1}(q) = 1; label(q) = -1; \phi(q) = -1$ 
            if  $label(q) == 3$  then
                 $L_1(q) = 1; label(q) = 1; \phi(q) = 1$ 

```

---

tion of  $label, \phi$  and  $L_0$ , one for initializing  $L_2$ , one for  $L_{-2}$  and one for initializing both  $L_{-1}$  and  $L_1$  since they share data dependencies with  $L_0$ . Each kernel is launched with two dimensional thread blocks with fixed size for efficiency purposes. These blocks are organized in bi-dimensional grids whose size is computed to traverse the entire volume domain. By appropriately offsetting the base address each 3D coordinate is converted into a 1D memory address. Moreover, we use atomic operations to safely update  $label$ , since it is read and written by neighbors threads.

### Updating procedure

In this phase we update  $\phi$  according to the evolution equation (4.5). Note that since the SF algorithm reinitializes the level-set to a signed distance function at each iteration, we assume that  $\|\nabla\phi\| = 1$ . Therefore we compute only the data term and the curvature term of our speed function and consequently update  $L_0$  in parallel. This procedure is summarized in Algorithm 9.

Once the previous step of the updating procedure is done, we then traverse in parallel each layer  $L_i$  in order to identify the points whose status has changed. Each thread manages an element of the buffer corresponding to  $L_i$ . If it belongs to the layer the corresponding element of  $\phi$  is updated (adding or subtracting a unit), based on the values of the next inner layer  $L_{i\mp 1}$  (*i.e.* the next layer closest to the active set). By doing so, the values of  $\phi$  are updated so that they are exactly one unit from their nearest neighbor



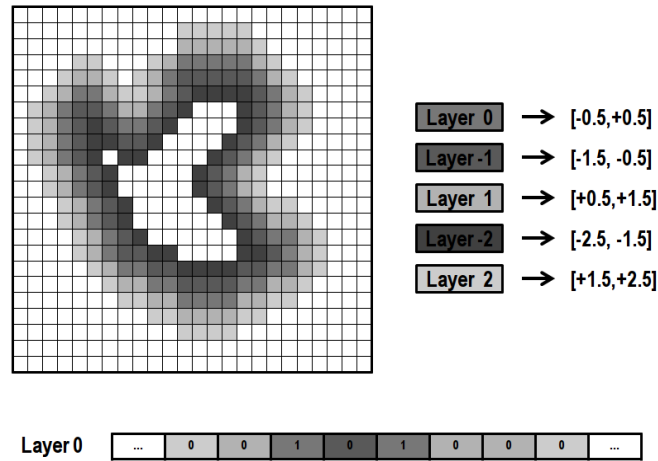


Figure 4.14: Illustration of the SF representation of the  $\phi$  domain, the use of layers of points  $L_i$ , and the interpretation of layers as 1D array adopted by the P-SF. Numeric range for each list refers to the  $\phi$  values for each layer.

---

**Algorithm 9:** Updating procedure: first step

---

**Output:**  $\phi, L_0, S_1, S_{-1}$

```

for all points  $p$  in  $\Omega$ , in parallel do
    if  $L_0(p) == 1$  then
        Compute  $F(p)$ ;
         $\phi_{new}(p) = \phi(p) + F(p) \cdot \Delta\tau$ ;
        if  $\phi_{new}(p) > 0.5$  then
             $L_0(p) = 0$ ;  $S_1(p) = 1$ ;
        if  $\phi_{new}(p) < -0.5$  then
             $L_0(p) = 0$ ;  $S_{-1}(p) = 1$ ;
    
```

---

in the next layer closest to the active set. If more than one neighbor exists, we use the maximum for the inside layers and the minimum for the outside layers. If no neighbor exists, the element is directly removed from  $L_i$  and added to the appropriate list  $S_i$  (or removed from all the lists by updating  $\phi$  and *label*). An example of such procedure applied to  $L_{-1}$  is given in Algorithm 10.

Finally, to complete the updating procedure, one needs to deal with the status lists  $S_i$ . This is achieved exploiting the parallelism as done for the layers management. An additional step could be required related on the level in the narrow band: points that belong to  $S_0$  are removed from the corresponding list and added in parallel to  $L_0$ . *Label* is then updated. Similarly, points that belong to  $S_{\pm 1}$  are moved in parallel to the appropriate  $L_i$  and *label* is updated. Moreover, if the points in  $L_{\pm 1}$  have neighbors outside or inside all layers, the neighbors must be added to the appropriate list  $S_{\pm 2}$  and  $\phi$  must be

---

**Algorithm 10:** Updating procedure: second step

---

**Output:**  $\phi, L_{-1}, S_0, S_{-2}$

```

for all points  $p$  in  $\Omega$ , in parallel do
    if  $L_{-1}(p) == 1$  then
        if there are no neighbors  $q$  in  $\eta(p)$  with  $label(q) == 0$  then
             $L_{-1}(p) = 0; S_{-2}(p) = 1;$ 
        else
             $g = \max(\phi(q))$  with  $label(q) \geq 0;$ 
             $\phi(p) = g - 1;$ 
        if  $\phi(p) \geq -0.5$  then
             $L_{-1}(p) = 0; S_0(p) = 1;$ 
        if  $\phi(p) \leq -1.5$  then
             $L_{-1}(p) = 0; S_{-2}(p) = 1;$ 

```

---

updated. Finally points in  $S_{\pm 2}$  are moved to the appropriate list  $L_{\pm 2}$  and  $label$  is updated. As in the previous step, these operations are split into different kernels to ensure the synchronization between different layers and status lists. Each kernel is launched with the same thread grid described above. Atomic operations are used to safely manage the updating of structures that are read and written by neighbors threads avoiding data race conditions. All the required buffers are stored in the global GPU memory. In fact, due to the characteristics of the proposed algorithm, memory coalesced access cannot be guaranteed, it is not clear how to take advantage of GPU shared memory for level-set computations. For this reason our implementation does not use GPU shared memory.

### Time step management

The time step  $\Delta t$  in (4.5) must be defined to ensure the stability of the solution which implies that small errors in the solution are not amplified during the evolution. The stability is enforced using the Courant Friedrichs Lewy (CFL) condition:

$$\Delta \tau \leq \frac{\Delta x}{\max |F(\mathbf{x}, \tau)|} \quad (4.17)$$

The computation of the CFL number is a time consuming step since it requires the determination of the largest element of  $|F(\cdot)|$  at each iteration. Therefore, an arbitrary step  $\Delta \tau$  is usually chosen to ensure the stability without significantly slowing down the computation time. Our framework provides the possibility of computing the time step following the CFL condition. We have experimentally investigated the influence of the choice of  $\Delta \tau$ : small changes of this value could dramatically affect the quality of the result. It is thus preferable to automatically evaluate the optimal value at each iteration using (4.17). A parallel implementation of the CFL condition can follow two

different strategies. The first solution consists in transferring the result of the speed function computation back to the host memory in order to determine  $\max |F(\mathbf{x})|$ . The second solution implies the use of specific kernel to find the largest value of  $F$  in parallel. Here, we decided to choose the latter approach and we have exploited parallel reduction techniques [76] to compute the highest value of  $F$ . Thus, an additional buffer to maintain values of  $F$  is required. The CFL condition computation at each iteration is enabled or disabled by a flag whose value is set by the user.

### Specific parallel implementation of speed functions

In order to ensure the flexibility of our solver, specific CUDA kernels were developed according to the speed function involved in the computation procedure. More particularly, when considering time-varying terms, efficient computations of the intrinsic statistical parameters are required. For this reason, to efficiently implement the Chan-Vese speed term we use four auxiliary buffers to mainly track the points that cross the zero level during the evolution. Two of them are used as auxiliary lists: the first is assigned to points changing from interior to exterior, whereas the other is assigned to points changing from exterior to interior. They contain only values '0' or '1' to keep track of points in the list, as done for layers and status lists. The remaining buffers are used to store the image intensity values of the changing status points. All these buffers are updated at each iteration checking the sign of  $\phi$  before and after it has been updated thanks to (4.5). Since at each iteration one can easily know the points that have moved inside or outside the zero level, it is easy to update the underlying statistical parameters. It has to be noticed that these buffers are also processed using parallel reduction techniques for efficiency reasons.

By combining the localized region-based approach with the SF method implies that at each algorithm iteration a local neighborhood of size  $\rho$  must be identified for each point in  $L_0$  and local statistics must be computed on it. Since the direction of  $\hat{N}$  changes during the evolution, this direction and the corresponding neighborhood must be computed at each iteration. For an efficient parallel implementation we associate each thread to one local neighborhood. Thus each thread manages a voxel of the active set  $L_0$  so that at each iteration all the required local statistics can be computed in parallel, while computation in each neighborhood are performed sequentially by the associated thread. The computation of 4.15 proceeds as follows. First, each thread manages an element of the domain and checks if it belongs to  $L_0$ . If it is the case the thread computes the components of  $\hat{N}$  in that point and consequently identifies the neighborhood for the local statistical computation. Then, based on the corresponding  $\phi$  values, the thread checks if each neighbor is inside or outside the contour and uses this information to compute the local statistics inside and outside. Since the neighborhood and the statistics must be computed at each iteration no additional buffers are needed.

Concerning the use of shape prior knowledge we observe that, introducing the shape constraint, the segmentation procedure consists in a two stage procedure: (i) fitting the

reference shape to the level-set; (ii) evolving the level-set through the following equation:

$$\frac{\partial \phi}{\partial t}(\mathbf{x}) = \delta(\phi(\mathbf{x})) \cdot g(\mathbf{x}, \lambda) \quad (4.18)$$

The fitting of the reference shape is done by finding the parameter vector  $\lambda$  at each iteration solving the linear least square fitting problem:

$$\hat{\lambda} = \underset{\lambda}{\operatorname{argmin}} \int_{\Gamma} \psi^2(s, \lambda) ds \quad (4.19)$$

To solve in parallel the fitting problem (4.19) we use CULA [172], an implementation of the Linear Algebra PACKage (LAPACK) interface for CUDA-enabled NVIDIA GPUs. The CULA GELS routines solve overdetermined or underdetermined real linear systems involving an M-by-N matrix  $\mathbf{A}$ , or its transpose, using a QR factorization. Using appropriate options we find the least squares solution of the overdetermined system,  $\min \|\mathbf{B} - \mathbf{A}\mathbf{X}\|$ , where  $\mathbf{A}$  is the ellipsoid parameters matrix and  $\mathbf{B}$  is a vector of constant terms.

Note that fitting the ellipsoid to the level-set using the sparse field method means to consider the set of points belonging to the list  $L_0$ . In order to reduce the segmentation time, the ellipsoid parameters updating can be performed only during the first algorithm iteration. In fact, the adoption of a localized segmentation approach imposes an initialization not too far from the desired solution to obtain meaningful results. Thus, it can be assumed that the ellipsoid fitted during the first iteration is a reliable reference shape model to drive the evolution.

In order to perform the segmentation with ellipsoidal shape prior the level-set evolves according to the following equation:

$$\frac{\partial \phi}{\partial t} = \delta(\phi(\mathbf{x})) [f(\mathbf{x}) + \beta \cdot g(\mathbf{x}, \lambda)] \quad (4.20)$$

where  $f(\cdot)$  is the data attachment term,  $g(\cdot)$  the shape prior term and  $\beta$  a parameter that weights the influence of the shape prior term. The choice of the weight term  $\beta$  is dependent on the specific application. Specific CUDA kernels are then used to compute the shape prior term as well as the overall speed term including the shape prior and the data attachment term.

## 4.6 Experimental results

To assess the segmentation accuracy of our solver and analyze its performance we first compared our parallel sparse field implementation (P-SF) with both its sequential counterpart (S-SF) and the temporally coherent sparse field algorithm (TC-SF) proposed by Roberts [161]. This performance comparison was carried out on both simulated and real datasets. As practical application on real medical dataset particular attention has been paid to the detection of vessels in 3D rotational angiography (3DRA) images. Then, to

demonstrate the applicability of the proposed segmentation framework in clinical environment for our target application, *i.e.* to perform an automated quantitative analysis of 3D cardiac morphology and function, an extensive clinical validation that compares our approach with manual contouring from expert physicians, has been carried out. In this application, to evaluate time performance of our segmentation framework, and demonstrate the effectiveness of using GPU for 3D LV segmentation, we compared our approach with its sequential counterpart.

All the experiments were performed on a quad-core Intel 2.4 GHz Xeon processor with 8GB of memory and an NVIDIA GTX580 graphic card with 512 streaming processors (CUDA cores) and 1.5 GB of DRAM. We implemented both our P-SF and the TC-SF algorithm using CUDA in Matlab<sup>TM</sup> environment. Our implementations of the TC-SF and the S-SF were developed based on a code provided by Roberts\* and on the open source implementation proposed by Lankton [173], respectively. We used a single CPU core to run the CPU level-set solver. The metric used to assess the segmentation accuracy corresponds to the standard Dice Coefficient [174]. This coefficient measures the correspondence between two segmented regions, and varies from 0 to 1: it is 1 when the two regions are coincident and 0 when they have null intersection. By defining  $\Omega_1$  and  $\Omega_2$  as the set of pixels enclosed by the closed contours  $C_1$  and  $C_2$  the Dice coefficient can be written as:

$$D(\Omega_1, \Omega_2) = \frac{2\#(\Omega_1 \cap \Omega_2)}{\#(\Omega_1) + \#(\Omega_2)} \quad (4.21)$$

where  $\#(\cdot)$  extracts the set cardinality. For all the experiments, the three methods were initialized with the same binary mask, corresponding to a sphere manually drawn by a user. It has to be noticed that the TC-SF solver proposed in [161] has been implemented and validated using a fixed time step. For a fair comparison, in all the experiments, the solvers work with a fixed time step whose value is manually fixed according to the image in order to maximize results accuracy. However, due to the poor image quality of ultrasound data, the segmentation of RT3DE images have been carried out using a dedicated initialization and the computation of the time step  $\Delta t$  according to the CFL condition described in subsection 4.5.2.

#### 4.6.1 Simulated experiments

In order to quantify the performance of our solver, we first segmented a set of simulated 3D images where the object of interest is known and used as reference. The accuracy of each solver has been measured by computing the Dice coefficient between the given result and the reference. Figure 4.15 provides a slice of each simulated image.

We first evaluated the speed up of our approach using 3D images with low level of difficulty, *i.e.* images where the object of interest is well defined by its intensity

---

\*The author would like to thank Mike Roberts for making the source code of his GPU level-set method available.

Table 4.2: Parameter values used for the segmentation of simulated images using Lefohn speed term.

	$\epsilon$	$T$	time step
Binary squirrel	90	255	0.4
MRI brain	40	145	0.4

Table 4.3: Segmentation results using a time constant speed function.

