UNIVERSIDAD DE CÓRDOBA

Programa de Doctorado:

COMPUTACIÓN AVANZADA, ENERGÍA Y PLASMAS

GPU ACCELERATION FOR LIVER ENHANCEMENT AND SEGMENTATION

OPTIMIZACIÓN EN GPU DE ALGORITMOS PARA LA MEJORA DEL REALCE Y SEGMENTACIÓN EN IMÁGENES HEPÁTICAS

Directores:

DR. JOAQUÍN OLIVARES BUENO
DR. JUAN GÓMEZ-LUNA

Autor

D. NITIN SATPUTE

Fecha de depósito de la tesis
SEPT 2020

TITULO: *GPU ACCELERATION FOR LIVER ENHANCEMENT AND SEGMENTATION*

AUTOR: *Nitin Satpute*

Campus de Rabanales
Ctra. Nacional IV, Km. 396 A
14071 Córdoba

https://www.uco.es/ucopress/index.php/es/
ucopress@uco.es

INFORME DE LOS DIRECTORES

Título de la tesis: GPU acceleration for liver enhancement and segmentation

Doctorando: Nitin Satpute

Informe razonado de los directores de la tesis:

El doctorando ha realizado un trabajo de investigación novedoso en el que ha mostrado la evolución personal y académica desde ingeniero hasta investigador. El doctorando ha realizado numerosas propuestas y discusiones con los directores que han llevado a obtener las aportaciones científicas incluidas en esta tesis doctoral. El doctorando ha realizado una revisión bibliográfica de antecedentes científico-técnicos relativos al campo de investigación de la tesis.

Todas estas aportaciones científicas son novedosas y han producido 3 artículos indexados en JCR. Y con ello demuestran la capacidad del doctorando en el ámbito investigador.

Por todo ello, se autoriza la presentación de la tesis doctoral.

En Córdoba, a de Sept 2020.

Los Directores de la tesis doctoral:

Fdo. Joaquín Olivares Bueno Fdo. Juan Gómez-Luna

ABSTRACT

This doctoral thesis deepens the GPU acceleration for liver enhancement and segmentation. With this motivation, detailed research is carried out here in a compendium of articles. The work developed is structured in three scientific contributions, the first one is based upon enhancement and tumor segmentation, the second one explores the vessel segmentation and the last is published on liver segmentation. These works are implemented on GPU with significant speedups with great scientific impact and relevance in this doctoral thesis

The first work proposes cross-modality based contrast enhancement for tumor segmentation on GPU. To do this, it takes target and guidance images as an input and enhance the low quality target image by applying two dimensional histogram approach. Further it has been observed that the enhanced image provides more accurate tumor segmentation using GPU based dynamic seeded region growing. The second contribution is about fast parallel gradient based seeded region growing where static approach has been proposed and implemented on GPU for accurate vessel segmentation.

The third contribution describes GPU acceleration of Chan-Vese model and cross-modality based contrast enhancement for liver segmentation.

SCIENTIFIC PUBLICATIONS

The ideas, images and data that support this doctoral thesis have been previously published in the following works:

- Nitin Satpute, Rabia Naseem, Egidijus Pelanis, Juan Gómez-Luna, Faouzi Alaya Cheikh, Ole Jakob Elle, Joaquín Olivares *GPU acceleration of liver enhancement for tumor segmentation*. Computer Methods and Programs in Biomedicine, 184 (2020) 105285.
  DOI: https://doi.org/10.1016/j.cmpb.2019.105285. Impact Factor: 3.632

- Nitin Satpute, Rabia Naseem, Rafael Palomar, Orestis Zachariadis, Juan Gómez-Luna, Faouzi Alaya Cheikh, Joaquín Olivares *Fast Parallel Vessel Segmentation*. Computer Methods and Programs in Biomedicine, 192 (2020) 105430.
  DOI: https://doi.org/10.1016/j.cmpb.2020.105430. Impact Factor: 3.632

- Nitin Satpute, Juan Gómez-Luna, Joaquín Olivares *Accelerating Chan-Vese Model with Cross-Modality Guided Contrast Enhancement for Liver Segmentation*. Computers in Biology and Medicine, 124 (2020) 103930.
  DOI: https://doi.org/10.1016/j.compbiomed.2020.103930. Impact Factor: 3.434

GRATITUDE

I would like to thank all the people who have helped me in this personal project, for their support and patience, especially in the most difficult times.

A very special thanks to my parents and brother. As always, even without understanding destiny very well, they have adorned my path with their best wishes.

This Doctoral Thesis would not have been possible without having had the personal and professional support of my directors to whom I dedicate a few words of thanks. To Joaquín Olivares Bueno, because without counting the hours spent in front of the board he has managed to turn engineering into science in my way of thinking. And Juan Gómez-Luna, for his pragmatic and optimistic vision that makes everything easier. Both, who have believed in me even when I have not been able, are more my friends than my directors.

Thank You.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

CDF Cumulative Distributive Function

CFL Courant, Friedrichs, Lewy

CPU Central Processing Unit

CT Computed Tomography

CUDA Compute Unified Device Architecture

DS Dice Score

GPU Graphical Processing Unit

HE Histogram Equalization

HL Histogram Length

HM Histogram Matching

IBS Inter Block GPU Synchronization

KTRL Kernel Termination and Relaunch

MRI Magnetic Resonance Imaging

PT Persistent Threads

RoI Region of Interest

SDF Signed Distance Function

SRG Seeded Region Growing

SM Streaming Multiprocessor

TMS Task Mask Stack

Part I

INTRODUCTION

1

INTRODUCTION

Index

1.1 Introduction to Medical Imaging with GPU(s)

Computer aided diagnosis and treatment is the challenging problem in medical imaging because of the intensity values of the structures of interest and the nearby regions in image. Medical image enhancement and segmentation are the problems faced by clinicians to achieve real time treatment of the patients. There are works targeting medical image enhancement and segmentation methods [12, 23, 25, 31, 36, 54, 67, 75].

Liver images contain a lot of unnecessary data. This impacts the speedup of overall process and accuracy of liver segmentation. There are many algorithms developed recently for liver segmentation. We classify these approaches in two categories; traditional and learning approaches. Seeded region growing is one of the traditional approaches for segmentation and volume assessment [43, 50]. Xuesong et. al. [37] have proposed automatic liver segmentation using multi-dimensional graph cut with shape information. Chan-Vese algorithm for liver segmentation is discussed in [10, 53]. Unet is a convolutional neural network architecture used for biomedical image segmentation [47]. Unet based liver segmentation sometimes segments part outside the liver boundary and the Chan-Vese based liver segmentation is slow due to the unwanted data in the medical image [10, 47, 53]. Dong et. al. [69] have proposed learning based automatic liver segmentation using an adversarial image to image network.

Learning based algorithms require huge dataset during training and validation. They use complete computed tomography (CT) liver image of fixed size as an input to the network. But the traditional approaches can operate on region of interest (RoI) to avoid operations on unnecessary elements [56]. These unwanted elements in liver image makes it difficult to implement liver segmentation in real time. It impacts the accuracy of the segmentation. The medical image during processing contains large amount of data and takes lot of memory. When implemented on graphics processing unit (GPU), the memory on the GPU sometimes is not sufficient to process the whole image and it slows down the overall process of segmentation [50, 56, 58]. Cropping helps to remove unwanted elements from the medical image and improves the speedup of overall process of segmentation.

Seeded region growing (SRG) is a widely used approach for semi automatic segmentation [43, 50]. Delibasis et. al. [12] have proposed a tool based on a modified version of SRG algorithm, combined with a priori knowledge of the required shape. SRG starts with a set of pixels called seeds and grows a uniform, connected region from each seed. Key steps to SRG are to define seed(s) and a classifying criterion that relies on the image properties and user interaction [58].

SRG is an iterative process. SRG is invoked continuously until region can not be grown further. Iterative process in SRG, when implemented on GPU requires terminating kernel and relaunching from CPU (Kernel Termination and Relaunch (KTRL)) and data transfers between CPU and GPU [50, 58]. GPU based implementation of SRG needs a dynamic queue (stack). CPUs provide hardware support for stacks but GPUs do not [8]. Any queuing system has a large number of pieces of work to do and a fixed number of workers corresponding to the fixed number of computing units. Pieces are then assigned dynamically to the workers. The problem is deciding the maximum number of pieces of work in the queuing system. GPU implementation of a stack re-

quires continuous changes in memory allocations which in turn requires iterative GPU kernel invocation from CPU in other words kernel termination and relaunch as discussed in the algorithms IVM backtracking and work stealing phase by Pessoa et al. [44]. Task-parallel run-time system, called TREES, that is designed for high performance on CPU/GPU platforms by Hechtman et al. [28] have shown the invocation of GPU kernels from CPU iteratively for updating task mask stack (TMS) in TREES execution. The loop involved while implementing data flow through the stream kernels of the rendering system (involving stack) on GPU controlled by CPU (that is KTRL) is proposed by Ernst et al. [18].

