Carnegie Mellon University

CARNEGIE INSTITUTE OF TECHNOLOGY

THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF Doctor of Philosophy

TITLE **Development of Computational Tools for Computerized Training of Computerized Training of Cryosurgery**

PRESENTED BY **Robert L. Keelan**

ACCEPTED BY THE DEPARTMENT OF Mechanical Engineering

_____	_____
ADVISOR, PROFESSOR YOED RABIN	DATE
_____	_____
ADVISOR, PROFESSOR KENJI SHIMADA	DATE
_____	_____
DEPARTMENT HEAD	DATE

APPROVED BY THE COLLEGE COUNCIL

_____	_____
DEAN	DATE

Development of Computational Tools for Computerized Training of Cryosurgery

Submitted in partial fulfillment of the requirements for

the degree of

Doctor of Philosophy

in

Mechanical Engineering

Robert L. Keelan

B.S., Mechanical Engineering, Carnegie Mellon University
M.S., Mechanical Engineering, Carnegie Mellon University

Carnegie Mellon University
Pittsburgh, PA

May, 2015

Acknowledgements

I would like to express my gratitude to my research advisors, Professors Yoed Rabin and Kenji Shimada, for providing me the opportunity to pursue a Ph.D. degree. Their guidance and patience over the past four years has been instrumental to both my professional and personal development.

I would like to thank Professors Phil Campbell and Jessica Zhang for agreeing to be on my thesis committee. The constructive comments and suggestions they have provided throughout the course of this research has been invaluable.

This project is supported by Award Number RC01CA134261 from the National Cancer Institute

I would also like to thank my CMU colleagues: Dr. Soji Yamakawa and Tomotake Furuhata for providing key insights and numerical tools, without which this research would not have been possible, Dr. Anjali Sehrawat and Dr. Hong Zhang for their collaboration on various papers, Dr. Michael Rossi and Dr. Daigo Tanaka who provided the foundation for this work, as well as the rest of the members of the Biothermal Technology Laboratory. I am very grateful to the staff members in the Mechanical Engineering Department who have assisted me over my many years there, especially Jim Dillinger, John Fulmer, and Edward Wojciechowski.

I would like to thank my mother and father, Charlene and Jack Keelan, for their love and support, and for teaching me the value of hard work. I would also like to thank my mother and father in law, Bob and Lisa Hyland, for all of their help and guidance throughout this process.

Most importantly, I want to thank my wife Ellen, who has given me more love and support throughout my studies at Carnegie Mellon than I could have ever imagined. Your sacrifices made this degree possible. I cannot thank you enough.

This thesis is dedicated to my wife and best friend, Ellen Paige Keelan.

Abstract

Cryosurgery is the destruction of undesired tissue by freezing. Modern cryosurgery is performed by strategically inserting a number of cooling probes (cryoprobes) shaped like long hypodermic needles into the target region. Minimally invasive cryosurgery presents unique challenges to the clinician associated with the use of ultrasound (US) as the imaging modality, the selection of the cryoprobe configuration, the accurate placement of the cryoprobes, and the monitoring of the procedure through imaging and temperature measurements. Currently the cost of training clinicians to perform cryosurgery is exceptionally high because the procedure itself is very complex, and the learning process consists of a long residency period.

The work presented in this thesis is part of an ongoing project between the Biothermal Technology Laboratory and the Computational Engineering and Robotics Laboratory at Carnegie Mellon University, to develop a computerized training platform that will teach clinicians how to perform the minimally invasive cryosurgery effectively and efficiently. This work uses prostate cryosurgery as a developmental model for verification and benchmarking of all algorithms.

Towards that goal, this work provides many of the tools needed to create a virtual training environment. A cryosurgical training framework was presented to enable the diverse functionality of such a computerized training system. Next, a method for generating synthetic ultrasound images from a small library of samples was presented. This method is based off of a texture synthesis technique called “Image Analogies”. The algorithm generates realistic 3D ultrasound images from a small library of samples. Then, an improvement was made to an efficient bioheat transfer simulator. The implementation used GPU computing to achieve a 15x increase in performance over the previous state of the art approach. The enhanced bioheat simulator was leveraged in a cryosurgically relevant ultrasound simulator. The ultrasound simulator uses nonlinear ray-tracing and a novel energy propagation step to simulate imaging artifacts associated with cryoprobe insertion and the thermal field in real-time. Finally, an enhancement to the cryoprobe placement algorithm “bubble packing” was performed to enable the addition of geometric constraints. These geometric constraints allow more clinically relevant plans to be generated. It is the hope of the author that these tools will be the foundation of a fully immersive surgical simulator.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

The work presented in this thesis is focused on the development of computation tools for computerized cryosurgical training. This chapter provides a brief overview of the mechanisms of tissue injury and discusses the practices and challenges of modern prostate cryosurgery. Additionally, this chapter introduces computerized cryosurgical training, along with a sample work flow that this thesis aims to address. Finally, the unique contributions are covered in the specific aims.

1.1 Mechanisms of Cryosurgical Tissue Injury

Cryosurgery aims to destroy undesired biological tissue through freezing. In order to give context and understanding to modern cryosurgical procedures, it is of key importance to understand the mechanisms of tissue injury. There are many different mechanisms in which cell death may be induced at low temperatures. Cryosurgery utilizes the complex interactions between biological tissues and low temperatures in order to destroy undesired tissues, while leaving healthy tissues unaffected. Mechanisms of tissue injury may be roughly divided into three temperature regions; a hypothermic range (20°C to 0°C), a sub-zero range (0°C to -22°C), and a low freezing range (-22°C to -40°C) [1].

In the hypothermic range temperatures are too high for freezing to occur. However, exposure to temperatures in this range causes the cell metabolism to gradually decay, eventually leading to cell death. Cell survival at 0°C is a few days at most, but even short exposure to non-freezing temperatures may destroy the cell. Although tissue injury due to hypothermia has received little attention in cryosurgery, it still plays a role in cryosurgical cell death.

Hypothermia is present many minutes before and after cryosurgical procedures, especially in the regions surrounding the frozen volume [2].

As temperatures enter the sub-zero range, water in the extracellular fluid begins to crystalize. The conversion from liquid water to solid ice works to increase the effective electrolyte concentrations outside the cell. This change in extracellular fluid solute concentration induces an osmotic pressure that draws water out of the cells leading to cell dehydration. Also, the increase in solute concentration outside of the cell leads to a similar increase in solute concentration inside the cell. This increase in solute concentration, combined with cell dehydration that accompanies the formation of extracellular ice, significantly damages the cell, which is known as solution-effect injury [3]. Cells are resistant to forming ice crystals down to temperatures of -10°C.

Once low freezing temperatures are reached, many cells develop intracellular ice (heterogeneous nucleation), however; temperatures are required to drop to -40°C for ice to form with certainty. The formation of intracellular ice is almost always lethal. Temperatures below -45°C are clinically considered lethal, because cell viability after this point is highly unlikely. As a result, temperatures in this range are known as “lethal temperatures.” Faster cooling rates make intracellular ice more probable while slower cooling rates tend to injure cells through solution effect injury. This occurs because fast cooling rates do not allow time for water to escape from the cell. Additionally, whether the ice is intracellular or extracellular, ice formation will induce shear stresses on the cells, which causes further damage to its walls and organelles [1].

Perhaps a more important parameter to cell injury is the warming rate. A slow warming rate allows ice crystals to fuse and reform in a process called recrystallization. Large ice crystals are injurious to cell membranes. Also, as ice melts in the extracellular environment it becomes

hypotonic. This hypotonic environment allows water to enter the dehydrated cell, increasing the cell volume until it ruptures [4].

In addition to these immediate causes of cryosurgical tissue death, there are also delayed causes which take effect after the tissue thaws. One mechanism of delayed injury is the disruption of the microcirculation caused by previously acquired cell injuries. Cells may also die due to apoptosis (programmed cell death) after sustaining injuries due to hypothermia [5].

In the context of cryosurgical procedures a considerable number of studies have found that cycling the tissue between multiple freeze thaw cycles is beneficial. Multiple freeze thaw cycles destroy a greater percentage of tissue within the frozen region [6]. Tissues further from the cryoprobes have a slower thermal response. As result of these lower cooling rates, they are less likely to produce intracellular ice and cell death. Multiple freeze thaw cycles are used in cryosurgery because often times, the tissue that needs to be destroyed is very close to critical healthy tissues. It would be impossible to induce a lethal temperature within the target tissue without also producing an unacceptable amount of collateral damage to the surrounding healthy tissue. By using multiple freeze thaw cycles, warmer temperatures may be induced in the target and surrounding tissues, while increasing the probability of causing cell destruction within the cooled region [7].

Following cryosurgery, an edema is present within 24 hours. Necrosis of the tissue is evident after two days. The cryogenic lesion that forms closely matches the volume of tissue that was previously frozen. In the following days more cells die due to delayed effects, achieving maximum cellular injury within 10 days. Treated tissue will be replaced by scar tissue as healing progresses [8].

1.2 Modern Prostate Cryosurgery

Prostate cancer is the most common cancer found in men, and the second leading cause of cancer-related deaths among men. Prostate Cryosurgery has had practical applications since the beginning of the 20th century [9]. The development of cryosurgery as an invasive procedure started in 1961, when Cooper and Lee invented the first cryoprobe (then cryostat) based on liquid nitrogen boiling [10]. Cryoprobes are shaped like long hypodermic needles with a sharp tip. The cooling effect is created near the tip of the cryoprobe. Contrary to common belief, the Joule-Thomson-based cryoprobe was presented shortly thereafter. Joule-Thompson-based cryoprobes are cooled by a sudden release of pressurized gas resulting in a temperature change [11,12]. Ever since, the technology of cryosurgical devices has shown tremendous improvements, following trends of miniaturization while improving cooling capabilities. Dramatic developments in medical imaging in the 1980s eventually led to a quantum leap in cryosurgical practice in the early 1990s, from a superficial treatment (often combined with surgical excision) to an image-guided, minimally invasive procedure on the prostate [13]. While minimally invasive cryosurgery has been experimented on virtually every tissue of the body, the prostate was the first organ to be treated with this application, for which a wide database is already available [6,14,15].

Cryo-therapy is an option for men who have non-metastatic cancer (stage T3 or earlier). Often cryosurgery is performed as a salvage surgery after another treatment such as brachytherapy has failed. Cryosurgery is also commonly used among patients with a high bleeding risk due to its minimally invasive nature. Often it can be performed as an outpatient procedure with only minor swelling and tenderness after surgery.

Modern prostate cryosurgery is performed by placing a number of cryoprobes (modern procedures can use upwards of 12) into the prostate through a registration grid. This registration

grid is used to place cryoprobes in known locations relative to the trans-rectal ultrasound probe used for imaging [6]. The urethra travels through and is encircled by the prostate, so urethral tissue must be protected during the freezing process. In order to prevent the urethra from freezing, a warming catheter (urethral warmer) is inserted through the urethra and into the bladder. A saline solution is then circulated through this urethral warmer during the procedure. A diagram of a typical surgical setup is shown below in Figure 1-1.

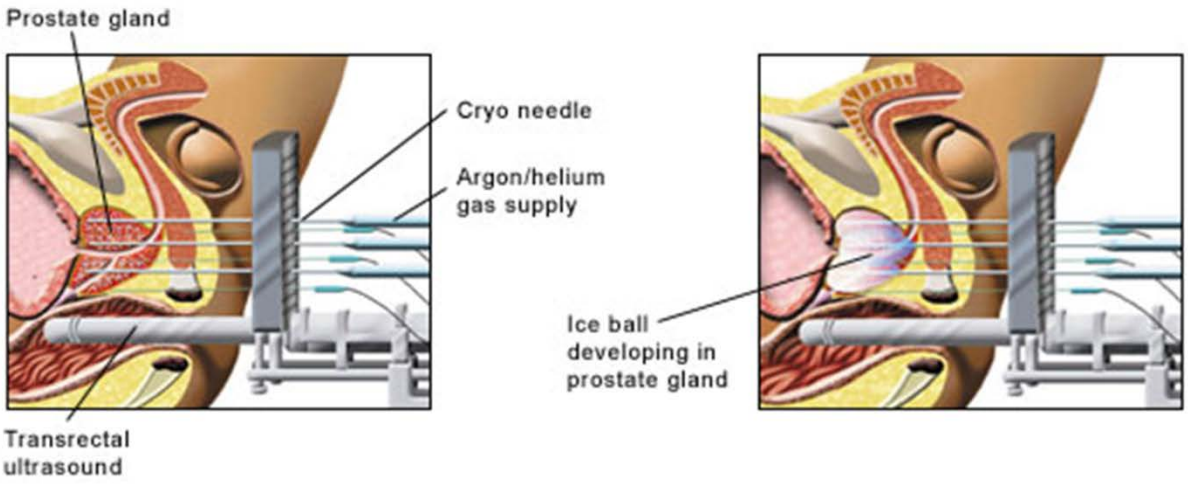

Figure 1-1: Diagram of modern minimally invasive prostate cryosurgery [16]

Once cooling has begun, the extent of the frozen tissue is monitored through the use of both temperature sensors and ultrasound imaging. As shown in Figure 1-2, frozen tissue creates a distinct shadow in the ultrasound where clinicians can track the extent of cooling in the lower half of the prostate. Usually, multiple freeze-thaw cycles are performed in order to maximize damage to the target tissue and minimize collateral damage to surrounding healthy tissue [17].

The prostate is similar in size and shape to a chestnut, and has a wide base close to the bladder and a narrow apex away from the bladder. Because of this irregular shape, a “pullback”

procedure is often performed, where cryoprobes are inserted to their maximum depth to freeze the wide base of the prostate with a number of freeze thaw cycles. Then, some of the periphery probes are removed, while centrally located probes are pulled back in depth to freeze the narrow apex of the gland [18].

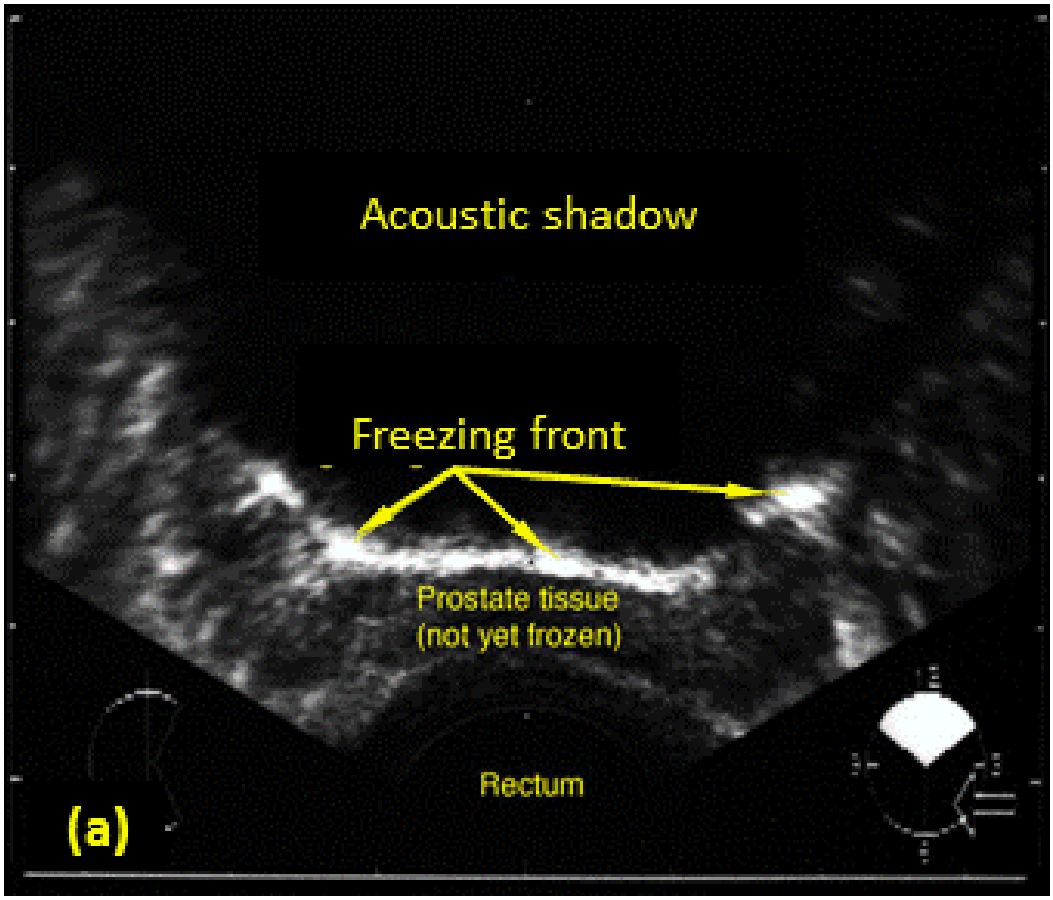

Figure 1-2: Ultrasound shadowing caused by freezing of tissue [19]

1.3 Computerized Training of Cryosurgery

The application of cryosurgery is a four-stage process: (i) reconstruction of the target region based on an imaging technique; (ii) evaluation of the optimum number of cryoprobes and their layout; (iii) insertion of the cryoprobes according to the optimal plan; and, (iv) orchestration of cryoprobe operation to achieve the optimum match between the target region and the forming frozen region. Cryosurgical success is the product of the successes of each of the above stages. To date, these surgical steps are performed manually, relying solely upon the cryosurgeon's experience. While, current surgical training practices necessitate an elongated learning curve and extended practice time.

The new minimally invasive procedure has presented practical challenges associated with selecting the most effective cooling technology (i.e., liquid nitrogen versus Joule-Thomson cooling), optimizing the cryoprobe for its diameter and active length, optimizing the cryoprocedure for the total number of cryoprobes and their layout, identifying the most effective cooling protocol (also known as *thermal history*), and monitoring the extent of treated tissue during the surgery by using ultrasound imaging and temperature sensors. While the literature is saturated with clinical reports on best cryosurgery practices [1], often sponsored by device manufacturers, an unbiased holistic approach to cryosurgery training and practice remains an unmet need.

It is the view of the author that cryosurgery technology has far surpassed cryosurgery practice, and that modern cryosurgery devices could be used at far higher levels of efficiency, effectiveness, and sophistication. Concurrent developments in computer hardware and computation techniques are likely to facilitate the next quantum leap in cryosurgery applications. One current barrier is the level of complexity of the procedure. Optimizing cryosurgery is essentially a multi-parameter optimization process, where clinical constraints are intimately

linked with technological limitations. For example, even when the cooling technology is predetermined for a particular procedure and the number of cryoprobes along with their operational parameters are chosen, the optimal cryoprobe layout alone is a complex optimization problem [20,21,22,23]. The design of the frozen region based on the configuration of cryoprobes is probably the difficult obstacle to overcome. The clinician must visualize how the frozen region formed by a specific cryoprobe layout compares to the 3D shape of the target region. Here, a suboptimal cryoprobe layout may leave areas in the target region untreated, lead to cryoinjury of healthy surrounding tissues, increase the duration of the surgical procedure, or increase the likelihood of post-cryosurgery complications; all of which affect the quality and cost of the medical treatment. The minimally invasive nature of the procedure also introduces challenges. These difficulties may further affect the quality and duration of clinical training.

A holistic approach to improve modern cryosurgery necessitates the development of computerized tools. For example, the current research team has devoted efforts in recent years to develop a computerized planning scheme for cryosurgery, based on bioheat-transfer simulations. This scheme integrates a bioheat simulation [24,25]with two planning algorithms, known as *bubble packing* [26,27] and *force-field analogy* [23,27]. The planning scheme has used an early prototype method for organ model reconstruction [28,29]. However, those prior efforts have been aimed at developing the fundamental building blocks for computerized planning, while bypassing challenges associated with current clinical practice and human-machine interface—the interface necessary for the clinician to effectively use the computer code. Challenges in human-machine interaction for a minimally invasive thermal procedure arise from the need to combine medical imaging data with thermal data, while simulating the clinical procedure in order to provide a realistic training environment.

As a part of our ongoing program to develop a computerized training tool for cryosurgery, the objective of the current study is to provide the foundation for a computerized training platform for cryosurgery, while benchmarking and integrating previously developed building blocks [23,24,25,26,27,30,29,28]. Developing the human-interface module [31] and integration of clinical data on the abnormal growth of the organ with the progress of the disease [32] are subject matters of parallel efforts. It is envisioned that future integration of the key elements developed in this and other parallel studies will enable computerized training of cryosurgery in a virtual setup.

The structure of the thesis roughly follows a typical workflow of a virtual surgical procedure. The following list details the necessary steps performed by the user and tutoring program during a virtual procedure.

1. The tutor generates the most valuable 3D US for the user to perform.
2. The user segments the US in order to extract the 3D shape of the prostate that will be used for surgical planning.
3. The tutor provides feedback on the segmentation quality based on the known ground truth shape of the organ.
4. The user generates their best cryoprobe configuration for the surgery based on the segmented prostate shape.
5. The tutor evaluates the user's plan and gives guidance on how to modify the cryoprobe layout to improve the surgical outcome. This requires that the program has the ability to generate its own high quality plan that meets a given set of clinical constraints.
6. The user virtually inserts cryoprobes into the correct locations in the 3D ultrasound image.

7. The user begins the virtual cooling of the tissue. This requires the tutor to simulate the intra-operative heat transfer and US imaging effects.
8. The user uses feedback from the simulated 3D ultrasound image along with virtual temperature sensors placed in various locations within the domain in order to track the extent of the freezing.
9. The program rates the quality of the procedure based on the amount of damage to both cancerous and healthy tissue.

The overall structure and requirements of a surgical planner are presented in Chapter 2. Chapters 3, 4, and 5 explain how the ultrasound images and heat transfer physics are simulated. Chapter 6 explains how high quality surgical plans are automatically generated, and also describes how the user generated plans and surgical procedures are evaluated. A summary and conclusion to the thesis work is presented in Chapter 7. Finally, Chapter 8 proposes the future direction and improvements to the foundation of work laid out in this thesis.

1.4 Research Objectives

The primary objective of this research is to develop efficient tools for cryosurgical training that enable a fully immersive computerized learning environment for clinicians. Toward that goal, the following contributions have been made.

1. To develop a robust cryosurgery training framework
2. To develop a method for generating synthetic ultrasound images
3. To develop an efficient implementation for an established finite difference (FD) bioheat simulation on a graphics processing unit (GPU)
4. To validate and benchmark the GPU bioheat simulation against state of the art central processing unit (CPU) based FD and finite element (FE) implementations.
5. To develop a method for simulating intraoperative US imaging artifacts for the purpose of medical training
6. To enhance the Bubble-Packing surgical planning algorithm to allow for the addition of clinically relevant constraints.

Chapter 2: Cryosurgical Training Framework

The purpose of this section is to lay out the necessary requirements of an effective cryosurgical trainer, and to present the structure of the current cryosurgical framework. Chapter 6 discusses one practical example of a prototype implementation of this framework.

2.1 Requirements

The purpose of the proposed computerized trainer is to more quickly and efficiently teach clinicians the necessary skills needed to perform cryosurgery. In order to meet this goal, the skills and knowledge gained in the virtual training environment must translate into useable skills in the operating room. To accomplish this, the training tool must give the user the correct information at the right time and provide a realistic practice environment for clinicians to hone their skills.

To begin training, the tutor must first present the user with the necessary background knowledge regarding the mechanics of the surgery. The delivery and content of this information is part of a parallel study [31]. After the user has been presented with the knowledge necessary to perform the surgery, he or she can begin performing test cases. This is the area in which tutor intelligence and realistic simulations are necessary. The requirements listed in the following paragraphs will list the components necessary for a successful cryosurgical training simulator.

At the start of the test case the tutor must intelligently select the most useful exercise for the trainer to perform. The prostate can take on a range of sizes and shapes, due to tumor growth. Additionally each patient can have a different set of physical constraints concerning clinical practice. The tutor must select the most beneficial training case for the user to perform based on previously acquired data. This is an important step because it ensures that the user is receiving

the maximum benefit from each case performed. As of the time of this writing, there is not an extensive database of prostate ultrasounds or segmented prostate surface geometry to present to users as test cases. Additionally, such a database would be prohibitively expensive to create. There are only a few small libraries of 3D prostate ultrasounds, and they are far too small to help clinicians learn all of the necessary cryosurgical skills. To overcome this obstacle the tutor must generate synthetic 3D US data from the small library of scans. This synthetic US data should correspond to the geometry that is most useful for the user to practice on. Once the user is presented with the 3D US image, they must segment the target shape. Then, the tutor must compare the user's segmented shape to the "ground truth" model the US was synthesized from. Next the user begins planning the surgery and selecting the configuration of the cryoprobes. An example of the planning stage in a prototype prostate cryosurgical system is shown in Figure 2-1.

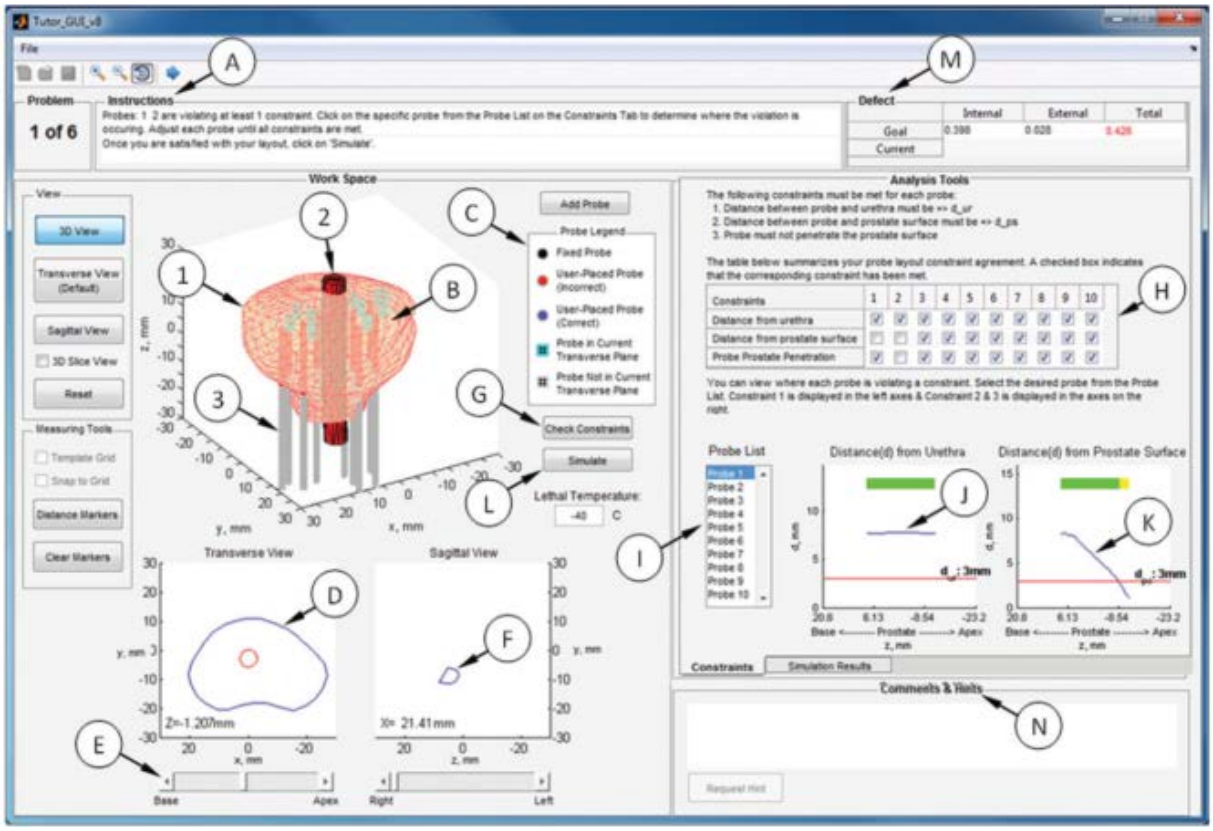

Figure 2-1: A representative screen of the cryosurgery ITS used to evaluate rules and constrains in a tested cryoprobe layout: (A) instructions panel; (B) geometrical presentation of the prostate to be treated, where (1) is the prostate capsule, (2) is the urethral warmer, and (3) is a cryoprobe; (C) color-coded scheme for probes at various stages of placement; (D) a transverse cross section of the prostate and urethra, with an sliding bar (E) for sagittal location; (F) a sagittal cross section of the prostate, with adjustable transverse location with the sliding bar below; (G) a button to request the trainer for a constraints-violation check; (H) table summarizing all probe violations; (I) probe-selection list for detailed analysis; (J) and (K) are distance tools, where the red line represents the distance constraint, blue line represents measured distance by the training software, green corresponds to the portion of the active surface of the cryoprobe which does not violate any constraints, yellow bar corresponds to a constraint-violating portion of the probe; (L) a button to request the trainer for a bioheat simulation of the current layout; (M) table displaying the total defect value results; and (N) is an area to for the trainer to provide comments and hints after an execution of a simulation. The viewing direction with trans-rectal ultrasound transducer during an actual prostate cryosurgery case can be switched between D and F **[33]**

Then the tutor generates its own optimal plan and gives feedback on how to improve the user's plan. After the cryoprobe configuration is planned, the intraoperative simulation begins. Here the user will use simulated US images with coupled heat transfer effects in order to virtually place

the probes in the tissue and monitor the frozen region progress. The user should use a combination of simulated US artifacts as well as virtual temperature sensor readings to accomplish this. Finally, the system should be able to output an overall score for the virtual procedure which corresponds to the expected clinical outcome [32]. An example of the verification stage in a prototype prostate cryosurgical system is shown in Figure 2-2.

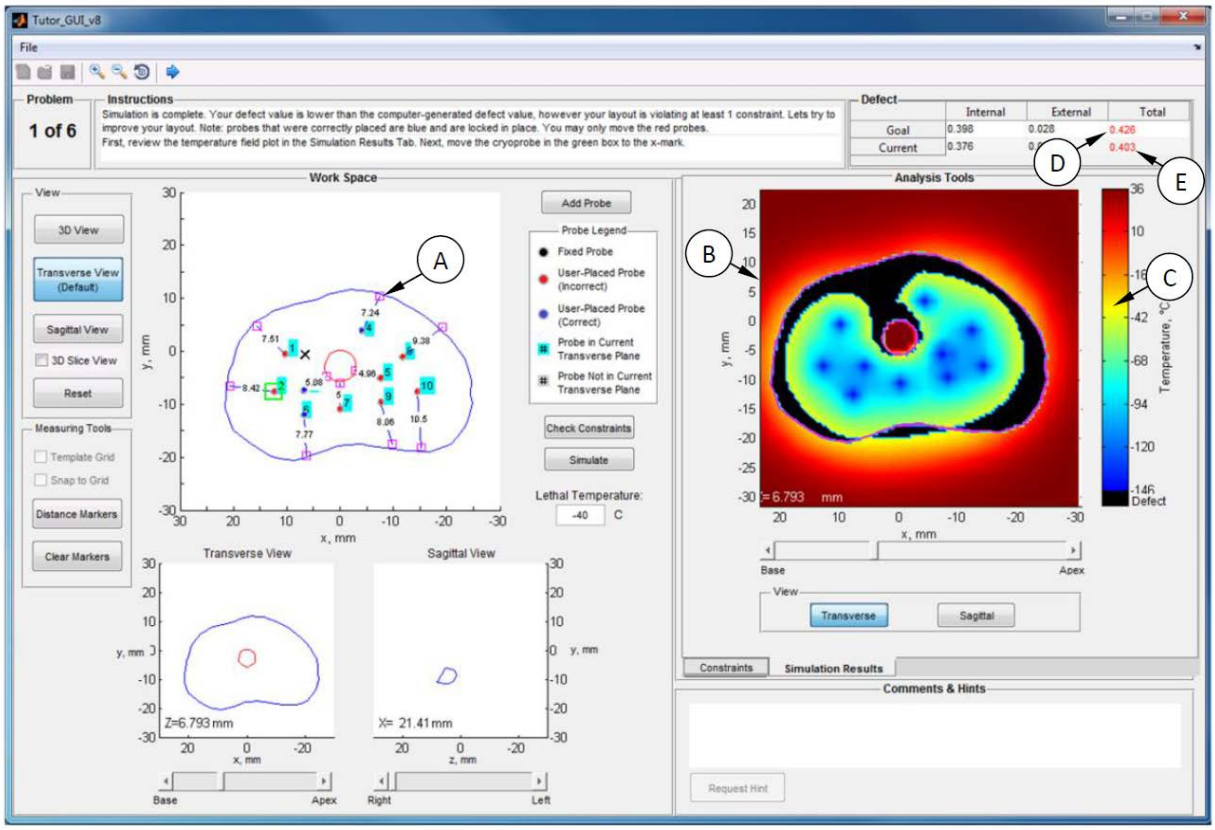

Figure 2-2: A representative screen of the cryosurgery ITS to evaluate rules and constrains in a tested cryoprobe layout: (A) prostate contour with user-selected distance measurements; (B) temperature map presented at the end of a cryosurgery simulation, where pink represents the prostate contour, black represents defect region, and the red-blue spectrum is correlated with temperature according to color scale (C); (D) goal defect, which is equal to the defect value of the optimized, computer-generated cryoprobe layout; and, (E) total defect value of the most recent trainee planning **[33]**.