	DICE			Execution Time			# iter.
	S-SF	TC-SF	P-SF	S-SF	TC-SF	P-SF	
Binary squirrel	1	1	1	190.74	0.57	0.55	450
MRI brain	0.92	0.90	0.92	5093.27	5.93	7.76	2000

values. These images correspond to the binary squirrel and the simulated brain MRI<sup>†</sup> images shown in Figure 4.15-a and Figure 4.15-d, respectively. The size relative to the squirrel and the brain experiments are 148x108x147 and 181x217x181, respectively. These images have been segmented using the time-constant speed function of Lefohn (4.9) where the two underlying parameters  $\epsilon$  and  $T$  were manually fixed. We reported in Table 4.2 the chosen values for each experiment along with the selected time step value. For each experiment, the parameter  $\alpha$  is fixed to 0.1.

From the results provided in Table 4.3, we observe that for each segmentation task our solver obtains important speed-up, equal to 347 and 657 times *w.r.t.* its sequential counterpart while maintaining exactly the same level of segmentation accuracy. Compared to the TC-SF, our solver gives comparable time performance and segmentation accuracy. Figure 4.16 shows the 3D results we obtained using our solver.

In order to prove the flexibility of the proposed method, we then evaluate the performance of our solver using the two time-varying speed functions presented in section 4.4.3. The global region-based Chan-Vese speed function was evaluated on the noisy squirrel shown in figure 4.15-b (with a corresponding time step equal to 0.2) whereas the localized region-based speed function of Lankton was evaluated on the inhomogeneous squirrel given in figure. 4.15-c (with a corresponding time step equals to 0.5).

It has to be noticed that the latter segmentation approach is greater sensitive to initialization than the former: a good initialization should ensure that the localized contours use enough information and should avoid that some parts of the contours collapses due to a lack of support. To avoids these problems the initialization should be relatively close to the desired object boundaries [154].

On the other hand, a fair comparison with the TC-SF and with previous experiments, requires initializing the segmentation with a seed sphere inside the image. By initializ-

<sup>†</sup>The MRI brain image has been simulated from the BrainWeb [175]

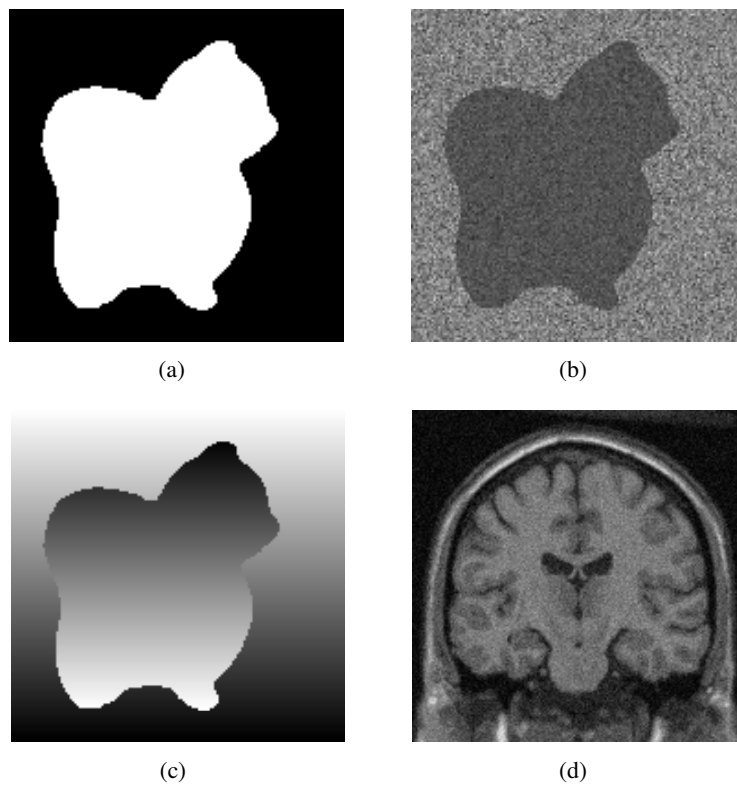


Figure 4.15: Simulated images used for the algorithms evaluation. Each image corresponds to a slice extracted from the corresponding volume.

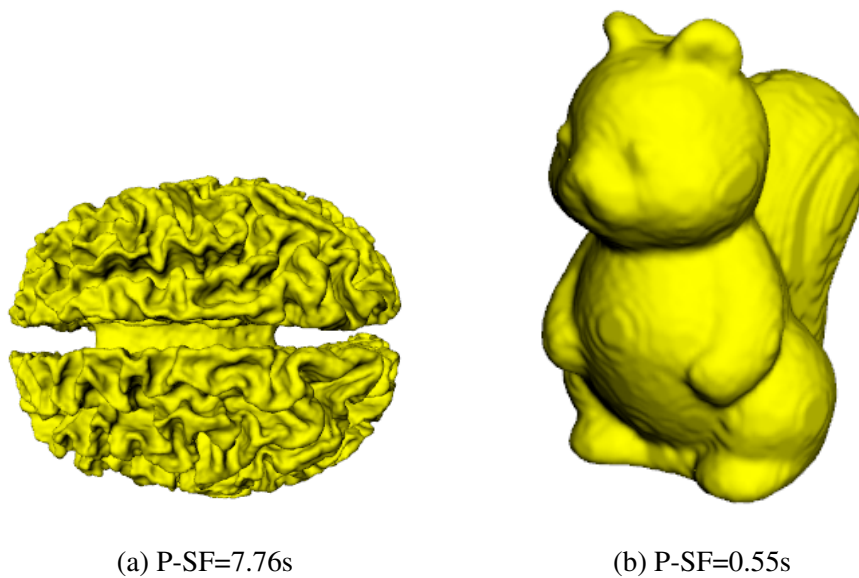


Figure 4.16: Segmentation results using our GPU level-set solver P-SF along with the corresponding execution time.

Table 4.4: Segmentation results using a time varying speed function.

	DICE			Execution Time			# iter.
	S-SF	TC-SF	P-SF	S-SF	TC-SF	P-SF	
Noisy squirrel	0.99	0.96	0.99	459.39	3.59	2.72	1000
Inhomogeneous squirrel (sphere)	0.96	0.83	0.99	17739.00	12.23	15.32	3000
Inhomogeneous squirrel (box)	0.99	-	0.99	777.10	-	3.59	1350

ing the localized region-based segmentation with such a sphere limits the accuracy of the results that could be obtained because it is not possible to create a sphere large enough to be uniformly close to the squirrel boundaries without reaching the image boundaries. For this reason, we compared all solvers performing the same segmentation task but using two different initialization strategies: a sphere for all codes and then a rectangular box for the S-SF and the P-SF.

For the inhomogeneous experiment, a localized version of the Chan-Vese energy functional was used and the size of the underlying neighborhood mask was manually set to 35 pixels when a spherical initialization is used, and to 13 otherwise.

From the results shown in Table 4.4, we first can see that the proposed solver still maintains the segmentation accuracy provided by its sequential counterpart (Dice coefficient either the same or very close). This confirms the effectiveness of our approach in managing different type of speed functions. Moreover, our implementation outperforms its sequential counterpart in terms of computational time with a speed up of 169 for the global approach and 1.158 times for the localized region-based approach initialized with a sphere. Note that the segmentation accuracy of our method is appreciably greater than the one obtained with the TC-SF model for both speed functions with similar execution time. This could be explained by the fact that the use of a time varying speed term could make the TC-SF work on false assumptions since there is no guarantee that a converged part of the surface will not move again during the segmentation process. Finally, the last row of the table shows the results for the inhomogeneous squirrel segmentation task initialized with a rectangular box, performed by P-SF and its sequential counterpart. The maintaining of segmentation accuracy with a speed up of 217 times confirms the ability of our approach to be really efficient and flexible in the interface evolution.



### 4.6.2 Detection of vessels with aneurysm in 3D rotational angiography images

Intracranial aneurysm are pathological dilations of the major arteries crossing through the subarachnoid space at the basis of the brain (circle of Willis). Aneurysmal rupture, leading to subarachnoid hemorrhage, is a devastating event and is associated with a high mortality and morbidity rate [176, 177]. Unruptured brain aneurysms do not have indicative symptoms associated and are therefore difficult to diagnose before rupture. Three-dimensional rotational angiography (3DRA) is a powerful technique for imaging blood vessels and is used as intra- and inter-operative tool for assessing intracranial aneurysms. Both the size and the shape of aneurysm as well as hemodynamics parameters are important risk factors for rupture and therefore are valuable for clinical evaluation and treatment. As a consequence an automatic and objective quantification of aneurysmal volume and shape parameters from medical images is highly desirable for planning preventive strategies in clinical practice. These measurements require accurate initial vessel segmentation which is investigated here using our GPU level-set solver.

In particular, we tested our method on two different 3DRA images centered on an aneurysm. For these two images, an expert manually delineated the different structures of interest and we used the corresponding results as reference. We then performed the segmentation task with the three solvers (S-SF, TC-SF, P-SF) using the global region-based Chan-Vese speed function. In each case, we evaluated the accuracy of the results both in terms of shape (through the Dice coefficient) and computational time. From the results presented in Table 4.5 we first can see that our solver provides exactly the same segmentation accuracy than its sequential counterpart which corresponds to a solution very close to the manual contours drawn by the expert (Dice coefficients higher than 0.9 in both cases). Moreover our algorithm outperforms its sequential counterpart in terms of computational time with a mean speed-up of 21 times. Finally, since a time-varying speed function has been used, it can be observed, as in the simulated cases, that our approach provides better results than the ones obtained from the TC-SF model. In order to better quantify this difference, we provide in Figure 4.17 the segmentation results we obtained for the first patient. From these results, we can observe that the TC-SF model leads to an oversegmented result, and this could be explained again by the fact that in the context of time varying speed function this algorithm works on false assumptions.

### 4.6.3 Left-ventricle segmentation in Real-Time 3D echocardiography

As stated above, despite the efforts made by medical vendors and researchers, 3D LV segmentation remains a time consuming task. In this context, we exploit our GPU level-set solver to achieve fast and accurate detection of the left-ventricle borders. In order to achieve this goal we used the localized segmentation approach described in section 4.4.3. In particular, we used the modified version of the localized Yezzi's energy function recently proposed by Barbosa *et al.* [140], which takes into account the darker appearance

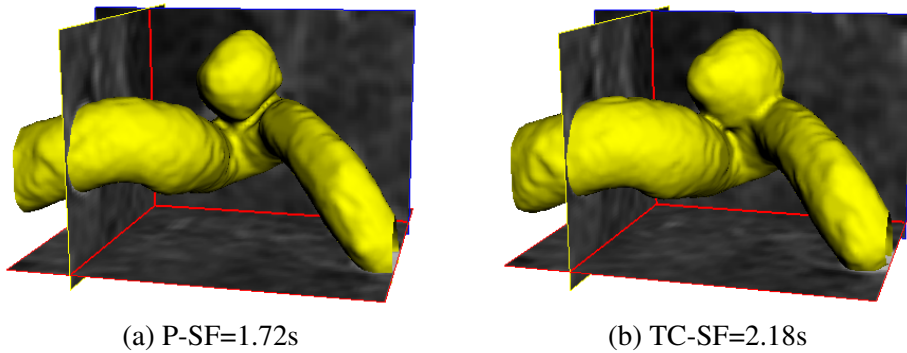


Figure 4.17: 3DRA segmentation results performed on Patient1 using P-SF (a) and TC-SF (b) algorithms along with the corresponding execution time.

Table 4.5: Vessel with aneurysm segmentation results obtained from a three-dimensional rotational angiography.

	DICE			Execution Time			# iter.
	S-SF	TC-SF	P-SF	S-SF	TC-SF	P-SF	
Patient 1	0.98	0.91	0.98	46.76	2.18	1.72	1000
Patient 2	0.90	0.87	0.90	21.14	1.45	1.42	1000

of the blood *w.r.t* the myocardial tissues, whose corresponding speed function is reported in (4.15).

To verify the clinical applicability of our method we measured clinical volumetric indices from 24 RT3DE exams. We then computed the agreement between the proposed method and the surfaces manually drawn by experts physicians. The dataset comprises both healthy subjects (5) and patients with different cardiac dysfunctions (19) and each exam is constituted by two volumes, one at the end-diastolic (ED) instant and one at the end-systolic (ES) instant. These data were acquired with a Siemens Acuson SC2000 (Siemens US, Mountain View, CA) using a 4Z1c matrix transducer. Volume sequences were acquired from an apical window. In the manual analysis each sequence is manually contoured by three expert physicians using a semi-automated segmentation tool (eSie LVA pre-release software, Siemens US, Mountain View, CA, USA). The reference LV volumetric parameters (end-diastolic volume [EDV], end-systolic volume [ESV], stroke volume [SV] and ejection fraction [EF]) were estimated as the average of the values estimated by the three experts. The automated analysis was performed using the proposed segmentation framework combined with the automatic initialization scheme proposed by Barbosa *et al.* in order to reduce the total LV analysis time and the user-dependency

Table 4.6: Left ventricle segmentation performance obtained from 3D Echocardiography imaging.

	Linear Regression Analysis		Bland-Altman Analysis		Time performance		
	Regression Equation	R	Bias	LOA ( $\mu \pm 1.96\sigma$ )	P-SF ( $\mu \pm \sigma$ )	S-SF ( $\mu \pm \sigma$ )	Speed up
EDV (ml)	0.86x+9.42	0.93	9.51*	[-25.80,44.82]	0.86±0.1s	13.49±9.82s	15.68 times
ESV (ml)	0.79x+7.65	0.93	6.24*	[-19.87,32.35]	0.82±0.09s	5.61±4.89s	6.84 times
SV (ml)	0.78x+12.14	0.83	3.27	[-25.66,32.20]			
EF (%)	0.68x+17.50	0.84	-0.77	[-13.45,11.91]			

[140]. In particular, we used an ellipsoidal mask to initialize the ED segmentation. We then used the corresponding segmentation result to create an initialization mask for the ES segmentation. Linear regression and Bland-Altman analysis were performed in order to investigate accuracy and agreement assessment between the volumetric indices estimated with our P-SF method and the reference ones (average of the volumetric indices extracted manually by the three experts). A significance level of  $p < 0.05$  was used in the statistical tests. In order to evaluate the time performance of our segmentation framework, we compared the results given by our approach with its sequential counterpart (S-SF) running on a single CPU core. Finally, it wasn't possible to investigate the performances of the TC-SF method on RT3DE images since it is not suited for segmentation tasks involving localized region-based velocity function. A summary of the obtained results is reported in Table 4.6.