Nevertheless, there is an alternate GPU implementation of queuing system (stack) using dynamic kernel launching. Chen et. al. [2015] have proposed free launch based dynamic kernel launches through thread reuse technique [8]. This technique requires no hardware extensions, immediately deployable on existing GPUs. By turning subkernel launch into a programming feature independent of hardware support, free launch provides alternate approach for subkernel launch which can be used beneficially on GPUs.

In this thesis, I discuss the GPU optimization strategies for liver enhancement and segmentation. The block diagram of the overall contributions is given below in Figure 1.1. This describes the objectives such as liver enhancement and segmentation of liver, tumor and vessels and the proposed approaches such as cross-modality for enhancement, region growing for tumor and vessel segmentation and Chan-Vese based active contour approach for liver segmentation.

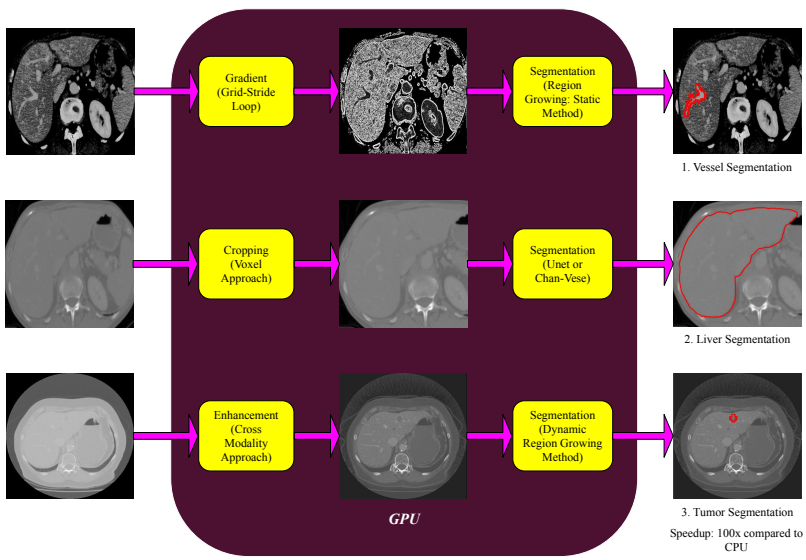


Figure 1.1
Block Diagram of The Overall Contributions

In the next section, I discuss the thesis approach in detail.

1.2 Thesis Approach

The theoretical and technological antecedents introduced help to understand the origin and tendency of the medical image analysis. As for the future, the forecasts are clear: the liver enhancement and segmentation require efficient approaches which will make the diagnosis and treatment faster and accurate; It will be introduced into hospitals, industries, and even in research. The conjunction of artificial intelligence, Big Data, and the humanization of interfaces with technology will produce a change in computer aided diagnosis and treatment. The challenges arising from such a paradigm shift require analysis, vision and decision in all aspects of human knowledge and doing: politics, science, ethics, society, economy, etc.

One of the numerous research fronts of the technical dimension of the challenge has to do with the huge volume of medical data that is expected to be channeled from the CT or MRI scans. The scientific implication in this problem encompasses all branches of knowledge related to proposing approaches oriented to the different phases of the treatment: data collection, clinical perspective, preprocessing, information fusion, strategies, priority schemes, etc. Likewise, the heterogeneity of liver data and their purpose makes it difficult to propose general purpose solutions, which encourages the emergence of efficiency solutions for specific purposes. This extensive wealth of the scientific landscape implies the impossibility of predicting the technological future with precision; it seems that the future medical image diagnosis and treatment infrastructure is called to deal with huge amount of data along with more intensive analysis, and the intelligence to adapt to the context.

Due to the abstract of its definition, the concept of adaptation to the context is subject to interpretation. The word context refers to any kind of circumstance that characterizes the environment of a fact. That is why addressing this purpose from the rigor of science requires the concretion of the fundamental parameters that model the supposed context to which the proposal refers. In the case of the liver segmentation, we propose three levels of context:

Technological Context The technological context refers to the technical details of the liver cancer diagnosis being adapted. At this level we refer to the characteristics of the deployed GPU platform, approaches, the technologies employed, the environment, etc..

Medical Context The quality of the treatment matters a lot in medical imaging. The quality of results widely explored by the scientific community in medicine come into play here.

Logical context The logical context is related to the meaning and use of the volume of liver image data. The question of the purpose of the information is not accessible if a general purpose adaptation strategy is intended. However, in application specific research, knowing the importance and objective of each medical process makes it possible to incorporate this logic into the adaptation mechanisms.

Understanding that the context is an evolving reality, the adaptability of a system is a process that starts with its characterization as input information, and as an output it requires the availability of resources to act accordingly, with the final objective of optimizing the operation. On the other hand, the

process requires some intelligence to correctly manage diagnoses, actions and repercussions.

In this conceptual framework, this doctoral thesis is developed, in which issues related to the fast and efficient liver image enhancement and segmentation are addressed, limiting the logical context to the evaluation of quality of liver, tumor or vessel segmentation. In the research described in this thesis, the enhancement and segmentation algorithms and analysis related to the processing of iterative computations are provided as a part of GPU optimization strategies, proposing adaptive techniques that have a place in the previous phases of obtaining knowledge of greater abstraction, based on the result of quality of segmentation.

1.3 Thesis Structure

The content of this thesis is organized in three blocks. Block I is divided into 4 chapters and contains the introduction (which ends in this description of the structure of the document), the description of the background of this work in a theoretical framework (Chapter 2), the motivation and objectives of the proposals developed (Chapter 3), and the hypotheses and methodology proposed for their demonstration (Chapter 4). Block II contains 4 chapters, the first is a description of the 3 publications contributed as support for this doctoral thesis (Chapter 5), while the remaining three chapters contain these contributions separately (Chapters 6, 7 and 8). Finally, block III concludes with 3 chapters, addressing the discussion of the results obtained (Chapter 9), conclusions (Chapter 10) and future work (Chapter 11).

2

BACKGROUND

Index

2.1 Medical Imaging and High Performance Computing

Medical images contain large amounts of data that are not part of a region of interest (RoI). Processing over medical image includes redundant memory transfers and unnecessary computations over complete image. Memory on the GPU is limited and may not be enough for processing large medical dataset. As computing threads processes each element in the medical image, removing the unnecessary data not only reduces the memory usage, but also the execution time [50, 56]. This can be done by reducing the number of memory transfers between CPU and GPU. The number of intermediate memory transfers are involved in implementing iterative kernels on GPU. By reducing the number of intermediate transfers between CPU and GPU reduces overall computation time. Apart from speedup, accuracy of the results plays a vital role in medical imaging. One of the applications is tissue detection where we care not only about speedup but also about accuracy of the results.

Boundary of the object in RoI is diffused sometimes in a CT liver image due to poor image quality. Hence it becomes difficult to find the RoI automatically. Boundary detection plays a vital role in cropping.Yap et. al. [71] have discussed convolutional neural networks for automated breast ultrasound lesions detection. Zhao et. al. [74] have provided the study of knowledge aided convolutional neural network for small organ segmentation and Yan et. al. [68] have proposed three stage deep learning model for accurate retinal vessel segmentation. Smistad et. al. [54, 56] have discussed parallel cropping for image segmentation. They crop the medical image dataset before processing. CPU allocates the memory equivalent to the cropped size to copy data to cropped image on the GPU. Further seeded region growing is performed for image segmentation. This is the simplistic representation of the work by Smistad et. al. [56]. They have proposed non-PT (non persistent thread) approach for cropping and seeded region growing. Persistence imply the number of active computing threads that can process complete image. If the number of active computing threads are lesser than the image elements, then the task of remaining image elements are distributed amongst the threads. The total number of active persistent threads imply the grid-size.

Smistad et. al. [56] have implemented slice based parallel cropping algorithm. Slice based parallel cropping needs three kernel invocations for calculating cropping points and one kernel invocation to generate cropped data within cropping points. There are four kernel calls for 3D parallel cropping which impacts the performance.

Once the boundary is detected using cropping points, it is necessary to crop image properly. Over cropping of liver crops part of the liver destroying the quality of the segmentation. Whereas under cropping crops the small amount of data from the image resulting in the limited speedup. It is necessary to balance the speedup and accuracy using cropping and parallelization to avoid above mentioned problems.

2.2 Enhancement and Segmentation

The performance of liver segmentation can be made better after preprocessing CT images [52, 76]. Methods of image enhancement include linear contrast adjustment, filtering (median, Wiener and unsharp mask filter), morphological operators, histogram equalization, contrast limited adaptive histogram equalization and decorrelation stretch.