2.2 Structure

With reference to Figure 2-3, the proposed trainer in the current study is composed of a graphical user interface (GUI), which accesses three additional modules: Tutor, Simulator, and Database. The GUI is the portion of the program that presents the user with information, and accepts input. The GUI is driven by the other three modules. Its purpose is to keep the other modules well separated and concise, and to eliminate unnecessary coupling between tutor system logic and algorithmic implementation. The Tutor contains much of the logic on how workflows within the program are executed. It is the tutor's job to present the user with the most suited sample surgery cases and relevant information. Additionally, it allows the user to select a suggested cryoprobe layout, and then automatically generates an optimal layout, and provides feedback on how the user could improve performance. The Tutor module uses the Simulator to accurately create a virtual procedure based on relevant data selected from the Database. The Simulator is capable of simulating both the bioheat transfer process during cryosurgery and the clinical operation via US imaging guidance. The Simulator returns the thermal history in the target region to the Tutor, in order to evaluate the effectiveness of the specific surgical case. Additionally, the Simulator evaluates the outcome of a particular surgical setup. This functionality is contained in the Simulator because it is used as a stopping criterion to end the simulation. Following an optimization scheme, the Tutor may execute the Simulator multiple times with variable input datasets, until a preferred simulated protocol is identified. At the end of each simulation, the thermal history is stored in the Database with the evaluation of results by the Tutor.

Examples of the data stored in the Database in conjunction with a particular test case are: ultrasound voxel data of the organ to be treated, volumetric prostate and urethra models, thermal history of simulated procedure, history of cryoprobe insertion, cryoprobe operational parameters,

and the planning objectives for the particular case study. At the end of each tutoring session, the new data generated during the session is added to the Database. Database management, exchange with the Tutor and Simulator, presentation, and interaction with the trainee is the subject of a parallel study [31]. This effort is done in concert with parallel studies, to ensure compatibility and efficiency in data sharing, all internally-controlled.

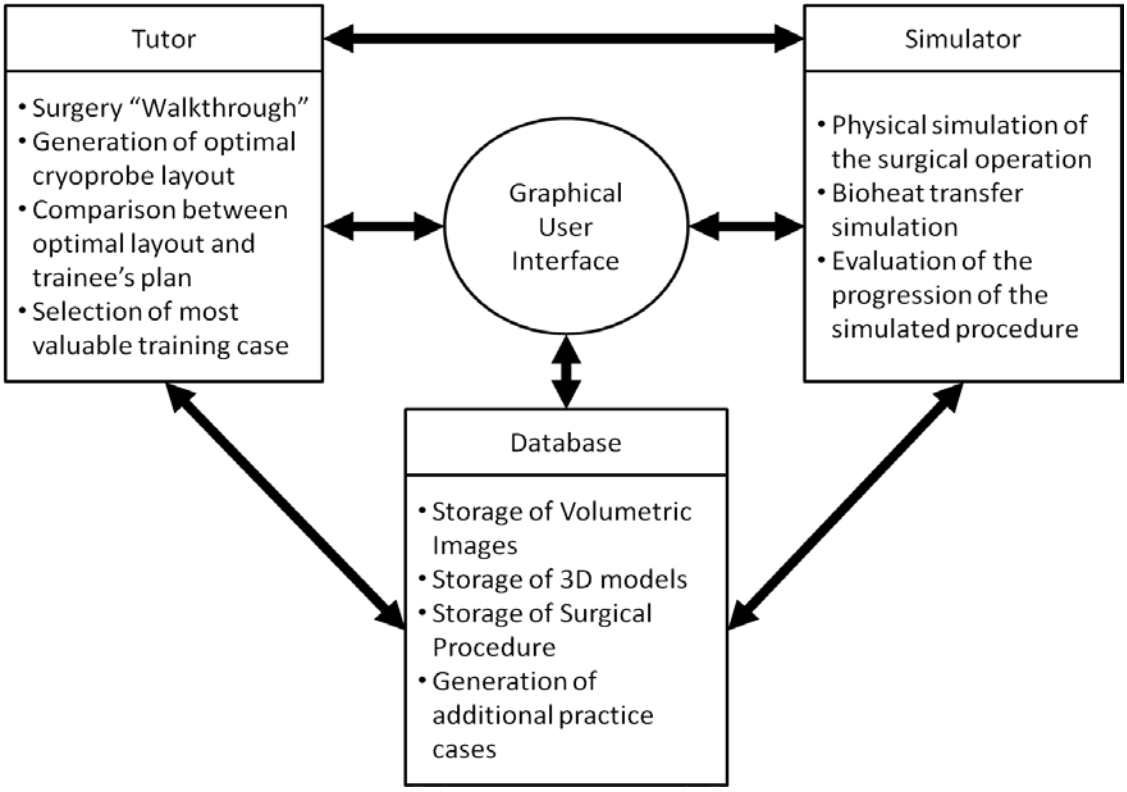

Figure 2-3: Structural design of the cryosurgery training code **[34]**

Chapter 3: An Efficient Implementation of Cryosurgical Bioheat Simulations

For the purposes of computerized training of cryosurgery, a fast and efficient method to carry out cryosurgically relevant bioheat simulations is of utmost importance. The heat transfer simulation is used by many different portions of the tutor. As a result it is one of the most important components to optimize. First and foremost, the bioheat simulator rates the quality of a user plan. In order for the user to improve their planning abilities the simulator must provide rapid feedback as the user iterates to the desired solution. Next, during the surgery the bioheat simulator is used in conjunction with the US artifacts simulator to simulate the effects of temperature on tissue in the resulting US image. Finally, many advanced planning algorithms such as the "Force-Field Analogy" require many iterations of full bioheat simulations to run in order to produce optimal plans [23]. These methods typically require a long runtime. As a result, more approximate methods such as "Bubble-Packing" are often used when runtime is a critical factor [30]. If significant speedups to the heat transfer simulation are made, the applicability of these more accurate yet computationally expensive planning methods could be broadened.

3.1 Previous Work on Bioheat Simulations

An initial implementation of a multiple grid size, multiple time-step size finite difference (FD) algorithm was created by previous members of the Biothermal Technology Laboratory and the Computational Engineering and Robotics Laboratory [25]. This scheme was specifically designed to simulate cryosurgery. The goal of their research was to develop a numerical scheme that reduces the overall number of numerical operations needed to perform a given simulation. Since the introduction of the technique, additional work was performed on optimizing and

parallelizing the implementation on the CPU [32]. However, the previous work lacked fair comparisons to the commonly used Finite Element (FE) method, which is the "gold standard" technique for heat transfer simulations. The current study addresses this issue by implementing a FE simulator and benchmarking it against the developed FD simulator. Although the FD method was previously optimized on the CPU, there are other more suitable platforms to perform the computation on. The current study implements the algorithm on the GPU and is able to achieve significant speedups. For completeness of presentation, the mathematical formulation of the FD method is presented in Section 3.1.1.

3.1.1 Mathematical Formulation

Bioheat transfer in this study is modeled with the classic bioheat equation [34]:

$$C\frac{\partial T}{\partial t} = \nabla \cdot (k\nabla T) + \dot{w}_b C_b (T_b - T) + \dot{q}_{met} \tag{3-1}$$

where C is the volumetric specific heat of tissue, T the temperature, t the time, k the thermal conductivity of the tissue, $\dot{w}_b$ the blood perfusion rate (ml_{blood}/ml_{tissue} per second), C_b the specific heat of blood, T_b the blood temperature entering the treated area, and $\dot{q}_{met}$ is the metabolic heat generation. Table 3-1 lists values of the thermophysical properties used for benchmarking in the current computerized investigation. Note that the functional behavior of the specific heat presented in Table 3-1 follows the enthalpy approach [34], where the latent heat effect is implicitly expressed in the form of an effective specific heat within the phase transition temperature range.

Table 3-1: Representative thermophysical properties of biological tissues used in the current study, where T is given in degrees Kelvin **[25]**

Thermophysical Property	Value	
Thermal conductivity k, (W/m K)	0.5	273 K < T
	15.98 - 0.0567 × T	251 K < T < 273 K
	$1005 \times T^{-1.15}$	T < 251 K
Volumetric specific heat, C (MJ/m^3 K)	3.6	273 K < T
	880 - 3.21 × T	265 K < T < 273 K
	2.017 × T - 505.3	251 K < T < 265 K
	0.00415 × T	T < 251 K
Blood perfusion, w_bC_b (kW/m^3 K)	40	273 K < T
	0	T $\leq$ 273 K
Metabolic heat generation, $\dot{q}_{met}$ (kW/m^3 K)	0	

The validity and mathematical consistency of the classical bioheat equation has drawn a lot of attention in the literature of the past six decades, with greater interest around the 1970s and 1980s [35,36]. Despite the above controversy, the classical bioheat equation is deemed adequate for the current study, whereas more sophisticated mathematical models may not warrant higher accuracy for cryosurgery computation but will involve greater mathematical complications [37,38]. Typical to bioheat transfer simulations of cryosurgery, the heating effect resulting from blood perfusion is far more significant than the metabolic heat generation, with the latter neglected in the current study [39].

Further note that the dependency of blood perfusion on temperature may represent a more complex behavior than the step-like change described in Table 3-1. While the specific behavior

may be unknown for the case of freezing in the prostate, Rabin and Shitzer [39] have already demonstrated that the overall effect of blood perfusion on the size of the frozen region is measured in only a few percent in the absence of major blood vessels, and as a result of an extremely high blood perfusion rate. Given these prior findings, the simplistic temperature-dependent blood perfusion selected is assumed adequate to predict the size of the frozen region in the current study, which is the monitored parameter via medical imaging during prostate cryosurgery.

An efficient numerical scheme to solve Equation 3-1 has been published previously [25], and is presented here in brief for the completeness of presentation. The numerical scheme is based on a finite differences (FD) approach, using a variable grid size and grid-dependent time intervals, which is well suited for parallel computation [32]. The variable grid size is used in order to reduce the computational cost, where a fine grid is only necessary in regions with steep thermal gradients, such as those around the urethral warmer and cryoprobes, while a coarse grid is used in the rest of the domain. Since the time interval to ensure stability is proportional to the typical grid size to the second power, an area with a coarse grid permits a much larger time step.

The characteristic FD equation for solving Equation 3-1 is shown below in Equation 3-2 [39]:

$$T_{i,j,k}^{p+1} = \frac{\Delta t}{\Delta V_{i,j,k}\left[C_{i,j,k} + (\dot{w}_b C_b)_{i,j,k}\Delta t\right]} \sum_{l,m,n} \frac{T_{l,m,n}^{p} - T_{i,j,k}^{p}}{R_{l,m,n-i,j,k}} + \frac{\Delta t\left[(\dot{w}_b C_b)_{i,j,k} T_b + (\dot{q}_{met})_{i,j,k} + q_{input}/\Delta V\right] + C_{i,j,k} T_{i,j,k}^{p}}{C_{i,j,k} + (\dot{w}_b C_b)_{i,j,k}\Delta t} \tag{3-2}$$

where *i*, *j*, *k* are the indices of the grid point under consideration, the indices *l*, *m*, *n* represent its neighboring grid points, *p* is the time level, *V* is the element volume associated with the grid point under consideration, and *R* is the thermal resistance to heat transfer by conduction between

grid point *i*, *j*, *k* and its neighbor *l*, *m*, *n*. The thermal resistance to heat transfer by conduction in a Cartesian geometry is given by Equation 3-3:

$$R_{l,m,n-i,j,k} = \left[\frac{L}{2kA}\right]_{l,m,n} + \left[\frac{L}{2kA}\right]_{i,j,k} \tag{3-3}$$

where L is the space interval in the direction of heat flow, and A is the representative cross-sectional area perpendicular to the same direction.

Figure 3-1 schematically illustrates a 2D domain, representative of a cross-section during prostate cryosurgery—the focus of the current line of research. For demonstration purposes, Figure 3-1 includes two grid sizes: a general coarse grid and a fine grid around the cryoprobes and the urethra; the urethra runs through the prostate and is warmed by a special heater during cryosurgery. Figure 3-1 also illustrates the thermal resistance network around a fine grid point and at a transition area between fine and coarse grid. Figure 3-1 presents a coarse-to-fine grid ratio of 1:3, which corresponds to a grid point ratio of 1:9 in 2D. These ratios illustrate the potential run time reduction by varying grid size. Consistent with [25], the simulated domain is assumed much bigger than the frozen region, such that the boundary condition represent constant temperature throughout the simulation—equal to the core-body temperature.

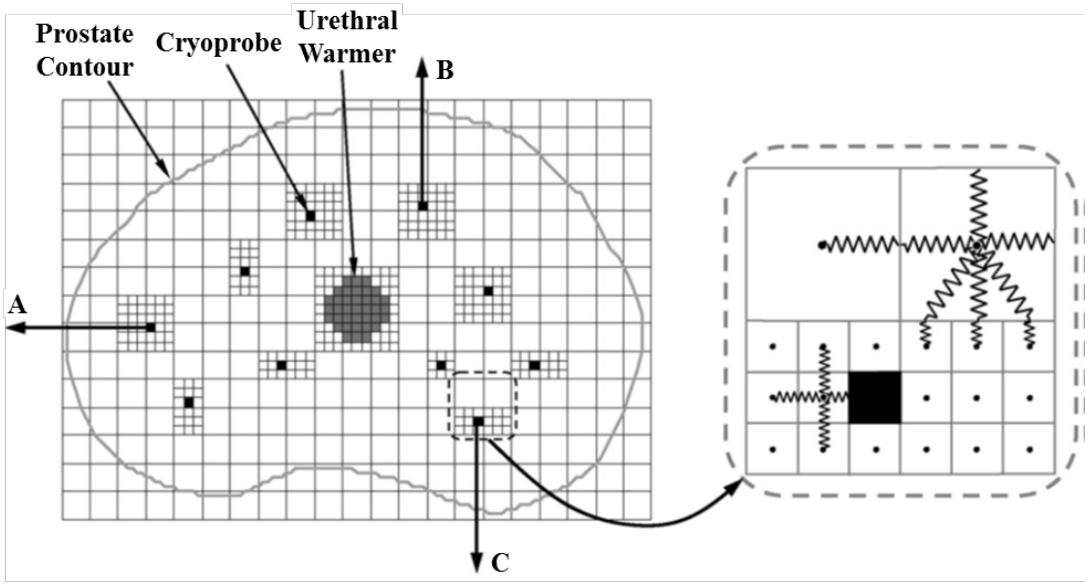

Figure 3-1: Schematic illustration of a variable grid of 1:3 ratio in the 2D case, representative of a prostate cryosurgery, and the thermal resistance network used for numerical simulations (adopted with permission from **[25]**).

The stability criterion for applying Equation 3-2 is given by Equation 3-4 [25]:

$$\Delta t \leq \left[\frac{(C\Delta V)_{i,j,k}}{\sum_{l,m,n} 1/R_{l,m,n-i,j,k}}\right] \tag{3-4}$$

In the current study, the maximum allowable time interval for the finest grid, Δt_{fine}, is evaluated first, to be used in the finest grid regions. Next, the maximum allowable time interval for the coarse grid is calculated, Δt_{max}, where the time interval for the coarse grid, Δt_{coarse}, is selected as the product of the truncated ratio $\Delta t_{max}/\Delta t_{fine}$ and Δt_{fine}.

With reference to Figure 3-2, the cryosurgical simulation is composed of a main loop that marches in time subject to six key operations: (i) field-properties update, where the boundary conditions and material properties are recalculated during each computation cycle, based on the corresponding temperature field; (ii) fine-grid temperature calculations subject to small time step intervals (calculated every loop cycle); (iii) coarse-grid temperature calculations subject to a

corresponding larger time step intervals (in practice calculated only every *m* loop cycles, where *m* is the ratio of time step intervals); (iv) interpolation of the irregular simulation grid into a regular fine grid in preparation for the next step; (v) calculation of the defect field and its integrated value; and, (vi) periodic data to disk for logging and further thermal analyses. Only operations (i) and (ii) are executed every time main loop cycle, where the other operations are performed periodically, based on numerical stability criteria and operational parameters. For this purpose, special tests are scheduled to indicate whether the execution of operation (iii) through (vi) is due in the current time step.

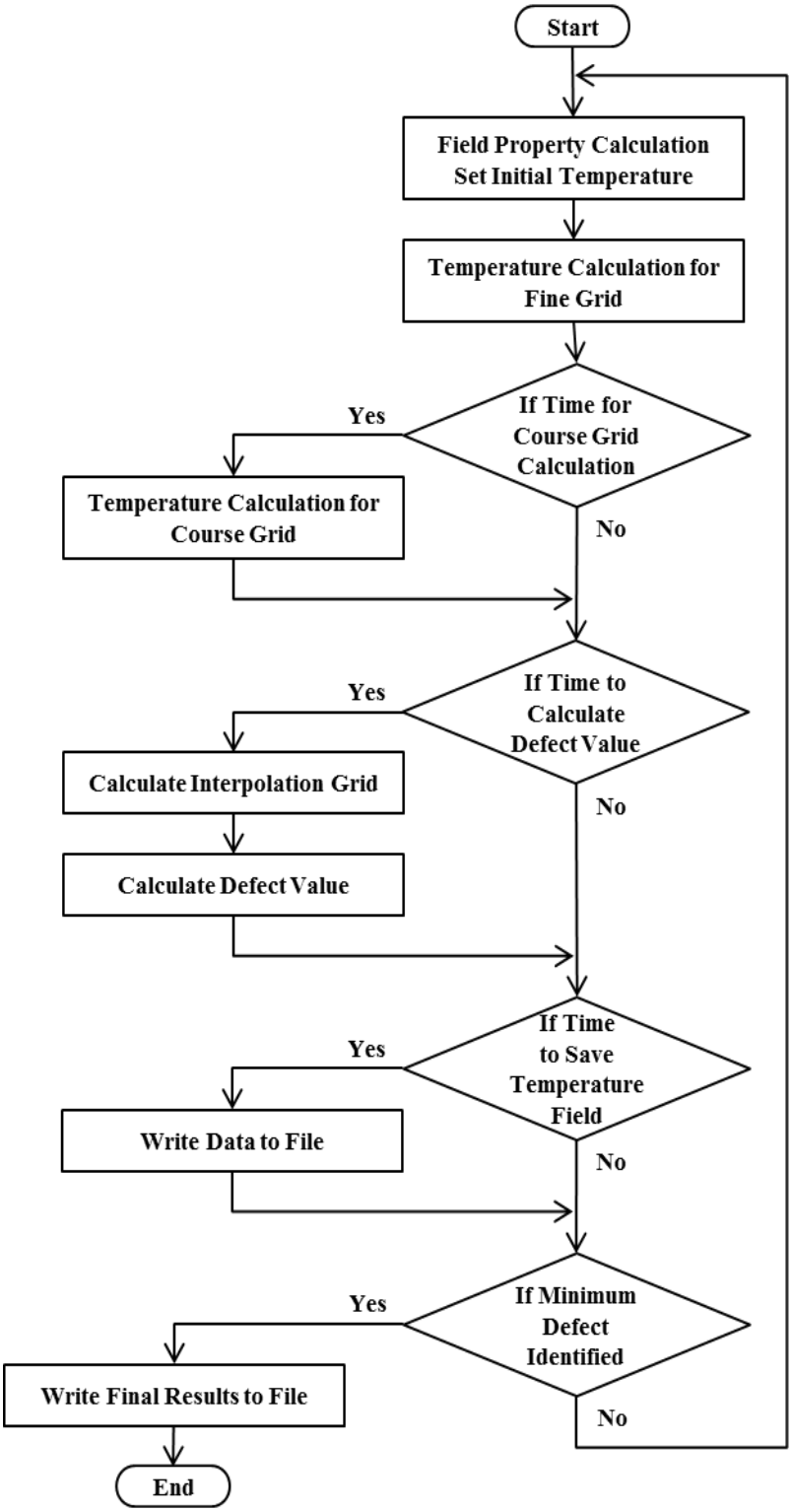

Figure 3-2: A flow chart representing the cryosurgery simulation framework **[34]**

3.2 Finite Element Cryosurgical Simulator

Conceptually, at least two approaches could be applied for bioheat simulations, using either finite-elements (FE) analysis or finite-difference (FD) scheme, where discussions about the preferred approach attract significant attention in professional meetings [40,41,42]. When an FE approach is taken, one could use a commercial code which may guarantee robustness and user friendliness [43]. Equally important is efficiency in calculations and short runtime, where an FD scheme may be of superior performance. While the application of a particular numerical scheme may simply be a matter of personal preference, the application of commercial software versus a proprietary code may carry further implications on the cost and complexity of development. From considerations of cost, efficiency, availability, development time, user friendliness, and robustness, it appears to the author that a commercial code would be a better choice for the current development, should the FE approach be taken. By contrast, developing a proprietary code would be required should the FD approach be taken. Of course, the opinion of other developers in the field may vary based on their own experience and record of achievements, and they may wish to explore additional methods, should they be challenged with the task of developing competitive computerized platforms. It was therefore warranted that both simulation approaches be investigated in the current study.

For the purpose of development in the current study, two bioheat transfer simulation cores were benchmarked, using the FE commercial code ANSYS [43] and a proprietary FD code presented previously [25]. Table 3-2 lists key comparison points for those alternatives, where more discussion on the pros and cons of these applications is provided in the Discussion section. While some of the comparison criteria are unique to the current selection, many others are quite general. It is acknowledged that other FE commercial codes could be used, but comparison of the various commercial codes is beyond the scope of the current study. Nevertheless, based on the

author's experience and the literature, ANSYS would serve the objectives of this comparison adequately. It is further acknowledged that other FD schemes could also be used for heat transfer simulations [44]. Nevertheless, the scheme presented by Rossi et al. [25] has already been demonstrated superior for the purpose of cryosurgery simulations. This FD scheme has been further verified against experimental data on gelatin solution [24,26], based on 24 independent experiments in eight representative cryoprobe layouts (n=3). Results of that study indicated an average uncertainty of 0.4 mm in freezing front location, and a mismatch of less than 5% between the simulated and the measured frozen region areas (average mismatch value of 2.9% from all experiments). By comparison, identifying the freezing front location by means of medical imaging is estimated to be not better than 1 mm. More information about the expected level of certainty during a clinical procedure is included in Results and Discussion.

With the above acknowledgements in mind, it appears that the selection between the FE and the FD approaches to develop the bioheat transfer simulation core is not a trivial decision. This decision calls for benchmarking the two schemes as discussed below.

Table 3-2: Comparison of a FE simulator based on ANSYS [43] and a proprietary FD simulator [25]. As presented in **[34]**.

Comparison Criteria	FE-Based Simulator		FD-Based Simulator	
	Pros	**Cons**	**Pros**	**Cons**
Geometrical representation of complex shapes	Natural conformity of grid points to curved surfaces			Difficult with the more intuitive orthogonal grid system
Simulation of the classical bioheat equation		Requires a special operations to account for the heating effect due to blood perfusion	Similar difficulty to the ordinary conduction problem; blood perfusion increases system stability [15]	
Computation cost at each time step		High, requires matrix solvers and high number of operations	Low for an explicit scheme, straightforward calculations	
Memory resources		High demand, matrices size is orders of magnitude larger than the number of grid points	Low demand, matrices size and number of grid points are of the same order	
Defect region calculations		Very high cost	Low, can be calculated simultaneously	
Time step interval for bioheat simulation	Relatively long	Stability does not guaranty convergence nor consistency; have to be evaluated iteratively	Stability is a necessary and sufficient condition for convergence and consistency	Explicit formulation requires very small time steps to ensure stability
Computational Complexity	Linear with numerical grid points		Linear with numerical grid points	
Pre-processing & Post-processing		Complex due to grid points registration	Trivial with an orthogonal grid system	
Scalability	Parallel kernels possible	Bottlenecks due to matrices operation	Trivial for an explicit scheme, no bottlenecks	
Availability of commercial codes for heat transfer simulations	A selection is available			None
User friendliness	High when an FE commercial code is used by a proficient user with technical background	Low when a commercial code is used by a novice user with no technical background	High when a proprietary code is tailored for a novice user from a medical background	Impractical without developing a tailored system

3.2.1 Finite Element Simulator Structure

The simulator framework must be modular in nature, to enable data transfer with the other components of the training software package Figure 3-2, and to integrate alternative heat transfer simulation cores. As a requirement, the simulator must be able to evaluate defect values through the process of simulation. A schematic illustration of the heat transfer simulation framework is presented in Figure 3-3. The description below highlights the FE implementation with ANSYS, where source code of the simulation core is unavailable. With reference to Figure 3-3, the simulating core is naturally subdivided into: pre-processing, computing, post-processing, and a main control loop. In the pre-processing phase, three input files are read into the simulator: (i) ".sim" which contains the simulation settings, such as material properties, operational parameters, and cryoprobe layout; (ii) ".csv" which contains the numerical grid; and (iii) ".agf" which contains the 3D surface data of the target region—the prostate and urethra in the current study.

The simulator core encapsulates two time cycles, the first is the sequence of time steps for progression of the numerical simulation, and the second is the periodic evaluation of the defect value. While the cost of defect evaluation is negligible for an FD scheme, and can be done in parallel, a FE scheme necessitates a post-processing phase for defect evaluation, which may create a bottleneck in calculations. Using ANSYS, the second cycle is performed by analyzing the results file ".rst" at the end of the cycle, following the execution of the commands listed in ".lgw". In either simulator, heat transfer simulation progresses until a minimum defect is identified. Once the simulation terminates a ".srt" results file is generated, containing the temperature and defect history of the simulation domain, to be used by the Trainer and stored in the Database Figure 3-2. While the above process of file sharing and data transfer may appear cumbersome and computationally expensive, it is necessary in the development of the post-

processor functionality when integrating a commercial code structure.

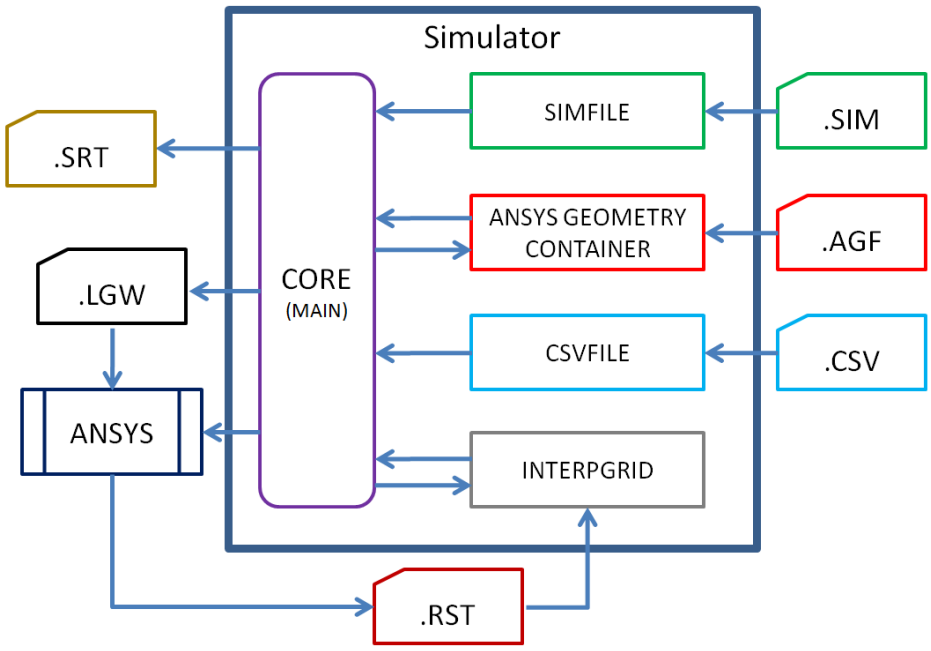

Figure 3-3: Implementation of the FE bioheat transfer simulation core with ANSYS. Data transfer scheme is made identical for the FD simulator for the purpose of benchmarking **[34]**.

3.2.2 Finite Element Simulator Optimization

Before benchmarking, the FE simulation core must be optimized for runtime while maintaining acceptable accuracy. Due to the large number of parameters, a sequential optimization process was performed for: (1) mesh size, (2) time step size, and (3) solver type and associated convergence tolerance. An acceptable accuracy level in all optimization steps was considered to be 2 mm in the location of the freezing front, which corresponds to about 3% change in frozen region volume in most cases. Such an error in freezing front location coincides with the uncertainty range with medical imaging—the sum of imaging resolution of 1 mm and 1 mm positional error in the process of shape reconstruction. The accuracy in freezing front location was evaluated by a comparison with a base case, which was selected as a converged

solution that matched the results of the FD simulation core (discussed in [25] and compared with experimental results in [24]). In addition, the average temperature difference within the frozen region was also investigated to ensure the convergence of the temperature-field. The optimization test cases were based on fourteen cryoprobes in a layout optimized with bubble packing, for a realistic cancerous prostate geometry. The test case was run until minimum defect had occurred for each parameter choice for identical cryoprobe layouts.

FE optimization for mesh size: In order to select the optimal mesh distribution, mesh sizes are set on various control surfaces within the model, then the mesh size is interpolated within the volume between the surfaces. Selected mesh surfaces include the active cooling portion of the cryoprobes (1 mm), the heated area of the urethral warmer (3 mm), the prostate contour (5 mm), and the external boundary of the simulated domain (10 mm); those numerical values are scaled with the respective temperature gradients. The mesh convergence study was carried out by proportionally increasing the mesh on all of the above surfaces, until the maximum difference in the location of the lethal temperature isotherm exceeded one millimeter from the converged solution. This convergence criteria was chosen because it is the most clinically relevant metric for prostate cryosurgery. A tetrahedral mesh was found to best populate the domain defined by those surfaces.

FE optimization for time step: One minute is deemed the largest allowable interval between simulation results for effective planning from defect-calculation considerations. As specified above, the thermal history of the cryoprobes consists of a rapid cooling stage and cryogenic temperature hold stage. Consistently, the freezing front grows at a faster rate at the beginning of the procedure. The two-stage cooling protocol was matched with a simulation scheme based on two time step intervals. Figure 3-4 displays selected results of the time step convergence study, where the average temperature difference within the frozen region (temperatures below 0°C) is

calculated with reference to the FD solution. It can be seen from Figure 3-4 that the average temperature difference is initially large, due to the small volume of the frozen region and the high temperature gradients surrounding the cryoprobes; that temperature difference monotonically decays as the simulated process approaches steady state. Since the early stage of freezing plays a lesser role in evaluating the effects of the cryoprocedure, time step intervals of 10 sec during cryoprobe cooling and 60 sec during cryogenic temperature hold period are found acceptable for the purpose of the current study. While the 60 sec interval is deemed adequate for the numerical solution, the parallel search for minimum defect necessitated intermediate values, which eventually led the investigation to also include 30 sec intervals. This illustrates a difficulty in integrating a commercial code for heat transfer simulations with parallel calculations for the quality of simulation.