The results of the linear regression analysis are shown in Figure 4.18. Regression analysis shows a strong correlation between LV volumetric parameters values estimated using the P-SF algorithm and the manual reference ones (Pearson product-moment correlation coefficients of 0.931, 0.931, 0.834, 0.837 for EDV, ESV, SV and EF respectively). Good agreement via linear regression and high correlation coefficient confirm the effectiveness of our method in segmenting the LV volumes for quantitative analysis assessment. Bland-Altman analysis, illustrated in Figure 4.19, reveals small biases and tight limits of agreement especially for EF, and only biases for EDV and ESV are statistically significant (paired t-test,  $p < 0.05$ ). Examples of the LV segmentation using our P-SF method are shown in Figure 4.20 for both end-diastolic and end-systolic instants.

Among the several studies that, as discussed above, have also used level-set model to try to assess clinical indices through the segmentation of the LV borders, the work proposed by Angelini *et al.* has produced one of the best results. In [130], they compared the results of their level-set model with manual tracing on 10 RT3DE exams. They reported a confidence interval ( $\mu \pm 1.96\sigma$ ) equals to  $16.11 \pm 50.07$  for EDV,  $-6.64 \pm 34.37$  for ESV, and  $-0.59 \pm 22.17$  for EF. They also reported correlation coefficients equal to 0.63, 0.62, and 0.45, for EDV, ESV and EF respectively. Therefore, our method compares favorably *w.r.t.* the Angelini's one showing tighter limits of agreement and stronger correlation coefficients.

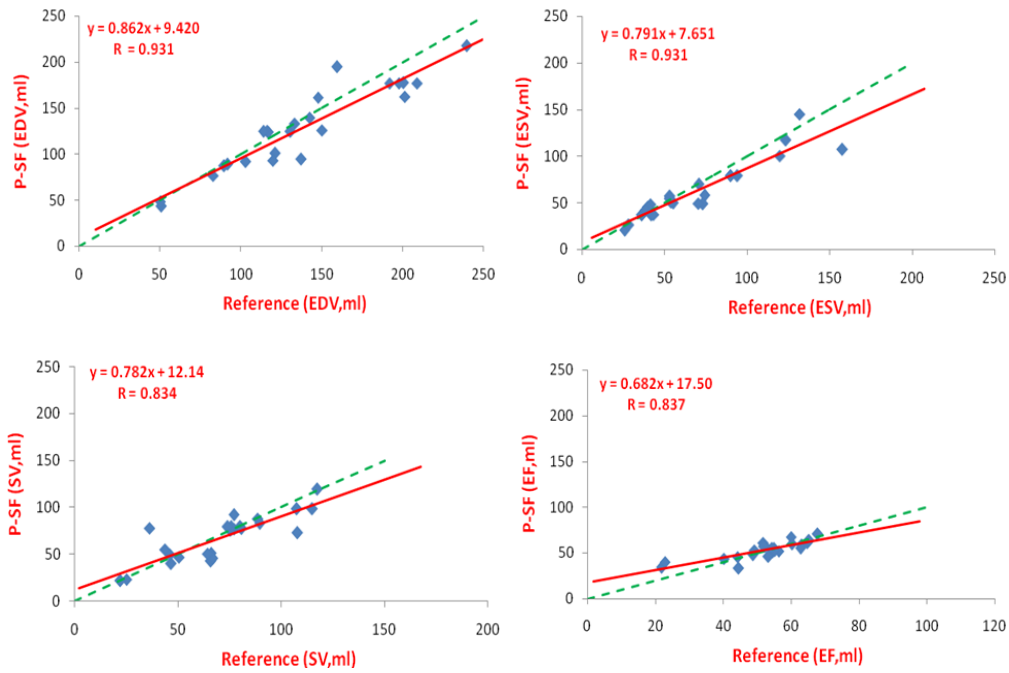


Figure 4.18: Linear regression analysis for EDV, ESV, SV and EF.

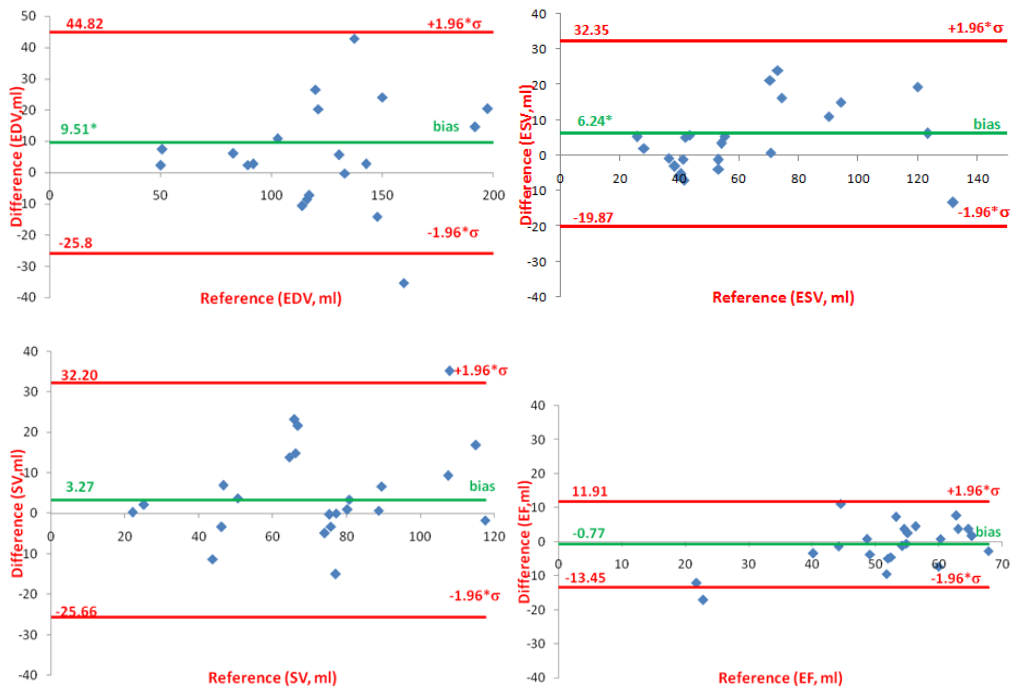


Figure 4.19: Bland-Altman analysis plots for EDV, ESV, SV, and EF, comparing the measured errors against the reference values. The green line corresponds to the bias (\*,  $p < 0.05$ ) and the red ones to the limits of agreement ( $\mu \pm 1.96\sigma$ ).

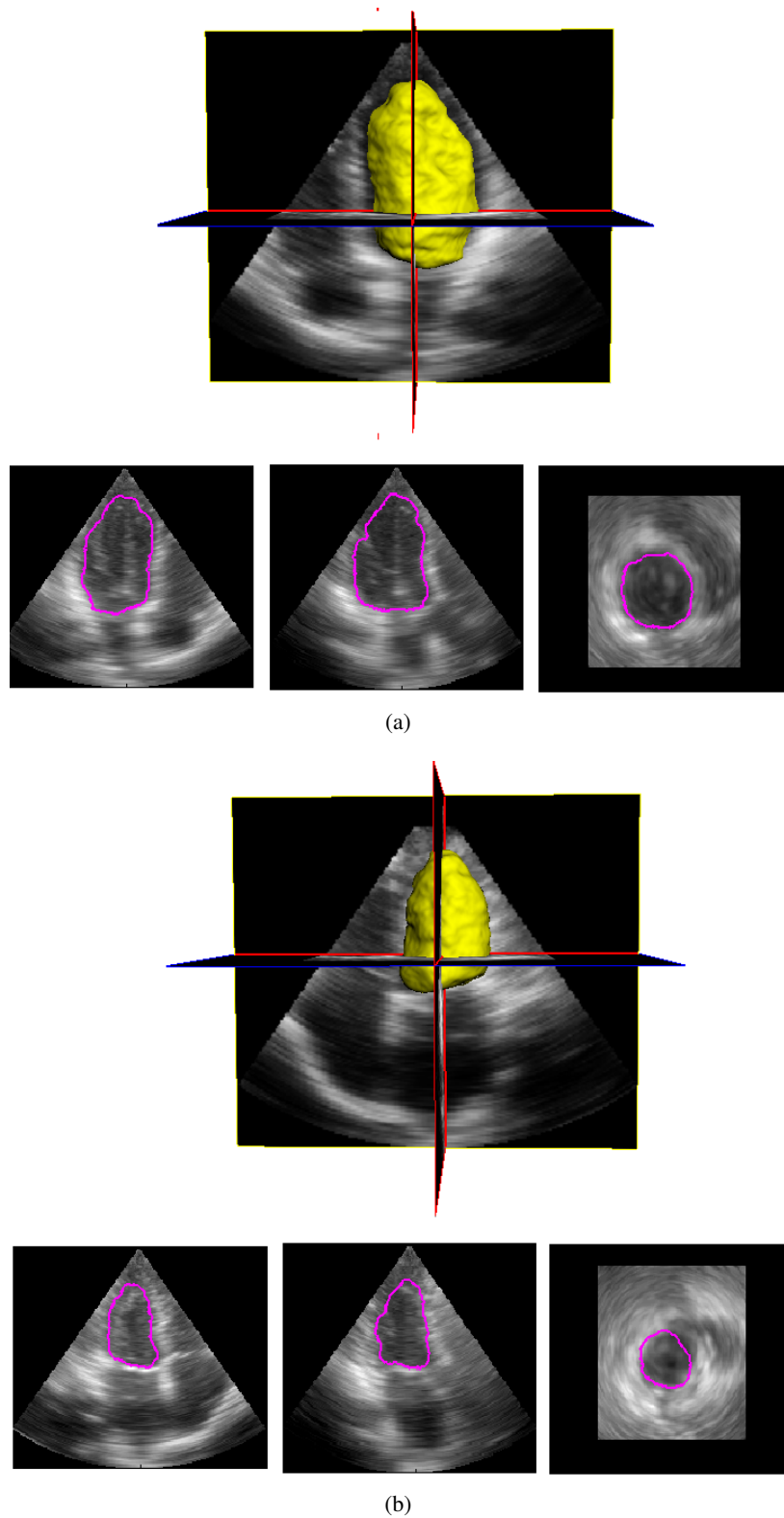


Figure 4.20: Segmentation of a RT3DE exam using P-SF. (a) EDV (Ref.=113.8ml, P-SF=124.5ml), (b) ESV (Ref.=39.8ml, P-SF=45.6ml).

Table 4.7: Left ventricle segmentation performance obtained from 3D Echocardiography imaging using a shape prior constraint.

	Linear Regression Analysis		Bland-Altman Analysis		Time performance
	Regression Equation	R	Bias	LOA ( $\mu \pm 1.96\sigma$ )	P-SF ( $\mu \pm \sigma$ )
EDV (ml)	0.85x+13.63	0.93	7.57	[-28.68,43.81]	0.68±0.09s
ESV (ml)	0.78x+6.31	0.93	8.56*	[-18.25,35.36]	0.66±0.09s
SV (ml)	0.74x+19.73	0.81	-0.99	[-31.56,29.59]	
EF (%)	0.63x+23.08	0.83	-3.74*	[-16.83,9.30]	

Table 4.6 also reports the execution time averaged across all the exams for both the P-SF and the S-SF as long as the corresponding speed up. The segmentation of each volume require 250 iterations and took about 0.84 seconds on average. Speed up equal to 15.68 times and 6.84 times were obtained respectively for the EDV and ESV segmentation. It can thus be stated that the proposed segmentation framework allows efficient and accurate segmentation of the myocardium in RT3DE with significant reduced computational cost.

Finally, to further evaluate the flexibility of our solver, and further improve segmentation results, we introduced a shape prior term in the level-set evolution process. Indeed, from the manual contours drawn by the experts, it can be observed that locally the extracted surfaces are very close to an ellipsoid shape. In order to be visually closer to the manual contours, we thus constrained the level-set to evolve according to (4.20). The value of the parameter  $\beta$  that weights the influence of the shape prior term was manually fixed to 1.2. A summary of the obtained results is reported in Table 4.7. As in the previous case, the regression analysis, depicted in Figure 4.21, shows a strong correlation between LV volumetric parameters values estimated using the P-SF algorithm and the reference ones. Moreover, the Bland-Altman analysis, illustrated in Figure 4.22, shows small biases and tight limits of agreement especially for SV and EF. Using the prior knowledge information only biases for ESV and SV are statistically significant (paired t-test  $p < 0.05$ ). Therefore, by comparing the given values with the ones provided in Table 4.6 we can observe that the segmentation with shape prior gives comparable results in terms of clinical indices measurements than the one without any prior knowledge. Interestingly, when using the shape prior, the segmentation of each volume requires only 100 iterations to converge (in comparison to the 250 iterations for the segmentation performed without any shape prior) and thus took about 0.67 seconds on average. This result can be explained by the fact that involving both local contrast and shape prior terms overconstrained the level-set and thus make it converge more rapidly to a local minimum. Examples of the LV segmentation using our P-SF method with the shape prior term are shown in Figure 4.23 for both ED and ES volumes. Moreover, Figure 4.24 shows the segmentation results obtained with and without the prior constraint for volume slices in x, y, and z directions.

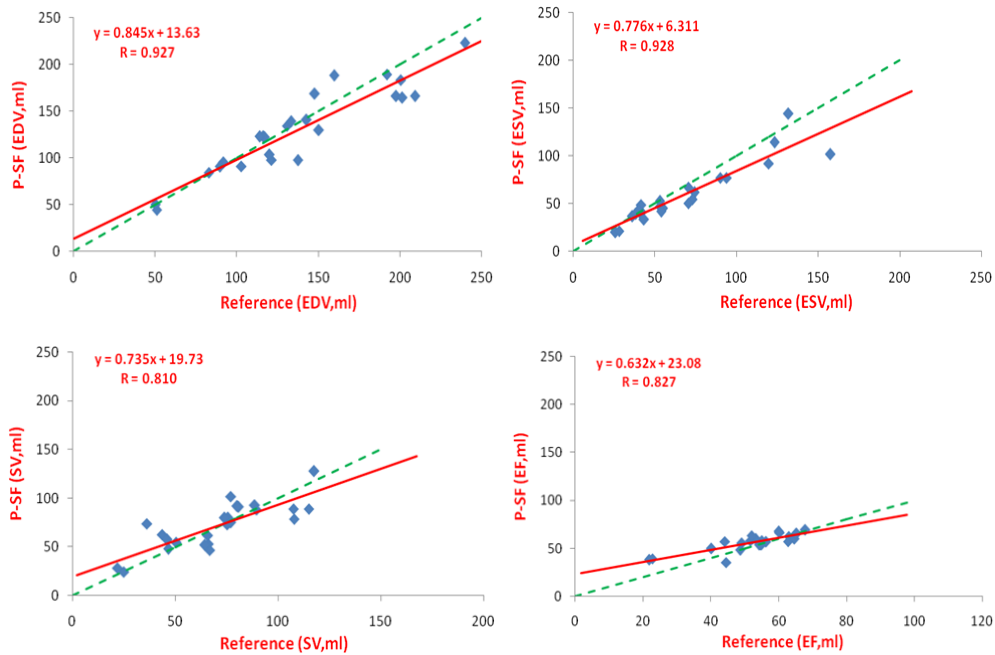


Figure 4.21: Linear Regression Analysis for EDV, ESV, SV and EF obtained adding the shape prior constraint.