A set of operations that process images based on shapes is termed as morphological operators. Value of each pixel in an image is adjusted based on neighborhood pixels. Wiener deconvolution is used to deblurr the images. Adaptive histogram equalization is image processing technique used to improve contrast in image. Adaptive method computes several histograms. Each corresponding to distinct section of image. Decorrelation stretch enhances the color separation of an image. Original image color values are mapped to new set of color values with a wider range. Hence primary purpose of decorrelation stretch is visual enhancement. In median filter, we keep the pixels in window in order and replace the candidate pixel with the middle pixel [11, 38, 60].

The image enhancement techniques are essential to reduce the noises and improve the contrast of the image specially in medical imaging for detecting and segmenting the region of interest [25, 31, 36, 67, 75] CT scans of liver images are sometimes low in contrast. It becomes essential to improve the contrast for better image guided surgery [26, 52, 68, 71]. In order to improve the contrast of the image, histogram based methods such as histogram equalization, histogram specification and histogram matching are analyzed [5, 17, 61].

There 2D histogram techniques along with 1D histogram methods for the image enhancement which enhance the target image with the help of guided image [65, 70]. The task of 2D histogram based liver enhancement is computationally expensive and time consuming [33]. There are literatures to enhance an image using guided image for natural images [36, 52, 67, 76]. This process uses the additional information contained in the other image with better contrast or minimal noise and helps in segmenting the image accurately.

Image segmentation is a popular research topic in medical imaging, as it has number of applications such as tissue segmentation, reconstruction, registration etc [12, 43, 50, 53]. There are many methods proposed for image segmentation such as region growing, thresholding, gradient approach, contour methods, automatic clustering etc [29, 57, 58]. These are either based on edge or region. They are further categorized based on histogram, spatial information of the image, convergence active contours etc [53, 63]. The active contour models are essential when the edge of the region of interest in the image is not clear and diffused [53]. Computed tomography (CT) scans sometimes provide poor quality images where liver boundaries are not clearly visible and liver segmentation is essential for clinicians for the treatment of the patients.

Computer-aided diagnosis (CAD) in medical field is playing vital role in clinical practice. Accordingly, medical therapy using CAD is becoming a more challenging research task. Image segmentation for the identification of features and/or objects is a challenging data-dependent task in medical imaging. Image segmentation approaches are classified as edge-based approach, region-based approach and finally segmentation by thresholding on pixel intensities [16]. Types of thresholding based segmentation are given in Figure 2.1. It is divided into two parts: 1. Global and 2. Local Thresholding [48]. Global thresholding

can be applied by using either of threshold binary, threshold binary and inverted, truncation, thresholding to zero, thresholding to zero and inverted and Otsu method which is an adaptive thresholding way of binarization [15]. If pixel value is greater than threshold, then the new pixel intensity is set to maximum value else they are set to zero. This is the functioning of thresholding approach. Global thresholding algorithm includes the Otsu algorithm for segmentation. Otsu encourages automatic threshold selection approach for image segmentation. Local thresholding can be obtained in terms of mean and gaussian values of pixels. But region growing (RG) is one of the most conventional and efficient approaches in image segmentation, which is the first step of image analysis.

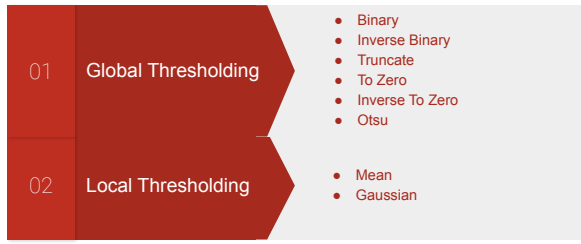


Figure 2.1
Thresholding based Segmentation

Medical imaging is one of the prominent fields in today's world. In medical imaging the identification of connected regions is a challenging task. Region growing starts with a set of pixels called seeds and grows a uniform, connected region from each seed. The key steps to region-growing methods are to define seed(s), number of seeds and a classifying criterion that relies on image properties and user interaction [39]. Among the various studies, seeded region growing (SRG) is a widely used image segmentation method in identifying identical objects due to its lesser computational complexity [19, 39]. However, in practice, it demands high computational cost to the large amount of dependent data to be processed in medical image analysis and still requires efficient solutions [64].

The recursive process in region growing (RG) based medical image segmentation is time consuming. Kernel RG is invoked recursively till the region can not be grown further. So our main objective is to reduce this recursive time using different inter block GPU synchronization methods resulting in efficient parallel medical image segmentation.

Kernel Termination and Relaunch/CPU based synchronization includes terminate a GPU kernel and invoke it from CPU if the region can be grown further. It includes the overhead of data transfer from CPU to GPU and vice versa while invoking a GPU kernel. Hence we have implemented inter block synchronization methods to improve performance of RG based segmentation. Happ, Patrick Nigri et al. have proposed two parallel algorithms, i.e., Parallel Best Fitting (PBF) and Parallel Local Mutual Best Fitting (PLMBF), for region growing segmentation on GPUs [22]. They are based on different heuristics for merging adjacent segments. Their computational performances are evaluated experimentally on two distinct GPUs, and the outcome of the segmentation compared to the sequential output.

Chan-Vese algorithm is developed on active contour models using level set approach [14, 53]. This works on the initial contours, average intensity values inside and outside the curve and try to optimize the energy based on level set approach [3, 10]. The algorithm works on the principle of energy minimization problem which relies on calculus and partial differential equation. It is one of the influential and effective methods in order to optimize Mumford-Shah function which includes energy terms defined in image space and contour space [27, 63, 73]. Chan-Vese is flexible and robust to segment the CT liver image which is difficult to segment using classical segmentation techniques [29, 56].

There are many algorithms developed recently for liver segmentation. We classify these approaches in two category; traditional and learning approaches. Seeded region growing is one of the traditional approaches for segmentation and volume assessment [43, 50]. Xuesong et. al. have proposed automatic liver segmentation using multi-dimensional graph cut with shape information [37]. Chan-Vese algorithm for liver segmentation is discussed in [10, 53]. Unet is convolutional networks used for biomedical image segmentation [47]. The unet based liver segmentation sometimes segments the part outside the liver boundary and the Chan-Vese based liver segmentation is slow due to the unwanted data in the medical image [10, 47, 53]. Dong et. al. [69] have proposed learning based automatic liver segmentation using an adversarial image to image network. Our contribution with respect to Chan-Vese is more accurate and fast liver segmentation using cross-modality guided image enhancement on GPU. It gives flexibility to play with the initial contour depending on the shape of the liver on various slices resulting in improved accuracy of the results in real time on GPU.

Medical image segmentation has many applications such as diagnosis of brain cancer [4, 34], breast cancer [2] and liver cancer [55]. Similarly it plays vital role in cardiac imaging [7, 51]. Many algorithms have been developed for segmentation detection and classification of medical images. Smart imaging plays vital role when the quality of the CT, MR or ultrasound image is low. It makes it necessary to increase the contrast of low quality images for better quality of results. In cardiac images, segmentation and detection of ventricles, aorta and myocardial mass are necessary for computer aided diagnosis [7, 62]. This also requires huge amount of labeled data for supervised learning. If data is small then it is important to gather more images by data augmentation techniques or image generative models [7]. Multi task learning can be useful for understanding multiple similar tasks in more effective way. Transfer learning helps to transfer the learned features from one task to another in order to increase the accuracy of segmentation, detection or classification. Medical image analysis is essential for liver cancer diagnosis and planning of preoperative and intraoperative stages [62].

Planning during preoperative and intraoperative stages include processing CT or MRI or ultrasound images obtained from scanners at runtime so that it provides useful information about tissues and assist clinicians for the analysis of tumors and important blood vessels. This analysis includes the location and volume of tumors and blood vessels. In order to obtain the location, it can be done using bounding box and the process is called localization. If multiple tumors or vessels are need to be localized then it is called detection. Once the organs or tissues are detected then the required volume of the tissues need to be extracted for the further analysis. This has to be done on real time for the proper planning of the treatment and hence makes it necessary to use GPU resources efficiently. Hence fast, efficient and accurate implementation

of segmentation algorithms on GPU need to be explored and analyzed. This is the important base of the thesis. Later, the impact of preprocessing stage enhancement has to be analyzed for the speedup and accuracy of segmentation. For the analysis, we explore liver cancer diagnosis as an application.

Liver cancer diagnosis and treatment includes understanding and analysis of liver, vessels and tumors [40]. It involves following steps. First classification of liver images with non liver images. Then removal of unnecessary parts from liver images which will crop the liver from volume of slices. Once the liver is cropped then segmentation of liver so that further analysis of tumor and vessel will be fast and efficient. Further the detection and segmentation of tumors and vessels are applied to extract the necessary information. This will assist the clinicians to analyze the liver images for tumor cases and important blood vessels. The detection, segmentation and classification of liver images are impacted by modality of images and image scanners which may result in low quality scans. Hence this is necessary to apply proper preprocessing steps in order to enhance the quality of images and/or remove noise from the image. The quality of liver images can be enhanced using 1D or 2D histogram approaches [42, 46]. 2D histogram approach is also called a cross-modality approach where features of one modality can be utilized to improve the low quality image from different modality. The traditional and machine learning approaches are developed to enhance and analyze the liver images.