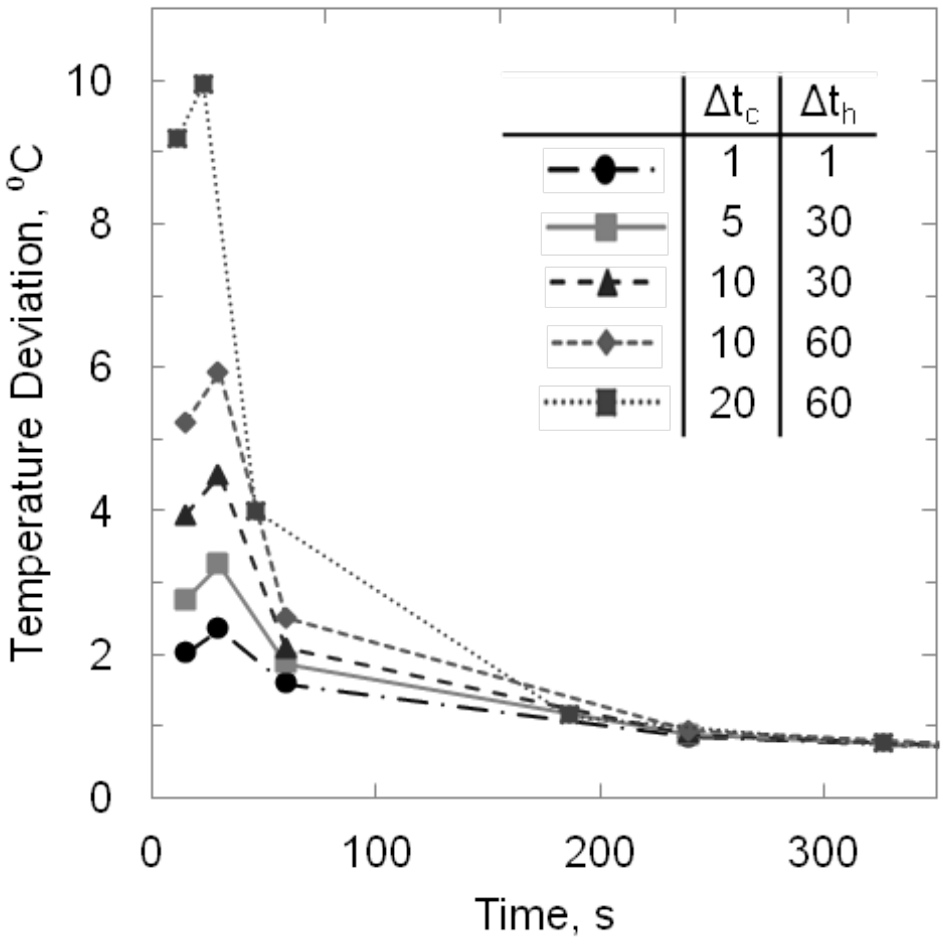

Figure 3-4: Average temperature difference in the frozen region (T<0°C) between the FE and FD simulation results, for a FD time step of 0.02 sec and various FE time intervals: Δt_c is the time step during cryoprobe cooling from 37°C to -145°C, and Δ_{th} is the time step size when the cryoprobe temperature is held at -145°C **[34]**.

FE optimization for numerical solver and convergence tolerance: For the above mesh selection and time intervals, the following solvers were tested: Sparse Solver, an Advanced Multigrid Solver, a proprietary preconditioned solver (conjugate gradient), a Jacobi Solver (conjugate gradient), and an Incomplete Cholesky Conjugate Gradient Solver (ICCGS). Most effective was found the ICCGS, running at least 8% faster than any other tested solver (infinity norm=10^{-6} for Newton-Raphson tolerance).

Computational Complexity: As could be expected from theoretical analysis of the FE scheme, Figure 3-5 displays almost linear increase in complexity with the increased number of nodes and, independently, with the increased number of cryoprobes (i.e., increased surface area of fine mesh). The deviation from linearity is resulted from changes in the cryoprobe layout when the number of cryoprobes varies in the same selected domain. A linear computational complexity for the FD scheme has been demonstrated previously with the increasing cryoprobes number [25].

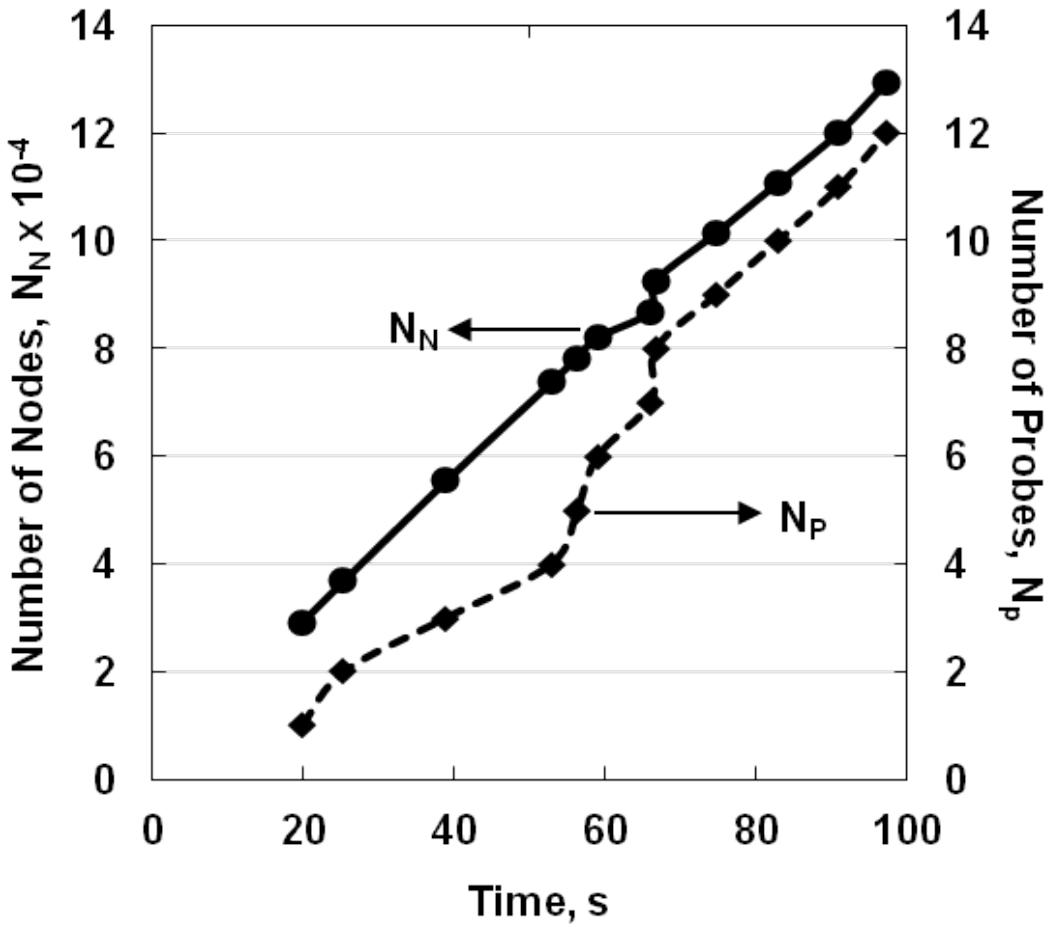

Figure 3-5: Computation time of the FE simulator as a function of the number of cryoprobes NP, and for the corresponding number of nodes NN using optimal mesh sizes, for cryoprobe layouts determined using bubble packing [34].

Scalability: For the case of 30 sec FE time step, the FD simulator is about 6.7 to 8.3 times faster than the FE simulator Table 3-3. Given the fact that the FD simulator actually advances in time steps in the order of 10^{-2} sec, this result is extraordinary. The reason for the superiority of the FD runtime is in its explicit formulation, where the short time interval is traded with simplicity in calculation. With regard to defect calculations, the 30 sec intervals of the FE simulator may lead to unnecessary simulation time, whereas the small time intervals of the FD simulator provide continues information of the defect region progression.

Multi-Core Processing: While one could expect the simulation runtime to be inversely proportional to the number of computation cores in parallel processing, increasing the number of computation cores from 1 to 4 reduces simulation runtime by a factor of 2 for both the FD and FE schemes Table 3-3, but from different reasons (both codes are specifically written for parallel processing). For the FD simulator, suboptimal acceleration is primarily due to the i7 chip architecture, where all the cores share the same amount of L3 cache in parallel computing as one core would use in a single core run. For the FE simulator, the suboptimal acceleration is primarily due to the sequential steps necessary for the complex pre- and post-processing. Due to the explicit nature of the FD scheme, it is an excellent candidate for parallel computing, providing that the amount of L3 cache is large enough. It is concluded that with increased cache size, the FD scheme runtime will become inversely scaled with the number of cores, which represents a significant advantage with the current trend of increased number of computation cores on personal computers (six and eight cores are already widely available on mid-range machines).

Geometry representation: An area in which the FE simulator is superior to the FD simulator is that of geometrical modeling fidelity. The FD simulator is restricted to placing nodes on a rectilinear grid, which necessitates a finer discretization to represent the non-rectangular

geometry of the prostate. By contrast, the FE method has fewer restrictions on node placement and, thereby, the pre-processor of the commercial code is able to represent a complex shape with fewer nodes. For the current test case, approximately 95,000 nodes were needed for the FD simulator as opposed to 80,000 nodes for the FE simulator to obtain an accurate solution. Despite the apparent superior geometrical representation in the FE simulations, the FD simulator displayed a higher accuracy in temperature field calculations, which was obtained in a shorter runtime.

Key development objectives for the cryosurgery trainer are short runtime on a personal computer and user friendliness in operation, in order to make the trainer application practical for a large community of novice users. Given the computation cost of the bioheat transfer simulation with phase change, meeting this objective requires innovation in code writing and selection of an efficient computer framework. Given the computation cost and recent developments in computer hardware, it appears that we are currently at the verge of meeting those development objectives with the FD simulation core.

3.2.3 Finite Element Simulator Benchmark

Benchmarking studies were performed on a cancerous prostate model created previously Figure 3-6, using 14 cryoprobes at a uniform insertion depth, with optimal layout generated by bubble packing [30]. The probe layout is further detailed in Figure 3-7 All simulations were performed on an Intel® i7-950 (four-core) machine, running at 3.07GHz, with 9GB of RAM. All heat transfer simulations were run for 255 seconds of simulated runtime, which coincides with the point of minimum defect for the FD-based code.

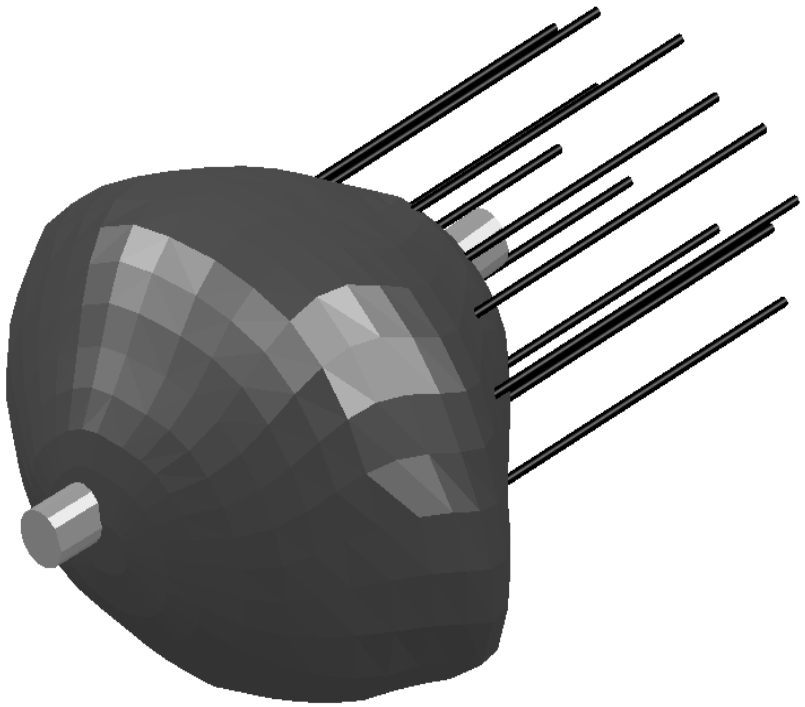

Figure 3-6: Illustration of a fourteen-cryoprobe layout in the cancerous prostate model used in the current study for benchmarking. The urethral warmer—a cylindrical heater in the form of a catheter— is designed to protect the urethra from cryoinjury and is modeled as a heated tube **[34]**.

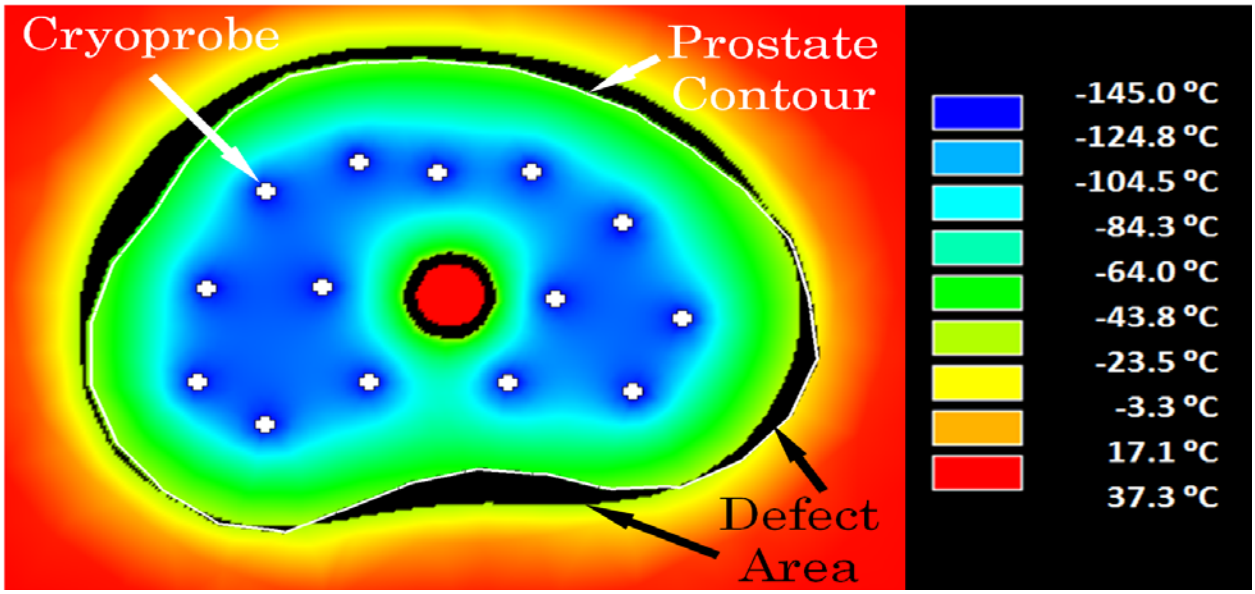

Figure 3-7: Temperature regions of interest for the purpose benchmarking in the current study **[34]**.

It can be seen from Table 3-3 that the FD and FE schemes identify different values for the time to minimum defect. This difference is due to the way that the phase-transition temperature range is handled in each scheme. Note that for both selected schemes, the freezing front location is interpolated from temperature data and it is not a parameter of the solution. It is also noted that the optimal cryoprobe layout has shown to be insensitive to the simulated time to minimum defect—computationally [25] and experimentally [24]. Based on the high certainty in predicting the freezing from location with the FD solution (0.4 mm, see Bioheat Simulation Technique, above), and given the fact that stability is a necessary and sufficient condition for convergence in the particular FD scheme [44]. Figure 3-8 displays the resulting temperature field at the point of minimum defect, where Figure 3-9 displays the evolution of the defect for the same simulation. Comparison of advantages and disadvantages in using the FE and FD simulators is listed in Table 3-2, based on the specifics of benchmarking. Furthermore, Table 3-3 lists simulation

results for parallel computation on a number of cores ranging from one to four, for the same cryosurgery case displayed in Figure 3-6.

Table 3-3: Benchmarking results of the FE and FD simulators on a representative case of a cancerous prostate model. Cryoprobe layout is based on a bubble packing method (illustrated in Fig. 4). For the FE simulator, a time step of the 60 sec represents an upper limit for convergence, whereas the 30 sec time step is deemed practical for concurrent defect region calculations. Time step for the FD simulator is taken at the stability limit **[34]**.

Simulator		FE		FD
Time step		60 sec	30 sec	.02 sec
Minimum defect value		24.5%	24.7%	24.3%
Total simulated time		255 sec	255 sec	255 sec
Simulated time to minimum defect		180 sec	199 sec	255 sec
Actual runtime	1 core	613 sec	966 sec	144 sec
	2 cores	423 sec	689 sec	83 sec
	4 cores	304 sec	521 sec	67 sec
Simulated time ratio = comp. runtime / procedure duration	1 core	236%	379%	56%
	2 cores	165%	270%	32%
	4 cores	119%	204%	26%

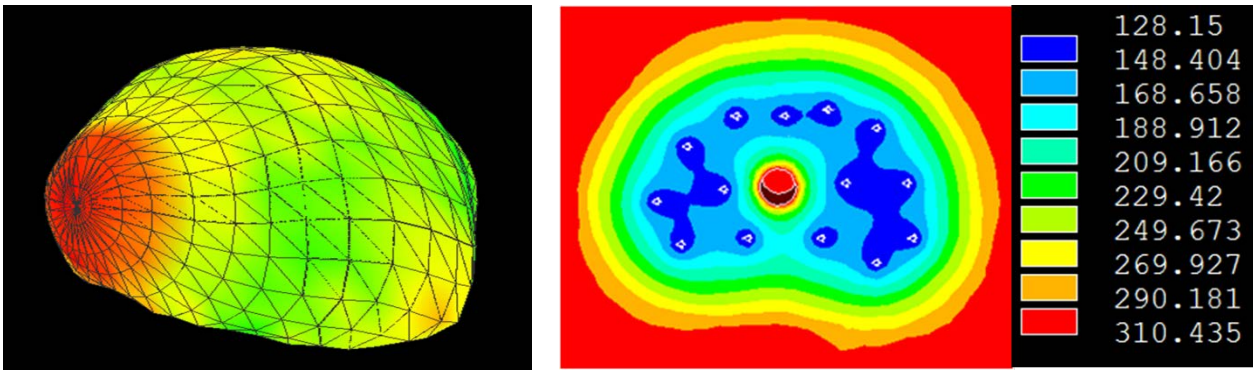

Figure 3-8: Temperature field at the point of minimum defect: (left) temperature field superimposed on the prostate outer surface; (right) cross section passing through the center of the active surface of the cryoprobes **[34]**.

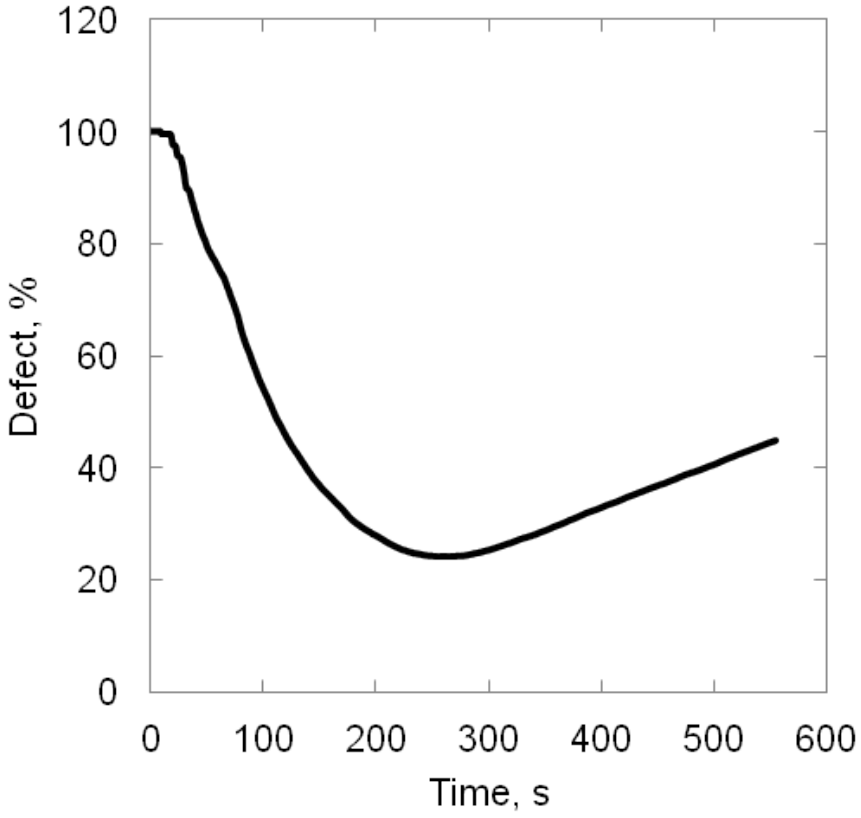

Figure 3-9: Evolution of total defect for the simulated case displayed in Figure 3-6 **[34]**.

3.3 GPU Based Finite Difference Implementation

The numerical scheme previously presented was parallelized and optimized for the CPU [32]. However, significant changes to the implementation were necessary to achieve optimal results on the GPU. The following sub-sections will give the necessary background for today's GPU architecture as well as explain the structure and optimization process of the numerical scheme. The GPU implementation and optimization have been conducted in an iterative manner. Below, the key elements of GPU implementation are presented first, the key optimization steps and their relative impact on run-time acceleration are presented next, and benchmarking results against an optimized CPU implementation on selected machines are presented last. Implementation results, optimization analyses, and benchmarking in this study are based on a

cancerous prostate model created previously, using a urethral warmer and 12 cryoprobes at a uniform insertion depth, with optimal layout of the probes generated by bubble packing [30]. Figure 3-11(a) displays typical results obtained with the simulation code [32] at the point of minimum defect, presented at the largest prostate cross section. The C++ AMP language was selected for GPU implementation as a choice of practice, based on its user friendliness as well as its advantage of future graphics interoperability with DirectX.

Conceptually, the current study represents an advanced-stage development, where validation with experimental data is established in two consecutive steps. The first step has been completed in previous studies, where the applied FD numerical scheme [25] has been validated against experimental results [24]. That previous study demonstrated mismatch between the predicted freezing front location and the simulated freezing front location of less than 4.8%, with an average mismatch of 2.9% (n=24). Note that the freezing front location is the monitored parameter during minimally invasive cryosurgery. That 2.9% mismatch has been translated to 0.3 mm of disagreement in the actual freezing front location [24], which is much smaller than the resolution in medical imaging for minimally invasive cryosurgery (in the range of 1 to 2 mm for ultrasound). The second step of comparison with experimental data is presented here, where any new computational development in the current study is validated against the established FD scheme and CPU-based implementation. For example, Figure 3-11(b) displays a mismatch of up to 0.4ºC between GPU-based and CPU-based simulations of the same case, with a negligible difference in the freezing front location (less than 0.01 mm, which has no clinical meaning). A wider discussion on the propagation of uncertainty in measurements into cryosurgery simulations has been presented by Rabin [45].

3.3.1 GPU Background

CPUs and GPUs are two of the most common processor types used in computers. The CPU usually contains a few very powerful cores. CPU cores typically have a large instruction set and are optimized to perform a single task very quickly. GPUs are made up of many (tens to thousands) cores. Each core is relatively simple, has a smaller instruction set, and is much less powerful compared to a single CPU core. The primary purpose of GPUs is to calculate the color of each pixel on the screen. To perform this task, the same set of calculations must be repeated for every pixel. As a result, GPUs are optimized for computationally intensive work flows that require the same independent calculation to be performed on a large amount of data (data parallel). Despite the fact that each GPU core is much less powerful than a CPU core, the combined computational power of all of the cores on a GPU is much higher than that of a CPU. The problem of simulating heat transfer during cryosurgery is also a data parallel computation as the same independent calculation must be performed for every node in the domain within each time step [46].

As a result of the different class of problems for which CPUs and GPUs have been developed, the way in which they address memory latency, the largest bottleneck in computing, has diverged. Memory latency is the time it takes between a request for and a delivery of data from the processor's memory (RAM). Memory fetches are extremely expensive operations to perform on both the CPU and GPU, typically taking in the range of one hundred cycles. During this time the processor is idle waiting for the data due to the long latency. CPUs are designed to perform a single task very quickly. What results is a processor with a lot of fast local cache to store frequently used data and large complicated cores that can predict what memory and instructions to pre-fetch to reduce the amount of idle time. Only a relatively small percentage of the total transistors are dedicated to performing numeric computation. Because GPUs are

designed for large independent data parallel computation, chip designers were able to take a different approach to the memory latency problem. Instead of using extensive caches or complicated cores, GPUs hide memory latency with other work. While a processor is waiting for a memory fetch request to complete, there is other data parallel work to perform as the number of independent calculations to complete is orders of magnitude higher than the number of cores. The GPU is able to switch back and forth between many different calculations every time a long memory request is issued, effectively hiding it with computation. By using this strategy GPU chip designers were able to improve performance by adding more computational cores, but not necessarily by making each core more complicated. Many simplifications are made from an architectural standpoint to allow more transistors to be dedicated to computing [47]. A depiction of the difference between GPU and CPU architectures is shown below in Figure 3-10.

Figure 3-10: Comparison between CPU and GPU architecture. The portions of the figures labeled ALU (algebraic logic units) are the parts of the chips responsible for computation **[50]**.

GPUs are composed of groups of processors called multiprocessors. Each multiprocessor typically contains a group of cores (typically 32) that all operate in lock-step. These cores all execute the same instruction synchronously. If there are conditional statements in the code all of

the cores will need to traverse all of the code paths taken by the cores within the multiprocessor. For example, if a multiprocessor is executing an “if else” statement and 31 of the 32 cores executes the “if” branch then the one remaining core waits for those instructions to execute. While the “else” executes, the final core performs work while the 31 cores that executed the “if” branch wait. As a result conditional statements in GPU computing are relatively expensive and should be minimized [48].

Each multiprocessor shares a memory controller that aggregates memory requests from the individual cores. These memory controllers are extremely efficient at fetching a contiguous chunk of memory. Despite the fact that memory fetches typically overlap with other computation, many applications are still memory bound. Therefore, care should be taken in designing the data layout in order to maximize the number of contiguous memory fetches a group of processors make [49].

To further address the issues of memory latency, each multiprocessor has a small amount of shared memory cache, which can be accessed by all of the cores within the multiprocessor. It is much quicker to access this shared memory cache than it is to get the data from GPU memory. Therefore, algorithms should be designed to have cores in a multiprocessor cooperatively load and unload frequently used shared data into this cache [47].

3.3.2 GPU Implementation

The GPU simulation algorithm stores temperature and thermal conductivity data for each node in the simulation. Other thermal properties, like heat generation and specific heat, are calculated ad-hoc based on the nodal temperature. In addition, two matrices are used to store the nodal connectivity and thermal resistance data. The iterative optimization process in this study starts with a GPU implementation base-case, designated as *naïve*, and then progresses through

various optimizations stages to improve runtime performance.

Since most of the simulation algorithm can be modeled as a sparse-matrix vector operation (SPMV), the naïve implementation is based on a compressed sparse row (CSR) matrix format for both the matrices of nodal connectivity, thermal resistance, and interpolation. The access pattern for both of these matrices follows the scalar SPMV algorithm [47], where each thread is responsible for a row in the matrix-vector multiplication operation. In order to calculate the defect value, a reduction operation is applied to the sum of defect values from all interpolation nodes. A cascade algorithm [46] is used for the reduction, where each thread sums a fixed number of elements from the interpolation grid, each processor group sums up an intermediate defect value from all of its individual threads, and finally each processor group copies its partial sum to the CPU where the final reduction occurs. Only one of the above steps were run at a time in the naïve implementation.

3.3.3 GPU Optimization

Optimization of the GPU-based simulation scheme progressed based on a prioritized list of expected improvements in terms of runtime, with the outcome displayed in Figure 3-12:

1. *Float precision:* Converting from double-precision in the naïve implementation to float-precision for all applicable parameters dramatically reduced runtime by 44%. No significant degradation of the quality of calculated temperature field (Figure 3-11(b)), the defect field, or the time to minimum defect, was identified due to the decrease in numerical precision. As can be seen from Figure 3-11(b), the maximum mismatch between the float precision and the CPU-based implementation is 0.4ºC, which is negligible in the great majority of the field. Reducing the amount of data transferred from the global memory by half speeds up all of the steps in the heat transfer simulation. Most significantly, this

change speeds up the defect value calculation, which is the most global memory-intensive process. It should be noted that, in the case of bioheat-transfer simulations for cryosurgical training the additional resolution of the double-precision floating point representation is unnecessary. However, if one were to solve the bioheat equation with the previously described scheme with a smaller time-step size and smaller grid size, double-precision may be needed in order to prevent the buildup of numerical error.

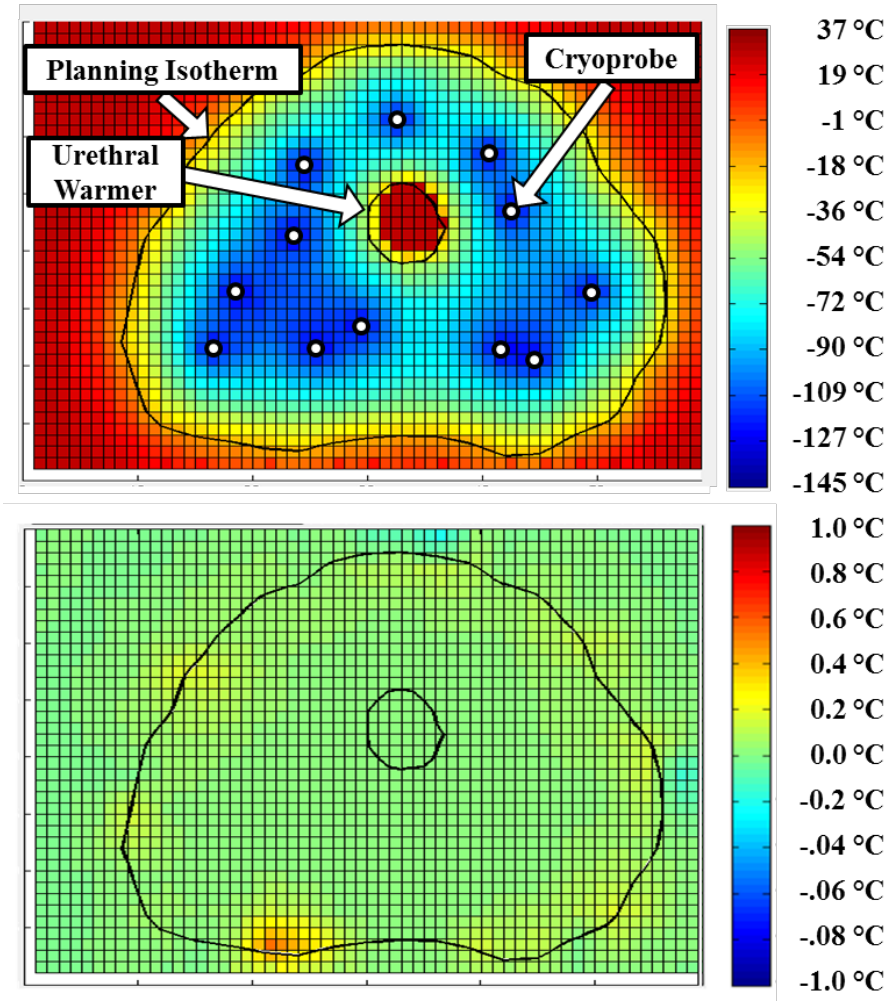

Figure 3-11: A typical simulated temperature field at the largest prostate cross section, subject to the operation of a urethral warmer and 12 cryoprobes: (top) temperature field results obtained with a CPU-based parallel computation, using double precision; (bottom) temperature difference between the CPU-based parallel simulation and the optimized GPU-based simulation. All temperature values are given in Celsius degrees **[53]**

2. *Asynchronous I/O:* Since file I/O can be performed in a separate thread on the CPU, in parallel to GPU operations, asynchronous file operations were tested in the next step of optimization, shifting this costly step to the background. Asynchronous I/O saved an additional 14% in terms of run time with reference to the float-precision implementation.
3. *Defect calculation:* Defect calculations are based on a tri-linear interpolation scheme, where each interpolated grid point is calculated based on eight neighboring nodes. This allows the use of an ELLPACK (ELL) matrix representation instead of CSR. The ELLPACK matrix more efficiently represents the sparse matrix, as it saves only the non-zero entries (eight in this case). Anytime that the ELL representation is used to store a matrix on the GPU, the non-zero entries are arranged so that the first non-zero entry in each row are laid out sequentially in the ELL array, with the pattern continuing for each additional non-zero per row. This layout significantly improves global memory coherency. In the third step of optimization, temperature interpolations and defect calculations were further combined into a single kernel, which dramatically reduced the number of global memory operations. Unfortunately, the current optimization effort did not significantly shorten runtime Figure 3-12 since the current effort covered only a small portion of the total time spent within this kernel. In practice, the majority of the computation time is devoted the scheme of defect calculations, which could not be accelerated any further. Interestingly, because of the asynchronous I/O operations, modifying this kernel actually had an adverse effect on the runtime of the field property update kernel.
4. *Texture memory:* Texture memory is a specially mapped location of global memory, which reduces lookup time and writing time. To take advantage of this special GPU feature, f such as temperature and thermal properties, was stored in texture memory. Applying texture memory directions reduced the runtime of all kernels by 28% compared to the

previous optimization stage, and overall reduction of 12% compared with the naïve implementation. In order to use texture memory with C++ AMP, one needs a 2D texture structure to contain the large data structures typical to cryosurgery simulations.