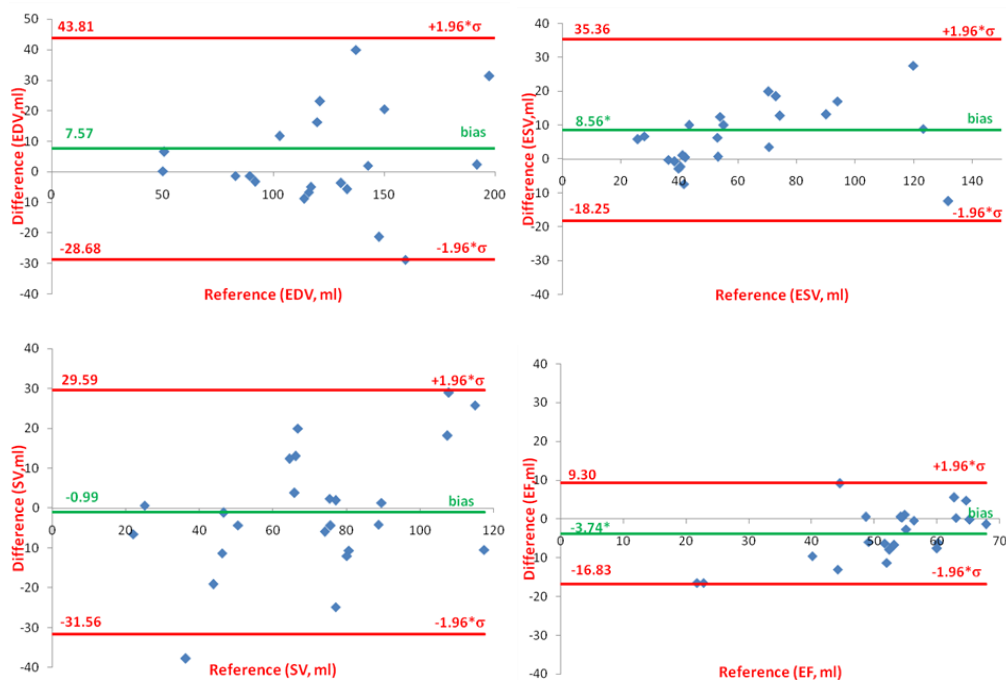


Figure 4.22: Bland-Altman Analysis plots for EDV, ESV, SV, and EF, comparing the measured errors against the reference values in case of using the shape prior constraint [green line: bias (\*, $p < 0.05$ ), red lines: limits of agreement ( $\mu \pm 1.96\sigma$ )].

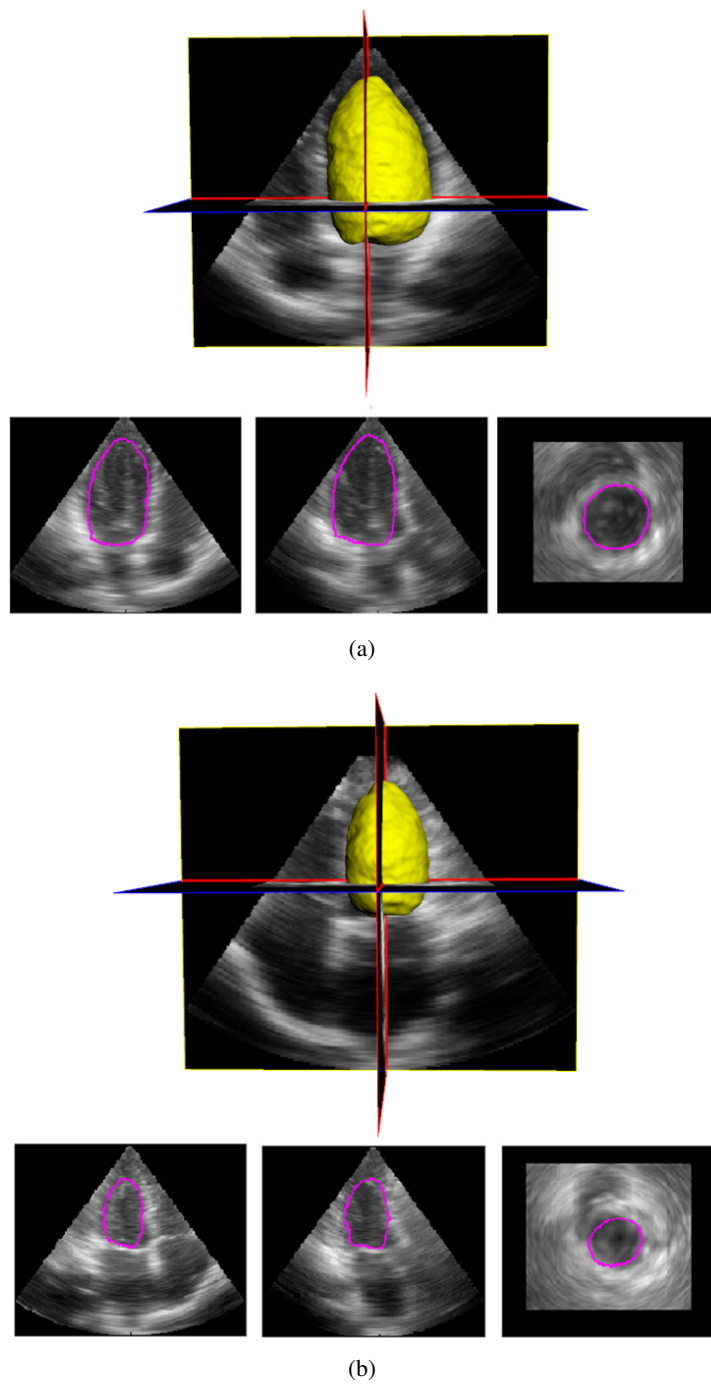


Figure 4.23: Segmentation of a RT3DE exam using P-SF with the shape prior constraint. (a) EDV (Ref.=113.8ml, P-SF=122.7ml), (b) ESV (Ref.=39.8ml, P-SF=42.6ml).



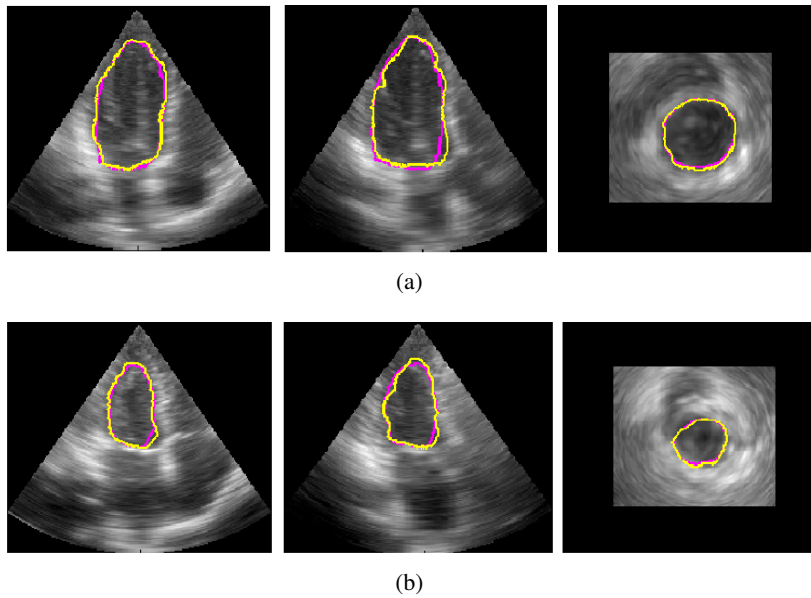


Figure 4.24: Segmentation results obtained with (magenta) and without (yellow) the prior constraint for volume slices in x, y, and z directions, for both EDV (a) and ESV (b).

From these results, it is possible to appreciate the smoother extracted surfaces that locally look closer to an ellipsoidal shape. Therefore, the practical utility of the prior knowledge introduction is twofold: reduce the analysis time, and increase the smoothness of the segmentation results since it allows to avoid local irregularities, linked to the attraction of the active contours to small scale noisy structures.

## 4.7 Conclusions

Cardiovascular disease is the main cause of morbidity and mortality for both genders in industrialized nations. Among the several tools for cardiac morphology and function assessment, Real-Time 3D Echocardiography is the preferred imaging modality. Fast and automatic segmentation of LV myocardium is essential to enable an efficient quantitative analysis of the RT3DE data. In this chapter a novel GPU based level-set segmentation framework aimed at achieving this goal has been introduced and described. It is, to the best of our knowledge, the first rigorous parallel implementation of the level-set sparse field algorithm. Our GPU level-set solver presents an interesting compromise between efficiency and flexibility in the interface evolution that makes it able to handle different segmentation tasks. Thanks to its flexibility, by using a localized region-based segmentation approach we made the solver specifically designed for the segmentation of strongly inhomogeneous images as this is the case for RT3DE. Moreover, combining our GPU level-set solver with an efficient automated initialization procedure we obtained a com-

plete efficient and automated segmentation framework suitable to be used as a support tool for cardiac morphology and function assessment in clinical environment. Also, we first introduce the shape prior knowledge in the context of GPU based level-set segmentation, to further improve segmentation accuracy in the application of our interest.

We first evaluated performance of our method at segmenting real and simulated 3D images of different nature, *i.e.* its performance at working as general purpose GPU level-set solver. A comparison with its sequential counterpart demonstrates the effectiveness of our solver at achieving high computational performance without losing segmentation accuracy. From a time performance point of view our solver is comparable with TC-SF algorithm which is, to the best of our knowledge, the actual fastest GPU level-set solver. High segmentation accuracy was obtained using both time constant and time varying speed functions demonstrating the flexibility of our solver in handling different segmentation tasks. As a practical application, particular attention has been paid to the detection of vessels in 3D rotational angiography images. Case studies performed on real data from this medical application demonstrate the applicability of our method in clinical environment.

Then, we evaluated performance of our segmentation framework in the context of LV myocardium segmentation in 3D echocardiography images. An extensive clinical validation which compares the volumetric indices estimated from manual segmentation by expert physicians and from our method results confirms the effectiveness of our framework at achieving accurate 3D LV segmentation near real-time, meaning that our GPU segmentation framework could be used for a quantitative analysis of 3D cardiac morphology and function in clinical applications.

---

# Conclusions

---

This thesis described the research activity carried out by the author during her Ph.D. program at Advanced Research Center on Electronic Systems for Information and Communication Technologies E. De Castro (ARCES) of Bologna University, Italy, from January 2010 to December 2012. Part of the work was carried out during an internship at CREATIS, Centre de Recherche et d'Application en Traitement de l'Image et du Signal, in Lyon, France.

This research, focused on efficient ultrasound signal processing techniques for medical diagnostics, has been addressed toward two important clinical applications: the improvement of the standard biopsy protocol for prostate cancer detection, and the improvement of the quantitative analysis of cardiac morphology and function usually performed to diagnose and treat cardiovascular diseases and heart defects. To achieve these targets two major topics have been investigated in this research: the ultrasound tissue characterization, in order to develop a computer aided detection tool for prostate cancer detection, and the ultrasound segmentation, in order to provide cardiologists with software tools for easing reading and analysis of echocardiographic sequences.

The designed CAD system for PCa detection was aimed at improving the standard biopsy protocol efficiency by providing physicians a second opinion, allowing them to perform a lesion-directed biopsy. To achieve this goal the CAD has to be extremely accurate and fast in order to preserve the real-time nature of the echographic examination and to be used as biopsy guide. A reliable medical ground truth, an appropriate learning scheme, and an efficient implementation of the CAD processing flow are essential to satisfy these requirements. All these aspect of the CAD design have been object of the conducted research.

In particular, in the first part of this thesis we proposed a real-time Computer Aided Biopsy tool based on a binary classification system that exploits information extracted from the ultrasound echo signal to discriminate between healthy prostate tissue and tissue diagnosed with adenocarcinoma. The classification result is provided as a false color overlay image representing a binary cancer risk map that, superimposed to the standard B-Mode image, suggests physicians were to sample thus reducing the number of biopsy cores requested for an accurate diagnosis. The system was trained on a large and consis-

tent ground truth database, collected in collaboration with the department of Urology of the S.Orsola Hospital in Bologna. An expressly designed supervised learning procedure based on CV was used to train the classifier on this ground truth. In addition, the CAD processing flow was implemented exploiting CUDA parallel processing acceleration to speed up the image analysis process. Experimental results confirmed the effectiveness of the proposed CAD tool at improving the standard prostate biopsy protocol and its applicability in clinical environment. In fact, the tool is able to reduce the number of biopsy cores from 8-12 to 7 preserving the same diagnostic power of the standard biopsy protocol. Furthermore, the tool is characterized by a minimum frame rate of 38 fps suitable for clinical applications.

However, the intrinsic ground truth uncertainty, due to the unknown distribution of the tumor in each core, prevents the use of the whole amount of collected data reducing the CAD training performance and increasing the data collection effort necessary to further improve the CAD system reliability. This has motivated the research described in the second part of this thesis, focused on the development of a new learning procedure aimed at enhancing classification performance reducing the data collection effort and avoiding collected data wasting. To this end a novel semi-supervised learning algorithm, named SelfCo3, was designed. By using unanimous agreement among three different classifiers, and class balance preservation, the algorithm enlarges the original labeled training set and selects the final classification model. Experiments on both benchmark datasets and on ground truth images from biopsy findings demonstrated both the effectiveness of SelfCo3 as general purpose SSL procedure, and its ability to enhance computer aided detection of PCa outperforming the previously developed application-driven classification framework without requiring any additional labeled data and making full use of the whole ground truth dataset. The new SSL algorithm was also specified to deal with the PCa detection in ultrasound images exploiting tumor core volume knowledge to further improve prediction performance. Exploiting this new version of the algorithm, named SelfCo3Core, an innovative mixed supervised/semi-supervised learning procedure was designed, and a new detection tool based on this procedure was developed. This new CAD scheme is able to improve the PPV of the standard biopsy protocol of 55% consequently reducing the number of biospy cores, from 8-12 to 6 without losing diagnostic power.

Future research in this context will be targeted to test the classification system on a larger medical database in order to evaluate its performance with higher accuracy, and to the installation of the developed CAD tool on a standard machine in order to carry on a reliable clinical evaluation of the method.