Traditional methods include edge detection [56], region growing method [57], otsu method [13] for segmentation and histogram based techniques for image enhancement. Machine learning methods include deep learning (DL) and non DL approaches. DL approaches include convolutional neural network, fully connected networks, transfer learning, adversarial networks etc. Non DL approaches include KNN (K nearest neighbour) [41, 45], K means clustering algorithms [35], decision trees [66] and random forest classifiers [20] which are employed to learn the features from liver images. Researchers have also tried fusion of DL and non DL methods to improve the accuracy of the results. But the problem with all these approaches are their implementation is slow on CPU. Hence it makes necessary to have fast and efficient implementation on GPU(s). Hence it is necessary to understand the GPU architecture for effective implementation of medical imaging algorithms [30, 34].

GPU implementation of traditional algorithms include fast and efficient use of shared memory, thread blocks, and synchronization strategies. GPU memory hierarchy consists of device, shared, local, cache, constant, texture and register memory. Constant and texture memory are the read only memory. Texture memory is cached on chip and provide faster access than the global memory due to higher effective bandwidth. It is necessary to understand the difference between CPU and GPU architecture in order to exploit GPU architecture heavily for achieving huge computational speedup [1, 34].

CPU and GPU both have L1 and L2 cache but they are smaller in size and higher in bandwidth on GPU in comparison to CPU. GPUs have SMs (Streaming Multiprocessors). Each SM has a L1 cache. But it is not coherent on GPU. If two different caches from different SMs are reading and writing to the same memory location, it may be possible that one SM may not immediately see the changes done by other SM. GPU launches the grid of blocks of threads. Blocks of threads are mapped on SM. Each SM can have more than one block of threads. Threads in the same block have access to the shared memory inside SM. Threads are also grouped in warps Group of 32 threads is one warp. All 32 threads in a warp can execute only same instruction concurrently in the

form of Single Instruction, Multiple Data (SIMD). But CPU threads can work on different instructions concurrently. Warp has fixed number of threads but number of threads in a block can vary depending on application and programmer [4, 34].

The theoretical comparison analysis between automatic deep learning methods and semi automatic Chan Vese model includes flexibility of fine tuning in semi automatic method based on results in real time by clinician [53]. Deep learning models need huge dataset for training to increase the accuracy of results. This in many cases are difficult to obtain. Even if the model is accurate on sample of slices but there are possibilities when the results will be poorer. Clinicians do not get the flexibility to tune the parameters in real time during testing stage. The deep learning model need to be trained again for new sample slices which is time consuming. Hence we chose traditional algorithms and their parallel implementations on GPU which provides flexibility to fine tune the parameters in real time based on the accuracy of results [34, 56].

Learning based algorithms require huge dataset during training and validation. They use complete computed tomography (CT) liver image of fixed size as an input to the network. But the traditional approaches can operate on region of interest (RoI) to avoid operations on unnecessary elements [56]. These unwanted elements in liver image makes it difficult to implement liver segmentation in real time. It impacts the accuracy of the segmentation. The medical image during processing contains large amount of data and takes lot of memory. When implemented on graphics processing unit (GPU), the memory on the GPU is not sufficient to process the whole image and it slows down the overall process of segmentation [50, 56, 58].

3

MOTIVATION AND OBJECTIVES

Index

In recent years we have witnessed the emergence of Computer Aided Diagnosis and Treatment. From the conceptual seed of high performance computing to the current architectural proposals for the Medical Imaging, a melting pot of techniques and concepts emerged from different fields of technology combine to shape the visions of an already palpable future. The recent but firm involvement of the industry is a clear indication of development maturity, the path of standardization of the services and products that are called to implement this reality. On the other hand, the scientific community has participated very actively in the development since the beginning of the ideas, diversified in multiple areas of research that have contributed the perspective of its field of study. With the purpose of specifying and limiting the scope of the contributions of this doctoral thesis, this chapter addresses the motivation and objectives that support it.

3.1 Motivation

The technical dimension of Medical Imaging addresses the problems arising in real time implementation of medical image enhancement and segmentation. Faced with this challenge, GPU optimization strategies are developed as a mantra to achieve fast and accurate results, regardless of the level of abstraction, application or focus. On the one hand, in order to enhance the quality of liver image, it is about analysis of distribution of image elements over the image. On the other hand, increasing the quality of segmentation in real time, thus distributing and scheduling the tasks of image enhancement and segmentation to achieve real time implementation. In both cases it is necessary to compute the tasks in parallel to decrease the time for computation without or rather increasing the accuracy of the liver segmentation.

With regard to the liver segmentation, CT liver images are need to be preprocessed sometimes, to increase the accuracy of the results. This include cropping, enhancement or denoising. A predominant approach here is the coordination of strategies to achieve the accurate segmentation. It becomes necessary to identify and analyze the preprocessing step for liver segmentation and evaluate the impact.

There are three applications which can be targeted: 1. Liver segmentation 2. Vessel Segmentation 3. Tumor Segmentation. They all need good quality liver images for better accuracy of the results. Hence enhancement plays vital role for the images which are low in contrast. This requires assessment of image enhancement techniques. Then the selection of the segmentation algorithm plays the role to segment the desired organ. It can be semi-automatic or automatic. Further the results should be evaluated for the quality assessment by comparing with the ground truths. These tasks are compute intensive and hence GPU computing is necessary to speedup the overall process of segmentation. This includes GPU optimization strategies for liver enhancement and segmentation.

Seeded region growing is one of the semiautomatic segmentation methods where it starts from user defined seeds and evolves using threshold criteria until the region can not be grown further. This is iterative process and requires efficient mapping of the iterative tasks on GPU for achieving enough speedup. The enhancement of liver images requires histogram analysis and if it is cross-modality based contrast enhancement, 2D histogram analysis becomes necessary which is computationally expensive and hence GPU optimization strategies need to be explored for run time implementations of volume of images.

3.2 Objectives

In this conceptual framework, the research of this doctoral thesis is proposed, which aims to deepen the GPU optimization strategies for liver enhancement applied for liver, vessel or tumor segmentation, focusing on the following main objective.

- GPU acceleration for liver enhancement and segmentation

From this main objective the following secondary objectives are derived.

- Establish a high performance liver enhancement model, focused on the increasing the accuracy of segmentation.
- Develop fast parallel segmentation techniques based on top of the proposed model for enhanced images to reduce the errors and increasing the accuracy of each of the tumor, vessel and liver segmentation.
- Provide quantitative metrics that allow to evaluate the performance of the enhancement and segmentation model with different settings

4

HYPOTHESIS AND METHODOLOGY

Knowing the quality and real time implementation of liver segmentation depends on preprocessing of CT or MR images and the segmentation algorithm, it requires contrast enhancement and the effective approach to get the required quality of segmentation. Segmentation approach such as seeded region growing is an iterative process which starts from a seed and expands until region can not be grown further. While this SRG strategy is simple, but the implementation of iterative procedure on GPU is costlier. The implementation of enhancement to achieve better quality of segmentation adds more computations and hence it is desirable to have real time implementation.

In the discipline of GPU acceleration of liver segmentation, we analyze the impact of enhancement, initial seed points and threshold criteria used for segmentation. This interesting line of research aims to reduce false positives and increase the accuracy of the liver, vessel or tumor segmentation. Although aspects such as stability or convergence have no direct translation to the problem that motivates this doctoral thesis, the underlying concept of heavy computations used in enhancement and segmentation can be applied and implemented on GPU for real time implementations and analyzed in the proposed scenario.

On the other hand, as a final result of the evaluation of quality of segmentation, the segmented image is compared with the ground truths. Mathematically, the quantitative assessment is done to evaluate the quality of segmented image in comparison to ground truth. To avoid unnecessary computations, preprocessing and GPU optimization strategies can be useful for liver segmentation. In this way the importance of a liver data is conditioned with its potential effect on the segmentation results.

These ideas support the approach with which this doctoral thesis has been designed. The research work carried out is built on the logic of combining the concepts to achieve the proposed objectives, based on the following hypotheses.

Hipótesis 1 *In the evaluation of a high performance liver segmentation whose accuracy is dependent on the preprocessing step such as enhancement, a model can be established that relates the liver contrast enhancement which increases the quality of segmentation in terms of accuracy and completeness of the segmentation of the desired organ.*

Hipótesis 2 *Knowing this model, an algorithm that segments the desired organ such as liver, vessel or tumor can be designed to significantly reduce the number of false positives needed to correctly assess the quality of segmentation.*

To demonstrate these hypotheses, the following methodological objectives are proposed:

1. Conduct a study of the state of the art in three lines of research in the field of medical image segmentation: a) liver enhancement to improve the quality of the image, b) segmentation to obtain the desired organ, and c) quantitaive assessment between the segmented image and the ground truth.