5. *Network connectivity:* While any node within a uniform grid, either fine or coarse, has connections with six neighboring nodes, more connections may develop at the interface between the fine and coarse regions. To calculate this connection network more effectively, a hybrid matrix representation was formulated as follows. The six connections within uniform grid areas are stored in the memory efficient ELLPACK matrix representation. For interface nodes (more than six neighbors), these connections are stored in a permutated CSR (PCSR) format, with rows in descending order based on the number of non-zero elements. This hybrid representation is similar to the one presented in [47], except for the non-zero entries that do not fit into the ELLPACK representation, which in turn are saved in a PCSR format rather than the coordinate (COO) format suggested in [47]. In general, the COO format saves a row number, a column number, and a value for each non-zero entry. The PCSR representation has less memory coherency than the COO format when performing sparse matrix vector-like operations. However, for the application of cryosurgery, the PCSR format significantly reduces the amount of computation compared to the COO format, since calculations of the temperature-dependent terms in the update step only need to occur once per row rather than once per non-zero element. The PCSR format does, however decrease code divergence within a processor group compared to the original CSR format, by processing rows with a similar number of non-zeroes within one group. The real acceleration due to the improved connectivity comes about by storing the majority of the connections in the network in the more efficient ELLPACK representation. Improved network connectivity as described above reduced runtime by 10% compared with

the naïve implementation.

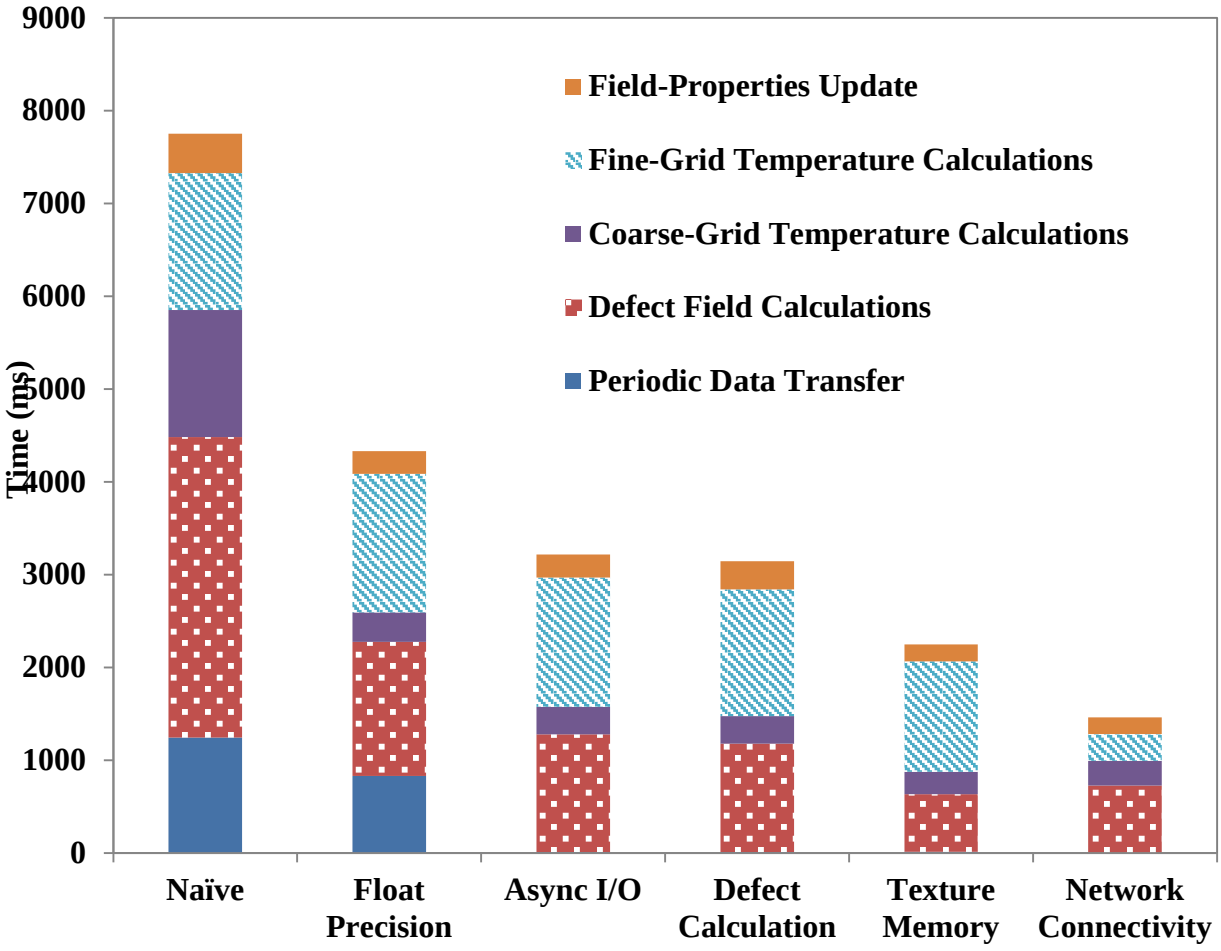

Figure 3-12: Typical runtime results of successive optimization steps for the GPU implementation, where Naïve represents the non-optimized base-case GPU implementation **[53]**.

3.3.4 GPU Benchmark

Keeping in mind the above optimization steps, the optimized GPU-based code was benchmarked against a CPU-based parallel computation code [32] on the selected computation platforms as listed in Table 3-4. . These machines were selected to represent the typical performance of a midrange laptop, a high-end home PC, a gaming PC, and a workstation PC. All heat transfer simulations were run for 201 seconds of simulated runtime, which coincides with the point at which the minimum defect was identified. The grid size used in this study is 1 mm × 1 mm × 1 mm for the fine grid and 3 mm × 3 mm × 1 mm for the course grid, resulting in a total of about 80,000 grid points in the entire 3D domain. These grid sizes were chosen in a previous study in order to reduce computational cost [24].

Table 3-4: **Computer platform specifications used for benchmarking in the current study** where the results are displayed in Figure 3-13 [53].

System	CPU	Memory	CPU Cores/Threads	GPU	Video Memory	GPU Cores
Laptop	Intel Core i7 M620	4 GB DDR2	2/4	AMD Mobility Radeon HD 5450	512 MB dedicated 2202 MB Total DDR3	80
High-end home PC	Intel Core i7 950	9 GB DDR3	4/8	AMD HD Radeon 5570	1 GB DDR3	400
Gaming PC	Intel Core i7 960	9 GB DDR3	4/8	Nvidia GeForce GTX 580	1.5 GB GDDR5	512
Workstation	Intel Core i7 870	8 GB DDR3	4/8	Nvidia Quadro 600	1GB DDR3	96

In all tested cases, the GPU-based implementation significantly outperformed the optimized CPU implementation. Simulation speed increases ranged from 3× on the laptop and up to 13× on the gaming PC, which contained the most powerful GPU. Note that the CPU-based computation has already been optimized for multi-processors [32], where the implementation of the applied FD scheme is trivial using parallel *for* loops with OpenMP. In the un-optimized CPU implementation, every node was represented as an object that contained all of the material properties and connectivity information with the surrounding nodes. This data structure was optimized by rearranging the more commonly used properties, such as conductivity, volume and specific heat together. This Data Separation allowed the implementation to more effectively

cache frequently used data and reduce the system memory load. For example, this CPU-based optimization resulted in 3.9× faster runtime than the non-optimized serial CPU implementation, when using Hyper-Threading on all 4 cores of an Intel Core i7 960. This makes the runtime of the most sophisticated GPU-based simulation about 50 times faster than the serial CPU simulation. Figure 3-11(b) displays very good agreement between the two simulation approaches.

Parametric investigation using Nvidia GeForce GTX 580 (512 GPU cores) reveals that simulation runtime is linearly proportional to the number of numerical nodes for larger problems, like the one used in the current benchmark study (about 80,000 nodes). This linear relationship holds as long as the problem is big enough to effectively use the entire GPU memory bandwidth. If additional computations are performed while memory fetches occur, much of the work can be fully hidden [46]. However, there is a problem-size threshold below which the computational performance will start to degrade, because the large number of cores will become underutilized during memory fetches. This occurs when the problem becomes smaller than about 30,000 numerical nodes for the current numerical scheme and the above computation unit.

A full-scale GPU-based simulation of cryosurgery can already be achieved in less than 2 seconds Figure 3-13. Such a short time can be considered instantaneous for the purpose of providing an early prediction of the cryoprocedures outcome when compared with the actual cryosurgical procedure, which is measured in several minutes or longer. This capability of rapid simulations advances the discussion about cryosurgery computation from merely analyzing a special case to analyzing a battery of cases in effort to improve decision making, which is an integral part of any computerized planning [27,23,50], computerized training [32,31], or an intelligent approach to control of the procedure. Such a battery of cases may contain a varying set of problems in the search for an optimum, or a set of “what-if” scenarios for the benefit of

training and medical education.

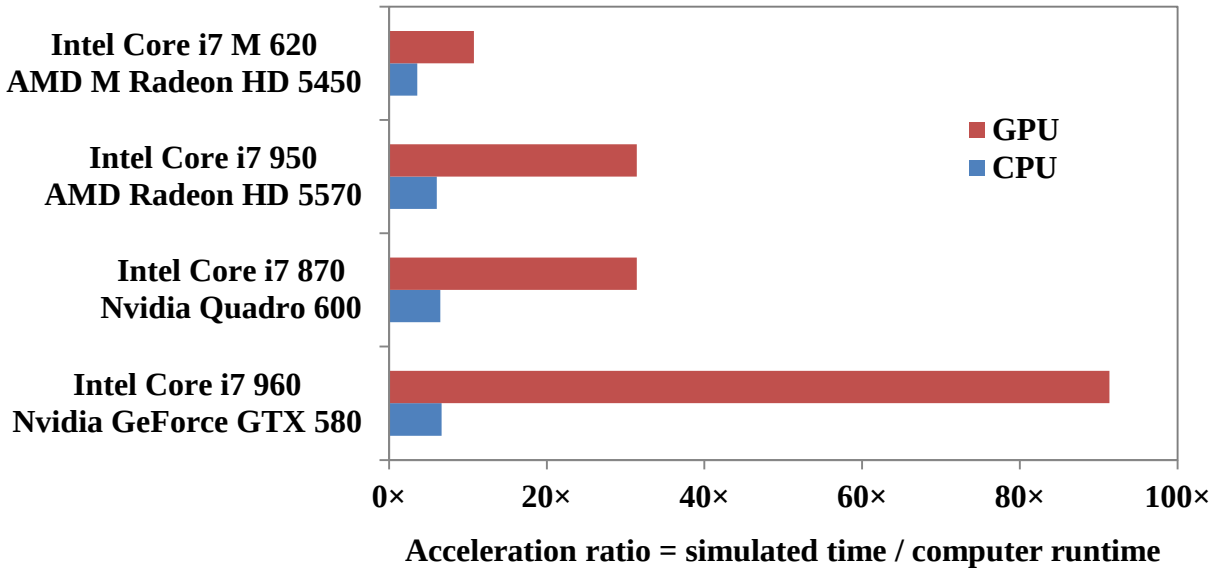

Figure 3-13: Simulated to actual cryoprocedure runtime ratio, for GPU-based (optimized) and CPU-based (optimized) simulations on various platforms, with platform specification listed in **Table 3-4 [53]**.

3.4 Results and Discussion

While developing a computerized training tool for cryosurgery, two fundamentally different approaches were considered for the cryosurgery simulator; integration with either an FE commercial code (ANSYS) or a proprietary FD code. While a wide selection of FE commercial codes could be selected and, independently, a wide range of FD schemes could be adopted, the significance of the current study is in the comparison of these two approaches. Nonetheless, each alternative is an excellent (and maybe the best) candidate for the task in its own category.

Benchmarking studies were performed on a cancerous prostate model created previously, using 14 cryoprobes at a uniform insertion depth, with optimal layout suggested by the bubble-packing method. Benchmarking data were developed on a mid-range, four-core machine, which is commonly available in hospitals and medical clinics. Results of this study indicate that the FE

simulator is superior for geometric modeling, but the FD simulator is superior for runtime. Benchmarking results further indicate that the FD simulator is also superior in the areas of memory resources, defect region calculations, pre-processing, post-processing, and multi-core processing. Both the FE and FD simulators operate on an identical computation framework. However, the FD simulator is a tailor-made code, which makes it more user-friendly and more efficient during operation.

After discovering that the FD based code surpasses the FE based code, an optimized FD algorithm was implemented on a GPU machine, using C++ AMP. C++ AMP was chosen for the current study as a choice of practice. Most significant acceleration was achieved by switching to float-precision, implementing asynchronous I/O, saving frequently used data in texture memory, and integrating a hybrid ELLPACK matrix structure to store the nodal connectivity grid. Results of this study demonstrate runtime acceleration of up to 13× compared with multicore-CPU processing, up to 50× compared with sequential CPU processing, up to about 90× compared with actual cryosurgery operation runtime, and an actual runtime of about 2 seconds for a single procedure [51]. The GPU-based implementation reproduced previously simulated cryosurgery cases on a multicore CPU-based implementation within 0.4ºC margins, where results of that earlier implementation have been validated experimentally [24]. Additional modifications, like the reduction of stair-casing artifacts [52] may further be applied to the proposed GPU implementation to improve accuracy and tailor the scheme for other applications. Further performance improvements to the proposed algorithm may include modification of the resistance matrix storage scheme. This change could make better use of the processors shared memory cache [47]. Advanced optimization techniques have been proposed for finite difference methods on regular grids in which the calculations are carried out to reduce the processor's global memory load [53]. Similar methods could be envisioned for the irregular grid used in the current

proposed implementation, and is an option for future work. This level of performance makes GPU-based computation ideal for computerized planning and training of cryosurgery.

The use of this advanced GPU simulator opens up new applications for computationally expensive planning algorithms. One such example is the “force-field analogy” [23]. In this method, one must run an average of 45 full heat transfer simulations. Each simulation takes approximately 30 seconds using the current CPU implementation, which makes total planning time over 20 minutes long. The new GPU simulator has a runtime of less than 2 seconds, so the total runtime of the planning algorithm will be reduced from 20 minutes to less than 90 seconds. In Chapter 5, this simulator will be used in conjunction with an ultrasound artifact simulator in order to produce realistic temperature based artifacts in real-time.

The current trend of increasing the number of cores on modern GPU platforms, combined with a relatively fixed problem size, suggest that there will become a point of diminishing returns. The GPU may become “too big” for the mathematical problem and further speedups will not be realized. However, this would only be true for a single execution of the simulator, a large array of simulations run in parallel could fully utilize the available resources (each simulation is itself parallelized for GPU execution). Hence, computer-based decision making associated with advanced GPU systems does not only call for rapid computation techniques but also for advanced decision-making frameworks, where many simulations are concurrently executed on the same system. Consistently, it is suggested that the next generation of computational cryosurgery research should focus on new ways to utilize the rapid advancing technology of GPU-based systems.

Chapter 4: Generation of Synthetic Ultrasound Images

In order for clinicians to develop the skills necessary to perform cryosurgery, they need access to a wide variety of test cases on which they can practice in a virtual training environment. There is no existing database of 3D US prostate images, and assembling such a database would be prohibitively expensive to create and maintain, since there are almost infinite sizes and shapes of tumor morphologies that may exist. The current study addresses this issue by implementing an algorithm that generates synthetic ultrasound images from a small library of examples. This study is based off of previous work in the field of texture synthesis which allows large visual textures to be generated or filled in based on a small example patch.

4.1 Algorithm

There has been significant work in the field of texture synthesis [54,55,56]; however, there has been very little work done toward the generation of synthetic US images for medical training [57,58]. Many of the algorithms used for US image synthesis rely on storing blocks of textures from image samples for each tissue type to be synthesized. However, this method often leads to unwanted visual artifacts; for example distinct discontinuities between different regions in the synthesized image. This occurs because of the abrupt change in texture types on region boundaries. Methods like radial blurring have been proposed to address this issue, but results are often unsatisfactory [57]. To address this need for realistic 3D US image creation, the current study proposes a modified version of the “Image Analogies” method [59]. The algorithm, as it applies to synthetic prostate US image creation, is described below.

To produce the desired synthetic image, portions of an example 3D US image are remapped to different locations in the output image. In order to construct a realistic output

image, one must maximize the number of input patches that can be used to generate a particular region in the output. This intuitively makes sense, because the more input patches the algorithm has to choose from, the more likely it is to find a suitable patch to create the next region in the output. One problem with sector US images is that the visual texture changes as you move further away from the US transducer. To address this, the sector US image was transformed as shown in Figure 4-1 to make the texture appear more uniform across the image.

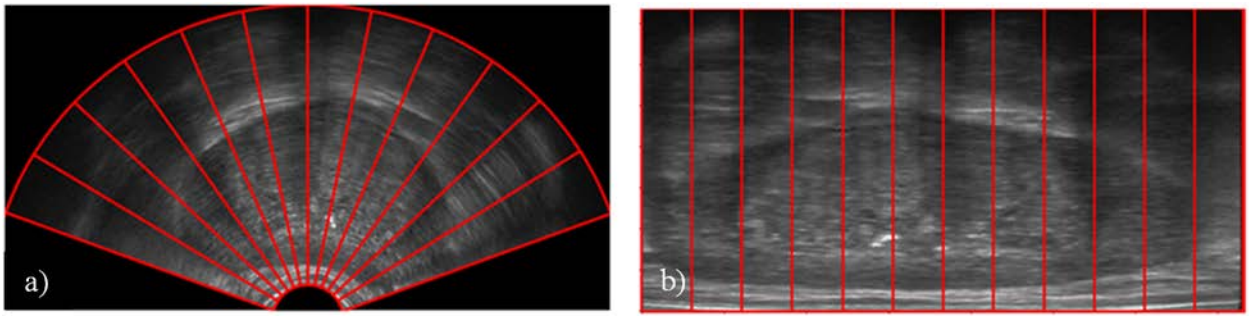

Figure 4-1: Transformation of sector shaped US image a) to rectangular representation b).

The Original US image must also be segmented into clinically relevant regions. These regions include the prostate, the urethra, and the rectal wall. These segmented regions are also transformed into rectangular images. The image below shows a slice from the transformed 3D input data which was used to generate the synthetic image. In Figure 4-2, segmented regions are depicted by color. Red represents regions enclosed by the urethra, regions in green lie within the prostate, blue regions are areas within the rectum, and regions in black represent the other surrounding tissue. The transformed 3D US image along with its corresponding transformed segmentation will be referred to as an input pair. It is from one of these pairs that new synthetic US images can be created.

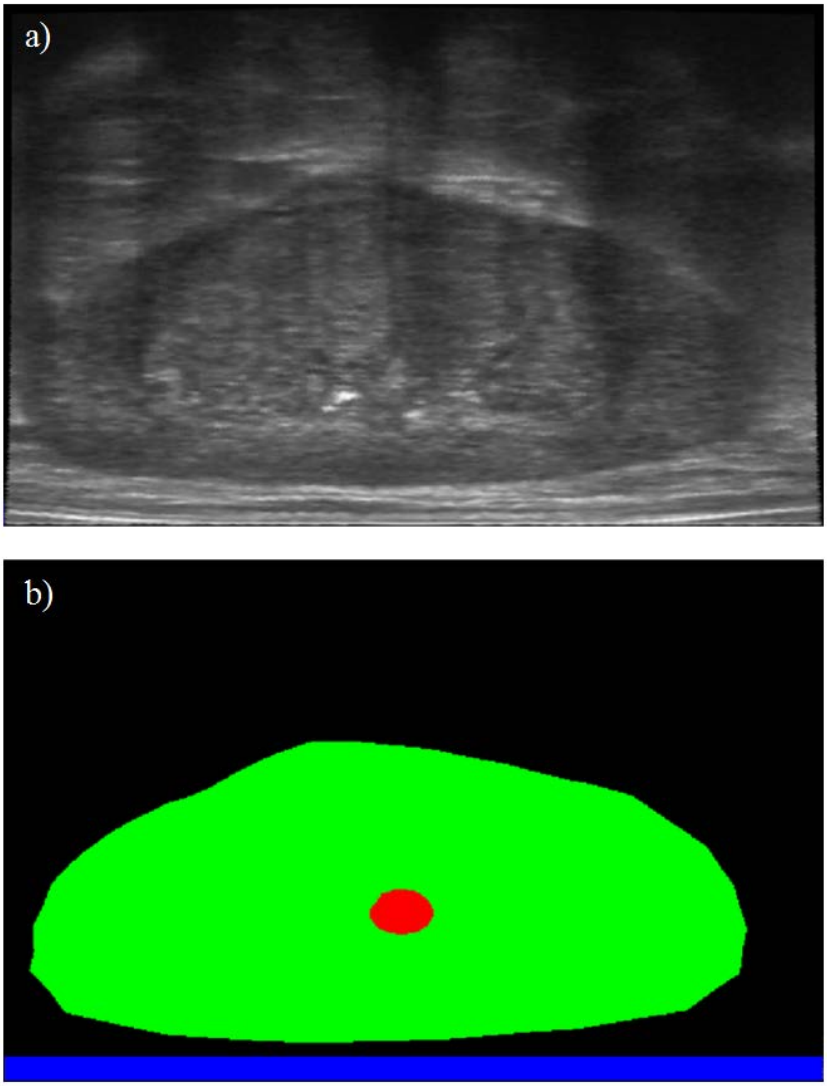

Figure 4-2: Input data pair, consisting of a 3D US image a) and segmented geometry b). Segmentation data is generated using the semi-automated method presented in **[28]**.

For completeness of presentation, the "Image Analogy" algorithm [59] will be explained in the context of prostate US image generation. With reference to Figure 4-3, the algorithm begins with an input data pair. This input data pair represents the image "analogy". When an US is generated, synthetic region data is given as an additional input. The algorithm uses the mapping from region to US texture given by the input data pair to determine how to generate an output US from the synthetic region data. The algorithm generates the synthetic US image a

pixel at a time on a lower resolution representation of the original data. The synthetic low resolution US images are then used to build progressively higher resolution images.

The algorithm begins with creating image pyramids which consist of progressively lower resolution representations of the input data pair. Then a feature vector for every voxel in the current level of the image pyramid is created. This feature vector is a concatenated vector consisting of region data and US image data for a voxel and its neighbors in the current level in the pyramid. This feature vector also contains region and US image data for the corresponding voxel at the next lowest resolution. The algorithm then creates an image pyramid for the synthetic segmentation data corresponding to the US to be generated. The US image is generated one voxel at a time in raster scan order starting at the lowest resolution representation in the image pyramid. The process for generating a single voxel begins with the creation of a feature vector around the location being generated in the output data pair. Then a search is performed for the closest feature vectors from the input data pair. In this work the approximate nearest neighbors library FLANN [61] was used to find the nearest feature vector. Once it is found, the US image voxel value corresponding to that feature vector is written to the current location in the output. There is an additional wrinkle to the Image Analogy algorithm, to maintain coherence in the output US image there is a bias to select neighboring voxels in the input image for neighboring voxels in the output image. By adding this preference to feature vector selection, the output image maintains more of the visual appearance of the input image.

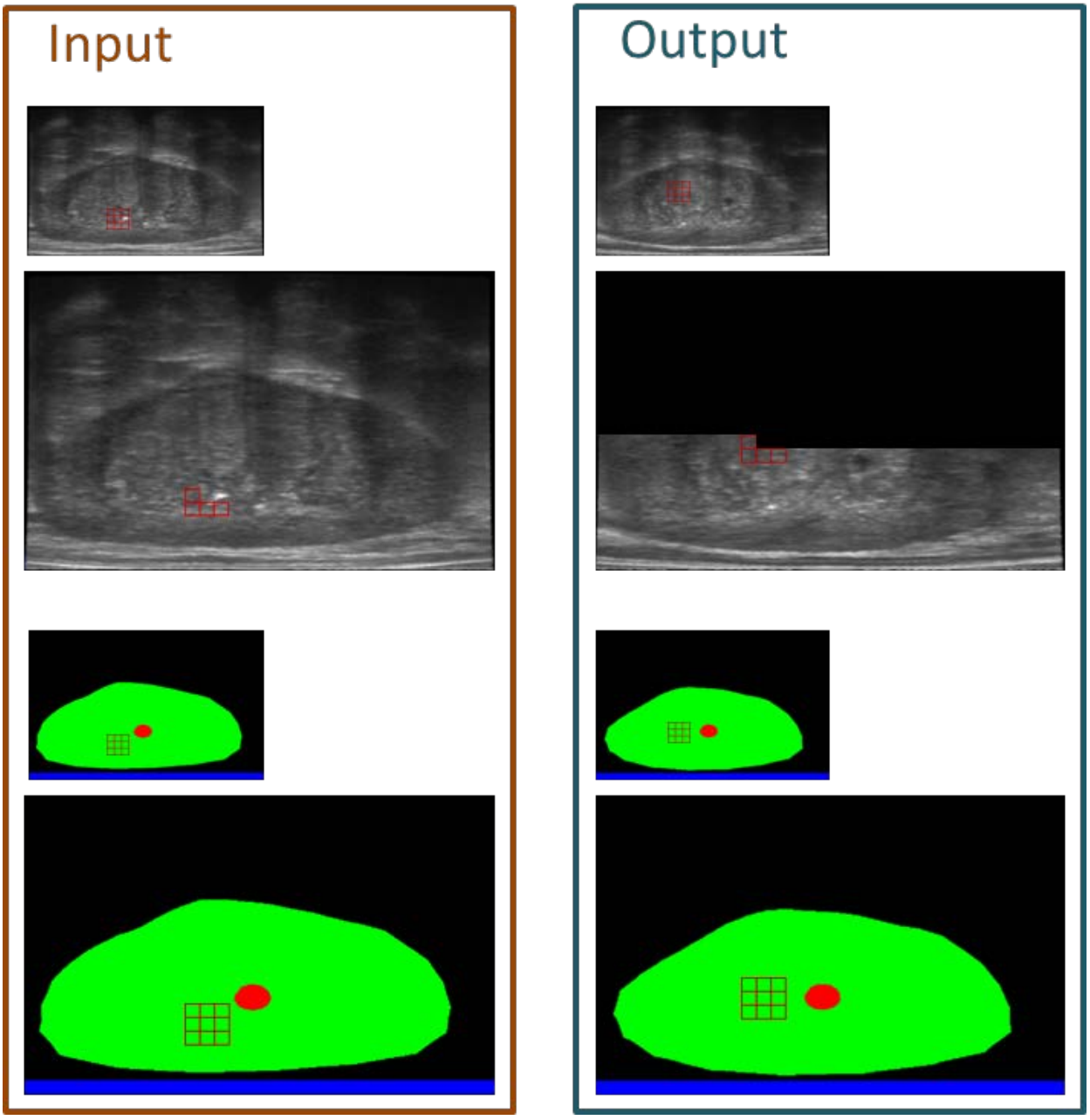

Figure 4-3: Depiction of 2D Image Analogies algorithm from which the proposed method is based. Input data consists of image pyramids of region and US image data. Output US image data is generated a voxel at a time. Highlighted voxels shown in red windows represent voxel values that make up feature vectors in input and output.

There are three main differences between the proposed algorithm and the original "Image Analogy" algorithm. The first is the extension to produce 3D images. This change consisted of

using neighbors from adjacent slices in the image in order to find the best output voxel value. This change causes an issue as it dramatically increases the size of the feature vector. For example for a 3x3 grid of neighbors in 2D, the feature vector for each point in the input contains 67 entries. For a 3x3x3 grid, each feature vector contains 202 elements. Additionally, the number of feature vectors is two orders of magnitudes higher for a 3D image than a 2D image because there are hundreds of 2D slices that make up a 3D image. As a result, the algorithm uses excessive amounts of memory to store all of this information. It was noticed that for the application of cryosurgery, many voxels in the input image contain neighbors that all lie within the same region. As a result, if a voxel and its neighbors all lie within the same region, only the US feature vector data will be saved, and it will be placed into a separate list of feature vectors. This effectively reduces the amount of data that needs to be saved for each voxel by approximately 75%. When an output voxel is to be generated and all of its neighbors lie within the same region, the value will be computed from that regions list of feature vectors. If a voxel contains neighbors from different regions, then its feature vector is placed into a list of feature vectors identical to the original algorithm. Finally, the algorithm was parallelized by using the assumption that at any given time during the computation, many voxels are ready to be processed because all of their raster scan neighbors have already been calculated.

4.2 Results and Discussion

Figure 4-4 shows a slice from the transformed 3D results of the algorithm. What can be seen from this data, is that unlike many parametric approaches, the algorithm used does not “grow garbage” and it clearly shows the regions of interest to the users with a realistic texture.

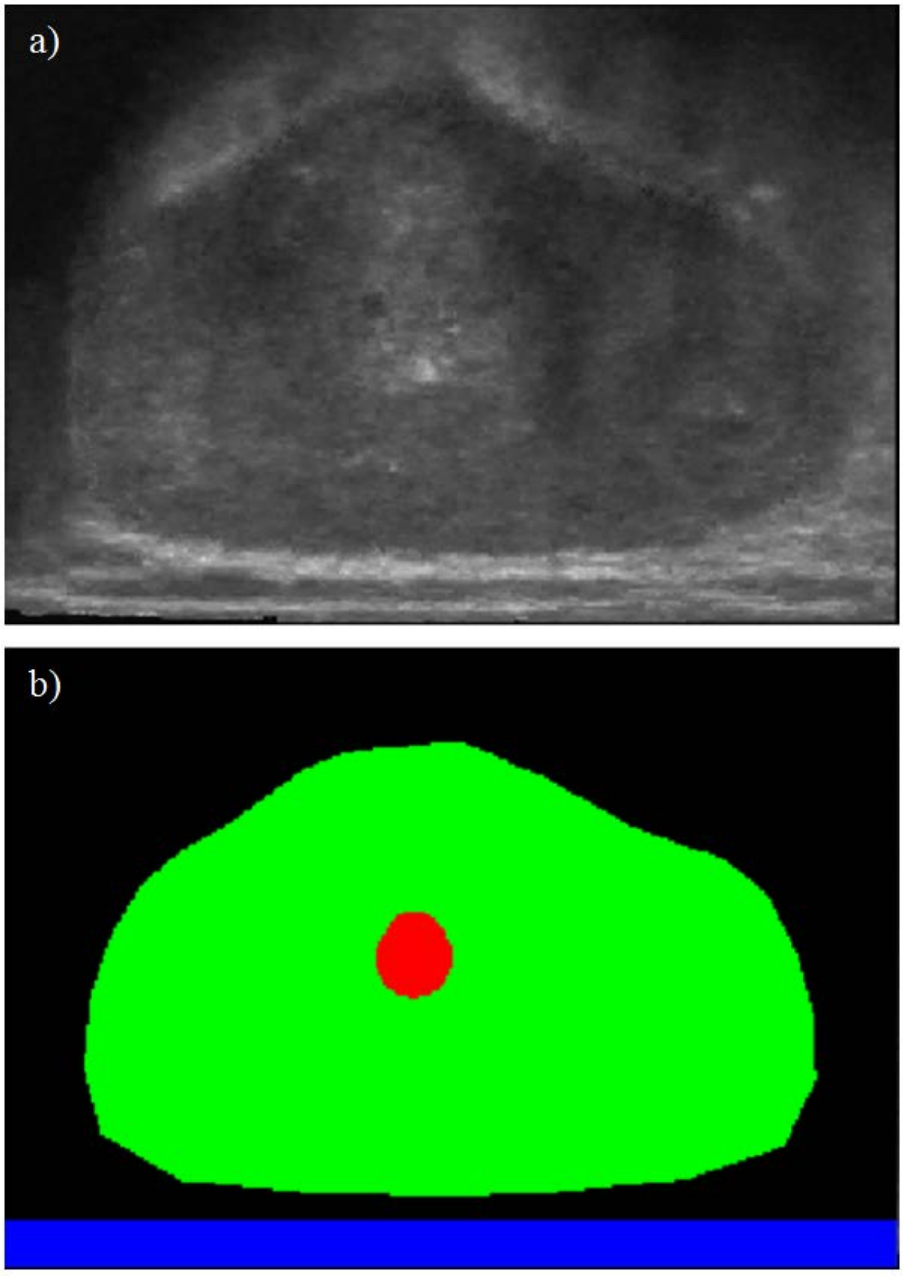

Figure 4-4: Output data pair, consisting of 3D US image a) and segmented geometry b). This example uses the data pair shown in Figure 4-2. as input.