Concerning the application of ultrasound segmentation in the context of heart functioning assessment, the research described in this thesis focused on the development of an automated segmentation framework for delineating the ventricle endocardial borders in real-time 3D echocardiography images. To address this task with high effectiveness and efficiency we exploited the accuracy of the popular level-set segmentation methods

and the GPU computational power. The result was the development of a new GPU-based level-set solver for 3D images which is, to the best of our knowledge, the first rigorous implementation of the popular sparse field level-set algorithm. The high efficiency and flexibility in the interface evolution of the proposed solver make it able to deal with different segmentation tasks.

Thanks to its flexibility, by using a localized region-based segmentation approach and combining the solver with an efficient automated initialization procedure, we developed an efficient and automated segmentation framework suitable to be used as a support tool for cardiac morphology and function assessment in clinical environment. In addition, a shape prior knowledge, that constrains the segmentation result to anatomically meaningful shapes, was exploited to further improve segmentation accuracy in the application of interest. Experimental results on both simulated and real datasets demonstrated the effectiveness of the proposed method as general purpose GPU level-set solver providing the same accuracy as its sequential counterpart, and comparable efficiency and higher flexibility than existing GPU level-set methods. An extensive clinical validation demonstrated the effectiveness of the proposed segmentation framework in the context of left-ventricle myocardium segmentation in 3D echocardiography images. This clinical validation, which compares the volumetric indices estimated from manual segmentation by expert physicians and from the designed method results, confirms the effectiveness of our framework at achieving accurate 3D LV segmentation near real-time, meaning that it could be used for a quantitative analysis of 3D cardiac morphology and function in clinical applications.

Thanks to the the low computational burden of the proposed method, future research on this activity can include its use to segment temporal sequences to track the myocardium during a whole cardiac cycle. Also the combination with motion estimation techniques can lead to the development of a new, complete motion and tracking imaging method for echocardiography, as well as the corresponding diagnostic tool for supporting physicians in the detection of cardiac abnormalities.



---

# Publications

---

1. N. Testoni, S. Maggio, F. Galluzzo, L. De Marchi, N. Speciale. "rtCAB: A tool for reducing unnecessary prostate biopsy cores," Ultrasonics Symposium (IUS), 2010 IEEE , pp.49-52, 11-14 Oct. 2010
2. N. Testoni, N. Speciale, A. Bertaccini, D. Marchiori, M. Fiorentino, F. Manferrari, R. Schiavina, R. Cividini, F. Galluzzo, S. Maggio, E. Biagi, L. Masotti, G. Masetti, G. Martorana, "A retrospective study to reduce prostate biopsy cores by a real-time interactive tool," Archivio italiano di urologia, andrologia: organo ufficiale [di] Società italiana di ecografia urologica e nefrologica/Associazione ricerche in urologia, vol. 82, no.4, pp. 238-241.
3. N. Testoni, N. Speciale, A. Bertaccini, D. Marchiori, M. Fiorentino, F. Manferrari, R. Schiavina, R. Cividini, M. Giampoli, S. Maggio, F. Galluzzo, G. Masetti, G. Martorana, "Reducing the number of prostate biopsy cores using a real-time tool," Anticancer research, vol. 31, no.5, pp.1943-1944, 2011.
4. F. Galluzzo, N. Testoni, L. De Marchi, N. Speciale, G. Masetti, "Improving Prostate Biopsy Protocol with a Computer Aided Detection Tool based on Semi-Supervised Learning," Anant Madabhushi, Jason Dowling, Henkjan Huisman, and Dean Bar- rat editors, Prostate Cancer Imaging. Image Analysis and Image-Guided Interventions, Lecture Notes in Computer Science, vol. 6963, pp. 109-120, 2011, Springer Berlin Heidelberg, 2011.
5. F. Galluzzo, N. Speciale, O. Bernard, "A rigorous and efficient GPU implementation of level-set sparse field algorithm," Image Processing (ICIP), 2012, 19th IEEE International Conference on, pp. 1705-1708, 30 Sept.- 3 Oct. 2012.
6. F. Galluzzo, D. Barbosa, H. Houle, N. Speciale, D. Friboulet, J. D'hooge and O. Bernard, " A GPU Level-Set Segmentation Framework for 3D Echocardiography," Ultrasonics Symposium (IUS), 07-10 Oct. 2012.
7. F. Galluzzo, D. Barbosa, N. Speciale, H. Houle, J. D'hooge, G. Courbebaisse, G. Masetti, and O. Bernard, "A rigorous and efficient GPU implementation of level-

set sparse field algorithm and its application in medical diagnostics, " submitted for publication in IEEE Transactions on Visualization and Computer Graphics.



---

# Bibliography

---

- [1] M. Garagnani and C. Lamberti. *Ecografia. Principi fisici e sviluppi tecnologici*. Athena Audiovisuals, 2003. [cited at p. 6]
- [2] A sound vision corporation Cardiovascular Sales. Mindray ultrasound probes. <http://www.cvsales.com/cvs/PartsProbes/MindrayProbes.aspx>, 2012. [Online;]. [cited at p. 10, 163]
- [3] G. Schmitz, H. Ermert, and T. Senge. Tissue-characterization of the prostate using radio frequency ultrasonic signals. *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on*, 46(1):126–138, jan. 1999. [cited at p. 10, 44, 45]
- [4] U. Scheipers, H. Ermert, HJ. Sommerfeld, M. Garcia-Schurmann, T. Senge, and S. Philippou. Ultrasonic multifeature tissue characterization for prostate diagnostics. *Ultrasound in Medicine & Biology*, 29(8):1137–1149, 2003. [cited at p. 10, 44, 45]
- [5] A. Fenster and D.B. Downey. 3-d ultrasound imaging: a review. *Engineering in Medicine and Biology Magazine, IEEE*, 15(6):41–51, nov/dec 1996. [cited at p. 11, 12]
- [6] A. Fenster, D.B. Downey, and Cardinal H.N. Three-dimensional ultrasound imaging. *Physics in Medicine and Biology*, 46(5):R67–R99, 2001. [cited at p. 11, 12]
- [7] O. von Ramm and S.W. Smith. Real time volumetric ultrasound imaging system. *Journal of Digital Imaging*, 3(4):261–266, 1990. [cited at p. 11]
- [8] T.L. Szabo. *Diagnostic ultrasound imaging: inside out*. Elsevier Science, Boston, 2004. [cited at p. 12, 167]
- [9] K. Doi. Computer-aided diagnosis in medical imaging: Historical review, current status and future potential. *Computerized Medical Imaging and Graphics*, 31(4-5):198–211, June 2007. [cited at p. 15]

- [10] iCAD. iCAD products. <http://www.icadmed.com/products/products.htm?expandable=1>, 2012. [Online;]. [cited at p. 15]
- [11] Hologic and the women's Health Company. <http://www.hologic.com/it/>, 2012. [Online;]. [cited at p. 15]
- [12] L.D. Jacobs, I.S. Salgo, S. Goonewardena, P. Weinert, L. and Coon, D. Bardo, O. Gerard, P. Allain, J.L. Zamorano, V. de Isla, L.P. and Mor-Avi, and R. M. Lang. Rapid online quantification of left ventricular volume from real-time three-dimensional echocardiographic data. *European Heart Journal*, 27(4):460–468, 2006. [cited at p. 15, 17]
- [13] J. Hansegård, S. Urheim, K. Lunde, S. Malm, and S. Rabben. Semi-automated quantification of left ventricular volumes and ejection fraction by real-time three-dimensional echocardiography. *Cardiovasc Ultrasound*, 7(18):379–84, 2009. [cited at p. 15, 17]
- [14] J.A. Noble. Ultrasound image segmentation and tissue characterization. *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, 224(2):307–316, 2010. [cited at p. 16]
- [15] C. Jenkins, K. Bricknell, L. Hanekom, and T. H. Marwick. Reproducibility and accuracy of echocardiographic measurements of left ventricular parameters using real-time three-dimensional echocardiography. *Journal of the American College of Cardiology*, 44(4):878–886, 2004. [cited at p. 17, 98]
- [16] O.I. Soliman, S.V. Kirschbaum, B.M. van Dalen, H.B. van der Zwaan, B.M. Delavary, W.B. Vletter, R.J.M. van Geuns, F.J.T. Cate, and M.L. Geleijnse. Accuracy and reproducibility of quantitation of left ventricular function by real-time three-dimensional echocardiography versus cardiac magnetic resonance. *The American Journal of Cardiology*, 102(6):778 – 783, 2008. [cited at p. 17, 98]
- [17] Victor Mor-Avi, Carly Jenkins, Harald P. Käijhl, Hans-Joachim Nesser, Thomas Marwick, Andreas Franke, Christian Ebner, Benjamin H. Freed, Regina Steringer-Mascherbauer, Heidi Pollard, Lynn Weinert, Johannes Niel, Lissa Sugeng, and Roberto M. Lang. Real-time 3-dimensional echocardiographic quantification of left ventricular volumes multicenter study for validation with magnetic resonance imaging and investigation of sources of error. *JACC: Cardiovascular Imaging*, 1(4):413–423, 2008. [cited at p. 17, 98]
- [18] E.G. Caiani, C. Corsi, J. Zamorano, L. Sugeng, P. MacEneaney, L. Weinert, R. Battani, J.L. Gutierrez, R. Koch, L. Perez de Isla, V. Mor-Avi, and Lang R.M. Improved semiautomated quantification of left ventricular volumes and ejection fraction using 3-dimensional echocardiography with a full matrix-array

- transducer: Comparison with magnetic resonance imaging. *Journal of the American Society of Echocardiography*, 18(8):779 – 788, 2005. [cited at p. 17, 98, 105]
- [19] J.D. Owens, M. Houston, D. Luebke, S. Green, J.E. Stone, and J.C. Phillips. Gpu computing. *Proceedings of the IEEE*, 96(5):879 –899, may 2008. [cited at p. 17, 18]
- [20] G. Pratz and L. Xing. Gpu computing in medical physics: A review. *Medical physics*, 5(4), 2011. [cited at p. 18]
- [21] L. Shi, L. Wen, H. Zhang, X. Yongming, and Defeng W. A survey of gpu-based medical image computing techniques. *Quantitative imaging in medicine and surgery*, 2(3):188 –206, september 2012. [cited at p. 18]
- [22] T.S. Crow. Evolution of the graphical processing unit. [http://www.cse.iitb.ac.in/graphics/~anand/website/include/papers/gpu/gpu\\_evolution.pdf](http://www.cse.iitb.ac.in/graphics/~anand/website/include/papers/gpu/gpu_evolution.pdf), 2004. [Online;]. [cited at p. 18]
- [23] W.R. Mark, R. S. Glanville, K. Akeley, and M. J. Kilgard. Cg: a system for programming graphics hardware in a c-like language. *ACM Trans. Graph.*, 22(3):896 – 907, jul 2003. [cited at p. 18]
- [24] E. Lindholm, J. Nickolls, S. Oberman, and J. Montrym. Nvidia tesla: A unified graphics and computing architecture. *IEEE Micro*, 28(2):39 – 55, mar 2008. [cited at p. 18, 20]
- [25] J. Nickolls and W.J. Dally. The gpu computing era. *Micro, IEEE*, 30(2):56 –69, march-april 2010. [cited at p. 18, 19]
- [26] NVIDIA Corporation. Fermi: Nvidia’s next generation cuda compute architecture. [http://www.nvidia.com/content/PDF/fermi\\_white\\_papers/NVIDIA\\_Fermi\\_Compute\\_Architecture\\_Whitepaper.pdf](http://www.nvidia.com/content/PDF/fermi_white_papers/NVIDIA_Fermi_Compute_Architecture_Whitepaper.pdf), 2009. [Online;]. [cited at p. 19, 20, 163]
- [27] NVIDIA Corporation. Technical brief. nvidia geforce gtx 200 gpu architectural overview. [Online;]. [cited at p. 19]
- [28] NVIDIA Corporation. Nvidia cuda compute unified device architecture programming guide, 2011. [Online;]. [cited at p. 21, 50, 53, 87, 163]
- [29] Wikipedia. Prostate — Wikipedia, The Free Encyclopedia. <http://en.wikipedia.org/Prostate>, 2012. [Online;]. [cited at p. 24]
- [30] Prostate cancer treatment (pdq). <http://www.uchospitals.edu/online-library/content=CDR62965>, 2012. [Online;]. [cited at p. 25, 163]
- [31] S. Curran, O. Akin, A.M. Agildere, H. Zhang, J.and Hricak, and J. Rademaker. Endorectal mri of prostatic and periprostatic cystic lesions and their mimics. *American Journal of Roentgenology.*, 188(5):1373–1379, 2007. [cited at p. 25, 163]

- [32] C. Abate-Shen and M.M. Shen. Molecular genetics of prostate cancer. *Genes & Development*, 14(19):2410–2434, 2000. [cited at p. 25, 26]
- [33] V. Kundra, P.M. Silverman, S.F. Matin, and H. Choi. Imaging in Oncology from The University of Texas MD Anderson Cancer Center: Diagnosis, Staging, and Surveillance of Prostate Cancer. *American Journal of Roentgenology*, 189(4):830–844, 2007. [cited at p. 26]
- [34] Wikipedia. Prostate Cancer — Wikipedia, The Free Encyclopedia. [http://en.wikipedia.org/wiki/Prostate\\_cancer](http://en.wikipedia.org/wiki/Prostate_cancer), 2012. [Online;]. [cited at p. 26]
- [35] J. Raja, N. Ramachandran, Munneke G., and Patel U. Current status of transrectal ultrasound-guided prostate biopsy in the diagnosis of prostate cancer. *clinical Radiology*, 61(4):142–153, 2006. [cited at p. 28, 29]
- [36] J. De La Rosette, R. Giensen, A. Huynen, R. Aornink, M. P. Van Iersel, F. Debruyne, and H. Wijkstra. Automated Analysis and interpretation of transrectal ultrasonography images in patients with prostatitis. *European Urology*, 27(1):47–53, 1995. [cited at p. 28]
- [37] P.A. Humphrey. Gleason grading and prognostic factors in carcinoma of the prostate. *Modern Pathology*, 17(3):292–306, 2004. [cited at p. 28]
- [38] M. Essink-Bot, H. de Koning, W. Nijis, W. Kirkels, P.V. der Maas, and F. Schroder. Short-term effects of population based screening for prostate cancer on health-related quality of life. *Journal of the National Cancer Institute*, 90:925–931, 1998. [cited at p. 28]
- [39] A.R. Webb. *statistical Pattern Recognition*. Wiley, 2002. [cited at p. 31, 39]
- [40] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press, 2006. [cited at p. 31, 39]
- [41] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K.R. Mullers. Fisher discriminant analysis with kernels. In *Neural Networks for Signal Processing IX, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop.*, pages 41–48, aug 1999. [cited at p. 31, 39]
- [42] N. Cristianini and J. Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000. [cited at p. 32]
- [43] H. Liu and L. Yu. Toward integrating feature selection algorithms for classification and clustering. *Knowledge and Data Engineering, IEEE Transactions on*, 17(4):491 – 502, april 2005. [cited at p. 32]