2. Propose some algorithms that model: a) contrast enhancement of the CT images, b) segmentation approach and the threshold criteria for the segmentation of liver, vessel or tumor, and c) metrics that quantify the quality of the segmentation result

3. Implement the enhancement and segmentation model on CPU and GPU for the scenario in question. This will allow you to experiment with the large volumes of medical data.

4. Validate the model through extensive experimentation that determines the impact of the parameters on the result, evaluated with the proposed metrics. This proof of concept will be oriented to a general purpose scenario, using medical data of liver images.

5. Compare the performance of the model with other existing techniques that can be applied to the proposed scenario based on the quality metrics.

6. In order to have quality data sets, find the open source dataset and from clinicians along with the ground truths, with which to obtain large volumes of quality data that are adequate for experimentation.

7. With the knowledge obtained from the experimentation with the proposed model, design an adaptive algorithm that establishes the parameters of the model at runtime, with the aim of linking the reduction of false positives to the quality of the segmentation.

8. Subject the algorithm to experimentation using an extensive and varied set of liver data.

9. Evaluate the experimental results to verify compliance with the starting hypotheses.

The realization of the methodological milestones described has resulted in three scientific contributions published in indexed journals that are described and included in the next block of this thesis.

Part II

CONTRIBUTIONS

5

OVERVIEW OF THE CONTRIBUTIONS

INDEX

This block compiles the scientific work done to cover the proposed methodology. The results of this research have been published in three articles that, put in common, include in their conclusions the demonstration of the hypotheses raised. As an introduction to this part, the content of each article and its implication in the achievement of the general and methodological objectives of this doctoral thesis are described below.

5.1 Article 1. *GPU acceleration of liver enhancement for tumor segmentation*

There are many methods proposed to improve the contrast of the image. Histogram equalization, histogram specification and histogram matching are some of the ways to improve the contrast in the image as discussed by [5, 17, 61]. We apply 2D histogram matching where Computed Tomography (CT) liver slice is the target image and the magnetic resonance imaging (MRI) liver slice is the guided image [65, 70]. The cross-modality based contrast enhancement exploits 2D histogram matching for liver enhancement. Once the image is enhanced then the task is to segment tumor from enhanced image. Seeded region growing for tumor segmentation is an easy and effective process. But the task of cross-modality based liver enhancement is computationally expensive and time consuming [33]. Hence it becomes necessary to use GPU for real time performance of liver contrast enhancement and tumor segmentation. We propose accelerated cross-modality guided liver enhancement scheme in this paper and demonstrate that our technique improves tumor segmentation on enhanced image.

The aim of this study is cross-modality based liver enhancement to improve the contrast of CT liver image for tumor segmentation. We propose parallel implementation of liver contrast enhancement. This is accomplished by 2D histogram matching using CT and MRI liver images. We propose dynamic region of interest (RoI) based seeded region growing (SRG) for tumor segmentation from enhanced CT image. The overall average speedup obtained by parallel implementation is 104.416 ± 5.166 times compared to the sequential CPU implementation of the contrast enhancement and tumor segmentation. The enhanced liver image improves the sensitivity and specificity of the lesion segmentation. This is the first work targeted towards the high performance multi-modality guided liver enhancement for tumor segmentation to the best of our knowledge.

5.2 Article 2. *Fast Parallel Vessel Segmentation*

In this paper, we propose persistent, grid-stride loop and IBS based GPU approach for SRG to avoid intermediate memory transfers between CPU and GPU. This also reduces processing over unnecessary image voxels providing significant speedup. Persistent thread block (PT) approach is basically dependent on number of active thread blocks and grid-stride loop becomes essential when the number of threads in the grid are not enough to process the image voxels independently [8, 21].

SRG is an iterative process. SRG is invoked continuously until region can not be grown further. Iterative process in SRG, when implemented on GPU requires terminating kernel and relaunching from CPU (Kernel Termination and Relaunch (KTRL)) and data transfers between CPU and GPU [50, 58]. So our main objective is to reduce these data transfers using different inter block GPU synchronization (IBS) methods resulting in an efficient parallel implementation of SRG. IBS provides flexibility to move all the computations on GPU by providing visibility to updated intermediate data without any intervention from CPU.

We implement parallel image gradient using grid-stride loop and propose gradient and shared memory based fast parallel SRG implemented entirely on GPU without any intermediate transfers between CPU and GPU. This is inspired by parallel processing on static region of interest (RoI) of tiles on GPU. We compare the proposed persistent based parallel SRG with KTRL for accurate vessel segmentation. The gradient based fast parallel SRG for 2D vessel segmentation is 1.9x faster compared to the state-of-the-art. We discuss gradient based seeded region growing and its parallel implementation on GPU. The proposed parallel seeded region growing is fast compared to kernel termination and relaunch and accurate in comparison to Chan-Vese and Snake model for vessel segmentation.

5.3 Article 3. *Accelerating Chan-Vese Model with Cross-Modality Guided Contrast Enhancement for Liver Segmentation*

The study proposes high performance Chan-Vese model for liver segmentation by avoiding intermediate memory transfers between CPU and GPU. But the Chan-Vese model alone is not sufficient for accurate liver segmentation as it sometimes results in many false positives, lowering the sensitivity and accuracy [68, 71] degrading the quality of liver segmentation. Hence, we employ enhancement module before Chan-Vese model for segmentation. The module is based on cross-modality based contrast enhancement. This works on target and guided image. We consider CT liver image as target image and MRI scan as guided image. The cross-modality approach approximate the histogram of target CT scan similar to guided MR image [6, 32]. The proposed parallel approach results in fast and accurate liver segmentation.

The contributions of the paper are as follows: We aim fast parallel Chan-Vese model for liver segmentation and apply Chan-Vese with and without liver contrast enhancement. The GPU implementation is faster compared to CPU and the liver contrast enhancement improves the quality of liver segmentation by reducing the false positives and increasing the sensitivity, dice score and accuracy of the segmentation. The average dice score, sensitivity and accuracy of the liver segmentation are 0.877 ± 0.036, 0.964 ± 0.037 and 0.956 ± 0.022 after liver contrast enhancement improving the quality of segmentation. GPU implementation of Chan-Vese segmentation algorithm improves the average speedup by 99.811 ± 7.65 times and 14.647 ± 1.155 times with and without enhancement in comparison to the CPU.

6

ARTICLE 1: GPU ACCELERATION OF LIVER ENHANCEMENT FOR TUMOR SEGMENTATION

INDEX

GPU Acceleration of Liver Enhancement for Tumor Segmentation

Nitin Satpute, R. Naseem, E. Pelanis, J. Gomez-Luna, F. A. Cheikh, O. J. Elle, J. Olivares, GPU acceleration of liver enhancement for tumor segmentation, Computer Methods and Programs in Biomedicine 184 (2020) 105285, DOI:https://doi.org/10.1016/j.cmpb.2019.105285.

6.1 Abstract

Background and Objective : Medical image segmentation plays a vital role in medical image analysis. There are many algorithms developed for medical image segmentation which are based on edge or region characteristics. These are dependent on the quality of the image. The contrast of a CT or MRI image plays an important role in identifying region of interest i.e. lesion(s). In order to enhance the contrast of image, clinicians generally use manual histogram adjustment technique which is based on 1D histogram specification. This is time consuming and results in poor distribution of pixels over the image. Cross modality based contrast enhancement is 2D histogram specification technique. This is robust and provides a more uniform distribution of pixels over CT image by exploiting the inner structure information from MRI image. This helps in increasing the sensitivity and accuracy of lesion segmentation from enhanced CT image. The sequential implementation of cross modality based contrast enhancement is slow. Hence we propose GPU acceleration of cross modality based contrast enhancement for tumor segmentation.
Methods : The aim of this study is fast parallel cross modality based contrast enhancement for CT liver images. This includes pairwise 2D histogram, histogram equalization and histogram matching. The sequential implementation of the cross modality based contrast enhancement is computationally expensive and hence time consuming. We propose persistence and grid-stride loop based fast parallel contrast enhancement for CT liver images. We use enhanced CT liver image for the lesion or tumor segmentation. We implement the fast parallel gradient based dynamic seeded region growing for lesion segmentation.
Results : The proposed parallel approach is 104.416 ($\pm$ 5.166) times faster compared to the sequential implementation and increases the sensitivity and specificity of tumor segmentation.
Conclusion : The cross modality approach is inspired by 2D histogram specification which incorporates spatial information existing in both guidance and input images for remapping the input image intensity values. The cross modality based liver contrast enhancement improves the quality of tumor segmentation.

6.2 Introduction

Computed tomography (CT) images of abdomen often possess low contrast [6, 9]. Radiologists often manually delineate lesions during segmentation of medical images, which can be difficult, time-consuming and prone to observer variability [29]. Some segmentation algorithms do not perform well when applied on the CT images and are time consuming [15, 28]. However, their performance can be made better once the CT images are preprocessed [17, 31]. Therefore, preprocessed CT images help in refining the lesions. One possible preprocessing step is image enhancement for the better visualization of tumors in undertaking surgical procedures [14, 25, 30].