For comparison, a slice from the untransformed input and output image are shown in Figure 4-5. Figure 4-6 depicts the transformed input and output US images into their final polar representation. The input and output image look qualitatively very similar. The most important

result of this study is not the algorithm, as many variants can be imagined, each of which with its own tradeoffs between accuracy, speed, and memory usage. What this study showed is that a model of the seemingly random background texture that comprise ultrasound images can be modeled to capture all of the relevant statistics of the input image.

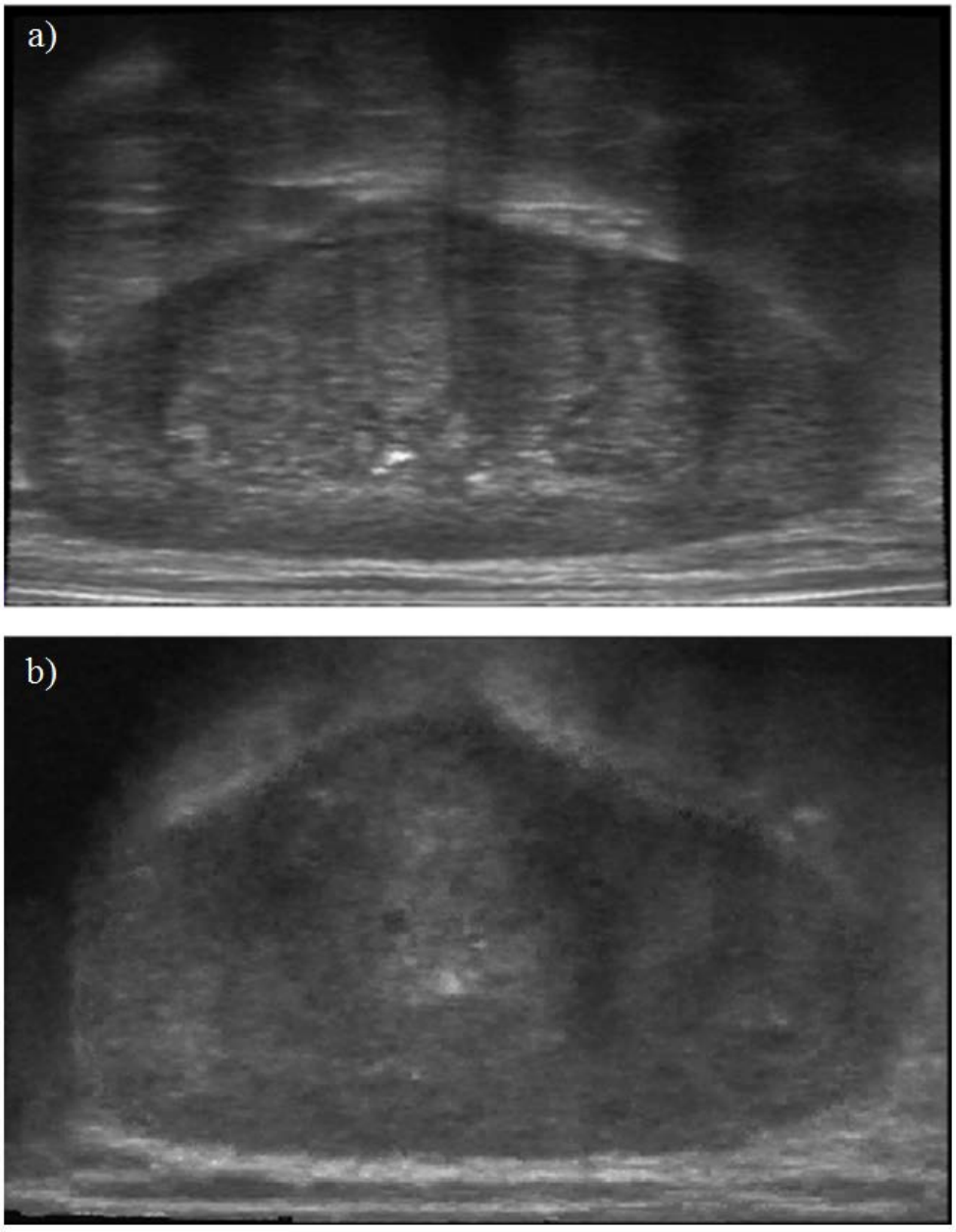

Figure 4-5: Comparison between input a) and output b) transformed US images.

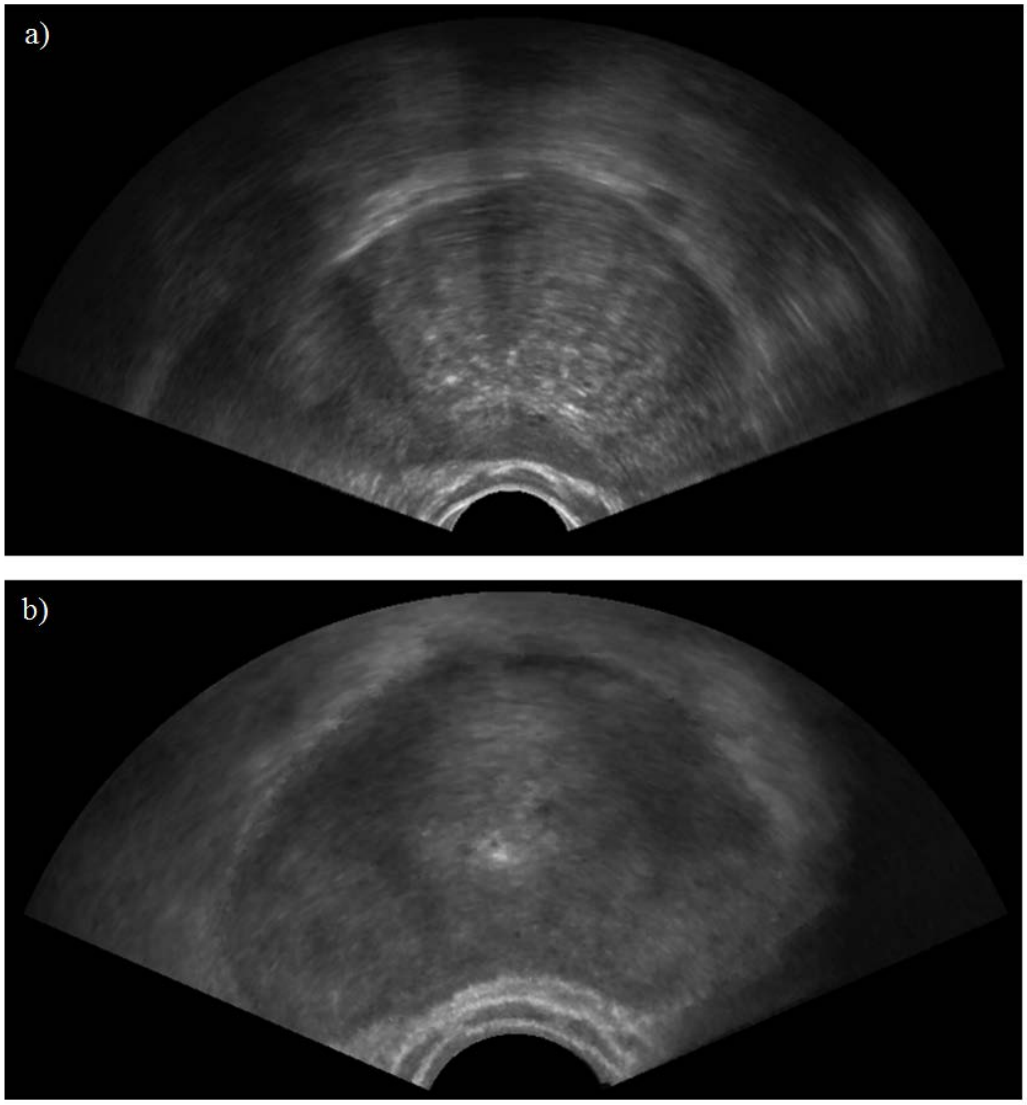

Figure 4-6: Comparison of input a) and output b) US images in polar representation.

The algorithm developed in this thesis is capable of generating synthetic trans-rectal prostate ultrasounds that contain all of the important physiological structures including; the prostate, the urethra, and the rectal wall. To generate the synthetic US image, the algorithm uses a single example 3D US and surface model pair. The surface models correspond to the regions generated by segmenting the image by various tissue types. What is then given is the surface geometry

which corresponds to the shape of the prostate for the desired test case. The algorithm searches the input data for regions that match the current portion of the output data. Once the closest matching region is found the algorithm uses the value of the corresponding voxel in the input to generate the synthetic image one voxel at a time. Memory optimizations were performed to allow the algorithm to run on a desktop computer. Additionally, by noticing that each voxel only requires the data from the preceding neighbors (using raster-scan ordering) to be generated in order for the calculation to begin allowed many voxels to be run in parallel. These optimizations greatly reduced the algorithm runtime. Using this algorithm in a computerized tutoring program would allow clinicians to practice on almost an infinite number of test cases.

Chapter 5: An Efficient Simulation of Ultrasound Imaging During Cryosurgery

In order to have an effective virtual environment for cryosurgical training, the visual input used by the clinician to both place the cryoprobes and monitor the extent of the frozen region must be accurately reproduced. TRUS imaging is most frequently used to monitor prostate cryosurgery. Placement of minimally invasive cryoprobes relies on the imaging artifacts created by said cryoprobes. Monitoring the frozen region formation also relies on US imaging artifacts, where the tissue becomes opaque upon freezing [60]. In order to make such a virtual setup a practical reality in surgical training, simulations of US images must be performed in real-time and their outcomes must appear realistic [31].

5.1 Previous Work on Ultrasound Imaging Simulation

The B-mode ultrasound images typically used in cryosurgery are produced by emitting a high frequency sound wave from the TRUS transducer into the tissue. As the ultrasound energy propagates through the tissue, portions of it may be transmitted, absorbed, or reflected, depending on the local material properties. Inhomogeneity in the local acoustic impedance of the tissue may cause some of the acoustic energy to scatter, and some to reflect back to the transducer. This reflected portion of the acoustic energy is processed to reconstruct the ultrasound image, while the signals are normalized to account for the fainter echoes from tissues that are further away from the transducer. This normalization process is called time-gain compensation (TGC), which is used to create the final image.

A significant body of literature exists on ultrasound imaging simulations [61], where simulation approaches are conveniently classified as wave-based, interpolation-based, or ray-

based, but mixed approaches have also been reported [62]. In general, the wave-based approach attempts to mimic an acoustic wave as it propagates through the tissue. This can be accomplished by either using spatial impulse response rendering, applying a K-space method for acoustic propagation, or directly solving the Westervelt equation, with the application of Green's functions as one possibility [63,64,65]. Unfortunately, the computational costs typically associated with acoustic-field simulations prohibit the use of regular wave-based approaches for real-time applications, and approximation methods must be devised.

The interpolative-based approach for ultrasound imaging simulations is likely the most popular approach for real-time applications [62,66]. Here, prerecorded volumetric images obtained from magnetic resonance imaging (MRI), computerized tomography (CT), or US are sliced along the desired plane to form a two-dimensional (2D) base image. Next, various texturing techniques are used to create the speckle pattern seen in native images. Finally, imaging artifacts are typically added ad-hoc, after the speckle pattern has been integrated. As a result of this sequential approach, simple artifacts can be quickly generated but complex artifacts—such as those originating from multiple reflections—cannot be realistically represented.

Ray-tracing methods have proven very successful in the field of optics, when the structures are large in comparison with the wavelength being used for imaging [67]. This is not necessarily the case for US imaging, since many small features within the tissue may nonlinearly affect ray propagation. While smoothing of the material properties, as is often done for seismic simulation, can improve the resulting images under some conditions [68], it is not required for nonlinear ray-tracing in biological tissues, as these small scale variations tend to add speckle texture to the image rather than changing the direction of the rays.

The ray-tracing approach is based on tracking the position of discrete points on a wave

front as it propagates, while being reflected and refracted along the path of propagation. Many state-of-the-art techniques for ray-tracing simulations require surface models of the geometry to define the boundaries between the various sub-regions in the domain [62]. In such techniques, each time that a projected ray hits a translucent boundary, the reflected and refracted rays are calculated. Unfortunately, these techniques may not be suitable for tissue freezing in cryosurgery, due to the non-discrete nature of the medium at the freezing front. An alternative technique that can potentially capture the nonlinear effects at the freezing front employs volumetric ray-tracing, which does not require definition of an explicit surface model. Reported volumetric ray-tracing algorithms [69,70,63] are based on tracing straight rays with uniform sampling points, which can only produce basic ultrasound artifacts such as reflection and shadowing. The available algorithms cannot represent more complex artifacts typical to cryosurgery, such as reverberation by cryoprobes and energy absorption during tissue freezing. Employing such approaches would necessitate adding cryoprobes to the image on an ad-hoc basis [57].

5.2 Methods

The current study aims at developing means for realistic simulations of US imaging artifacts for cryosurgery training. The conceptual design for integrating ultrasound imaging simulations with cryosurgery training is schematically illustrated in Figure 5-1. Towards this goal, the key building blocks for cryosurgery training have already been developed, including procedure planning methods [26,23,27], training strategies for cryoprobe placement [31], and bioheat simulations methods for the cryoprocedure [25]. Furthermore, the feasibility of a cryosurgery training approach has been presented recently. With reference to Figure 5-1, prerecorded ultrasound datasets of five individuals have already been developed in previous studies, and used to reconstruct the prostate shape [27,71,30] for computerized training [31,72]. Additionally, a realistic ultrasound scanning tool has been recently presented to enable intraoperative prostate reconstruction based on partial organ contours [28].

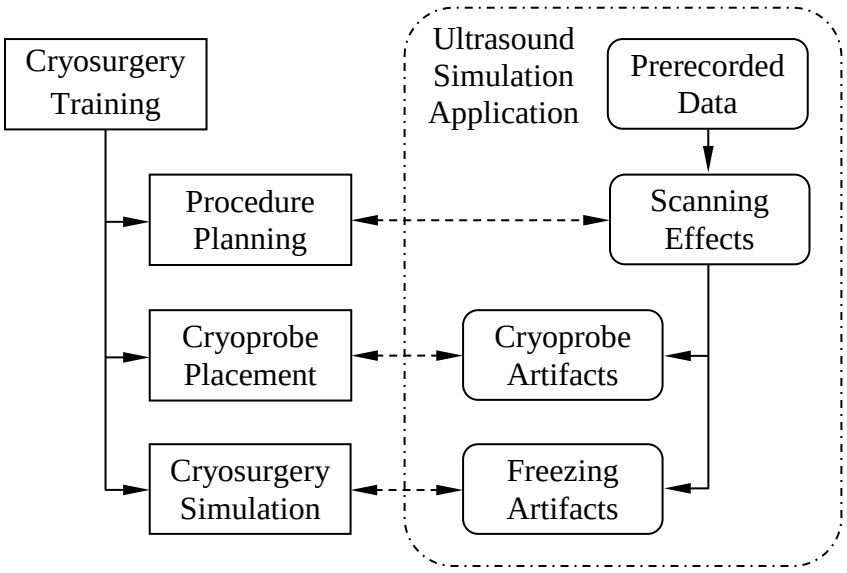

Figure 5-1: Schematic illustration of the conceptual approach to integrate ultrasound simulations with cryosurgery training, where cryosurgery simulation refers to the full-scale bioheat-transfer simulation **[76]**.

A hybrid method is proposed in the current study to produce realistic ultrasound imaging artifacts for cryosurgery training in real time, combining features from the interpolative and ray-tracing approaches. This method is designed to (i) display imaging artifacts resulting from cryoprobes insertion, and (ii) display the effects of tissue freezing as the cryoprocedure progresses. A general view of the targeted outcome is displayed in Figure 5-2, where the scanned prostate region (Figure 5-2(a)) and bioheat simulation results (Figure 5-2(b)) are combined to simulate freezing artifacts (Figure 5-2(c)). Figure 5-2(d) displays a hybrid ultrasound-thermal field presentation, to provide the trainees with insight into the thermal process affecting imaging artifacts. With such a presentation, the trainee can correlate the thermal field with criteria for cryosurgery success [31] and the location of cancerous tumors [72]. Figure 5-2 was generated using the mathematical formulation and code implementation described in the next two sections.

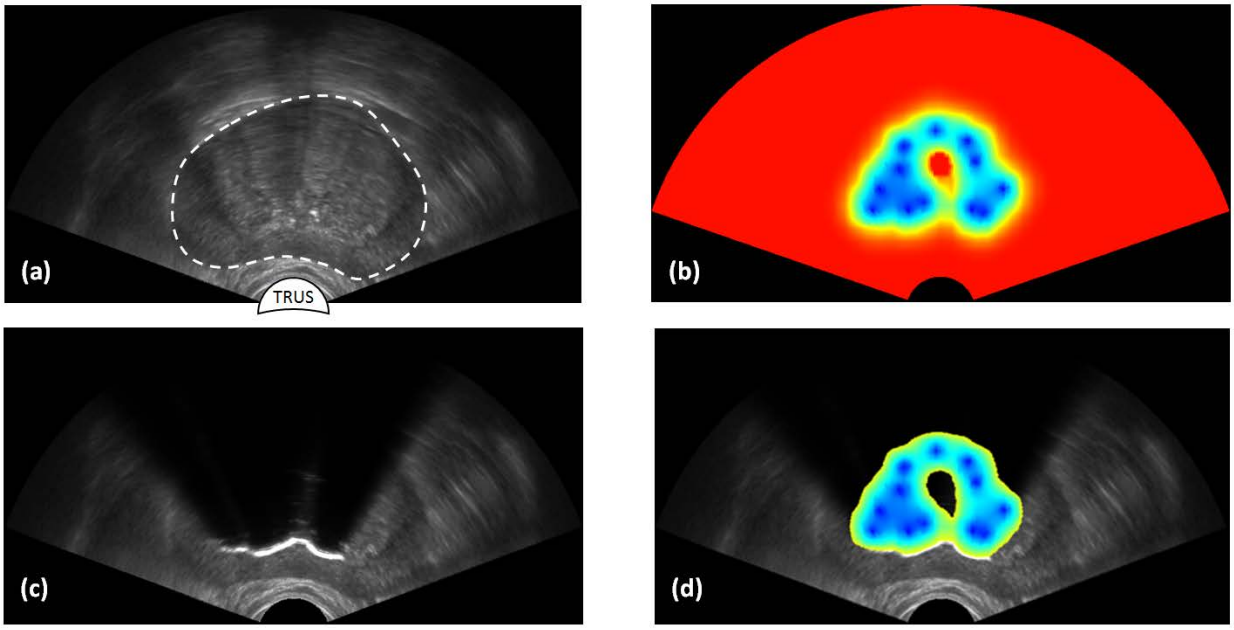

Figure 5-2: Application of ultrasound imaging for cryosurgery planning and training: (a) scanning of a prerecorded ultrasound imaging dataset, where the contour of the prostate is highlighted with a dashed line; (b) bioheat transfer simulation results of the cryoprocedure, (c) simulated frozen region effect based on the bioheat transfer simulation results and the artifacts-simulation method proposed in the current study, and (d) overlaid of the temperature the field in the frozen region on the simulated ultrasound image as a training means to explain cryosurgery artifacts **[76]**.

5.2.1 Mathematical Formulation

With reference to Figure 5-3, the proposed computation scheme for US image generation consists of six steps:

I. simulation of bioheat transfer in the target area;

II. determination of the acoustic properties within the domain, based on the local material properties and thermal history;

III. simulation of the US-ray paths from the transducer into the domain, Figure 5-3(a);

IV. calculation of acoustic energy propagation along the simulated US rays;

V. TGC evaluation of the energy returning to the transducer; and,

VI. modifications to the brightness of the previously recorded US texture according to account for the TGC results.

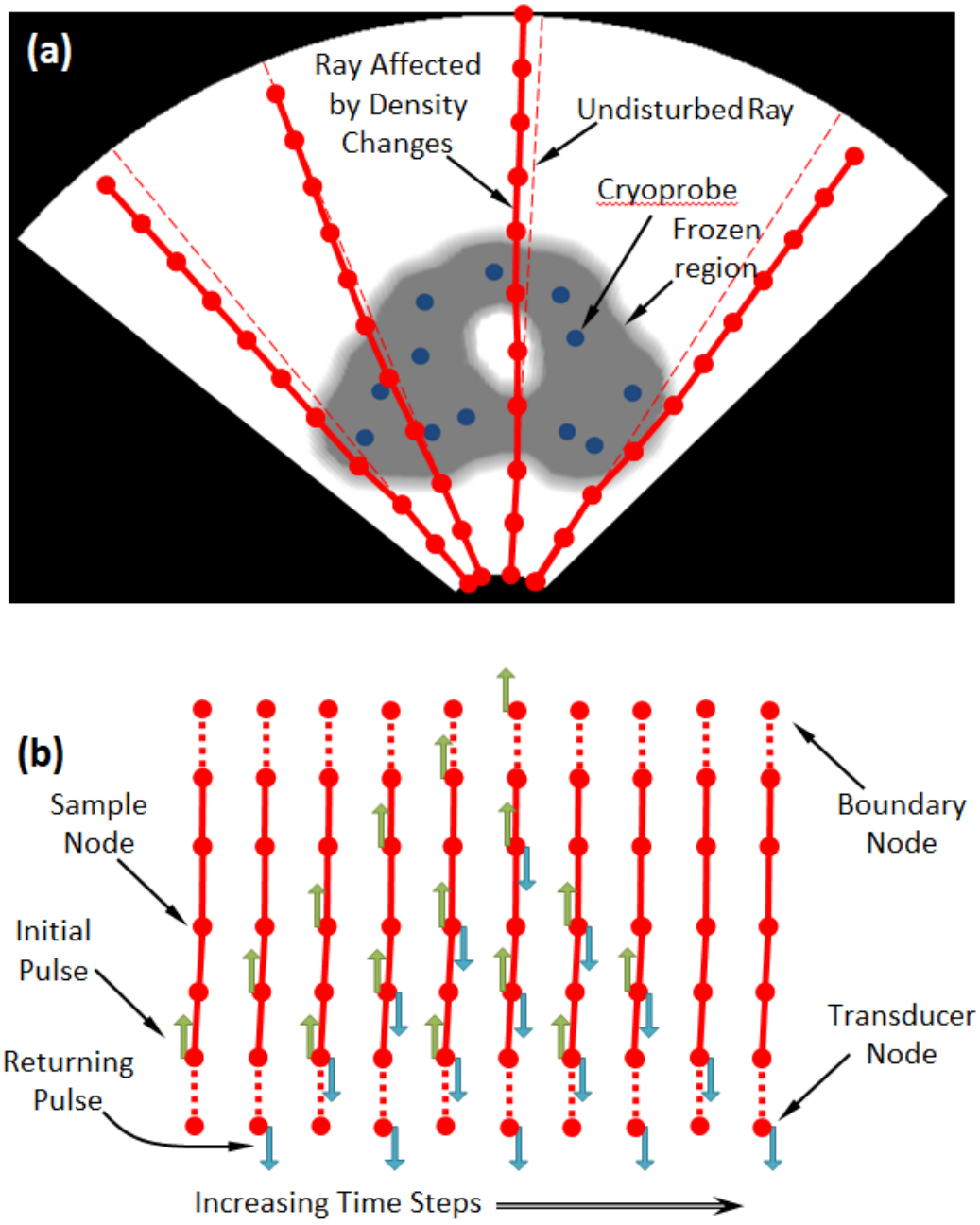

Figure 5-3: Schematic illustration of artifact-calculation process in a tissue of non-uniform acoustic impedance: (a) non-linear ray propagation with speed-of-sound dependent sampling in the original polar coordinate system, and (b) wave intensity propagation between sample points along a single ray over consecutive time steps, based on reflection and transmission properties after transformation to a Cartesian coordinate system **[76]**.

A framework for parallel computation of bioheat transfer during cryosurgery has been

established previously, and is adopted here as a choice of practice for Step I of the proposed method [51,32]. Results of this simulation are used to evaluate the local material properties in Step II. As appropriate, linear interpolation is used to determine the acoustic properties and their derivatives at the necessary locations within the domain.

The simulation of ray propagation from the transducer in Step III requires a set of acoustic properties, including echogenicity, speed of sound, acoustic impedance, and the tissue absorption of wave energy. In the case of prostate cryosurgery, these properties are further dependent upon the presence of cryoprobes and on the thermal history in the tissue. In particular, Step III is used to determine the advancement and deflection of the ultrasound wave as it propagates through the tissue. The ray's path of propagation through the domain is calculated by solving the Eikonal equation [67]:

$$\frac{d}{dl}\left(\frac{1}{V}\frac{d\vec{r}}{dl}\right) = \nabla\left(\frac{1}{V}\right) \quad \Rightarrow \quad \frac{d}{dl}\left(\frac{1}{V}\frac{dx_j}{dl}\right) = \frac{\partial}{\partial x_j}\left(\frac{1}{V}\right) \tag{5-1}$$

where l is the distance measured along the ray path, V is the velocity, $\vec{r}$ points to the direction ray propagation, and x_j are the principal orthogonal components. Equation 5-1 represents a generalization of Snell's law of refraction for a continuous and non-homogeneous medium, which is solved in this study with the application of the non-linear ray-tracing scheme described by Seron et al. [67]. In essence, this equation describes the change in ray direction as a function of the variation in speed of sound, as it propagates through the domain. Equation 5-1 is solved in a Cartesian geometry (Figure 5-3(b)), after transformation of the prerecorded imaging dataset from the polar coordinate system typical to TRUS imaging (Figure 5-3(a)).

Equation 5-1 is solved by using a forward Euler scheme discretization Equation 5-2:

$$\frac{dx_j^{i+1}}{dl} = \frac{\left(\frac{1}{V}\right)^i \frac{dx_j^i}{dl} + \Delta l_i \frac{\partial}{\partial x_j}\left(\frac{1}{V}\right)^i}{\left(\frac{1}{V}\right)^{i+1}} \tag{5-2}$$

where Δl is an incremental distance along the ray, and *i* is the increment index.

For convenience in presentation, the change in direction of the ray, $\vec{v}^i$, is defined as Equation 5-3:

$$\vec{v}^i \equiv \frac{d\vec{x}^{i+1}}{dl} \tag{5-3}$$

where the value of $\vec{x}^{i+1}$ can also be approximated using a forward Euler scheme (Equation 5-4):

$$\vec{x}^{i+1} = \vec{x}^i + \Delta l^{i+1} \frac{\vec{v}^i}{\|\vec{v}^i\|} \tag{5-4}$$

The procedure described by Seron et al. [67] is modified by applying fixed time intervals through the simulation, while computing the resulting distance of travel, Δl, as the product of the variable wave propagation speed, V, and the time interval. The selection of the time interval length is dictated by the required simulated image resolution. In essence, it is this dependency of the propagation distance on the speed of sound that allows the depiction of imaging artifacts, where objects may appear farther or closer to the ultrasound transducer [61].

Energy propagation calculations in Step IV represent one of the new contributions in the current study. After identifying the path of each ray, the amount of energy transmitted along the ray is calculated—from the simulated transducer into the tissue and back to the transducer. Here, the propagation of acoustic wave energy between the sample points (calculated in Step III) along each ray is calculated repeatedly, after being emitted from a simulated transducer. This novel step allows multiple reflections to be efficiently incorporated into a volumetric ray-tracing US simulator. This ability is necessary to simulate both temperature field and cryoprobe artifacts in real-time which was previously impossible. The material properties at each of those sampled

points are used to determine the portion of the energy transmitted along the ray and the portion of it that is reflected back along the same ray. The ratio of energy reflected between two sample points is given by Equation 5-5:

$$\rho_{imp} = \frac{(Z_{i+1} - Z_i)^2}{(Z_{i+1} + Z_i)^2} \tag{5-5}$$

where Z_i and Z_{i+1} are acoustic impedances at sample points *i* and *i*+1. The acoustic impedance is calculated by Equation 5-6:

$$Z = Vd \tag{5-6}$$

where *V* refers to the local and instantaneous speed of sound, and *d* is the mass density of the tissue at the specific location.

At every sample point, the amount of energy reflected, E_{ref}, absorbed, E_{abs}, and transmitted, E_{trans}, along the ray must be calculated from the total acoustic wave energy arriving at that particular point, E_{tot}. Only the portion of the energy reflected due to acoustic impedance mismatches between sample points will be reflected back along the ray. Qualitatively, this is analogous to a reflection off an interface between different media, where the normal to the interface and the incident ray direction are not parallel. This reflection is represented in the current study by modifying the Lambertian model [63], where the energy reflected along the ray is given by Equation 5-7:

$$E_{ref} = E_{tot}\left[\rho_{imp}\cos(\theta) + \rho_{echo}\left(1 - \rho_{imp}\right)\right] \tag{5-7}$$

where ρ_{echo} is the echogenicity reflectance and the reflection angle, θ, is calculated by Equation 5-8:

$$\theta = cos\left[\frac{x_n \cdot \frac{\partial}{\partial x}\left(\frac{1}{V}\right)}{\|x_n\| \times \left\|\frac{\partial}{\partial x}\left(\frac{1}{V}\right)\right\|}\right]^{-1} \tag{5-8}$$

Equation 5-7 represents a modification of the standard Lambertian reflectance model in two ways. The first modification concerns the addition of the echogenicity reflectivity term, ρ_{echo}. This term is designed to account for the portion of energy that is not reflected due to Lambertian effects but that is reflected due to small variations within the tissue, which give rise to the speckle US image pattern and to the image's overall brightness. It follows that the total reflectivity equals the sum of the Lambertian reflection component and the echogenicity component. The second modification is in the use of the gradient vector of the acoustic impedance instead of the normal-to-surface vector, as is typically the case for the standard reflection model. The gradient application can be viewed as a continuum representation of the discrete and ordinary equation. Conservation of energy dictates that the amount of transmitted energy must be equal to the difference between the total energy and the sum of the reflected and absorbed portions of it (Equation 5-9):

$$E_{trans} = E_{tot}\,(1-\alpha)\left\{1-\left[\rho_{imp}+\rho_{echo}\left(1-\rho_{imp}\right)\right]\right\} \tag{5-9}$$

where α is the absorption coefficient.

The above equations for calculating the transmission and reflection of wave energy between sample points are applied repeatedly in order to effectively integrate these quantities over time. The repeated calculations of those quantities allow for wave energy to be reflected back and forth multiple times along each ray, thereby providing a realistic simulation of both single and multiple reflection artifacts. Such multiple reflection artifacts may develop around the tissue-embedded cryoprobes, where the wave energy propagates between two highly reflective interfaces multiple times. When *N* samples are taken along a ray, energy propagation calculations must be repeated 2*N* times for each sample point, in order to allow for the simulated transmitted energy to travel all the way from the transducer to the furthest sample point and back. Figure 5-3(b) depicts the propagation of an initial pulse of energy along a single ray, where

the pulses of reflected and transmitted components at each sample point are calculated.