- [44] L. Masotti, E. Biagi, S. Granchi, L. Breschi, E. Magrini, and F. Di Lorenzo. Clinical test of rules (rules: radiofrequency ultrasonic local estimators). In *Ultrasonics Symposium, 2004 IEEE*, volume 3, pages 2173 – 2176, aug. 2004. [cited at p. 34]
- [45] G. Georgiou and F.S. Cohen. Tissue characterization using the continuous wavelet transform. part i. decomposition method. *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on*, 48(2):355 –363, march 2001. [cited at p. 34]
- [46] M. Shankar. A general statistical model for ultrasonic backscattering from tissues. *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on*, 47(3), 2000. [cited at p. 34]
- [47] J.W. Goodman. *Statistical Optics*. Wiley Classic Library Edition, 2001. [cited at p. 34]
- [48] M.K. Varanasi and B. Aazhang. Parametric generalized gaussian density estimation. *The Journal of the Acoustical Society of America*, 86:1404, 1989. [cited at p. 34]
- [49] R.M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *Systems, Man and Cybernetics, IEEE Transactions on*, SMC-3(6):610 –621, nov. 1973. [cited at p. 34]
- [50] M. Unser. Sum and difference histograms for texture classification. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8(1):118 –125, jan. 1986. [cited at p. 34, 35, 36, 55]
- [51] W.L. Lee, Y.C. Chen, and K.S. Hsieh. Ultrasonic liver tissues classification by fractal feature vector based on m-band wavelet transform. *Medical Imaging, IEEE Transactions on*, 22(3):382 –392, march 2003. [cited at p. 34]
- [52] P.M. Shankar. Ultrasonic tissue characterization using a generalized nakagami model. *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on*, 48(6):1716 –1720, nov. 2001. [cited at p. 34]
- [53] J. Cheng and N.C. Beaulieu. Maximum-likelihood based estimation of the nakagami m parameter. *Communications Letters, IEEE*, 5(3):101 –103, march 2001. [cited at p. 35]
- [54] G. A. Baker and P. Graves-Morris. *Padé approximants second edition*, in *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 1996. [cited at p. 35]
- [55] N. Testoni, S. Maggio, F. Galluzzo, L. De Marchi, and N. Speciale. rtab: A tool for reducing unnecessary prostate biopsy cores. In *Ultrasonics Symposium (IUS), 2010 IEEE*, pages 49–52, Oct. [cited at p. 35]

- [56] M. Moradi, P. Mousavi, and P. Abolmaesumi. Computer-aided diagnosis of prostate cancer with emphasis on ultrasound-based approaches: A review. *Ultrasound in Medicine & Biology*, 33(7):1010 – 1028, 2007. [cited at p. 36, 38, 43]
- [57] S. Maggio, A. Palladini, L. De Marchi, M. Alessandrini, N. Speciale, and G. Masetti. Predictive deconvolution and hybrid feature selection for computer-aided detection of prostate cancer. *Medical Imaging, IEEE Transactions on*, 29(2):455 –464, feb. 2010. [cited at p. 36]
- [58] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(8):1226 –1238, aug. 2005. [cited at p. 36]
- [59] M. M. Aizerman, E. Braverman, and L. Rozonoer. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25:821–837, 1964. [cited at p. 38]
- [60] B.E. Boser, I.M. Guyon, and V.N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, COLT '92, pages 144–152, New York, NY, USA, 1992. ACM. [cited at p. 39]
- [61] G. Baudat and F. Anouar. Generalized discriminant analysis using a kernel approach. *Neural Computation Journal.*, 12:2385 – 2404, 2000. [cited at p. 40]
- [62] F. Vojtech and H. Vaclav. Statistical pattern recognition toolbox for matlab,user's guide. <http://cmp.felk.cvut.cz/cmp/software/stprtool/stprtool.pdf>, 2004. [Online;]. [cited at p. 41]
- [63] F.L. Lizzi, E.J. Feleppa, s.K. Alam, and C.X. Deng. Ultrasonic spectrum analysis for tissue evaluation. *Pattern Recognition Letters*, 24(4 - 5):637 – 658, 2003. [cited at p. 43, 44]
- [64] O. Basset, Z. Sun, J.L. Mestas, and G. Gimnez. Texture analysis of ultrasonic images of the prostate by means of co-occurrence matrices. *Ultrasonic imaging*, 15(3):218 – 237, 1993. [cited at p. 44, 45]
- [65] A.L. Huynen, R.J.B. Giesen, J.J.M.C.H. de la Rosette, R.G. Aarnink, F.M.J. Debruyne, and H. Wijkstra. Analysis of ultrasonographic prostate images for the detection of prostatic carcinoma: The automated urologic diagnostic expert system. *Ultrasound in Medicine & Biology*, 20(1):1 – 10, 1994. [cited at p. 44, 45]
- [66] A.G. Houston, S.B. Premkumar, D.E. Pitts, and R.J. Babaian. Prostate ultrasound image analysis: localization of cancer lesions to assist biopsy. In *Computer-Based*

- Medical Systems, 1995., Proceedings of the Eighth IEEE Symposium on*, pages 94–101, jun 1995. [cited at p. 44, 45]
- [67] R. Llobet, J.C. Pérez-Cortés, A.H. Toselli, and A. Juan. Computer-aided detection of prostate cancer. *International Journal of Medical Informatics*, 76(7):547–556, 2007. [cited at p. 44, 45]
- [68] S.S. Mohamed and M.M.A. Salama. Prostate cancer spectral multifeature analysis using trus images. *Medical Imaging, IEEE Transactions on*, 27(4):548–556, april 2008. [cited at p. 44, 45]
- [69] E.J. Feleppa, J.A. Ketterling, C. R. Porter, J. Gillespie, C.S. Wu, S. Urban, A. Kalisz, R. D. Ennis, and P. B. Schiff. Ultrasonic tissue-type imaging (tti) for planning treatment of prostate cancer. In *SPIE Proceedings Vol.5373, Medical Imaging 2004: Ultrasonic Imaging and Signal Processing*, William F. Walker; Stanislav Y. Emelianov, Editors, pages 223–230, 2004. [cited at p. 44, 45]
- [70] S.M. Han, H.J. Lee, and J.Y. Choi. Prostate cancer detection using texture and clinical features in ultrasound image. In *Information Acquisition, 2007. ICIA '07. International Conference on*, pages 547–552, July. [cited at p. 44, 45]
- [71] S.S. Mohamed and M.M.A. Salama. Computer-aided diagnosis for prostate cancer using support vector machine. In *SPIE Proceedings Vol. 5744, Medical Imaging 2005: Visualization, Image-Guided Procedures, and Display*, Robert L. Galloway, Jr.; Kevin R. Cleary, Editors, pages 898–906, 2005. [cited at p. 44, 45]
- [72] M. Scabia, E. Biagi, and L. Masotti. Hardware and software platform for real-time processing and visualization of echographic radiofrequency signals. *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on*, 49(10):1444–1452, oct. 2002. [cited at p. 47]
- [73] C. Fritsch, A. Ibanez, and M. Parrilla. A digital envelope detection filter for real-time operation. *Instrumentation and Measurement, IEEE Transactions on*, 48(6):1287–1293, dec 1999. [cited at p. 50, 52]
- [74] J.H. Chang, J.T. Yen, and K.K. Shung. High-speed digital scan converter for high-frequency ultrasound sector scanners. *Ultrasonics*, 48(5):444–452, 2008. [cited at p. 50]
- [75] N. Testoni. Real-time approach to amplitude phase estimation. internal technical report. Technical report, DEIS- Department of Electronics, Computer Sciences and Systems, University of Bologna, Italy, 2010. [cited at p. 52]
- [76] M. Harris, S. Sengupta, and J.D. Owens. *Parallel Prefix sum (scan) with CUDA*. GPU Gems 3, Nguyen H, Addison Wesley, 2007. [cited at p. 55, 123]

- [77] O. Chapelle, B. Scholkopf, and A. Zien. *Semi-Supervised Learning*. MIT Press, Cambridge, Massachusetts, 2006. [cited at p. 65, 66]
- [78] X. Zhu, A.B. Goldberg, R. Brachman, and T. Dietterich. *Introduction to Semi-Supervised Learning*. Morgan and Claypool Publishers, 2009. [cited at p. 65, 66, 67]
- [79] X. Zhu. Semi-supervised learning literature survey. technical report. Technical report, 530, Department of Computer Sciences, University of Wisconsin, Madison, 2005. [cited at p. 66]
- [80] D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics, ACL '95*, pages 189–196, Stroudsburg, PA, USA, 1995. Association for Computational Linguistics. [cited at p. 67]
- [81] E. Riloff, J. Wiebe, and T. Wilson. Learning subjective nouns using extraction pattern bootstrapping. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4, CONLL '03*, pages 25–32, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. [cited at p. 67]
- [82] B. Maeireizo, D. Litman, and R. Hwa. Co-training for predicting emotions with spoken dialogue data. Co-training for predicting emotions with spoken dialogue data., 2004. [cited at p. 67]
- [83] C. Rosenberg, M. Hebert, and H. Schneiderman. Semi-supervised self-training of object detection models. In *Proceedings of the Seventh IEEE Workshops on Application of Computer Vision (WACV/MOTION'05) - Volume 1*, pages 29–36, Washington, DC, USA, 2005. IEEE Computer Society. [cited at p. 67]
- [84] S. Goldman and Y. Zhou. Enhancing supervised learning with unlabeled data. In *Proceedings of the Seventeenth International Conference on Machine Learning, ICML '00*, pages 327 – 334, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. [cited at p. 68, 69]
- [85] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory, COLT' 98*, pages 92–100, New York, NY, USA, 1998. ACM. [cited at p. 68]
- [86] T. Mitchell. The role of unlabeled data in supervised learning. In *Proceedings of the Sixth International Colloquium on Cognitive Science. San Sebastian, Spain.*, 1999. [cited at p. 68]
- [87] K. Nigam and R. Ghani. Analyzing the effectiveness and applicability of co-training. In *Proceedings of the ninth international conference on Information and knowledge management, CIKM '00*, pages 86–93, New York, NY, USA, 2000. ACM. [cited at p. 68]



- [88] Z.H. Zhou and M. Li. Tri-training: exploiting unlabeled data using three classifiers. *Knowledge and Data Engineering, IEEE Transactions on*, 17(11):1529 – 1541, nov. 2005. [cited at p. 68, 69, 72]
- [89] M. Collins and Y. Singer. Unsupervised models for named entity classification. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999. [cited at p. 68]
- [90] Jones. Learning to extract entities from labeled and unlabeled text (technical report cmu-lti-05-191). Technical report, Carnegie Mellon University. Doctoral Dissertation., 2005. [cited at p. 68]
- [91] Z.H. Zhou, Z. De-Chuan, and Y Qiang. Semi-supervised learning with very few labeled training examples. In *Proceedings of the 22nd National Conference on Artificial intelligence - Volume 1, AAAI'07*, pages 675–680. AAAI Press, 2007. [cited at p. 69]
- [92] Y. Zhou and S. Goldman. Democratic co-learning. In *Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference on*, pages 594 – 602, nov. 2004. [cited at p. 69]
- [93] R. Johnson and T. Zhang. Two-view feature generation model for semisupervised learning. The 24th International Conference on Machine Learning., 2007. [cited at p. 69]
- [94] A. Frank and A. Asuncion. UCI machine learning repository. <http://archive.ics.uci.edu/ml>, 2010. [online]. [cited at p. 71, 72, 167]
- [95] I. Guyon. A scaling law for the validation-set training-set size ratio. In *AT & T Bell Laboratories*, 1997. [cited at p. 80]
- [96] J. Malmivuo and Plonsey R. Bioelectromagnetism. Principles and applications of bioelectric and biomagnetic fields. Oxford University Press. <http://www.bem.fi/book/06/06.htm>, 1995. [Online;]. [cited at p. 93, 165]
- [97] Wikipedia. Heart — Wikipedia, The Free Encyclopedia. <http://en.wikipedia.org/wiki/Heart>, 2012. [Online;]. [cited at p. 93]
- [98] Cardiovascular visuals. <http://www.mhhe.com/biosci/ap/dynamichuman2/content/cardio/visuals.mhtml>. [Online;]. [cited at p. 94, 165]
- [99] Wikipedia. Cardiovascular disease — Wikipedia, The Free Encyclopedia. [http://en.wikipedia.org/wiki/Cardiovascular\\_disease](http://en.wikipedia.org/wiki/Cardiovascular_disease), 2012. [Online;]. [cited at p. 94]

- [100] Department of Public Health. European cardiovascular disease statistics, 2012 edition. <http://www.escardio.org/about/what/advocacy/EuroHeart/Pages/2012-CVD-statistics.aspx>, 2012. [Online;]. [cited at p. 95]
- [101] E. D. Folland, A. F. Parisi, P. F. Moynihan, D. R. Jones, C. L. Feldman, and D. E. Tow. Assessment of left ventricular ejection fraction and volumes by real-time two-dimensional echocardiography: A comparison of cineangiographic and radionuclide techniques. *Circulations*, 66:760–766, 1979. [cited at p. 97]
- [102] R. Jurcut, H. Wildiers, J. Ganame, J. D’hooge, J. De Backer, H. Denys, R. Paridaens, F. Rademakers, and Voigt J. Strain rate imaging detects early cardiac effects of pegylated liposomal doxorubicin as adjuvant therapy in elderly patients with breast cancer. *Journal of the American Society of Echocardiography*, 21(12):1283 – 1289, 2008. [cited at p. 98]
- [103] T. Kukulski, F. Jamal, L. Herbots, J. D’hooge, B. Bijmens, L. Hatle, I. De Scheerder, and G.R. Sutherland. Identification of acutely ischemic myocardium using ultrasonic strain measurements. a clinical study in patients undergoing coronary angioplasty. *Journal of the American College of Cardiology*, 41(5):810–819, 2003. [cited at p. 98]
- [104] R. Lee, L. Hanekom, T.H. Marwick, R. Leano, and S. Wahi. Prediction of subclinical left ventricular dysfunction with strain rate imaging in patients with asymptomatic severe mitral regurgitation. *The American Journal of Cardiology*, 94(10):1333 – 1337, 2004. [cited at p. 98]
- [105] C. Jenkins, K. Bricknell, J. Chan, L. Hanekom, and T.H. Marwick. Comparison of two- and three-dimensional echocardiography with sequential magnetic resonance imaging for evaluating left ventricular volume and ejection fraction over time in patients with healed myocardial infarction. *The American Journal of Cardiology*, 99(3):300 – 306, 2007. [cited at p. 99]
- [106] G.D. Garson, B. Li, Acton. S.T., and Hossack J.A. Guiding automated left ventricular chamber segmentation in cardiac imaging using the concept of conserved myocardial volume. *Computerized Medical Imaging and Graphics*, 32(4):321 – 330, 2008. [cited at p. 99]
- [107] V. Walimbe, V. Zagrodsky, and R. Shekhar. Fully automatic segmentation of left ventricular myocardium in real-time three-dimensional echocardiography. In *SPIE Proceedings, Medical Imaging 2006, Image Processing*, volume 61444H, pages 1–10, 2006. [cited at p. 99]