Efficient preprocessing can certainly help to attain accurate segmentation of the critical structures in medical images [10, 17]. High sensitivity and specificity indicates the improved quality of the segmentation [26, 28]. The liver images obtained from the CT scans are sometimes noisy, low in contrast and contains

high amounts of details. We consider contrast as the important feature. If the image is high contrast then it becomes easier to identify and segment the object of interest [9, 11]. In our case, lesion is necessary to be segmented.

There are many methods proposed to improve the contrast of the image. Histogram equalization, histogram specification and histogram matching are some of the ways to improve the contrast in the image as discussed by [1, 6, 22]. We apply 2D histogram matching where CT liver is the target image and the magnetic resonance imaging (MRI) liver slice is the guided image [23, 27]. Cross modality based contrast enhancement exploits 2D histogram matching for liver enhancement. Once the image is enhanced then the task is to segment tumor from enhanced image. Seeded region growing for tumor segmentation is an easy and effective process. But the task of cross modality based liver enhancement is computationally expensive and time consuming [12]. Hence it becomes necessary to use GPU for real time performance of liver contrast enhancement and tumor segmentation. We propose accelerated cross modality guided liver enhancement scheme in this paper and demonstrate that our technique improves tumor segmentation on enhanced image.

The aim of this study is cross modality based liver enhancement to improve the contrast of CT liver image for tumor segmentation. We propose parallel implementation of liver contrast enhancement. This is accomplished by 2D histogram matching using CT and MRI liver images. We propose dynamic region of interest (RoI) based seeded region growing (SRG) for tumor segmentation from enhanced CT image. The overall average speedup obtained by parallel implementation is 104.416 ± 5.166 times compared to the sequential CPU implementation of the contrast enhancement and tumor segmentation. The enhanced liver image improves the sensitivity and specificity of the lesion segmentation. This is the first work targeted towards the high performance multi-modality guided liver enhancement for tumor segmentation to the best of our knowledge.

The rest of the paper is organized as follows. Section 8.3 briefs the related works, background and motivation with respect to the liver image enhancement. Section 6.4 explains the proposed methodology for liver contrast enhancement and its parallel implementation on the GPU. Further, we discuss dynamic RoI based fast parallel SRG for tumor segmentation in Section 6.5. Performance results and comparison of contrast enhancement and seeded region growing for tumor segmentation are mentioned in the Section 6.6. Section 8.6 concludes summarizing the main results related to the cross modality based contrast enhancement and tumor segmentation.

6.3 Background and Motivation

Segmentation of lesions is a challenging problem in medical images because of the similar intensity values of structures of interest and the nearby regions in image. Research works are targeting various methods for the segmentation [4, 8, 18]. The results of the segmentation are subsequently used in patient specific model for diagnostics, surgery planning and navigation.. One such approach using gradient based SRG has been presented to segment the aorta and rib bones in thorax images by Rai and Nair [8]. Inspired by this idea, we propose parallel SRG to segment tumors from CT liver images.

Image enhancement is regarded as a precursor to the accurate segmentation. CT scans are commonly used due to the availability and quicker imaging time compared to MRI. CT scans often suffer from low contrast which limit their utility [6, 9]. In this work, we show through our experiments that corresponding MR image can be employed to improve the contrast of CT. The idea to enhance an image using another cross modal image has been witnessed in the literature for natural images [14, 17, 25, 31]. The motivation to use cross modality guided image enhancement is to use the additional information contained in the other image having similar contents in different imaging times or position but better contrast or minimal noise. Ultimately, the details in the enhanced image can be improved from the perceptual perspective. In the context of liver images, tumors can be easily seen in the enhanced CT image.

In this regard, the contrast of photographs was improved using the corresponding near infra red images [1, 31]. Histogram specification in combination with wavelet domain processing was used in this work. Yan et. al proposed a variational approach using anisotropic filter to eliminate noise in color images using infrared images [25]. The authors calculated cross correlation between input images and then used joint filtering for denoising in another approach [10, 17].

Deep learning is applied to multimodal image denoising recently [14]. A deep learning method consisting of three convolutional neural networks has been applied to denoise natural images. Various deep learning based approaches for CT denoising have been presented in the last few years, however, they do not incorporate the multimodality guidance and use the CT image alone for supervised learning [3, 23, 27]. Histogram based methods are useful to enhance the global contrast of image [1], however, they introduce bad artifacts in the processed images. Since it does not consider the neighborhood of the pixels while remapping, it does not necessarily gives the desired contrast[1, 9, 16]. Two dimensional histogram specification is presented recently to improve the 1D histogram specification [12]. It uses 2D cumulative distribution function of the input and target images for remapping intensity values in the original image.

We apply same notion to CT liver images by applying 2D histogram matching based cross modality approach for liver contrast enhancement in the following section.

6.4 Methodology: Liver Contrast Enhancement

We aim to improve the contrast of CT liver image considering MRI liver image as the guidance image to increase the quality of lesion segmentation. The methodology includes 2D contrast enhancement, gradient of enhanced image and segment the lesion using gradient based SRG. The parallel approach for liver enhancement and lesion segmentation makes the process faster in order to achieve real time implementation. In this section, we discuss parallel implementation of the cross modality based liver enhancement.

The flow of proposed GPU implementation of cross modality based contrast enhancement is shown in Figure 6.1. We load CT and MRI images of liver on CPU and transfer it to the GPU. They need not be registered slices. The first step of contrast enhancement of CT liver image is 2D (or pairwise) histogram

calculation (Hist_2d). We calculate parallel 2D histogram of both CT (hist_CT) and MRI (hist_MRI) images. A 2D histogram is a plot of pixel and its neighbouring element which allows us to discover, and show, the underlying 2D frequency distribution (shape) of image. This shows how often each set of values (pixel and neighbour) in the image occurs. Instead of just considering the individual pixel values, it considers every possible pixel pair in the input and guidance image and calculate 2D CDF accordingly [2, 12].

Further the calculation of cumulative distributive function (CDF_2d) of CT (CDF_CT) and MRI (CDF_MRI) images on GPU creates the input for the next step i.e. histogram equalization. 2D CDF is a function that describes the probability of a possible pixel pair in the input and guidance image. This helps in finding most frequent pairwise intensity values for histogram equalization [12].

Then we perform parallel histogram equalization (HE_2d). This step spreads out the most frequent pairwise intensity values increasing the global contrast of image. Hence it improves lower contrast areas to gain a higher contrast [2, 12].

The mapping (Map_2d) of histogram equalization onto the CT image gives the enhanced image. It maps the modified intensity values obtained from 2D histogram equalization to the corresponding pixels [12].

Inter block GPU synchronization (IBS) makes sure the updated values are sent to the next modules in GPU computing blocks. These parallel implementations of sub-modules of contrast enhancement are explained in following sections.

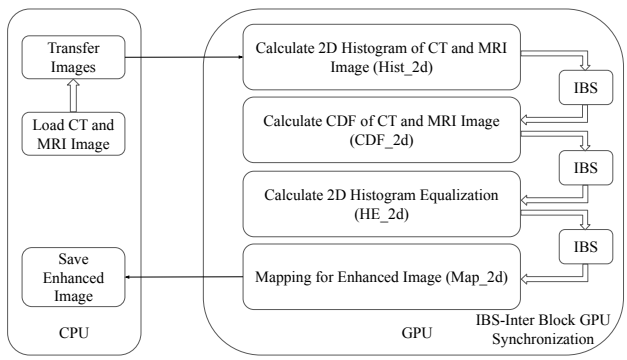


Figure 6.1
GPU Implementation of the Cross Modality based Contrast Enhancement

6.4.1 *2D Histogram*

In this section, we discuss the 2D histogram implementation on GPU as the first step of the contrast enhancement of CT liver image. The histogram length (HL) is 256. We launch HLxHL parallel threads and find the histogram of neighboring elements in pairs. Hence it is called as pairwise histogram. Pairwise histogram is stored in an array of size HLxHL.

For each thread in parallel, it takes the pixel (x,y) and neighbouring pixel (x+1,y) value. This represents one of the indices in the range of (0-HLxHL-1)

Algorithm 1 2D Histogram of CT and MRI Image (Hist_2d)

```
HL=256 and launch HL x HL parallel threads
ti and tj can be any thread id between 0-255
while x<width_of_image do
    while y<height_of_image do
        if ti == I[x][y] and tj == I[x + 1][y] then
            temp=ti*HL+tj;
            atomicAdd(histogram[temp], 1);
        end if
    end while
end while
```

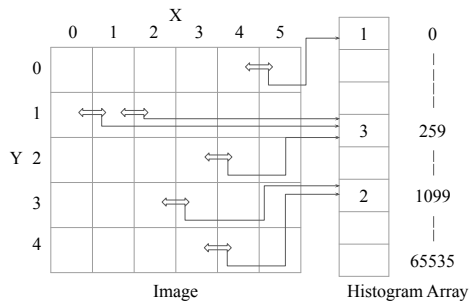


Figure 6.2
2D Histogram

in histogram array given by variable temp as shown in Algorithm 1. We increment corresponding value in the index position in histogram array as shown in Figure 6.2. This function hist_2d for CT and MRI images gives hist_CT and hist_MRI histograms respectively. These 2D histograms are the input to the cumulative distributive function which is the next step of contrast enhancement.