In Step V of the proposed scheme, the returning ultrasound simulated *echoes* are normalized. This normalization process is performed in order to account for the fainter echoes that arrive from distant portions of the tissue, which underlines the TGC process [61], where the gain applied to the signal is a function of time. In Step VI, TGC results are blurred using a Gaussian filter to take into account the finite ability to focus the waves used to facilitate ultrasound imaging. Finally pixel intensity values from the existing ultrasound image are modified based on the TGC signal strength compared to a reference value, to allow for a realistic representation of imaging artifacts.

5.2.2 Code Implementation

The most computationally expensive aspect of the ultrasound simulation is the calculation of the energy propagated along the rays. For example, for M rays and N sampling points along each ray, the number of required operations is proportional to $M \times N^2$ (typical values applied in the current study: M=500, N=300). Hence, efforts to optimize the number of sampling point calculations are highly warranted, in order to achieve real-time ultrasound simulations. This challenge is addressed by optimizing code implementation while taking advantage of the graphics processing unit (GPU). While various optimization steps were taken in order to speed up the simulation, summarized below are the effects of information-flow strategies, field-decomposition strategies, and memory usage effects, all with reference to a baseline, naïve implementation.

Naïve implementation: A baseline implementation of the proposed scheme consists of: (i) storing all the necessary data for each sample point in global memory; (ii) running a kernel for each sample point that would read the necessary values from global memory, perform the

corresponding calculations, and write the result back to global memory; and, (iii) repeating this kernel for 2*N* cycles, to account for acoustic wave propagation back and forth along a ray discretized into *N* sampling points. In terms of runtime, this naïve implementation performs very poorly due to extensive reads and writes to global memory. Furthermore, a considerable amount of application programming interface (API) overhead is associated with running the GPU kernel 2*N* times, which is inherent to the initialization of a GPU task. Note that it is necessary to call the kernel in a loop, since all sample points from a specific iteration must be calculated before the next iteration can executed, while there is no effective means to synchronize simultaneously running threads on different groups of GPU processors.

Information-flow strategies: It can be concluded from Fig. 3 that the propagation of information along a specific ray exhibits a finite speed. It follows that any numerical node that is farther than the signal-information penetration depth, or that information from it cannot reach the transducer in time to contribute to image formation, can be eliminated from calculations with no adverse effect. Those numerical points are said to be outside of the *signal information envelop* in the context of the current study. Eliminating numerical points outside of the information envelop accelerates runtime by a factor of 2. Further note that the numerical scheme has an even-odd information flow pattern, in the sense that energy flow calculations switch between even and odd nodes on successive iterations. If follows that energy calculations can alternate between even or odd nodes on successive iterations with no adverse effect, which is expected to accelerate runtime by another factor of 2. Hence, carefully accounting for information flow during code implementation can accelerate simulation runtime by a factor of 4, regardless of the computation platform selected for the task.

Field-decomposition strategies: For an instantaneous temperature field, energy calculations along each ray can be performed independently. It follows that decomposition of the

simulation domain, such that a single group of processors on the GPU are responsible for energy calculations along a given ray can improve parallel processing. Following this decomposition, processors synchronization within a specified group can be straightforwardly done with C++ AMP. This strategy further allows for the outer loop, which is used to call the kernel $2N$ times, to move into the kernel itself. Such domain decomposition reduces the API overhead to a single kernel invocation.

Memory usage: Domain decomposition and the need to only call the kernel function a single time for a given ray, enables the use of the lower latency shared memory instead of the much slower global memory. Once a kernel is invoked, all the necessary data is transferred from global to shared memory. Then, as the processor group performs the corresponding $2N$ calculation steps, all read and write operations are performed in the shared memory. It is not until the last stage of computation when the calculated results are transferred back to the global memory. As a result, all of the intermediate global memory operations are eliminated. Using the field-decomposition strategy in combination with the more efficient memory usage has accelerated ultrasound simulations by a factor of 5 in the current study.

5.2.3 Simulation Protocol

The proposed numerical scheme was analyzed based on a 3D prostate ultrasound dataset reported previously [27,71,30]. The 3D shape of the prostate was segmented using a previously developed semi-automated method [29]. Once the 3D shape of the prostate was reconstructed, an array of cryoprobes was virtually placed at a single insertion depth, using a computer-generated optimized layout based on the bubble-packing method [26]. Next, a bioheat transfer simulation was executed, using a GPU-based embodiment [73] of an efficient numerical scheme for the bioheat transfer process [25]. This combination of a numerical method and a coding framework

is characterized by a typical runtime of 2 seconds for a complete procedure [51].

Ultrasound image analysis in the current study is based on a typical case study of 12 identical cryoprobes having a diameter of 1.3 mm and active cooling length of 25 mm. The initial temperature in the entire domain is 37°C, and the domain is large enough such that the bioheat transfer problem can be considered as taking place in an infinite domain for the duration of freezing [25] (in practice, a domain volume three times as large as the prostate size satisfy this requirement). A urethral warmer, 6 mm in diameter, was simulated by maintaining the urethra wall at a constant temperature of 37°C throughout the simulation. The simulated thermal history at the cryoprobes is comprised of an initial cooling from 37°C to -145°C in the first 30 seconds, and constant temperature thereafter, until the completion of the simulated cryoprocedure and deactivation of the cryoprobes. The bioheat transfer simulation continued after cryoprobe deactivation, until the treated region returned to its initial temperature.

The cryoprobes were deactivated when the frozen region best-matched the prostate contour, using the defect-region concept as a measure [23]. In broad terms, a defect region represents unfrozen areas internal to the prostate and frozen areas external to the prostate. It follows that the entire prostate is considered a defect at the beginning of simulation, but if the cryoprobes are activated for long enough and the cryoprobes are powerful enough, most of the defect may become external. There is a point in time along this transition from internal to external defect that the overall defect value becomes minimum, which is the point of simulation termination in the current study. The bubble-packing planning method referenced above aims at uniform distribution of the defect along the edge of the target region (i.e., the prostate), which also results in a global minimum of the defect when variable cryoprobe layouts are considered. The resulting thermal field along the bioheat transfer simulation was used as input for the ultrasound simulation method described in the previous section.

5.3 Results

Figure 5-4 and Figure 5-5 display typical artifacts due to the presence of cryoprobes in the unfrozen prostate. From a cryosurgery training perspective, these artifacts are associated with the challenges typical to cryoprobe placement according the preplanning. Figure 5-4 displays reflection and reverberation artifacts along the cryoprobe in a sagittal plane, which is perpendicular to the transverse-plane images displayed in Figure 5-2 and Figure 5-3. Figure 5-4(a) displays an actual ultrasound image of a hypodermic needle in the prostate, having comparable dimensions to the cryoprobes specified above [21,26]. Figure 4(b) displays simulated cryoprobe artifacts following the mathematical formulation described above. It can be seen that the simulated artifacts contain the same features as the actual image. The most notable artifact being the periodic hyper-echoic regions extending behind the cryoprobe, with a period equals the probe diameter, d. This periodic repetition is marked as reverberation in Figure 5-4. Additionally, a brightening of the reflection occurs at twice the distance between the US transducer and the cryoprobe, L. The reverberation with the maximum brightness is marked as reflection in Figure 5-4, at a distance $2L$. This brightening effect is another multiple reflection artifact, but this time occurring between the highly reflective surface of the cryoprobe and the ultrasound transducer. Note that the overall brightness of the cryoprobe and its reverberations may vary among real procedures due to the specific clinical setup and tissue properties, and can be adjusted globally between repeated ultrasound simulations.

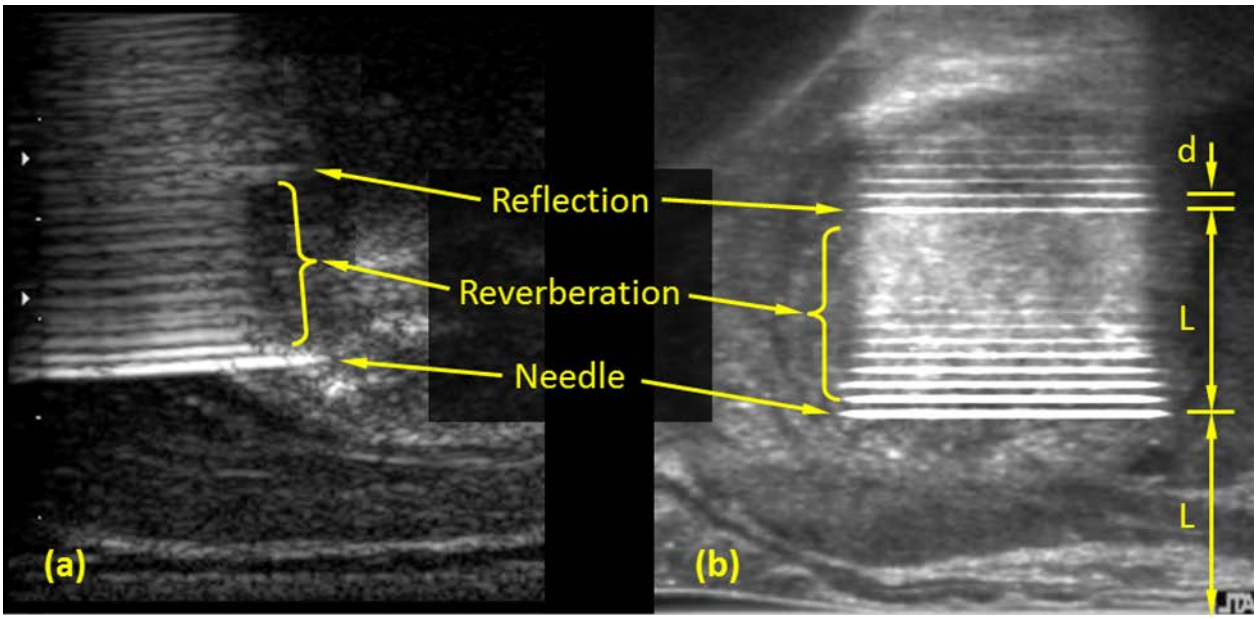

Figure 5-4: Sagittal view of cryoprobe artifacts: (a) ultrasound imaged needle in the prostate, and (b) integration of simulated cryoprobe artifacts with an undisturbed image of the prostate **[19]**. As shonw in **[76]**.

Figure 5-5 displays the corresponding artifacts in a transverse view. Similar to the results in Figure 5-4, cryoprobe reverberations are observed, radiating away from the US transducer with diminishing amplitude. Another feature displays in Figure 5-5 is the extended width of the cryoprobe image and its reverberations. The width of the imaged cryoprobe is 5 times its actual size, which is the result of the finite size of the focused ultrasound wave. Nevertheless, the same extended width effect is displayed both in the simulated image and the reference image. Note that the source ultrasound image used for Figure 5-5(b) is of higher resolution and quality with respect to Figure 5-5(a). Further note that the global intensity of each of the figures is adjustable.

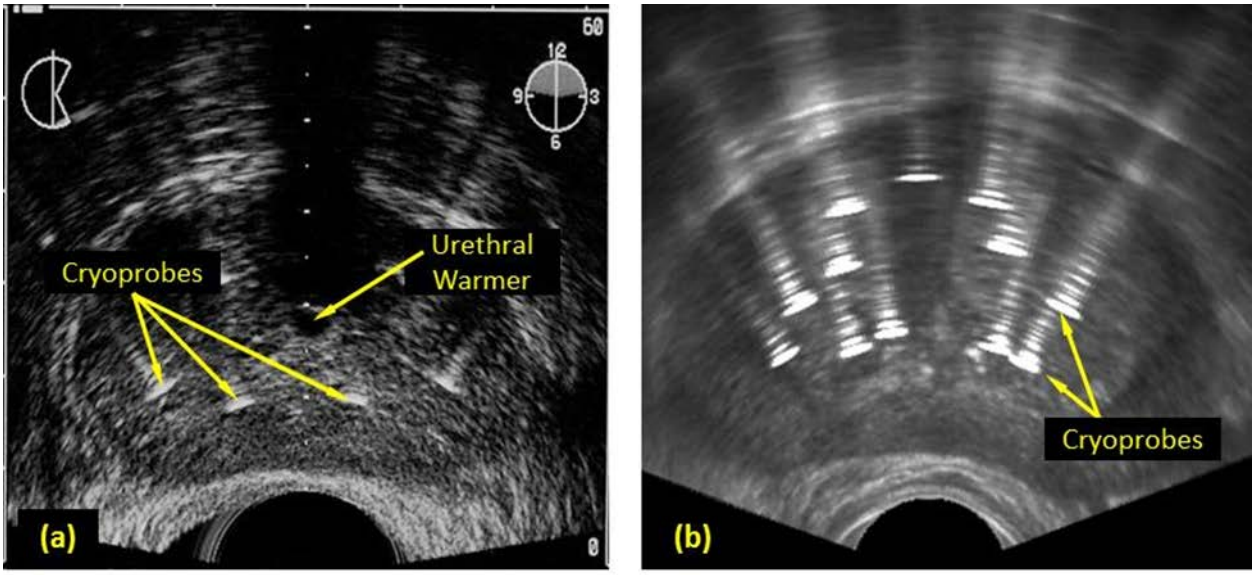

Figure 5-5: Transverse view of cryoprobe artifacts before the beginning of freezing: (a) ultrasound imaged cryoprobes placed within a prostate (with permission **[19]**), and (b) integration of simulated cryoprobe artifacts with an undisturbed image of the prostate **[76]**.

Probably the most important artifact to simulate for virtual cryosurgery training is the shadowing effect due to freezing [74,60]. In practice, this artifact is used to track the progression of the freezing front, from the cryoprobes towards the US transducer. This artifact is characterized by a hyper-echoic region at the freezing front, where the acoustic wave energy is reflected off due to the sharp change in the acoustic impedance between the unfrozen and frozen region. This artifact is further characterized by an opaque region behind the freezing front, because only a small portion of the wave energy is able to pass through the frozen region. Figure 5-6 displays the similarity between an actual transverse US image and a simulated image during cryosurgery. Note the granularity difference between both images displayed in Figure 5-6, which is due to the different US hardware used to create Figure 5-6(a) during cryosurgery, as opposed to the base texture created from an untreated tissue in Figure 5-6(b).

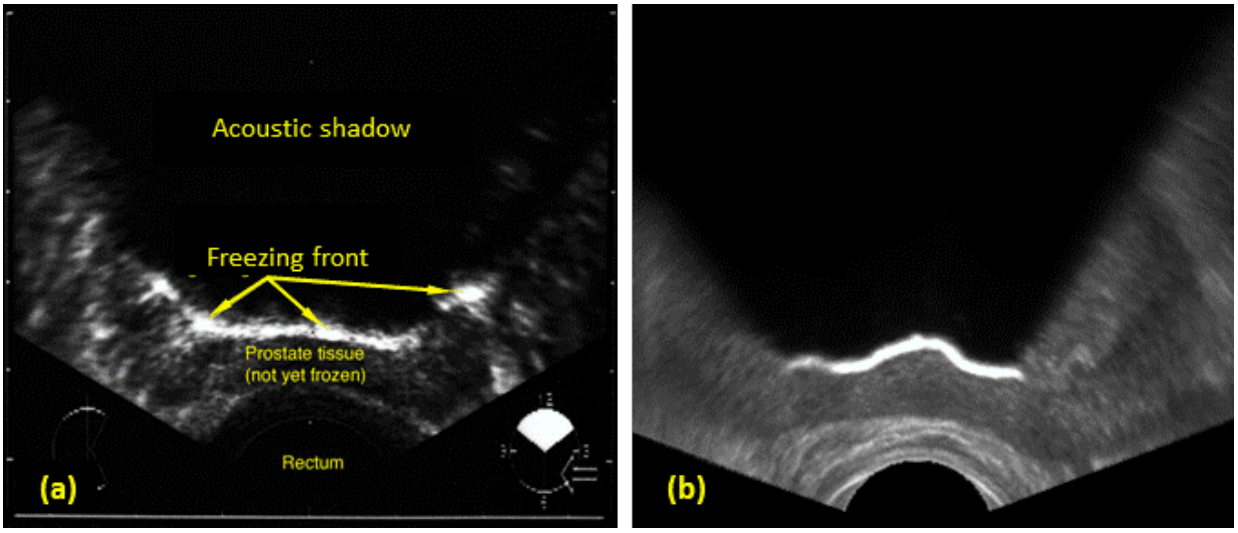

Figure 5-6: Transverse view of the frozen region during cryosurgery: (a) ultrasound imaged frozen region (with permission **[19]**), and (b) integration of simulated frozen-region artifacts with an undisturbed image of the prostate **[76]**.

Useful information can further be obtained from ultrasound imaging post cryosurgery, which is attributed to a reduction in echogenicity in the frozen-thawed region [74]. This artifact can help cryosurgeons track the regions of the tissue that have been frozen during the procedure, especially when multiple freeze-thaw cycles are performed. Figure 5-7(a)-(b) display this effect in a dog-prostate model, while Figure 5-7(c)-(d) display a somewhat similar effect in the current study. In order to produce the effect displayed in Figure 5-7(d), the lowest temperature achieved by each node during the heat transfer simulation is saved. Next, the localized echogenicity value is calculated based on the minimum temperature achieved, where frozen-thawed nodes receive an echogenicity value of 20% of that of a material that did not experience freezing at some point throughout the procedure.

Note that Figure 5-7(a)-(b) where created with an earlier US device (late 1980s) and a dog-prostate model, while Figure 5-7(c) has been created on a newer-generation device (mid 2000s) and a human-prostate model. These differences in hardware quality, prostate models, and

scanning procedures only permit a qualitative comparison of the post procedure effects on imaging. Nevertheless, the capability of the ultrasound simulation to capture the freezing thawing effect is important, and calibration of the material properties to create a more realistic presentation of this effect can be straightforwardly performed when comparable data becomes available. Note that the shadowing effect is not expressed when the entire tissue is unfrozen, in between freezing-thawing cycles.

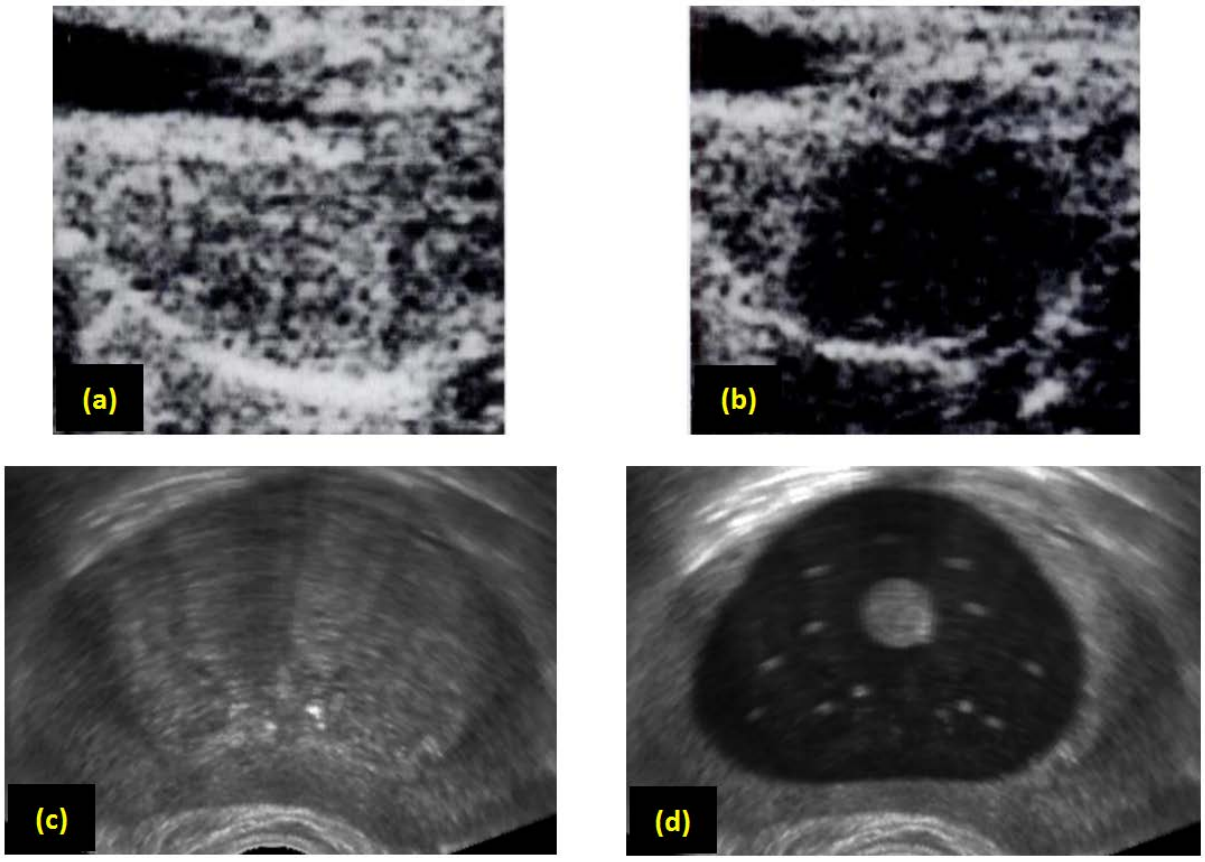

Figure 5-7: Transverse view of the prostate: (a) dog prostate before freezing, (b) dog prostate after thawing, (c) human prostate before freezing, (d) human prostate after thawing. Images for the dog prostate were taken before and after a real cryosurgical procedure (with permission **[74]**), while the human prostate post thawing is synthesized from a simulated thermal history of a cryoprocedure and an undisturbed image of the prostate **[76]**.

The benchmark cases described above were performed on a desktop computer (i7 960, 3.2 GHz, 9.0 GB RAM), housing a dedicated Nvidia GTX 580 GPU in real-time, at a rate of 100 fps. Between each two consecutive frames at that rate, the following processes where completed: (i) a complete 3D bioheat transfer simulation, (ii) imaging artifacts calculation for the entire field, and (iii) display of the results on the computer monitor. Toggling between the various presentations displayed in Figure 5-2 does not slow down those operations, which were performed on the same GPU. This performance makes the proposed mathematical formulation, combined with the proposed code implementation, excellent means for computerized training of cryosurgery. Furthermore, since the GPU implementation of the bioheat transfer alone is already two orders of magnitude faster than real time, with benchmarking data and scaling analysis presented previously [51], and since a complete ultrasound imaging artifacts calculations and display is done under 10 milliseconds (leading to a presentation rate of 100 fps), there is no apparent limitation to speed up the integrated bioheat-US simulation procedure described above by two orders of magnitude on the same machine.

5.4 Discussion

The current study presents a mathematical method for real-time simulations of ultrasound imaging artifacts. A six-step scheme is proposed to generate the imaging artifacts, and GPU-based real-time implementation strategies are discussed. The proposed implementation relies on a previously developed scheme for GPU-based bioheat transfer simulations of cryosurgery [51,32,31]. The temperature field generated by bioheat transfer simulations is used as an input for ultrasound artifacts calculations.

Simulated ultrasound imaging artifacts include reverberation and reflection of the cryoprobes in the unfrozen tissue, reflections caused by the freezing front, shadowing caused by

the frozen region, and tissue-property changes in repeated freeze-thaw cycles procedures. The intensity, brightness, and granulation of the recorded image may vary between clinical procedures, based on the specific ultrasound hardware, tissue properties, and scanning technique. Nevertheless, the simulated artifacts in the current study appear to preserve the key features observed in a clinical setting, which makes the proposed scheme suitable for cryosurgery training. It is proposed here that calibration means may further be developed, to create a wider trainee exposure to available ultrasound hardware and clinical conditions. This calibration is beyond the scope of the current study, and requires a survey of commonly available imaging devices.

This study displays an example of how training may benefit from toggling between (i) the undisturbed ultrasound image, (ii) the simulated temperature field, (iii) the simulated imaging artifacts, and (iv) a hybrid presentation of the temperature field superimposed on the ultrasound image to explain physical effects. The proposed scheme is demonstrated to simulate ultrasound-monitored cryoprocedure in real-time, at a rate of 100 fps, on a mid-range personal workstation. Simple calculations suggest that simulation runtime can exceed real-time cryosurgery operation by two orders of magnitude with no adverse effects.

Chapter 6: Computerized Planning for Cryosurgical Training

There has been significant work done to date on cryosurgical planning algorithms [20,21,23,27]. However, these algorithms have not accounted for current surgical practice. As a result, they are unsuitable for use in teaching clinicians how to generate more effective cryoprobe layouts while adhering to common surgical constraints. Currently surgeons are given general guidelines to determine acceptable cryoprobe configurations. An important rule for cryosugeons is to maintain a minimum distance between the active length of the cryoprobe and clinically relevant structures, like the urethra and the surface of the prostate [73]. Previous, heuristic methods of generating cryoprobe layouts such as bubble packing and the force-field analogy are able to quickly generate near optimal cryoprobe layouts. These methods are based on the premise of applying virtual forces to the cryoprobes and allowing the cryoprobes to move to minimize the total energy of the system. The addition of geometric constraints (such as enforcing a minimum distance to a surface) onto this force based system produces significant challenges that this work addresses. Additionally the integration of this work into an intelligent tutoring system prototype will be discussed.

6.1 Previous Work on Computerized Planning for Cryosurgery

The stated goal of most prostate cryosurgery procedures is to freeze and destroy the entire prostate gland, while keeping healthy tissue outside the prostate undamaged. In order to quantify the quality of a given cryosurgery, the idea of defect value has been previously presented [23],a small value of defect correlates to a high quality plan. The defect value shown in Equation 6-1 is a weighted sum of the total amount of healthy tissue outside the prostate that has been damaged due to freezing plus the amount of diseased tissue within the prostate that has been undamaged;

$$G = \frac{1}{V_t}\int_{V_s} wdV_s \;\; ; \;\; w = \begin{cases} 1 & -22℃ < T \;\; interior\ to\ the\ target\ region \\ 0 & T \leq -22℃ \;\; interior\ to\ the\ target\ region \\ 1 & T \leq -22℃ \;\; exterior\ to\ the\ target\ region \\ 0 & -22℃ < T \;\; exterior\ to\ the\ target\ region \end{cases} \tag{6-1}$$

where *G* is the defect volume, *w* is an associated volume weight, and *V* is volume. The simplest binary weighting function for the defect value is shown. More complicated functions that more heavily weight critical tissues can also be imagined. The locations that contain defects are essentially the areas in which there is mismatch between the lethal temperature isotherm and the prostate contour. Where Figure 6-1 depicts the typical defect regions at a given cross-section and described in the following equation.

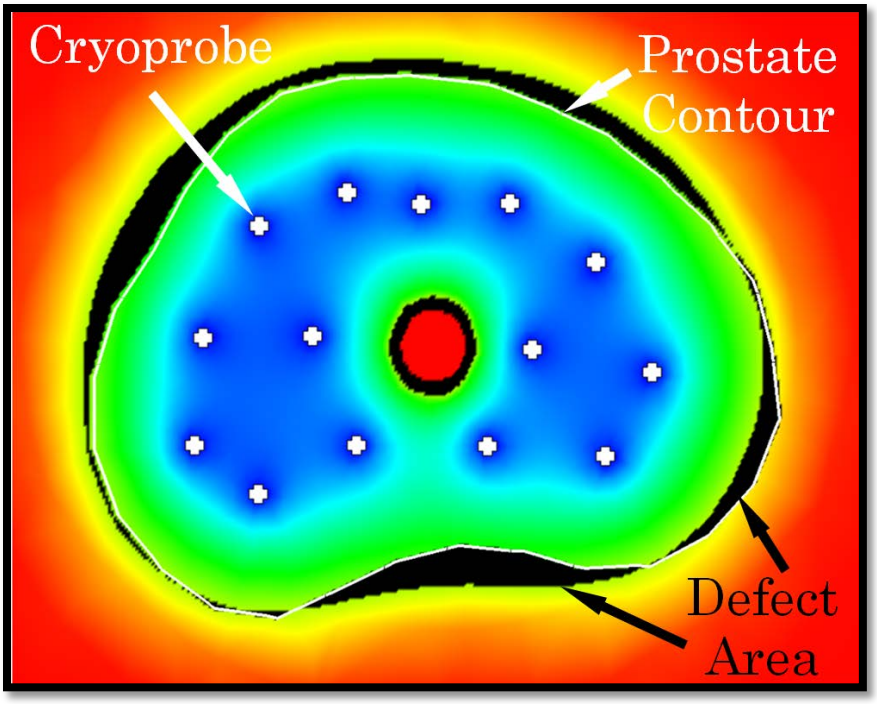

Figure 6-1: Defect area at a given cross-section **[78]**.

It has been previously shown that "bubble packing" is an effective method of generating high quality plans for prostate cryosurgery [26]. For completeness of presentation the basic algorithm will be presented below. Bubble packing places a number of fixed spherical bubbles (boundary bubbles) on the surfaces in which the cryoprobes will be constrained. For prostate cryosurgery, the constraint volumes are the areas inside the prostate and outside the urethra. Then a number of ellipsoidal bubbles corresponding to the number of cryoprobes are placed within the domain. A general depiction of bubble-packing is displayed in the following image (Figure 6-2), showing boundary bubbles on the surface of the prostate and urethra shown in green, and 8 ellipsoidal bubbles corresponding to a plan for 8 cryoprobes.

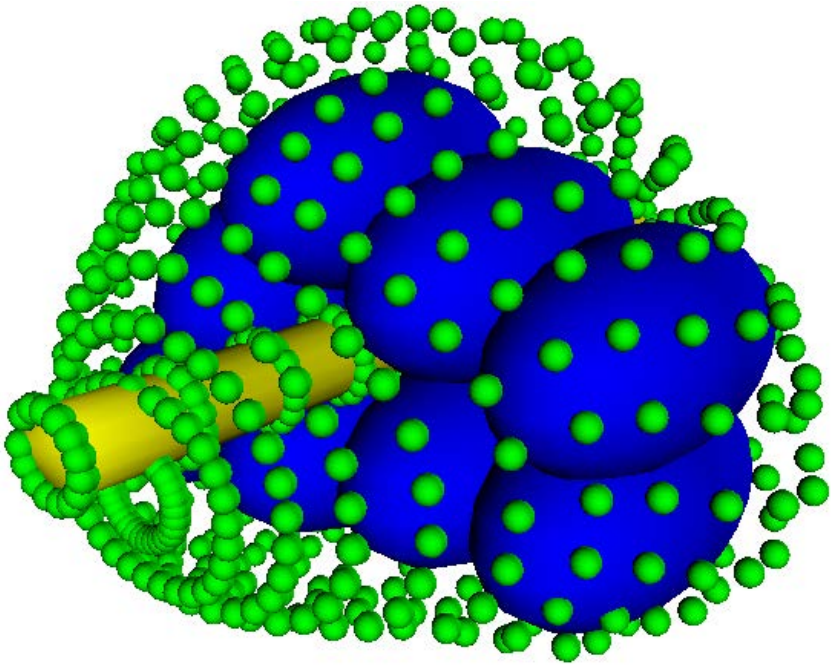

Figure 6-2: Depiction of bubble packing planning algorithm **[78]**

In the beginning of the simulation the ellipsoidal bubbles are placed within the domain. Then forces according to the Equation 6-2 and Figure 6-3 are imposed between the ellipsoidal bubbles and the boundary bubbles [75].