- [108] G. Coppini, R. Poli, and G. Valli. Recovery of the 3-d shape of the left ventricle from echocardiographic images. *Medical Imaging, IEEE Transactions on*, 14(2):301 –317, jun 1995. [cited at p. 99, 104]

- [109] M.Z. Song, R.M. Haralick, F.H. Sheehan, and R.K. Johnson. Integrated surface model optimization for freehand three-dimensional echocardiography. *Medical Imaging, IEEE Transactions on*, 21(9):1077–1090, sept. 2002. [cited at p. 99]
- [110] I. Wolf, M. Hastenteufel, R. de Simone, M. Vetter, G. Glombitza, S. Mottl-Link, C.F. Vahl, and H.-P. Meinzer. Ropes: a semiautomated segmentation method for accelerated analysis of three-dimensional echocardiographic data. *Medical Imaging, IEEE Transactions on*, 21(9):1091–1104, sept. 2002. [cited at p. 99]
- [111] J.A. Noble and D. Boukerroui. Ultrasound image segmentation: a survey. *Medical Imaging, IEEE Transactions on*, 25(8):987–1010, aug. 2006. [cited at p. 99]
- [112] K.Y.E. Leung and J.G. Bosch. Automated border detection in three-dimensional echocardiography: principles and promises. *European Journal of Echocardiography*, 12(2):97–108, 2010. [cited at p. 99]
- [113] J.M. Bland and D.G. Altman. Statistical methods for assessing agreement between two methods of clinical measurement. *The Lancet*, 327(8476):307–310, 1986. [cited at p. 100]
- [114] Wikipedia. Pearson product-moment correlation coefficient — Wikipedia, The Free Encyclopedia. [http://en.wikipedia.org/wiki/Pearson\\_product-moment\\_correlation\\_coefficient](http://en.wikipedia.org/wiki/Pearson_product-moment_correlation_coefficient), 2012. [Online;]. [cited at p. 100]
- [115] Wikipedia. Student's t-test — Wikipedia, The Free Encyclopedia. [http://en.wikipedia.org/wiki/Student's\\_t-test](http://en.wikipedia.org/wiki/Student's_t-test), 2012. [Online;]. [cited at p. 102]
- [116] J. Hansengard, F. Orderud, and S. Rabben. Real-time active shape models for segmentation of 3d cardiac ultrasound. In Walter G. Kropatsch, Martin Kampel, and Allan Hanbury, editors, *Computer Analysis of Images and Patterns*, volume 4673 of *Lecture Notes in Computer Science*, pages 157–164. Springer Berlin Heidelberg, 2007. [cited at p. 102]
- [117] M. Ma, M. van Stralen, J.H.C. Reiber, J.G. Bosch, and B.P.F. Lelieveldt. Model driven quantification of left ventricular function from sparse single-beat 3d echocardiography. *Medical Image Analysis*, 14(4):582–593, 2010. [cited at p. 103]
- [118] M. Ma, M. van Stralen, J.H.C. Reiber, J.G. Bosch, and B.P.F. Lelieveldt. Left ventricle segmentation from contrast enhanced fast rotating ultrasound images using three dimensional active shape models. In Nicholas Ayache, Hervé Delingette, and Maxime Sermesant, editors, *Functional Imaging and Modeling of the Heart, Lecture Notes in Computer Science*, volume 5528, pages 295–302. Springer Berlin Heidelberg, 2009. [cited at p. 103]
- [119] K.Y.E. Leung, M. van Stralen, M. M. Voormolen, N. de Jong, A. F. W. van der Steen, J. H. C. Reiber, and J. G. Bosch. Improving 3d active appearance model

- segmentation of the left ventricle with jacobian tuning. In *Proceedings of SPIE Medical Imaging*, pages 69143B–69143B–11, 2008. [cited at p. 103]
- [120] M. van Stralen, K.Y.E. Leung, M.M. Voormolen, N. de Jong, A.F.W. van der Steen, J.H.C. Reiber, and J.G. Bosch. P2a-6 automatic segmentation of the left ventricle in 3d echocardiography using active appearance models. In *Ultrasonics Symposium, 2007. IEEE*, pages 1480–1483, oct. 2007. [cited at p. 103]
- [121] J. Hansegard, S. Urheim, K. Lunde, and S.I. Rabben. Constrained active appearance models for segmentation of triplane echocardiograms. *Medical Imaging, IEEE Transactions on*, 26(10):1391–1400, oct. 2007. [cited at p. 103]
- [122] V. Lempitsky, M. Verhoek, J.A. Noble, and A. Blake. Random forest classification for automatic delineation of myocardium in real-time 3d echocardiography. In Nicholas Ayache, Hervé Delingette, and Maxime Sermesant, editors, *Functional Imaging and Modeling of the Heart, Lecture Notes in Computer Science, volume 5528*, pages 447–456. Springer Berlin Heidelberg, 2009. [cited at p. 103]
- [123] X. Papademetris, A.J. Sinusas, D.P. Dione, and J.S. Duncan. Estimation of 3d left ventricular deformation from echocardiography. *Medical Image Analysis*, 5(1):17–28, 2001. [cited at p. 103]
- [124] D. Boukerroui, O. Basset, A. Baskurt, and G. Gimenez. A multiparametric and multiresolution segmentation algorithm of 3d ultrasonic data. *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on*, 48(1):64–77, jan. 2001. [cited at p. 103]
- [125] X. Ye, J.A. Noble, and D. Atkinson. 3-d freehand echocardiography for automatic left ventricle reconstruction and analysis based on multiple acoustic windows. *Medical Imaging, IEEE Transactions on*, 21(9):1051–1058, sept. 2002. [cited at p. 103]
- [126] V. Chalana, D.T. Linker, D.R. Haynor, and Y. Kim. A multiple active contour model for cardiac boundary detection on echocardiographic sequences. *Medical Imaging, IEEE Transactions on*, 15(3):290–298, jun 1996. [cited at p. 104]
- [127] S. Malassiotis and M.G. Strintzis. Tracking the left ventricle in echocardiographic images by learning heart dynamics. *Medical Imaging, IEEE Transactions on*, 18(3):282–290, march 1999. [cited at p. 104]
- [128] J. Montagnat, H. Delingette, and N. Ayache. A review of deformable surfaces: topology, geometry and deformation. *Image and Vision Computing*, 19(14):1023–1040, 2001. [cited at p. 104]
- [129] E.D. Angelini, A.F. Laine, S. Takuma, J.W. Holmes, and S. Homma. Lv volume quantification via spatiotemporal analysis of real-time 3-d echocardiography. *Medical Imaging, IEEE Transactions on*, 20(6):457–469, june 2001. [cited at p. 104]

- [130] E.D. Angelini, S. Homma, G. Pearson, J.W. Holmes, and A.F. Laine. Segmentation of real-time three-dimensional ultrasound for quantification of ventricular function: A clinical study on right and left ventricles. *Ultrasound in Medicine and Biology*, 31(9):1143 – 1158, 2005. [cited at p. 104, 131]
- [131] L.A. Chan, T.F. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10:266–277, 2001. [cited at p. 104, 108, 111]
- [132] Y. Chen, F. Huang, H.D. Tagare, M. Rao, D. Wilson, and E.A. Geiser. Using prior shape and intensity profile in medical image segmentation. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, volume 2, pages 1117 –1124, oct. 2003. [cited at p. 105]
- [133] N. Paragios. A level set approach for shape-driven segmentation and tracking of the left ventricle. *Medical Imaging, IEEE Transactions on*, 22(6):773 –776, june 2003. [cited at p. 105]
- [134] Ning Lin, Weichuan Yu, and James S. Duncan. Combinative multi-scale level set framework for echocardiographic image segmentation. *Medical Image Analysis*, 7(4):529 – 537, 2003. [cited at p. 105]
- [135] C. Corsi, M. Borsari, F. Consegna, A. Sarti, C. Lamberti, A. Travaglini, T. Shiota, and J.D. Thomas. Left ventricular endocardial surface detection based on real-time 3d echocardiographic data. *European Journal of Ultrasound*, 13(1):41 – 51, 2001. [cited at p. 105]
- [136] C. Corsi, G. Saracino, A. Sarti, and C. Lamberti. Left ventricular volume estimation for real-time three-dimensional echocardiography. *Medical Imaging, IEEE Transactions on*, 21(9):1202 –1208, sept. 2002. [cited at p. 105]
- [137] A. Sarti, R. Malladi, and J.A. Sethian. Subjective surfaces: A geometric model for boundary completion. *International Journal of Computer Vision*, 46:201–221, 2002. [cited at p. 105]
- [138] Q. Duan, E.D. Angelini, and A.F. Laine. Real-time segmentation by active geometric functions. *Computer Methods and Programs in Biomedicine*, 98(3):223 – 230, 2010. [cited at p. 105]
- [139] D. Barbosa, T. Dietenbeck, J. Schaerer, J. D’hooge, D. Friboulet, and O. Bernard. B-spline explicit active surfaces: An efficient framework for real-time 3d region-based segmentation. *IEEE Transactions on Image Processing*, 21(1):241–251, 2012. [cited at p. 105]
- [140] D. Barbosa, T. Dietenbeck, B. Heyde, H. Houle, D. Friboulet, J.D’hooge, and O. Bernard. Fast and fully automatic 3d echocardiographic segmentation using b-spline explicit active surfaces: feasibility study and validation in a clinical setting. *Ultrasound in Medicine and Biology*, 39(1):89–101, 2013. [cited at p. 105, 114, 129, 131]

- [141] R. T. Whitaker. A level-set approach to 3d reconstruction from range data. *International Journal on Computer Vision*, 29(3):203–231, 1998. [cited at p. 106, 115, 117, 118]
- [142] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1:321–331, 1988. [cited at p. 106]
- [143] A. Blake and Isard M. *Active Contours*. Springer-Verlag, Cambridge, Massachusetts, 1998. [cited at p. 106]
- [144] N. Paragios and R. Deriche. Geodesic active contours and level sets for the detection and tracking of moving objects. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(3):266–280, mar 2000. [cited at p. 106]
- [145] N. Paragios, Y. Chen, and O. Faugeras. *Handbook of Mathematical Models in Computer Vision*. Springer-Verlag, New York, 2005. [cited at p. 106]
- [146] T. Zhang and D. Freedman. Tracking objects using density matching and shape priors. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1056–1062 vol.2, oct. 2003. [cited at p. 106]
- [147] O. Bernard. Segmentation in echocardiographic imaging using parametric level set model driving by the statistics of the radiofrequency signal. Ph.D. Thesis, INSA- University of Lyon. <http://www.creatis.insa-lyon.fr/~bernard/publications.html>, 2006. [Online;]. [cited at p. 106, 108, 109, 165]
- [148] A. E. Lefohn, J. M. Kniss, C. D. Hansen, and R. T. Whitaker. A streaming narrow-band algorithm: interactive computation and visualization of level sets. *IEEE Transactions on Visualization and Computer Graphics*, 10(4):422–433, 2004. [cited at p. 107, 111, 116, 117]
- [149] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22:61–79, 1995. [cited at p. 108, 110]
- [150] J. A. Yezzi, A. Tsai, and A. Willsky. A fully global approach to image segmentation via coupled curve evolution equations. *Journal of Visual Communication and Image Representation*, 13:195–216, 2002. [cited at p. 108, 113]
- [151] G. Aubert, M. Barlaud, O. Faugeras, and S. Jehan-Besson. Image segmentation using active contours: Calculus of variations or shape gradients? *SIAM Applied Mathematics*, 63(6):2128–2154, 2003. [cited at p. 108]
- [152] O. Michailovich, Y. Rathi, and A. Tannenbaum. Image segmentation using active contours driven by the bhattacharya gradient flow. *IEEE Transactions on Image Processing*, 15(11):2787–2801, 2007. [cited at p. 108]

- [153] D. Cremers, M. Rousson, and R. Deriche. A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape. *International Journal of Computer Vision*, 72(2):195–215, 2007. [cited at p. 108]
- [154] S. Lankton and A. Tannebaum. Localizing region-based active contours. *IEEE transaction on Image Processing*, 17(11):2029–2039, 2008. [cited at p. 108, 112, 113, 126]
- [155] M.E. Leventon, W.E.L. Grimson, and O. Faugeras. Statistical shape influence in geodesic active contours. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 316–323, 2000. [cited at p. 108]
- [156] Y. Chen, H.D. Tagare, S. Thiruvenkadam, F. Huang, D. Wilson, K.S. Gopinath, R.W. Briggs, and E.A. Geiser. Using prior shapes in geometric active contours in a variational framework. *International Journal of Computer Vision*, 50(3):315–328, 2002. [cited at p. 108, 114]
- [157] X. M. Pardo, V. Lebor, and R. Dosil. Integrating prior shape models into level-set approaches. *Pattern Recognition Letters*, 25(6):631–639, 2004. [cited at p. 108]
- [158] M. Alessandrini, T. Dietenbeck, O. Basset, D. Friboulet, and O. Bernard. Using a geometric formulation of annular-like shape priors for constraining variational level-sets. *Pattern Recognition Letters*, 32:1240–1249, 2011. [cited at p. 108]
- [159] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer Verlag, 2002. [cited at p. 110]
- [160] R. Malladi, J.A. Sethian, and B.C. Vemuri. Shape modeling with front propagation: a level set approach. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(2):158 – 175, feb 1995. [cited at p. 110]
- [161] M. Roberts, J. Packer, M. C. Sousa, and J. R. Mitchell. A work-efficient gpu algorithm for level set segmentation. In *Proceedings of the Conference on High Performance Graphics (HPG '10)*, pages 123–132, 2010. [cited at p. 111, 116, 118, 124, 125]
- [162] A.E. Lefohn, J.E. Cates, and R.T. Whitaker. Interactive gpu-based level sets for 3d segmentation. In *Proceedings of Medical Image Computing and Computer Assisted Intervention (MICCAI'03)*, pages 564–572, 2003. [cited at p. 111, 116, 117]
- [163] A.E. Lefohn, J.M. Kniss, C.D. Hansen, and R.T. Whitaker. Interactive deformation and visualization of level set surfaces using graphics hardware. In *Proceedings of 14th IEEE Visualization, VIS '03*, page 11, 2003. [cited at p. 111, 116, 117]
- [164] T. Chan and Wei Zhu. Level set based shape prior segmentation. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 1164–1170, June. [cited at p. 114]

- [165] D. Adalsteinsson and J. A. Sethian. A fast level set method for propagating interfaces. *Journal of Computational Physics*, 118(2):269–277, 1995. [cited at p. 115]
- [166] Joshua E Cates, Aaron E Lefohn, and Ross T Whitaker. Gist: an interactive, gpu-based level set segmentation tool for 3d medical images. *Medical Image Analysis*, 8(3):217–231, 2004. [cited at p. 116]
- [167] O. Klar. *Interactive GPU-based Segmentation of Large Medical Volume Data with Level-Sets*. PhD thesis, VRVis and University Koblenz-Landau, 2006. [cited at p. 116]
- [168] N. Cuntz, A. Kolb, R. Strzodka, and D. Weiskopf. Particle level set advection for the interactive visualization of unsteady 3d flow. In *Computer Graphics Forum*, volume 27, pages 719–726, 2008. [cited at p. 116]