6.4.2 *Cumulative Distributive Function (CDF)*

In this step of contrast enhancement, we calculate CDF of 2D histograms of CT and MRI liver images. The maximum number of histogram pairs can be (w-1)x(h) where w and h are width and height of the image.

Algorithm 2 Calculate CDF of CT and MRI Image (CDF_2d)

```
count= (width-1)*height i.e. maximum number of pairs
HL=256 and launch HL x HL parallel threads
ti and tj can be any thread id between 0-255
temp=ti*HL+tj;
while temp<HL*HL do
    for int j=0; j<=temp; j++ do
        cdf[temp]+ = histogram[j]/count;
    end for
end while
```

We launch HLxHL threads in parallel as shown in Algorithm 2. Each thread calculates its CDF from respective 2D histogram values. These CDF values for CT (CDF_CT) and MRI (CDF_MRI) images are the input to the next step of contrast enhancement which is 2D histogram equalization.

6.4.3 *2D Histogram Equalization (HE_2D)*

2D Histogram Equalization technique improves the contrast of image. It spreads out the most frequent intensity values. This method increases the global contrast of image. This improves the lower contrast areas to gain higher contrast. The pseudocode for 2D histogram equalization is shown in the Algorithm 3. We launch HLxHL threads in parallel. Each thread calculates the corresponding histogram equalization by taking the minimum difference between two CDFs (cdf1 for CDF_CT and cdf2 for CDF_MRI). It takes into the account the first minimum euclidean distance value between the indices when multiple minimum difference in CDFs are found. Again when multiple solutions are available, it further computes and find out the equalized value saved in array HE. This array is ready to get mapped for enhanced image which is the final step of contrast enhancement.

Algorithm 3 Calculate 2D Histogram Equalization (HE_2d)

```
1: HL=256 and launch HL x HL parallel threads
2: ti and tj can be any thread id between 0-255
3: index=ti*HL+tj;
4: for k=0; k<HL; k++ do
5:     for l=0; l<HL; l++ do
6:         temp8=k*HL+l;
7:         temp = cdf1[index]-cdf2[temp8]
8:         if temp is minimum then
9:             x = k
10:        end if
11:        if multiple minimum values found then
12:            temp2 = absolute((ti-k) + (tj-l))
13:            if temp2 is minimum then
14:                x = k
15:            end if
16:            if multiple minimum temp2 are found then
17:                temp3 = absolute((ti-tj) - (k-l))
18:                if temp3 is maximum then
19:                    x = k
20:                end if
21:            end if
22:        end if
23:    end for
24: end for
25: HE[index]=x;
```

6.4.4 *Mapping*

The mapping of 2D histogram equalization is essential for obtaining enhanced CT image as an output. We launch wxh threads where w and h are width and height of the image respectively. This is reverse process of 2D histogram calculation as explained in the psuedocode given by the Algorithm 4. The index value is generated from the neighbouring pixel values of the CT image. The pixel value in the CT image is changed by the corresponding value in the location (index) given by the 2D histogram equalization array. When all the threads are finished processing corresponding pixels, the enhanced image is sent back to the CPU.

Algorithm 4 Mapping for Enhanced Image (Map_2d)

1: launch (width)*(height) parallel threads
2: HL=256
3: tw can be any thread id between 0 to width-1
4: th can be any thread id between 0 to height-1
5: $temp1 = I[tw][th];$
6: $temp2 = I[tw + 1][th];$
7: $index = temp1 * HL + temp2;$
8: $I[tw][th] = HE[index];//EnhancedImage$

6.5 Application to The Tumor Segmentation

Seeded Region Growing is an easy approach to segment the various objects in an image. The result of the region growing relies mainly on the initial seed(s) and the criteria defined to end recursive or iterative region growing process [4, 15, 19, 20]. The parallel implementation of SRG based tumor segmentation is shown in Figure 6.3.

We load CT and MRI images and transfer it to the GPU. GPU performs cross modality based contrast enhancement and stores the enhanced CT image in GPU memory. The control comes back to the CPU. This is essential for the selection of seed(s) and to change the number of persistent blocks. These persistent blocks (i.e. number of available computing resources on the GPU) differ depending on the application. The next task is tumor segmentation. GPU computes the gradient of enhanced CT liver image. The gradient of enhanced liver image is communicated through IBS to the next module for tumor segmentation. We apply SRG on the gradient of enhanced liver image. Region grows and new seeds are formed from initial seed(s) based on the threshold criteria. This process is iterative until the threshold criteria is satisfied. The process stops when new seed(s) can not be formed and region can not be grown further.

In this work, we use threshold criteria defined by the homogeneity of region and region aggregation considering the pixel values and their gradient direction and magnitude. The criteria is defined via a cost function that uses few features of the image around seed. Value of the cost function is compared with homogeneity criteria specified to check if the value is smaller than 1. The pixel becomes part of the region if there is a match; otherwise it is excluded from the region. The cost functions for threshold criteria are given by Rai and Nair [8]. They select homogeneity criterion using gradient based cost function which

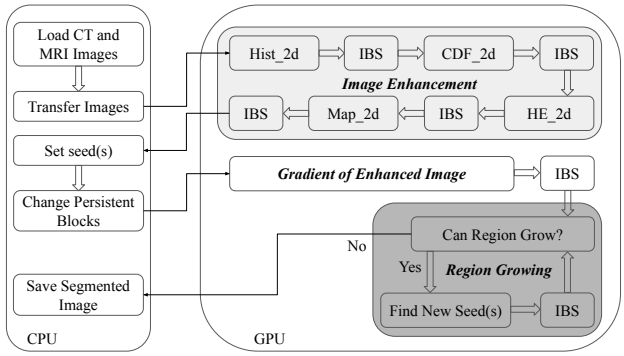
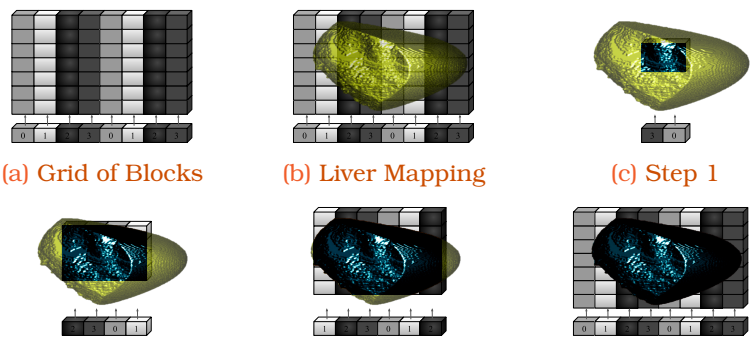


Figure 6.3
GPU Implementation of SRG based Tumor Segmentation

are dependent upon object contrast, texture features like shape and color, intensities values, gradient direction and magnitude. The cost function exploits features of image around the seed.

We apply parallel gradient based SRG algorithm on both enhanced images and original CT liver images. We propose dynamic RoI based parallel SRG.

6.5.1 *Dynamic SRG*



(a) Grid of Blocks (b) Liver Mapping (c) Step 1

(d) Step 2 (e) Step 3 (f) Step 4

Figure 6.4
SRG using Dynamic RoI of Tiles

Dynamic SRG as the name suggests, it increases the region of interest (RoI) in each iteration of SRG. The initial RoI is decided by number of active computing blocks or persistent blocks that can be launched on GPU. This represents the phenomenon of persistence. In order to communicate valid data in between the blocks, inter block GPU synchronization (IBS) is necessary. Persistence and IBS provide flexibility to exploit parallelism using grid-stride loop through constant increase in RoI. One grid-stride is number of active computing threads that can be launched on GPU device.

Gupta et al. [7] have explored persistent thread based GPU programming. The idea behind this is once the SRG kernel launched from CPU, the control returns from GPU when the region is grown completely. Intermediate data transfers

between CPU and GPU are avoided in this approach. SRG kernel on GPU is launched from the host CPU. Region is grown on GPU. Image elements are updated and communicated to the blocks via IBS. The region is grown again on GPU, if new similar neighbouring elements are found. This process continues until no similar neighbouring elements are available. The kernel terminates when the region can not be grown further and control returns to the CPU. Redundant data computations and communications are optimized on GPU using proposed approach. This process is explained in the Figure 6.4.