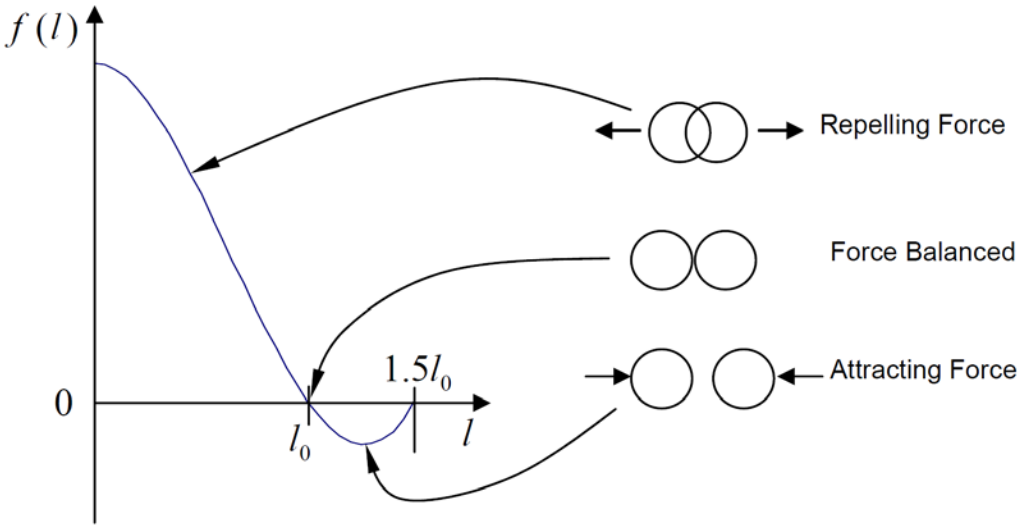

Figure 6-3: Inter-bubble force diagram **[79]**

$$f(l) = \begin{cases} al^3 + bl^2 + cl + d, & 0 \le l \le 1.5l_0 \\ 0, & 1.5l_0 < l \end{cases} \tag{6-2}$$

$$f(l_0) = f(1.5l_0) = 0, \quad f'(0) = 0, \quad f'(l_0) = -k_0$$

where l_0 represents the total distance between the centers of the bubbles divided by the sum of the radii of the two bubbles. The system behaves as a spring mass damper system according to the following equation (Equation 6-3).

$$m_i \frac{d^2 x_i(t)}{dt^2} + c_i \frac{dx_i(t)}{dt} = f_i(t), \quad i = 1 \dots n \tag{6-3}$$

where, *m* is the mass of the bubble, *c* is the damping coefficient, *x* is the position, and *f* is the external force imposed on the bubble. The damping ratio shown in Equation 6-4:

$$\zeta = \frac{c}{2\sqrt{mk}} \cong .7 \tag{6-4}$$

was found to be optimal because it produces quality plans in a reasonable amount of time. The system is then allowed to relax and reach its minimum energy state. Once this occurs the size of the bubbles are adjusted to optimize the overlap between bubbles. Bubble Overlap is defined by the Equation 6-5 and depicted Figure 6-4:

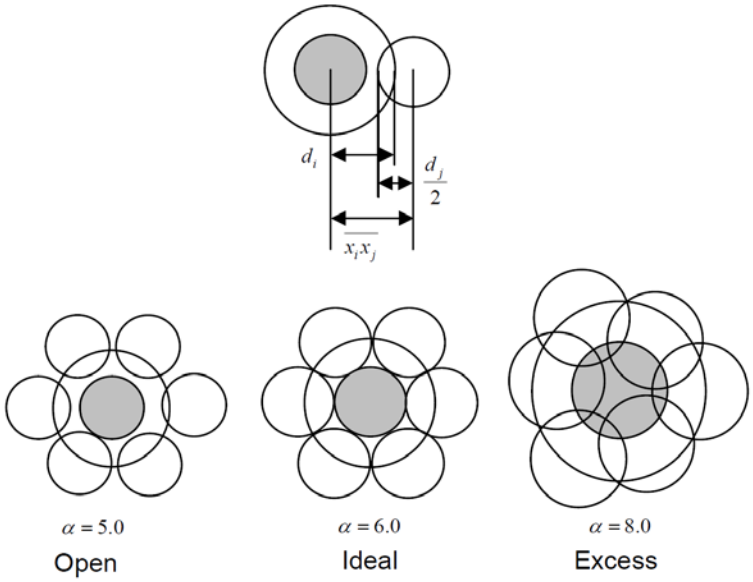

Figure 6-4: Overlap of bubbles. The ideal overlap ratio in the 2D case, α = 6.0, Bubbles are located too sparsely when α < 6.0, and too densely when α > 6.0.case, α = 6.0 [79],

$$\alpha_i = \frac{2}{d_i} \sum_{j=0}^{n} \left(d_i + \frac{d_j}{2} - \overline{x_i x_j} \right) \tag{6-5}$$

where α is the overlap ratio, d is the diameter of the bubble, i and j are the bubble indices, and x is the location of a bubble. The equation used previously to adjust the overlap ratio updates the diameter of the bubble such that the resulting overlap is closer to the ideal value. The simple update function is shown in the equation below (Equation 6-6).

$$d_{updated} = \frac{\alpha \cdot d_{current}}{\alpha_{ideal}} \tag{6-6}$$

This process is repeated until the maximum number of allowable iterations is hit or until an acceptable overlap ratio is found. The intuition behind this procedure is that each ellipsoidal bubble roughly corresponds to the lethal temperature isotherm which is produced by each

cryoprobe. Equally distributing these volumes of destroyed tissue within the prostate will cause maximum destruction to the undesired tissue while minimizing collateral damage to healthy tissue.

6.2 Addition of Constraints

Although the previously developed bubble-packing planner is able to create high quality plans in the absence of hard geometric constraints, significant modifications to the numerical scheme were needed in order to give satisfactory results. To begin, the simulation boundary bubbles are placed on the urethra and the surface of the prostate just as before. Then two constraint surfaces are added, one corresponding to the prostate and the other corresponding to the urethra. These two surfaces are offsets of the original surface models of the prostate and urethra by a user specified amount. Cryoprobe bubbles are placed the same as before as before, but with the additional constraint of having to lie between the offset constraint surfaces and not just within the original urethra and prostate geometry. A depiction of the constrained bubble packing setup is illustrated in Figure 6-5 with a few representative bubbles for clarity.

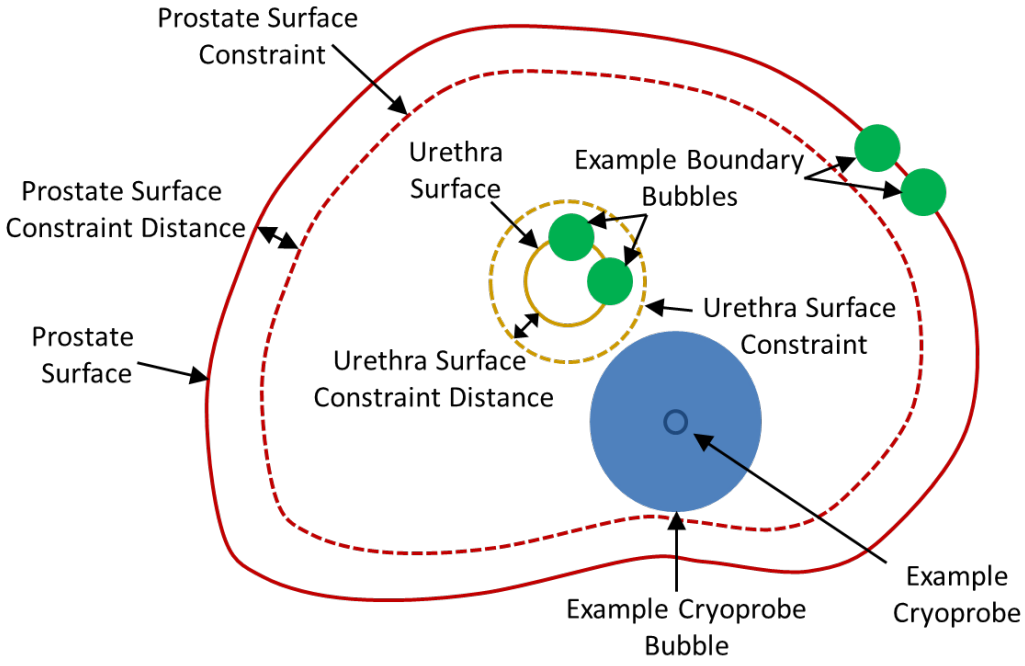

Figure 6-5: Depiction of constrained bubble packing setup

The simulation begins much the same as in the unconstrained case with the repelling and attracting forces being applied to each of the cryoprobe bubbles depending on its distance to neighboring bubbles. The bubble positions updated according to the spring mass damper equation previously presented, thus relaxing the system. The differences between the constrained and non-constrained implementations begin to arise as constraints are applied in the following step. First, collisions are detected between the cryoprobe and the constraint surfaces. If a collision is found, the cryoprobe is moved a minimum difference in the opposite direction of its current velocity until it is just within the constraint surface. This minimum distance is calculated using a bisecting line search in the direction opposite to the current velocity. Additionally, an inelastic collision velocity constraint is placed on the cryoprobe. With this constraint the component of the velocity that is in the normal direction of the surface at the point of contact is removed. This prevents the cryoprobe from violating the same constraint on the next iteration

while minimizing the numerical instability of the method. These constraints are iteratively checked every step in the simulation to ensure that the probe is not moved to a location that causes a constraint violation at another location on the surface. These steps of applying forces, updating the probe positions, and checking for/satisfying constraints are repeated until the minimum energy configuration of the system is reached.

Once the minimum energy configuration is reached, the overlap ratios for the bubbles are adjusted until they are closer to ideal. However, the additional hard geometric constraints placed on the location of the cryoprobes exacerbated a potential problem with the previously described bubble diameter update scheme. The issue is that the function for the overlap ratio is not continuous. As the size of the bubble increases and decreases, the neighbors that contribute to the calculation of the overlap ratio also changes. This can cause the overlap ratio to continually oscillate between too large and too small a value and never converge to the ideal solution. This problem did not surface previously, perhaps because without hard constraints, when the overlap ratio was too large, the bubbles were able to move away from each other and were less likely to acquire new neighbors. If no new neighbors are acquired, the function to calculate overlap is smooth and it is trivial to provide an update function that will iterate to the correct solution. To overcome this problem, the golden section line search was applied. This line search technique is advantageous because it maintains the size ratio of its stencil points which makes it well suited to finding the optimum value of a function.

6.3 Integration into Intelligent Tutoring System Prototype

In a collaborated study with Dr. Anjali Sehrawat, a simulation-based intelligent tutoring system (ITS) prototype for cryosurgery was developed [74,77]. The overall design and implementation of the intelligent tutoring system was the work of Dr. Sehrawat. Additionally Dr. Sehrawat also designed the studies to evaluate the effectiveness of the prototype system. This author's unique contribution was the implementation and integration of domain specific computerized planning and simulation tools into the intelligent tutoring system. The next sections describe the tutoring system created by Dr. Sehrawat, the effectiveness of the computational tools, and the outcome of the study.

6.3.1 ITS Learning Objective

ITS design is heavily dependent on the problem being represented, in this case cryosurgery. The ITS development in the current study focuses on planning of the cryoprobe layout for a cryosurgical procedure given a 3D shape of a prostate, while emphasizing thermal and geometrical considerations. The goal of this tutoring system is to teach clinicians how to create a cryoprobe layout that minimizes defects without violating minimum distance considerations between the cryoprobes and urethra, and between the cryoprobes and the surface of the prostate. This task consists of very intuitive (difficult to formalize) steps. Due to the very large parameter space of the resulting optimization problem it is nearly impossible for a clinician to come up with the optimal solution. As a result the clinician must use his or her intuition and acquired skill in order to come up with one of many acceptable solutions. For this prototype system, it is assumed the simulated patient is a good candidate for the cryosurgery treatment, and that the trainee is competent in the relevant medical imaging method. Despite these assumptions, it is still challenging to encode some of the highly intuitive concepts into computerized

instructions for an ITS. The tutoring in the two conducted studies combines trial-and-error interaction with customized ITS feedback. The feedback includes how to move cryoprobes in order to provide superior quality plans and to eliminate geometric constraint violations. In order to allow this functionality the previously presented planning algorithm was integrated. Additionally, the GPU accelerated bio-heat simulation was integrated into the tutoring system in order to allow the trainees to rapidly iterate and see the surgical outcomes their plans would produce.

6.3.2 ITS Architecture

The prototype cryosurgery trainer is comprised of two components: the ITS and the computerized cryosurgery tools (CCT), Figure 6-6. The CCT is comprised of the cryosurgery simulator and the cryoprobe-layout planner. The simulator takes a specific cryoprobe layout as input and uses the previously presented numerical scheme and GPU implementation to simulate the resulting bioheat transfer process in the treated tissue. The simulator's outputs are: the 3D temperature field at various times during the procedure and at the time of minimum defect, the mismatch between the planning isotherm and the shape of the target region, and the minimum defect value of the configuration. The cryoprobe-layout planner is based on the bubble-packing method, and is applied to the specific geometry of the target region (prostate and urethra) for a given number of cryoprobes.

The ITS has three components: a domain module, a student module, and a tutor. The domain module contains a set of rules and constraints that govern the cryosurgery planning process, combined with means to evaluate the trainee solution against the computer generated solution. The student module records the trainee actions throughout the problem solving process, including possible violations of rules and constraints. The tutor can be viewed as the

administrative portion of a physical instructor. It presents the student with a problem, asks relevant questions, passes the trainee-input to the domain module for evaluation, and returns feedback to the student based on the evaluator input. The database holds all case-related information, including organ geometries, previously solved cases, and student track record.

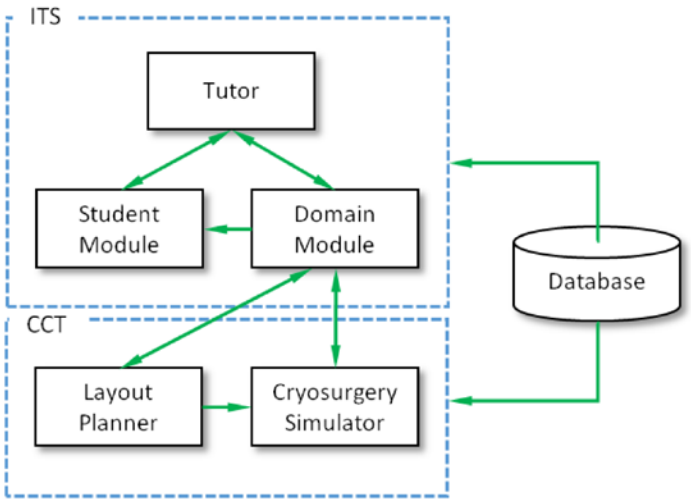

Figure 6-6: ITS prototype architecture **[77]**

6.3.3 Study Design

It is difficult to study the effectiveness of the computerized training concept since comparable data from current training methods (non-computerized apprenticeship based) is virtually nonexistent. To test the effectiveness of the ITS trainer, two studies were performed. The human subject studies were approved by the Allegheny Singer Research Institute, Allegheny Health Network (AHN), Pittsburgh, PA (IRB Protocol #14-006). The first study presents a novel benchmarking concept, where the ITS system performance is compared against the performance of a second but simplified computerized training system that does not contain ITS components. The non-ITS trainer is identical to the ITS trainer except it does not give feedback or guidance,

and it does not force the trainee to go through a sequence of instructional steps. Both training tools gave the users the ability to place probes, evaluate the 3D geometry of the problem, run bioheat simulations, inspect the resulting 3D temperature field, and evaluate the defect results relative to the computer-generated solution. The ITS tutor has additional functionality. It provides instructions on how the user can improve their layout using the constrained planning algorithm. Furthermore, the ITS tutor alerts the user when he or she violates geometric constraints. Both groups were given six cases for training and three cases for post-training test. For each case, the trainee was presented with a prostate and urethra model and was asked to place ten cryoprobes of a fixed length at a fixed insertion depth, effectively setting the x and y location of all the probes.

A second study was performed in which all students were given a version of the ITS without the iterative probe location feedback. Three pre-test cases were administered, in which the user had only a single attempt to complete the test. No feedback was given for these test cases. The students were then given 50 minutes to complete as many as 6 practice cases with the tutoring system to try and improve their skills. At the end of the 50 minutes, a post-test was conducted, consisting of the same three test cases originally given as the pre-test. For this study the problems were made more difficult. Here, the users were not constrained to place the probes at a fixed depth, now each probe could be placed at its own z depth [73].

6.4 Results and Discussion

The results of the first study showed that both groups, were able to improve the quality of their plans, in terms of reducing the violation of constraints while maintaining acceptable defect values. Although the difference in performance between the ITS and non-ITS was negligible, the group that performed the exercise with the ITS needed fewer iterations for each test case. A reduction in the number of attempts needed to learn a particular skill is by itself a valuable contribution. In the second study, individuals were able to significantly improve their performance with only 50 minutes of exposure to the training tool. In this study, 80% of individuals improved on case one while roughly 67% of individuals improved on cases two and three. It is very difficult to baseline these results because as there is no computerized alternative to cryosurgical training. Still, the users were able to significantly improve their planning skills in a relatively short period of time. using the ITS, the users saw a variety of surgical cases and were able to rapidly iterate and receive feedback while improving the quality of their surgical plans [73]. A key element in the computerized training experience is the ability to quickly simulate the surgical outcome based on a specific configuration of cryoprobes. The work performed in this thesis to speed-up simulation run-times greatly improved the user experience and allowed users to more quickly iterate to an acceptable solution. The success of the ITS in this collaborated study demonstrates the effectiveness of the computational training tools developed in this thesis.

Chapter 7: Summary and Conclusion

The goal of minimally invasive cryosurgery is to maximize destruction to diseased tissue within the target region, while minimizing collateral damage to the healthy surrounding tissue. Cryodevice manufacturers have designed increasingly advanced tools following trends in cryoprobe miniaturization in order to help clinicians control the size and shape of the frozen region. Although modern cryoprobes offer the clinician more control, they also add to the complexity of the procedure because they require more probes. The positions of these probes must be optimally planned and placed, and the cooling power must be monitored during the procedure. It is the view of the author that modern cryosurgical devices have advanced far beyond current surgical practices. These devices could be used with far greater effectiveness if clinical training was more efficient. Currently, clinicians go through an extensive residency process in order to learn how to perform this surgery from an experienced surgeon. This process is extremely costly and time consuming. Computerized training tools for cryosurgery could make this training process more efficient and help clinicians gain the surgical skills necessary to perform the surgery in a safe virtual environment.

As part of an ongoing effort by the Biothermal Technology Laboratory and Computational Engineering and Robotics Laboratory at Carnegie Mellon University, a long term goal has been set to develop a computerized trainer for cryosurgery. Toward that goal, the work presented in this thesis is focused on the development of the computational tools that would enable users to practice performing the surgery in a virtual environment and would provide users feedback on their performance. Note that many of these techniques have been previously published in greater detail. While these tools have been presented in the context of prostate

cryosurgery, they can be applied to any minimally invasive cryosurgical procedure that uses US as the primary imaging modality.

This work has presented the minimum set of requirements necessary for an effective cryosurgical trainer. An effective training tool must be able to select useful test cases based on the past performance of the user. Due to the lack of extensive patient libraries, the tool must then be able to generate the selected test case from a small library of data. The tutor should allow the user to segment the necessary geometry from the synthesized data and should provide feedback on their performance. The user generates a surgical plan based on the segmented geometry. The user will be given feedback on how to improve their planning technique based on comparisons to the computer generated optimal plan. To enable the user to hone their intraoperative skills, the system should be able to realistically simulate the procedure. It must take into account the important heat-transfer and imaging effects. Finally, the system should be able to score the entire procedure based on the expected clinical outcome. This thesis work has presented a software structure that would be able to address these many requirements.

In previous work a heat transfer simulation scheme was presented. It was based on the FD method that was specifically designed for cryosurgery. This numerical scheme significantly reduced the amount of numerical computation necessary to perform a cryosurgical simulation. However, the previous study did not do a direct comparison to the runtime and accuracy differences between the FD scheme and more commonly used FE programs. To address this, a simulator using the FE program Ansys was created and benchmarked against the FD implementation. It was shown that the FD method was able to achieve a comparable level of accuracy while significantly decreasing runtimes. This difference in speed is attributed to the fact that the FE program has extensive preprocessing operations needed to generate the mesh used for the simulation. Additionally, the implicit FE scheme excels at taking large time steps while

maintaining accuracy. However, smaller time steps are needed in order for the tool to be clinically relevant, thus further increasing the simulation time of the FE scheme. As a result of this work an additional study was completed to further enhance the speed of the FD scheme.

The FD scheme was previously optimized to run on a multi-core CPU. To further improve runtime, the numerical technique was implemented on the GPU using C++ AMP. Before beginning the study, it was determined that simulation run time would benefit from a GPU implementation. The memory access patterns for the FD scheme closely resemble those of a sparse matrix vector (SPMV) kernel operation. These SPMV kernels have previously been shown to map well onto GPU architecture. After an initial GPU implementation and subsequent optimization of the FD scheme a 15x performance improvement was realized over the previous implementation. This allowed a full 200s surgical simulation to be run in under 2s. This dramatic reduction in simulation run time opens up new applications for surgical simulations. This enhanced surgical simulation speed was leveraged in a later study which simulated intraoperative US artifacts. Some of the artifacts used by clinicians are temperature field based. As a result, heat transfer and ultrasound simulations must occur in real-time in order to be effective.

Next an algorithm was presented for generating synthetic 3D ultrasound images from surface models of relevant geometry and an example segmented 3D US. This method is based on a method called “Image Analogies” which was has been shown to generate realistic 2D textures. The work in this thesis extended the method into 3D and modified it for US imaging applications. Through the use of this method realistic US images can be generated for training purposes.

Once a synthetic US image is generated, there is an additional need to simulate the clinically relevant artifacts that the clinicians rely upon during the surgery. Surgeons use “comet tail” and reverberation artifacts to place cryoprobes within the tissue. Additionally, shadowing

and reflection artifacts are used to help track the progression of the freezing front during the surgery. To address this training need, an algorithm was presented that simulates these artifacts based on ray tracing. Unlike previous ray-tracing algorithms for US simulations, the presented method is able to simulate multiple reflections in a continuous media. This capability is critical for cryosurgical applications as the simulation of both cryoprobe and thermal field artifacts should be simulated using a single unified method to ensure their accurate interaction.

Finally, previous surgical planning methods were unable to cope with clinical constraints. It is common practice in cryosurgery today to set geometric constraints on the cryoprobes in order to ensure that they are a safe distance away from critical healthy tissue. Many of the advanced heuristic methods based on the application of virtual forces were not able to cope with these geometric constraints on cryprobe placement. To address this issue, the work presented changes to the previously developed planning scheme that would allow the integration of these forces. First, the method for adaptive size control of cryobubbles was modified to guarantee an adequate bubble size could be reached in a finite number of iterations. Finally, the geometric constraints were enforced by detecting collisions between cryoprobes and constraint surfaces, then moving the cryoprobes a minimum distance to resolve the collision and satisfy the constraints.

A collaborative study was performed with Dr. Sehrawat that employed the improved planning and simulation techniques presented in this thesis. These tools were used in a prototype training tool and tested on a number of medical residents. The results of the study showed improved outcomes to the users who were exposed to these advanced simulation and training tools. The work presented in this thesis allowed the users in the study to receive feedback on their proposed surgical plans and was able to rapidly simulate the outcome of a surgical

procedure. This rapid simulation capability allows users to iteratively improve their surgical plans more quickly.

As cryosurgical devices and techniques continue to advance, the complexity of cryosurgical planning will increase. This trend will demand advanced tools for surgical training. The work presented in this thesis, combined with other studies at the Biothermal Technology Laboratory, has led to the development of many foundational tools necessary for a cryosurgical training tool that has a virtual training environment. It is the hope of the author that these foundational tools will lead to more efficient and effective training for clinicians.

Chapter 8: Proposed Future Work

The previous chapters focused on the computerized training tools developed in this work. In this chapter the future direction of computerized training for cryosurgery will be discussed, including the presentation of a preliminary study. The preliminary study presents an example of the kind of tools that can be built off of the foundation presented in this work. The goal of this future work is to increase the effectiveness of the previously presented algorithms as well as to blur the line between effective training and intraoperative tools.

8.1 Increase Efficiency of Synthetic Ultrasound Generation

Future work in the area of synthetic US image generation should focus on creating more efficient models for 3D texture synthesis. The method proposed in this thesis relies on saving feature vectors for every voxel in the sample images. Similar to the original “Image Analogies” algorithm, when a new image is being generated it will be filled in a voxel at a time by finding the voxel in the input with the closest feature vector to the current voxel in the output being synthesized [76]. In the past many approaches to texture synthesis have been presented that attempted to build a model of the sample image through machine learning algorithms [77,78]. Typically these algorithms begin with the hand selection of a number of discriminative filters these filters are applied to the entire image and a feature vector is formed based on this compressed representation provided by the filters. These same filters are then applied around the next element in the output image that is being synthesized, and a value for that element is calculated using one of the following methods. One method is to use an element value from the input image that has the most similar feature vector to the location in the output image [59]. Other methods use artificial neural networks, support vector machines to build a model to predict

the output element value from the filter values [79,80]. However, these methods have had limited success despite their potential due to the inability to learn suitable representations for certain textures. Often times, these methods tend to "grow garbage" meaning a few poorly generated elements cause a chain reaction that produces a poor quality texture in the rest of the image.

As of this writing there has been a large movement toward a field of machine learning called deep learning [81]. The process of deep learning is in many ways similar to older machine learning techniques. However, deep learning is able to learn much more expressive and compact models than previous approaches. Using this technique the statistics of the original problem can be more effectively captured. These kinds of models tend to require a lot of data to train and it is unclear whether it would be able to generate a sufficient model from the available sample data that has been collected. These models also require a significant amount of time and computational power to train. However, once they are trained, they can generate data very quickly [82]. Future investigations should focus on implementing and training this class of potentially more effective texture synthesis methods.

8.2 Experimental Optimization and Validation of Ultrasound Imaging Algorithm

The algorithm previously presented for simulating cryosurgical imaging artifacts was designed to capture all of the relevant physical effects. As a result it was able to produce images that looked qualitatively similar to actual ultrasound images. However, there are a number of effective material properties used within the algorithm that require additional tuning. These properties include the effective echogenicity, acoustic impedance, speed of sound, and energy absorption properties of the frozen tissue and cryoprobes. Consequently, future work should include an optimization and validation study to set these physical properties and verify the applicability of the algorithm.

This optimization and validation study should consist of a physical experiment. A diagram of this experiment is shown in Figure 8-1. In this experiment a slab of tissue, or a tissue phantom is placed between two clear plastic plates and it will be frozen in much the same way as the actual surgery. A regular grid of holes will be drilled in the plate so that cryoprobes (blue) and thermocouples (green) can be inserted through the tissue in the same interval as a surgical registration grid. In the center, of the grid, a larger hole will be drilled to allow for a larger diameter plastic tube to be placed. This platic tube will allow a warm saline solution to circulate, to simulate the warming catheter in the actual surgery (yellow). Below the cryoprobes grid an additional hole will be drilled for an ultrasound imager to be placed (orange). Finally, some distance away a camera will be placed so that it can image the tissue as it is freezing. The experiment begins with the entire apparatus at room temperature. The cryoprobes are cooled using a protocol as close to the ones used in surgical conditions as possible. While the tissue is freezing, it will be imaged simultaneously by the US transducer as well as the color camera. Additionally, temperature measurements within the domain should be taken. Once the data is collected, a method such as the one developed by [42] can be used to reconstruct the temperature at various times throughout the experiment from the temperature measurements taken by the thermocouples, as well as the freezing front captured by the color camera. Once the temperature field has been reconstructed, it can be fed into the artifact simulation algorithm, then an optimization technique such as a quasi- newton method can be used to vary the unknown material properties so that the artifacts generated by algorithm match the actual artifacts captured by the US imager during the experiment. Once these optimized parameter are found there value as well as the validity of the imaging technique can be verified by performing a similar experiment in which the cyroprobe layout and the cooling protocols are different from the original experiment. If the algorithm still produces results that match those taken by the US

imager then we can assume the method and material properties are correct and capable of simulating a wide range of cryosurgical applications.

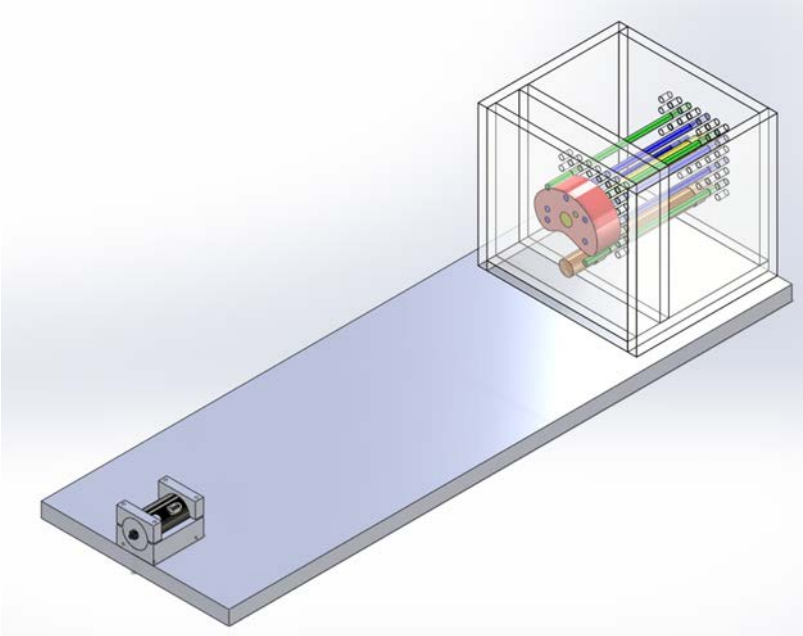

Figure 8-1: Experimental setup for the optimization of TRUS simulator

8.3 Leverage Efficient Bioheat Simulation Implementation

The efficient bioheat simulation implementation on the GPU allows for the development of many more practical cryosurgical tools. Now classes of problems such as partial differential equation constrained optimizations on the temperature field can be solved more effectively [53]. In the case of prostate cryosurgery, the quantity to be minimized could be defect value, and the constraints are the equations that govern the behavior of the temperature field. With this class of problems, most solution methods are iterative and require the simulation of the PDE to be carried out many times. As a result, a fast simulation tool could be invaluable to a training or intraoperative cryoprobe protocol planning algorithm.

Another application of a fast heat transfer simulation algorithm is that of intraoperative temperature field reconstruction. Studies have shown that prostate cryosurgical outcomes are dramatically improved by the use of temperature sensors during surgery [14,15,6]. These temperature sensors allow the clinician to more accurately determine the location of the freezing front in the shadowed regions of the ultrasound created by the frozen tissue. The goal of this research is to use the tools developed for surgical training and apply and extend their capability to both consolidate and enhance the information presented to the clinician during surgery. It is envisioned that this tool could augment intra-operative TRUS images with the best estimate of the temperature field and give the clinician a concise view of the current temperature state within the domain (See Figure 8-2 for illustration).

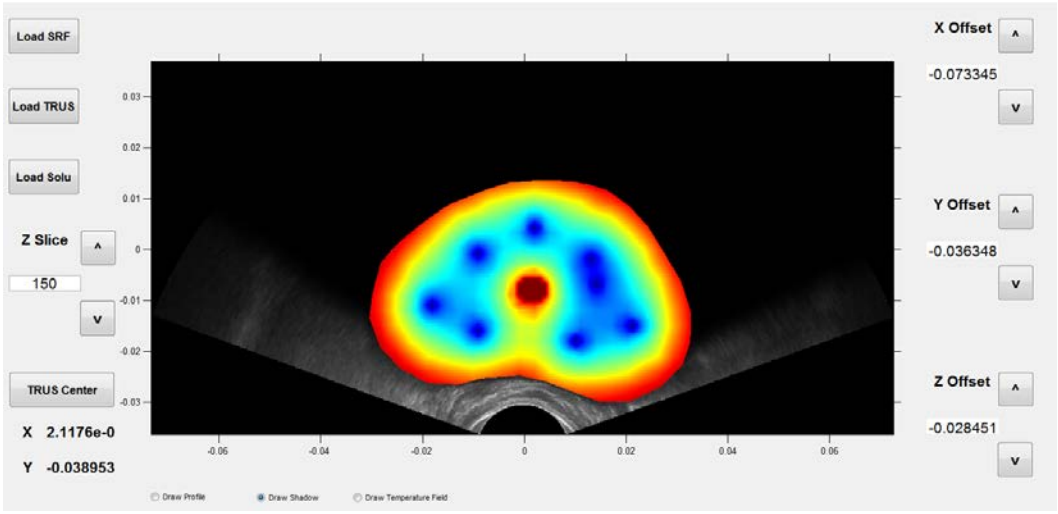

Figure 8-2: Depiction of augmented TRUS image with temperature field data

Very recently, algorithms been developed to track the location of the lethal isotherm from temperature probe readings and ultrasound shadowing [42]. Although these algorithms have been effective in some circumstances, they are based on heuristics which can potentially introduce inaccuracies into the estimation.