- [169] X. Mei, P. Decaudin, B. Hu, and X. Zhang. Real-time marker level set on gpu. In *Proceedings of the 2008 International Conference on Cyberworlds (CW '08)*, pages 209–216, 2008. [cited at p. 116]
- [170] W. Jeong, J. Beyer, M. Hadwiger, A. Vazquez, H. Pfister, and R. T. Whitaker. Scalable and interactive segmentation and visualization of neural processes in em datasets. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1505–1514, 2009. [cited at p. 116, 117]
- [171] A.C. Jalba, W. J. van der Laan, and J.B.T.M. Roerdink. Fast sparse level-sets on graphics hardware. *IEEE Transactions on Visualization and Computer Graphic*, 19(1):30–44, 2012. [cited at p. 117]
- [172] EM Photonics. Cula reference manual. <http://www.culatools.com>, 2012. [Online;]. [cited at p. 124]
- [173] S. Lankton. Sparse field methods-technical report. Technical report, Georgia Institute of Technology, 2009. <http://www.shawnlankton.com/wp-content/uploads/articles/lankton-sfm-TR-2009.pdf>. [cited at p. 125]
- [174] L. R. Dice. Measures of the Amount of Ecologic Association Between Species. *Ecology*, 26(3):297–302, July 1945. [cited at p. 125]
- [175] Brainweb: Simulated brain database. <http://www.bic.mni.mcgill.ca/brainweb>, 2011. [Online;]. [cited at p. 126]
- [176] J. Huang and J.M. van Gelder. The probability of sudden death from rupture of intracranial aneurysms: a meta-analysis. *Journal of Neurosurgery*, 51(5):1101–1105, 2005. [cited at p. 129]
- [177] J.L. Brisman, J.K. Song, and D.W. Newell. Cerebral aneurysms. *New England Journal of Medicine*, 355(9):928–939, 2006. [cited at p. 129]



---

# Abbreviations

---

Abbreviation	Description
3DRA	3D Rotational Angiography
ACC	Accuracy
ADC	Analog to Digital Converter
A-Mode	Amplitude Mode
B-Mode	Brightness mode
CAD	Computer Aided diagnosis
CT	Computed Tomography
CV	Cross Validation
DRE	Digital Rectal Examination
FLD	Fisher Linear Discriminant
GDA	Generalized Discriminant Analysis
GDA+FLD	Non-linear Classifier obtained combining Generalized Discriminant Analysis and Fisher Linear Discriminant
GPU	Graphic Processing Unit
KFD	Kernel Fisher Discriminant
LV	Left-Ventricle
M-Mode	Motion Mode
MRI	Magnetic Resonance Imaging
PCa	Prostate Cancer
PCA3	Prostate CAncer gene 3
PPV	Positive Predictive Value
PSA	Prostate Specific Antigen
P-SF	Parallel Sparse Field
QFLD	Non linear classifier obtained combining Fisher Linear Discriminant and Quadratic Data Mapping
RF	Radio-Frequency
ROI	Region of Interest
RT3DE	Real-Time 3D Echocardiography
SE	Sensitivity
SL	Supervised Learning
SMS	Supervised Model Selection
S-SMS	Semi-Supervised Model Selection

---

---

Abbreviation	Description
SP	Specificity
SSL	Semi-Supervised Learning
S-SF	Sequential Sparse Field
SVM	Support Vector Machine
TC-SF	Temporally Coherent Sparse Field
TRUS	Trans-Rectal Ultrasound
US	Ultrasound
UTC	Ultrasound Tissue Characterization
VE	Validation Error

---

---

# List of Figures

---

1.1	Longitudinal pressure wave. . . . .	2
1.2	Reflection and refraction of plane waves in Snell's law. . . . .	4
1.3	Diffuse reflection and Rayleigh scattering. . . . .	5
1.4	Block diagram of an ultrasound imaging system. . . . .	8
1.5	Configuration of multi-element probes and possible region scanned by the acoustic beam: sequential linear array (a), curvilinear array (b), linear phased array (c), 1.5D array (d) and 2D array (e). . . . .	9
1.6	Examples of medical ultrasound probes (picture from [2]). . . . .	10
1.7	NVIDIA's Fermi Architecture (picture from [26]). . . . .	20
1.8	CUDA threads hierarchy and example of the execution flow of a CUDA program (picture from [28]). . . . .	21
2.1	Prostate anatomy and its location in the body (picture from [30]). . . . .	25
2.2	Prostate schematic illustration in sagittal and coronal view. PZ = peripheral zone, TZ = transition zone, CZ = central zone. (picture from [31]). . . . .	25
2.3	Diagnostic tools currently used in PCa detection. . . . .	26
2.4	Schematic representation of prostate axial section with orientation inverted <i>w.r.t.</i> its real anatomy (left) and correspondent TRUS image (right). . . . .	27
2.5	Common scheme of UTC-based CAD systems architecture. . . . .	30
2.6	An example of separating hyperplane in the two-dimensional case. . . . .	31
2.7	Comparison of the power series approximation in the ML Nakagami estimator and the corresponding Padé Approximant. . . . .	35
2.8	Ultrasound signal processing chain in standard (a) and computer aided (b) biopsy. . . . .	45
2.9	Example of form used during biopsy sessions to collect information about patient history, clinical information and results of DRE and TRUS examination performed before and during biopsy. . . . .	48
2.10	Needle segmentation procedure: in figure (a) the penetration track identification in the RF space is shown. The ROIs automatically generated on the needle track are shown in figure (b). . . . .	49

2.11	Signal processing flow graph of rtCAB: from left to right the incoming RF signal undergoes envelope preprocessing, segmentation, feature estimation and classification. Preprocessing step comprises envelope detection and/or logarithmic compression depending on the kind of features that must be computed. Results are scan converted and superimposed to the original B-Mode image. . . . .	51
2.12	Schematic representation of the segmentation and features estimation procedure: from the input image the ROIs are identified based on the user defined parameters and stored per-column in the ROIs buffer; the ROIs buffer is the input of the features estimation software that computes $N_{feat} \times N_{roi}$ features and store the, in the features matrix. . . . .	54
2.13	An example of the histogram computation procedure for $D = 1$ in the angular direction $0^\circ$ : two pixels are extracted from the ROI buffer and their sum is used as array index to update the appropriate histogram location. Note that in the ROIs buffer each ROI is stored per-column, therefore, into an ROI of size $N_r \times N_c$ pixels consecutive in the horizontal direction are distant $N_r$ to each other in memory. . . . .	55
2.14	The core based PPV and SE of both the Nakagami (a) and Nakagami-Unser (b) classification tasks. The x-axis represents the 1000 threshold values. At right the color bar shows the color coding for the SE and PPV value. Each horizontal bar represents a patient from the database identified by a three letter code: the color in each bar indicates how the respective statistics changes as a function of the classifier threshold over the given patient. The vertical dashed line is the best threshold $Th_{best}$ selected by the learning scheme. 59	59
2.15	Examples of B-Mode images with superimposed needle trace and the corresponding image classified by the CAD for a healthy (a-b) and a unhealthy (c-d) patient. Images (a-c) refer to the Nakagami classification task while images (b-d) refer to the Nakagami-Unser one. For each image the frame on the top is the one corresponding to the biopsy sampling, while the previous frame with superimposed the binary risk map generated by the CAD is shown on the bottom. . . . .	60
2.16	rtCAB tool interface. . . . .	61
3.1	SelfCo3Core Flowchart . . . . .	79
3.2	Training and validation flows: training steps are represented by gray boxes while validations by white ones. . . . .	80

3.3	The core based PPV and SE of both the state-of-art (a) and the proposed detection tool (b). The x-axis represents the threshold values. At right the color bar shows the color coding for the SE and PPV value. Each horizontal bar represents a patient from the database identified by a three letter code: the color in each bar indicates how the respective statistics changes as a function of the classifier threshold over the given patient. The vertical dashed line is the best threshold $Th_{best}$ selected by the learning scheme. . . . .	86
3.4	Example of B-Mode images with superimposed needle trace and the corresponding image classified by the CAD tools for a unhealthy (a-b) and healthy (c-d) patient. Images (a-c) refer to the state-of-art CAD tool while images (b-d) refer to the proposed one. For each image the frame on the top is the one corresponding to the biopsy sampling, while the previous frame with superimposed the binary risk map generated by the CAD is shown on the bottom. . . . .	88
4.1	Heart position in the human body (picture from [96]). . . . .	93
4.2	Internal view of the heart anatomy (a) and path of blood through heart (b) (picture from [98]). . . . .	94
4.3	Workflow for cardiac functional indices computation considering the real-time 3D echocardiography as cardiac imaging modality. . . . .	97
4.4	Linear regression plot. . . . .	101
4.5	Bland-Altman analysis plot. . . . .	101
4.6	Example of segmentation using active contours (picture from [147]). . . . .	106
4.7	Level-set function for a two dimensional domain $\Omega$ represented as an elevation map (a) and the corresponding domain partition (b) (picture from [147]).	108
4.8	Evolution of the embedding function (b,d,f,h), along with the evolution of its zero level-set (red) while segmenting a synthetic image (a,c,e,g). The black curve on (a,c,e,g) corresponds to the projection of the zero level on the synthetic image (picture from [147]). . . . .	109
4.9	An example of a level-set function $\phi$ embedding a 2D contour and its discretization on a rectilinear grid of points where each point has a value of $\phi$ corresponding to its distance from the zero level-set. . . . .	111
4.10	Synthetic image with heterogeneous intensity profile: Initial contour (a) and unsuccessful result of region-based segmentation (b). . . . .	112
4.11	Local neighborhoods of contour points: circular (spherical) neighborhood (a) and neighborhood on the vector normal to the evolving surface (b). In each neighborhood local interior is indicated in blue, while local exterior is indicated in red. . . . .	113
4.12	Computational domain restriction obtained using the narrow band approach.	115
4.13	Illustration of the definition of the active set and its neighborhood in layers. . . . .	119

4.14	Illustration of the SF representation of the $\phi$ domain, the use of layers of points $L_i$ , and the interpretation of layers as 1D array adopted by the P-SF. Numeric range for each list refers to the $\phi$ values for each layer. . . . .	121
4.15	Simulated images used for the algorithms evaluation. Each image corresponds to a slice extracted from the corresponding volume. . . . .	127
4.16	Segmentation results using our GPU level-set solver P-SF along with the corresponding execution time. . . . .	127
4.17	3DRA segmentation results performed on Patient1 using P-SF (a) and TC-SF (b) algorithms along with the corresponding execution time. . . . .	130
4.18	Linear regression analysis for EDV, ESV, SV and EF. . . . .	132
4.19	Bland-Altman analysis plots for EDV, ESV, SV, and EF, comparing the measured errors against the reference values. The green line corresponds to the bias ( $*, p < 0.05$ ) and the red ones to the limits of agreement ( $\mu \pm 1.96\sigma$ ). . . . .	132
4.20	Segmentation of a RT3DE exam using P-SF. (a) EDV (Ref.=113.8ml, P-SF=124.5ml), (b) ESV (Ref.=39.8ml, P-SF=45.6ml). . . . .	133
4.21	Linear Regression Analysis for EDV, ESV, SV and EF obtained adding the shape prior constraint. . . . .	135
4.22	Bland-Altman Analysis plots for EDV, ESV, SV, and EF, comparing the measured errors against the reference values in case of using the shape prior constraint [ <i>green line</i> : bias ( $*, p < 0.05$ ), <i>red lines</i> : limits of agreement ( $\mu \pm 1.96\sigma$ )]. . . . .	135
4.23	Segmentation of a RT3DE exam using P-SF with the shape prior constraint. (a) EDV (Ref.=113.8ml, P-SF=122.7ml), (b) ESV (Ref.=39.8ml, P-SF=42.6ml). . . . .	136
4.24	Segmentation results obtained with (magenta) and without (yellow) the prior constraint for volume slices in x, y, and z directions, for both EDV (a) and ESV (b). . . . .	137

---

# List of Tables

---

1.1	Reflection coefficient for common interfaces in medical field. . . . .	5
1.2	Penetration depth at -3 dB for common materials and tissues. . . . .	7
1.3	Comparison of imaging modalities [8]. . . . .	12
2.1	Selected and ranked feature set obtained applying the MIHFS algorithm to the whole Nakagami-Unser feature set. . . . .	37
2.2	Comparison of the the published methods for prostate tissue characterization with ultrasounds. . . . .	45
2.3	Ground Truth Database. . . . .	50
3.1	Benchmark reference datasets from UCI machine learning repository [94]. .	71
3.2	Classification errors of the initial and final classification models and the corresponding improvements of SelfCo3, SelfCo2, Tri-training and self-training2 under different label rates. . . . .	73
3.3	Evaluation 1: classification errors of compared classification models and relative improvements. . . . .	76
3.4	Evaluation 2: classification errors of compared classification models and relative improvements. . . . .	77
3.5	Classification errors of compared classification models and relative improvements. . . . .	85
4.1	Functional indices of an healthy heart: typical values and normal ranges. . .	97
4.2	Parameter values used for the segmentation of simulated images using Lefohn speed term. . . . .	126
4.3	Segmentation results using a time constant speed function. . . . .	126
4.4	Segmentation results using a time varying speed function. . . . .	128
4.5	Vessel with aneurysm segmentation results obtained from a three-dimensional rotational angiography. . . . .	130
4.6	Left ventricle segmentation performance obtained from 3D Echocardiography imaging. . . . .	131

4.7	Left ventricle segmentation performance obtained from 3D Echocardiography imaging using a shape prior constraint. . . . .	134
-----	---------------------------------------------------------------------------------------------------------------------------	-----