The proposed solution is to use approximate optimal filtering, specifically the ensemble-Kalman filter [83]. The ensemble-Kalman filter is similar to the traditional Kalman filter whose equations are shown below. The only difference is that the propagation of the Gaussian state distribution is calculated from an "ensemble" of sampled possible states instead of being explicitly updated through linearized statistical equations.

The ensemble-Kalman filter is more robust to non-linearities than both the traditional Kalman filter and the extended Kalman filter because there is no need to linearize the solution. However, the real benefit of the ensemble-Kalman filter is that it often has a much lower computational cost than the traditional Kalman filters. In traditional Kalman filtering, a large system of equations must be solved in order to propagate the current state estimate forward in time. However, with ensemble-Kalman filtering, the state is propagated explicitly and there is no

need to solve a large system of equations. However, one needs to perform a state update for each of the ensemble members. Still, the number of members of the ensemble is often found to be much less than the number of states resulting in less computational cost.

The Ensemble-Kalman Filter has been proven to be effective in non-linear high dimensional sparse systems similar to the ones used in the simulation of cryosurgery such as weather forecasting. The algorithm has the following basic form as shown in Table 8-1. To begin, a sample of likely states is initialized, which is called an ensemble. The ensemble is then propagated in time according to the state update equation. The mean and covariance of these states is then calculated and measurements of the system are taken. Using the current state distribution and the measurements, each ensemble member can be updated to reflect the higher certainty of the current state based on the measurements. This process repeats as the states are propagated to the next time step. A sample of an update step in the ensemble Kalman filter Figure 8-3.

Table 8-1: Kalman filter equations

State Update Equation	$x_k = F_k x_{k-1} + B_k U_k + w_k$	
Measurement Equation	$z_k = H_k x_k + v_k$	
Prediction Equations	$x_{k\|k-1} = F_k \hat{x}_{k-1\|k-1} + B_k u_k$ $P_{k\|k-1} = F_k P_{k-1\|k-1} F_k^T + Q_k$	
Update Equations	$\hat{y}_k = z_k - H_k \hat{x}_{k\|k-1}$ $S_k = H_k P_{k\|k-1} H_k^T + R_k$ $K_k = P_{k\|k-1} H_k^T S_k^{-1}$ $\hat{x}_{k\|k} = \hat{x}_{k\|k-1} + K_k \tilde{y}_k$ $P_{k\|k} = (I - K_k H_k) P_{k\|k-1}$	
Variable Description	Variable	Description
	x_k	Current State
	F_k	State Transition Model
	x_{k-1}	Previous State
	B_k	Control Input Model
	U_k, u_k	Control Input
	w_k	Process Noise
	z_k	Current Measurement
	H_k	Measurement Model
	v_k	Measurement Noise
	$x_{k\|k-1}$	Current State Given Previous
	$\hat{x}_{k-1\|k-1}$	Updated Previous State Given Previous
	$P_{k\|k-1}$	Current Probability Distribution Given Previous
	$P_{k-1\|k-1}$	Updated Previous Probability Distribution Given Previous
	Q_k	Process Noise Covariance
	$\hat{y}_k$	Current Predicted Measurement Error
	H_k	Measurement Model
	$\hat{x}_{k\|k-1}$	Predicted Current State Given Previous State
	S_k	Innovation Covariance
	R_k	Measurement Noise Covariance
	K_k	Kalman Gain Matrix
	$\hat{x}_{k\|k}$	Updated Current Predicted State
	$\hat{x}_{k\|k-1}$	Current Predicted State Given Previous State
	$P_{k\|k}$	Updated Current Predicted Probability Distribution

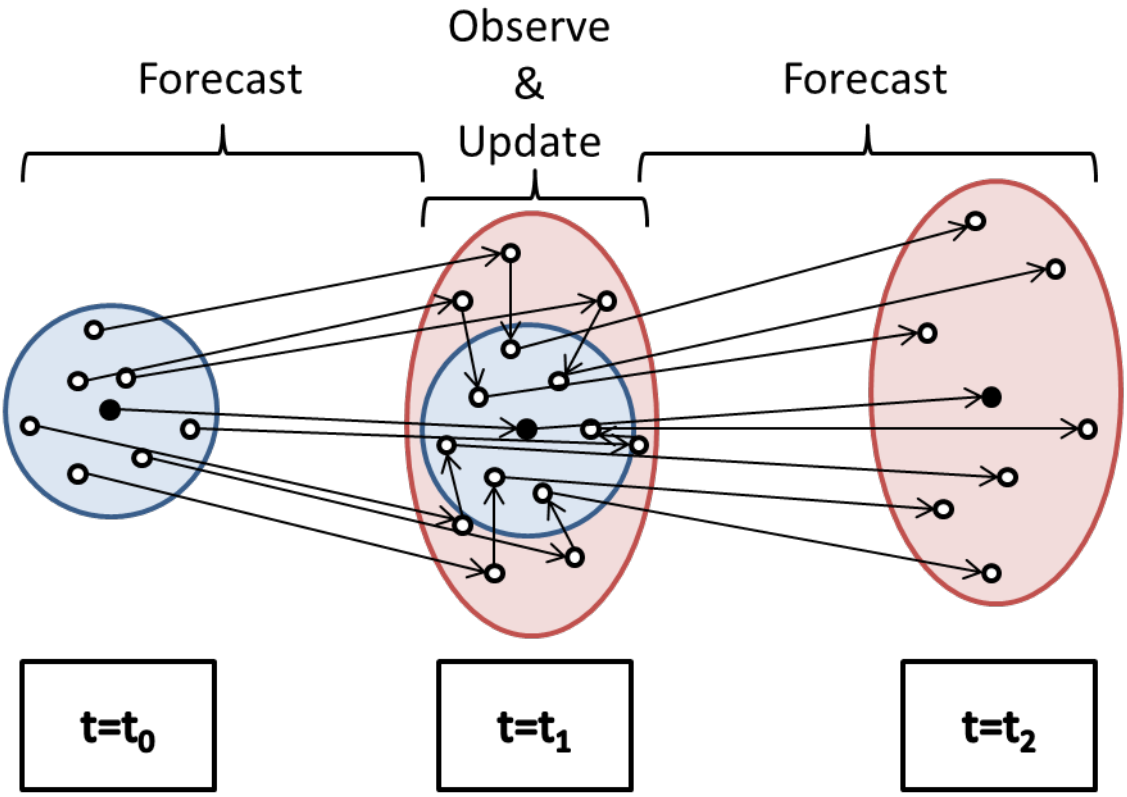

Figure 8-3: Depiction of Ensemble-Kalman filter

A proof of concept 2D version of the tool has been developed as a class project. This tool was developed in Matlab on a smaller system domain than what will be used in the final system and only included temperature probe readings and no imaging data. The system consisted of a 12x12 grid with the same material properties used to model frozen tissue in the previous chapters of this thesis. The domain was initialized with a temperature distribution shown in Figure 8-4. The system was allowed to naturally equilibrate back to body temperature. Four virtual temperature sensors were placed at grid locations (4, 4), (8, 4), (4, 8), and (8, 8). An initial

temperature estimate for the filter was set with a negative 10K estimate from the actual temperature distribution at the start of the simulation.

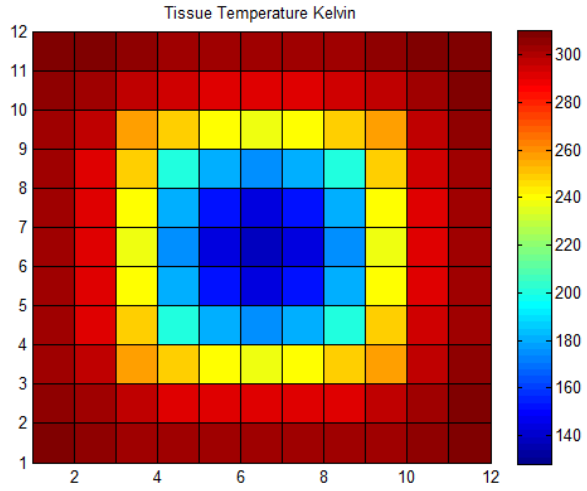

Figure 8-4: Initial state of experimental Ensemble Kalman Filter system

This work showed that temperature field estimates were improved by 30% over the uncorrected simulation estimates given the poor initial conditions. This is very promising as a starting point for future work.

8.4 Combine Computational Tools into a Virtual Surgical Environment

The ultimate goal of this work is to create a computerized tutoring system with the ability to provide a realistic 3D environment for clinicians to practice in. Many of the components for such a system have been developed by this author [51,32] and others [29]. However, there remains a significant amount of work to integrate these components into one cohesive system. A proof of concept version of such a system was implemented by Sehrawat [73], however, this implementation was a prototype and did not integrate all of the functionality necessary for a full virtual environment. For instance, the work done in this thesis in which the user can practice performing the surgery using realistic US images and artifacts was not integrated.

As of the time of this writing, there has been resurgence in the field of virtual and augmented reality. It is the hope of this author that the foundational simulation and rendering tools in this thesis can be used in conjunction with these virtual reality technologies in order to provide the clinicians with an immersive and realistic environment in which to practice performing cryosurgery.

List of Publications

Keelan, R., Shimada, K., Rabin, Y. (2011): Developing a framework for computerized training of cryosurgery based on finite elements analysis. ASME 2011 Summer Bioengineering Conference - SBC 2011, Farmington, PA, USA (June 22-25)

Keelan, R., Yamakawa, S., Shimada, K., Rabin, Y. (2012): Simulation of prostate cryosurgery for the purpose of surgical training: system design and benchmarking, 2012 Annual Meeting of the Biomedical Engineering Society, Atlanta, GA (October 23-27)

Sehrawat, A., Keelan, R., Shimada, K., Rabin, Y. (2013): Development of an Intelligent Tutoring System for Prostate Cryosurgery. CRYO2013-the 50th Annual Meeting of the Society for Cryobiology, N. Bethesda, DC (July 28-31). Cryobiology, 67(3):425-426

Keelan, R., Shimada, K., Rabin, Y. (2013): Generating synthetic ultrasound imaging artifacts for the purpose of computerized training of cryosurgery. CRYO2013-the 50th Annual Meeting of the Society for Cryobiology, N. Bethesda, DC (July 28-31). Cryobiology, 67(3):426

Keelan, R., Yamakawa, S., Shimada, K., Rabin, Y. (2013): Computerized training of cryosurgery - a system approach. Cryoletters, 34(4): 324-337

Keelan, R., Zhang, H., Shimada, K., Rabin, Y. (2014): Imaging artifacts, training interface and computational challenges, ACCryo 2014 – Annual Meeting of the American College of Cryosurgery and the Society for Cryobiology, Key Largo, FL (January 15-19); Cryobiology, 69(1):187

Keelan, R., Zhang, H., Shimada, K., Rabin, Y. (2014): Simulated ultrasound imaging for the purpose of prostate cryosurgery training, STM 2014 – Annual Meeting of The Society for Thermal Medicine, Minneapolis, MN (May 6-10)

Rabin, Y., Shimada, K., Keelan, R., Sehrawat, A., Zhang, H. (2014): Computerized training objectives and prototyping, ACCryo 2014 – Annual Meeting of the American College of Cryosurgery and the Society for Cryobiology, Key Largo, FL (January 15-19); Cryobiology, 69(1):187

Rabin, Y., Shimada, K., Keelan, R., Sehrawat, A., Zhang, H. (2014): Computerized training tools for cryosurgery, STM 2014 – Annual Meeting of The Society for Thermal Medicine, Minneapolis, MN (May 6-10)

Sehrawat, A., Keelan, R., Shimada, K., Wilfong, D.M., McCormick, J.T., Rabin, Y. (2015): Simulation-based cryosurgery intelligent tutoring system (ITS) prototype, Technology in Cancer Research and Treatment, published online (May 3), doi:10.1177/1533034615583187; PubMed http://www.ncbi.nlm.nih.gov/pubmed/25941163

Keelan, R., Zhang, H., Shimada, K., Rabin, Y. (2015): GPU-based bioheat simulation to facilitate rapid decision making associated with cryosurgery training, Technology in Cancer Research and Treatment, published online (May 3), doi: 10.1177/1533034615580694; PubMed http://www.ncbi.nlm.nih.gov/pubmed/25941162

Keelan, R., Shimada, K., Rabin, Y. (2015): GPU-based simulation of ultrasound imaging artifacts for cryosurgery training, Submitted

References

[1] A A Gage and J Baust, "Mechanisms of tissue injury in cryosurgery," *Cryobiology*, vol. 37, no. 3, pp. 171-186, 1998.

[2] J. Farrant and C. A. Walter, "The cryobiological basis for cryosurgery," *The Journal of dermatologic srugery and oncology*, vol. 3, pp. 403-407, 1977.

[3] G M Fahy, "Analysis of 'solution effect' injury: cooling rate dependence of the functional and morphological sequellac of freezing in rabbit renal cortex protected with dimethyl sulfoxide," *Cryobiology*, vol. 18, pp. 550-570, 1981.

[4] K Muldrew and L E McGann, "The Osmotic rupture hypothesis of intracellular freezing injury," *Biophysical Journal*, vol. 66, pp. 532-541, 1994.

[5] J G Baust and A A Gage, "The Molecular Basis of Cryosurgery," *the British Journal of Urology International*, vol. 95, pp. 1187-1191, 2005.

[6] K R Han et al., "Treatment of organ confined prostate cancer with third generation cryosurgery: preliminary multicenter experience," *Journal of Urology*, vol. 170, no. 4 Pt 1, pp. 1126-1130, 2003.

[7] K Tatsutani, B Rubinsky, G Onik, and R Dahiya, "Effect of thermal variables on frozen human primary prostatic adenocarcinoma cells," *Urology*, vol. 48, no. 3, pp. 441-447, 1996.

[8] Y Rabin, T B Julian, P Olson, M J Taylor, and N Wolmark, "Long-term follow-up post-cryosurgery in a sheep breast model," *Cryobiology*, vol. 39, pp. 29-46, 1999.

[9] H H Whitehouse, "Liquid air in dermatology; its indications and limitations," *Journal of the American Medical Association*, vol. 49, pp. 371-377, 1907.

[10] I Cooper and A Lee, "Cryostatic Congelation: A System for Producing A Limited, Controlled Region of Cooling or Freezing of Biologic Tissues," *Journal of Nerve and Mental Disease*, vol. 133, no. 3, pp. 259-266, 1961.

[11] Ralph E. Crump, Frank L Reynolds, and Carl R Tillstrom, 3,613,689, 1970.

[12] D Torre, "Alternate cryogens for cryosurgery," *Journal of Dermatologic Surgery*, vol. 1, pp. 56-58, 1975.

[13] G M Onik et al., "Transrectal ultrasound-guided percutaneous radical cryosurgical ablation

of the prostate," *Cancer*, vol. 72, no. 4, pp. 1291-1299, 1993.

[14] G M Onik, "Image-guided prostate cryosurgery: state of the art," *Cancer Control*, vol. 8, no. 1, pp. 522-531, 2001.

[15] W S Wong, D O Chinn, M Chinn, J Chinn, and W L Tom, "Cryosurgery as a treatment for prostate carcinoma," *Cancer*, vol. 79, no. 5, pp. 963-974, 1997.

[16] Bruce Montgomery. (2015, April) Prostate Cancer Centre. [Online]. http://www.prostatecancercentre.com/treatments/cryo.html

[17] J G Baust et al., "Issues critical to the successful application of cryosurgical ablation of the prostate," *Technology in Cancer Research and Treatment*, vol. 6, no. 2, pp. 97-109, 2007.

[18] J P Long et al., "Five-year retrospective, multi-institutional pooled analysis of cancer-related outcomes after cryosurgical ablation of the prostate," *Urology*, vol. 57, no. 3, pp. 518-523, 2001.

[19] M. R. Cooperberg. (2015, March) Medscape. [Online]. http://emedicine.medscape.com/article/458187-technique#aw2aab6b4b2

[20] R Baisallov et al., "A semi-empirical treatment planning model for optimization of multiprobe cryosurgery," *Physics in Medicine and Biology*, vol. 45, pp. 1085-1098, 200.

[21] R Baissalov, G A Sandison, D Reynolds, and K Muldrew, "Simultaneous optimization of cryoprobe placement and thermal protocol for cryosurgery," *Physics in Medicine and Biology*, vol. 46, pp. 1799-1814, 2001.

[22] R G Keanini and B Rubinsky, "Optimization of multiprobe cryosurgery," *ASME Journal of Heat Transfer*, vol. 114, pp. 796-802, 1992.

[23] D C Lung, T F Stahovich, and Y Rabin, "Computerized planning for multiprobe cryosurgery using a force-field analogy," *Computer Methods in Biomechanics and Biomedical Engineering*, vol. 7, no. 2, pp. 101-110, 2004.

[24] M R Rossi and Y Rabin, "Experimental verification of numerical simulations of cryosurgery with application to computerized planning," *Physics in Medicine and Biology*, vol. 52, no. 1, pp. 4553–4567, 2007.

[25] M R Rossi, D Tanaka, K Shimada, and Y Rabin, "An efficient numerical technique for bioheat simulations and its application to computerized cryosurgery planning," *Computer Methods and Programs in Biomedicine*, vol. 85, no. 1, pp. 41-50, 2006.

[26] M R Rossi, D Tanaka, K Shimada, and Y Rabin, "Computerized planning of cryosurgery using bubble packing: an experimental validation on a phantom material," *The International Journal of Heat and Mass Transfer*, vol. 51, no. 1, pp. 5671-5678, 2008.

[27] D Tanaka, K Shimada, and Y Rabin, "Two-phase computerized planning of cryosurgery using bubble-packing and force-field analogy," *ASME Journal of Biomechanical Engineering*, vol. 128, no. 1, pp. 49-58, 2006.

[28] Tomotake Furuhata, Inho Song, Hong Zhang, Yoed Rabin, and Kenji Shimada, "Interactive prostate shape reconstruction from 3D TRUS images," *Journal of Computational Design and Engineering*, vol. 1, no. 4, pp. 272-288, September 2014.

[29] K Shimada, T Furuhata, A Sehrawat, and Y Rabin, "Generation of 3D Prostate Models for Computerized Cryosurgical Trainer," *Cryobiology*, vol. 63, no. 3, pp. 315-316, 2011.

[30] D. Tanaka, M.R. Rossi, K. Shimada, and Y. Rabin, "Towards intra-operative computerized planning of prostate cryosurgery," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 3, no. 1, pp. 10-19, 2007.

[31] Anjali Sehrawat, Robert Keelan, Kenji Shimada, and Yoed Rabin, "Developing an intelligent tutoring system for prostate cryosurgery," in *Society for Cryobiology - Cryo 2013*, Bethesda, MD, 2013.

[32] Anjali Sehrawat, Kenji Shimada, and Yoed Rabin, "Generating prostate models by means of geometric deformation with application to computerized training of cryosurgery," *International journal of computer assisted radiology and surgery*, vol. 8, no. 2, pp. 301-312, 2013.

[33] A Sehrawat et al., "Simulation-based cryosurgery intelligent tutoring system (ITS) prototype," *Technology in Cancer Research and Treatment*, no. (in press), 2015.

[34] Robert L. Keelan, Soji Yanakawa, Kenji Shimada, and Yoed Rabin, "Computerized training of cryosurgery - a system approach," *Cryoletters*, vol. 34, no. 4, pp. 324-337, 2013.

[35] H H Pennes, "Analysis of tissue and arterial blood temperature in the resting human forearm," *Journal of Applied Physiology*, vol. 1, no. 2, pp. 93-122, 1948.

[36] Y. Rabin and E. Korin, "An efficient numerical solution for the multidimensional solidification (or melting) problem using a microcomputer," *International Journal of Heat*

and Mass Transfer, vol. 36, no. 3, pp. 673-683, 1993.

[37] K R Diller, "Modeling of bioheat transfer processes at high and low temperature," *Advances in heat transfer*, vol. 22, no. 1, pp. 157-357, 1992.

[38] C K Charny, "Mathematical models of bioheat transfer," *Advances in Heat Transfer*, vol. 22, no. 1, pp. 19-156, 1992.

[39] Y. Rabin and A. Shitzer, "Exact solution to the one-dimensional inverse Stefan problem in non-ideal biological tissues," *ASME Journal of Heat Transfer*, vol. 117, no. 2, pp. 425-431, 1995.

[40] Yoed Rabin and A. Shitzer, "Combined solution to the inverse Stefan problem for successive freezing/thawing in non-ideal biological tissues," *ASME Journal of Biomedical Engineering*, vol. 119, no. 2, pp. 146-152, 1997.

[41] Y Rabin and A Shitzer, "Numerical solution of the multidimensional freezing problem during cryosurgery," *Journal of Biomechanical Engineering*, vol. 120, pp. 32-37, 1998.

[42] Y Rabin, "Clinical and basic science of cryotherapy," in *Society for Thermal Medicine Annual Meeting*, Washington DC, 2007.

[43] Y Rabin, "Computational tools in the service of cryosurgery," in *Society for Thermal Medicine Annual Meeting*, Portland, 2012.

[44] C Thaokar and Y Rabin, "Temperature field reconstruction for the application of wireless implantable temperature sensors in cryosurgery," in *ASME 200 Summer Bioengineering Conference*, Farmington, 2011, pp. 41-42.

[45] Ansys Inc. Ansys Theory Reference. [Online]. http://www.ansys.com

[46] B. Carnahan, H. A. Luther, and J. O. Wilkes, *Applied Numerical Methods*. New York: John Wiley and Sons, 1969.

[47] Y. Rabin, "A general model for the propagation of uncertainty in measurements into heat transfer simulation and its application to cryobiology.," *Cryobiology*, vol. 46, no. 2, pp. 109-120, 2003.

[48] Kate Gregory and Ade Miller, *C++ AMP: accelerated massive parallelism with Microsoft Visual C++.*: Microsoft Press, 2012.

[49] Nathan Bell and Michael Garland, "Efficient sparse matrix-vector multiplication on

CUDA," *NVIDIA Technical Report NVR-2008-004*, 2008.

[50] Nvidia. (2015, April) Nvidia Corporate Web Site. [Online]. http://www.nvidia.com/object/what-is-gpu-computing.html

[51] Andrè R. Brodtkorba, Trond R. Hagena, and Martin L. Saetrab, "Graphics processing unit (GPU) programming strategies and trends in GPU computing," *Journal of Parallel and Distributed Computing*, vol. 73, no. 1, pp. 4-13, 2013.

[52] Guillem Pratxa and Lei Xing, "GPU computing in medical physics: A review," *Med. Phys.*, vol. 38, no. 5, pp. 2685-2697, 2011.

[53] R L Keelan, H Zhang, K Shimada, and Y Rabin, "GPU-based bioheat simulation to facilitate rapid decision making associated with cryosurgery training," *Technology in Cancer Research & Treatment*, p. in press, 2015.

[54] Yoed Rabin, David C. Lung, and Thomas F. Stahovich, "Computerized planning of cryosurgery using cryoprobes and cryoheaters," *Technol. Cancer Res. Treat>*, vol. 3, no. 3, pp. 227-243, 2004.

[55] E Neufeld, N Chavannes, T Samaras, and N Kuster, "Novel conformal technique to reduce staircasing artifacts at material boundaries for FDTD modeling of the bioheat equation," *Phys. Med. Biol.*, no. 52, pp. 4371-4381, 2007.

[56] M Christen, O Schenk, E Neufeld, P Messmer, and H Burkhart, "Parallel data-locality aware stencil computations on moder micro-architectures.," *Parallel & Distributed Processing*, pp. 1-10, 2009.

[57] A A Efros and T K Leung, "Texture synthesis by non-parametric sampling," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, Kerkyra, 1999, pp. 1003-1038.

[58] Jeremy S De Bonet, "Multiresolution sampling procedure for analysis and synthesis of texture images," in *SIGGRAPH '97 Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, Los Angeles, 1997, pp. 361-368.

[59] Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick, "Graphcut textures: image and video synthesis using graph cuts," in *SIGGRAPH '03 ACM SIGGRAPH 2003 Papers*, San Diego, 2003, pp. 227-286.

[60] Y Zhu, D Magee, R Ratnalingam, and D Kessel, "A virtual ultrasound imaging system for the simulation of ultrasound-guided needle insertion procedures," in *Proceedings of Medical Image Understanding and Analysis*, 2006, pp. 61-65.

[61] D Magee, Y Zhu, R Ratnalingam, P Gardner, and D Kessel, "An augmented reality simulator for ultrasound guided needle placement training," *Medical & Biological Engineering & Computing*, vol. 45, no. 10, pp. 957-967, July 2007.

[62] Aaron Hertzmann, Charles E Jacobs, Nuria Oliver, Brian Curless, and David H Salesin, "Image Analogies," in *SIGGRAPH '01 Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, Los Angeles, 2001, pp. 327-340.

[63] Marius Muja and David G Lowe, "Scalable nearest neighbor algorithms for high dimensional data," *Pattern analysis and Machine Learning*, vol. 36, 2014.

[64] Nikolai N. Korpan, Ed., *Atlas of Cryosurgery*, 1st ed. Wien, New York: Springer, 2001.

[65] Haim Azhari, *Basics of biomedical ultrasound for engineers*, 1st ed. Hoboken, N.J.: Wiley, 2010.

[66] B. Burger, S. Bettinghausen, M Radle, and J Hesser, "Real-time GPU-based ultrasound simulation using deformable mesh models," *Medical Imaging*, vol. 32, no. 3, pp. 609-618, 2013.

[67] Athanasios Karamalis, "GPU ultrasound simulation and volume reconstruction," Technischen Universität München, München, Master's Thesis 2009.

[68] Jørgen A Jensen, "Field: A program for simulating ultrasound systems," in *10TH NORDICBALTIC CONFERENCE ON BIOMEDICAL IMAGING*, 1996, pp. 351-353.

[69] Torbjørn Hergum, S Langeland, E Remme, and H Torp, "Fast ultrasound imaging simulation in k-space. Ultrasonics," *Ferroelectrics and Frequency Control*, vol. 56, no. 6, pp. 1159-1167, 2009.

[70] Jean-Louis Dillenseger, Soizic Laguitton, and Éric Delabrousse, "Fast simulation of ultrasound images from a CT volume," *Computers in biology and medicine*, vol. 39, no. 2, pp. 180-186, 2009.

[71] F. J. Seron, D. Gutierrez, G. Guitierrez, and E. Cerezo, "Visualizing Sunsets through Inhomogeneous Atmospheres," in *CGI '04 Proceedings of the Computer Graphics International*, D.C., 2004, pp. 349-356.

[72] Chad M Hogan and Gary M Margrave, "Ray-tracing and eikonal solutions for low-frequency wavefields," CREWES, Tech. rep. 2007.

[73] O. Kutter, A. Karamalis, W. Wein, and N. Navab, "A GPU-based framework for simulation of medical ultrasound," in *In SPIE Medical Imaging*, Princeton, 2009, pp. 726117-726117.

[74] Oliver Kutter, Ramtin Shams, and Nassir Navab, "Visualization and GPU-Accelerated Simulation of Medical Ultrasound from CT Images," *Computer Methods and Programs in Biomedicine*, vol. 94, no. 3, pp. 250-266, 2009.

[75] Daigo Tanaka, Kenji Shimada, Michael R. Rossi, and Yoed Rabin, "Cryosurgery planning using bubble packing in 3D," *Computer Methods in Biomechanics and Biomedical Engineering*, vol. 11, no. 2, pp. 113-121, 2008.

[76] R Keelan, K Shimada, and Y Rabin, "GPU-based simulation of ultrasound imaging artifacts for cryosurgery training," *submitted*.

[77] G Onik, C Cobb, J Cohen, J Zachbar, and B Porterfield, "US characteristics of frozen prostate," *Radiology*, vol. 168, no. 3, pp. 629-631, 1988.

[78] R Keelan, K Shimada, and Y Rabin, "Developing a framework for computerized training of cryosurgery based on finite element analysis," in *ASME 2011 Summer Bioengineering Conference*, Farmington, 2011.

[79] K Shimada, *Physically-Based mesh generation: Automated triangulation of surfaces and volumes via Bubble Packing.*: Massachusetts Institute of Technology, 1993.

[80] Anjali Sehrawat, *Computerized Training of Cryosurgery: Prostate Geometric Modeling and Training Framework*. Ann Arbor, United States of America: ProQuest LLC, 2015.

[81] Dalbhanjan, Peter. (2015, March) Overview of Deployment Options on AWS. [Online]. http://d0.awsstatic.com/whitepapers/overview-of-deployment-options-on-aws.pdf

[82] David J Heeger and James R Bergen, "Pyramid-based texture analysis/synthesis," in *SIGGRAPH '95 Proceedings of the 22nd annual conference on Computer Graphics and interactive techniques*, Los Angeles, 1995.

[83] Javier Portilla and Eero P Simoncelli, "A parametric texture model based on joint statistics of complex wavelet coefficients," *International Journal of Computer Vision*, vol. 40, no. 1, pp. 49-70, 200.

[84] Junyu Dong, Duan Yuanxu, Guimei Sun, and Lin Qi, "Texture synthesis by Support Vector

Machines," in *Pattern Recognition, 19th International Conference on*, Tampa, 2008.

[85] Volkan Atalay, Erol Gelenbe, and Nese Yalabik, "The random neural network model for texture synthesis," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 6, no. 1, pp. 131-141, 1992.

[86] G E Hinton, S Osindero, and Y Teh, "A fast learning algorithhm for deep belief nets," *Neural Computation*, vol. 18, pp. 1527-1554, 2006.

[87] H Luo, P Carrier, A Courville, and Y Bengio, "Texture modeling with convolutional spike-and-slab RBMs and deep extensions," in *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, Scottsdale, 2013.

[88] Geir Evensen, *Data assimilation: the ensemble Kalman filter*. New York: Springer, 2007.

[89] A A Gage, "Cryosurgery in th treatment of cancer," *Surgery Gynecology & Obstetrics*, vol. 174, no. 1, pp. 73-92, 1992.

[90] B Rubinsky, "Cryosurgery," *Annual Review of Cryotherapy for Localized Prostate Cancer*, vol. 10, no. 2, pp. 199-208, 1999.

[91] J C Saliken et al., "Outcome and Safety of Transrectal US-guided Percutaneous Cryotherapy for Localized Prostate Cancer," *Journal of Vascular and Interventional Radiology*, vol. 10, no. 2, pp. 199-208, 1999.

[92] Athanasios Karamalis, Wolfgang Wein, and Nassir Navab, "Fast ultrasound image simulation using the westervelt equation," *MICCAI*, pp. 243-250, 2010.

[93] Y Rabin, K Shimada, R Keelan, A Sehrawat, and H Zhang, "Computerized training objectives and prototyping," in *ACCryo 2014 Cryobiology*, Key Largo, 2014, p. 187.

[94] Y Rabin, K Shimada, R Keelan, A Sehrawat, and Zhang H, "Computerized training tools for cryosurgery," in *STM*, Minneaopolis, 2014.

[95] R Keelan, S Yamakawa, K Shimada, and Y Rabin, "Simulation of prostate cryosurgery for the purpose of surgical training: system design and benchmarking," in *Annual meeting of the Biomedical Engineering Society*, Atlanta, 2012.

[96] A Sehrawat, R Keelan, K Shimada, and Y Rabin, "Development of an Intelligent Tutoring System for Prostate Cryosurgery," in *Annual Meeting of the Society for Cryobiology*, N. Bethesda DC, 2013, pp. 425-426.

[97] R Keelan, K Shimada, and Y Rabin, "Generating synthetic ultrasound imaging artifacts for the purpose of computerized training of cryosurgery," in *CRYO2013 Cryobiology*, N. Bethesda DC, 2013, p. 426.

[98] R Keelan, H Zhang, K Shimada, and Y Rabin, "Imaging artifacts, training interface and computational challenges," in *ACCryo2014 Cryobiology*, Key Largo, 2014, p. 187.

[99] A Sehrawat, R Keelan, K Shimada, and Y Rabin, "Interactive tutoring on a prostate model and prototype development," in *ACCryo 2014 Cryobiology*, Key Largo, 2014, p. 187.

[100] R Keelan, H Zhang, K Shimada, and Y Rabin, "Simulated ultrasound imaging for the purpose of prostate cryosurgery training," in *STM*, Minneapolis, 2014.