Algorithm 5 Grid-stride Loop through Dynamic RoI

```
1: blockgrow=1;
2: while blockgrow==1 do
3:     blockgrow=0;
4:     unfinished=1;
5:     Increase RoI of Tiles;
6:     To Increase RoI of Tiles
       w=w+1; h=h+1; d=d+1;
7:     Ensure RoI within image dimensions;
8:     while unfinished==1 do
9:         unfinished=0;
10:        for int i=blockIdx.x; i<=w/blockDim.x; i+=gridDim.x do
11:            for int j=blockIdx.y; j<=h/blockDim.y; j+=gridDim.y do
12:                for int k=blockIdx.z; k<=d/blockDim.z; k+=gridDim.z
                   do
13:                    Region_Growing(arguments, unfinished, blockgrow);
14:                end for
15:            end for
16:        end for
17:        Inter_Block_GPU_Sync();
18:    end while
19: end while
```

There are four persistent blocks processing grid of blocks using grid-stride loop as shown in Figure 6.4a. We map 3D liver on grid of blocks as shown in Figure 6.4b and initialize RoI of tiles around the seed as shown in Figure 6.4c. Persistent blocks operate within RoI. First step of SRG takes place. Region is grown and RoI is incremented in all directions. This process makes necessary neighbouring voxels available for the second step of SRG as shown in Figure 6.4d. New neighbouring voxels perform same function and RoI is incremented again. This flow is repeated until region can not be grown further as shown in Figures 6.4e and 6.4f. This approach reduces compute and memory operations resulting in the increased performance. It is needed to ensure that the increase in RoI lies within the image dimensions.

Complete process is defined in the Algorithm 5. RoI should be initialized in such a way that all threads are busy performing SRG. Variable "blockgrow" is essential to check the increase the RoI. Increase RoI of tiles if value of "blockgrow" is "1", otherwise stop SRG as region is grown completely. This variable "blockgrow" along with the variable "unfinished" are updated in the SRG segmentation step. Lower and upper values of RoI (in x, y, and z directions) are

calculated when "blockgrow" is "1". It has to be made sure that the RoI should not increase beyond image dimensions in the successive steps of SRG.

Persistent blocks operate inside the RoI. Kernel SRG is called for the voxels within the RoI. IBS makes sure only updated values are communicated to the persistent blocks in each step of SRG. IBS can be atomic, quasi, lock free or based on cooperative groups from NVIDIA toolkit CUDA 10.1 [13, 21, 24]. We use quasi IBS for our approach due to its efficient implementation [13].

6.6 Results and Discussion

We discuss performance analysis of proposed parallel cross modality based liver enhancement for tumor segmentation. The enhanced liver images and segmented tumors are shown and the performance analysis of tumor segmentation is discussed based on quality assessment. We use Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz RAM 24 GB, NVIDIA GPU GeForce GTX 1050 (RAM 4GB), and CUDA Toolkit 10.1 to compare the proposed parallel GPU approach with CPU implementation.

6.6.1 *Liver Enhancement*

We propose fast parallel cross modality based contrast enhancement. 2D histogram of CT image is mapped to 2D histogram of guidance or MR image to get a better contrast image.

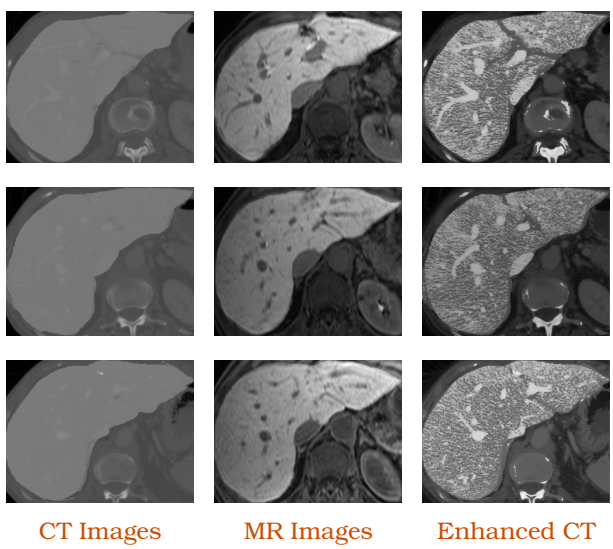


Figure 6.5
CT, MR and Enhanced CT Images

Figure 6.5 shows input CT, MRI and enhanced CT liver images without any tumors. Figure 6.6 shows enhanced CT liver images with tumors. Figures show the contrast is enhanced significantly to observe tumors clearly. Enhanced im-

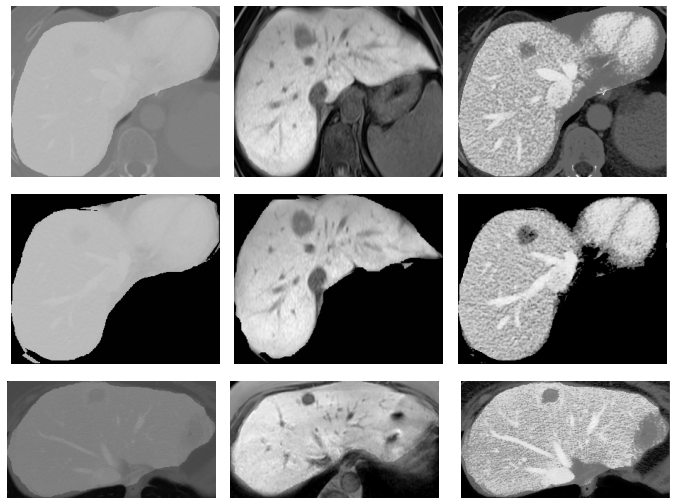


Figure 6.6
CT, MR and Enhanced CT Images showing Tumors

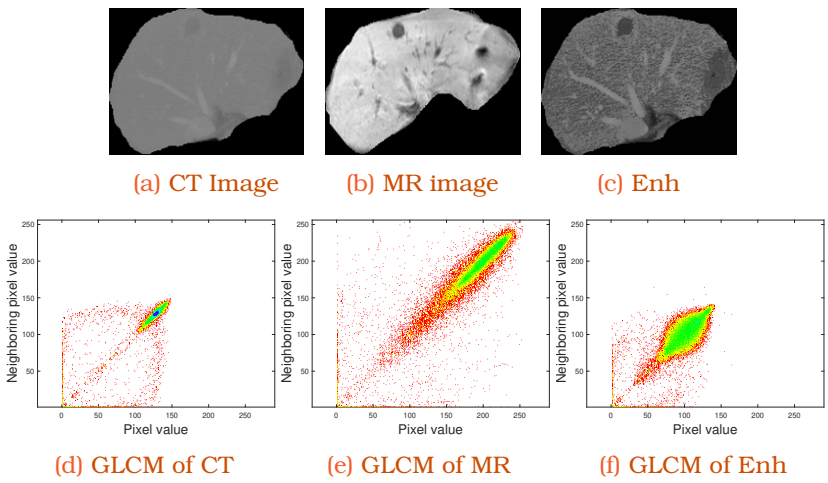


Figure 6.7
CT, MR and Enhanced CT (Enh) with GLCM Plots

age is further processed for tumor segmentation using SRG. Average time taken by NVIDIA GPU GeForce GTX 1050 is 1.976s $\pm$ 0.43s providing the average speedup of 104.416 $\pm$ 5.166 times over CPU implementation (208.082s $\pm$ 55.799s) for tumor segmentation using 2D cross modality based contrast enhancement.

In order to enhance the contrast in CT images, we investigate quality improvements by fusing the information that is available in one modality (e.g. liver inner structures in MRI) to guide the adaptive enhancement in other image modality (e.g. CT in our case). This provides better control over the enhancement and is more effective and efficient than the state of the art technique used by clinicians. Clinicians generally use manual histogram adjustment technique based on 1D histogram specification on CT or MRI scans. This process does not provide efficient distribution of pixels for contrast enhancement of CT or

MRI image. There are more chances of artifacts in 1D enhancement as it results in random histogram and is also a time consuming process.

However, 2D histogram specification incorporates spatial information while calculating 2D CDFs of both the guidance and input images and for remapping the input image intensity values. Instead of just considering the individual pixel values, it considers every possible pixel pair in the input and guidance image and calculate 2D CDF accordingly. Looking at the Gray Level Co-Occurrence Matrix (GLCM) plots in Figure 6.7, it can be observed that the distribution of pixel pairs in GLCM plot of the resulting enhanced image (Figure 6.7f) is expanded but concentrated along the diagonal in comparison to GLCM plots of CT and MR image (Figures 6.7d and 6.7e), which means it does not introduce artificial artifacts unlike 1D histogram specification or histogram equalization.

We provide the histogram comparison of images using 1D and proposed 2D technique as shown in Figure 6.8. The proposed 2D cross modality approach provides a proper distribution of pixel elements using guided MRI compared to 1D approach applied on CT or MRI image. 1D approach introduces unpleasant effects in the enhanced image. The histogram of enhanced CT using cross modality approach is similar to guided MRI image. There are more chances of artifacts in enhanced image using 1D approach as clinicians use manual adjustment which may result in any random histogram of the enhanced image. In the next section, we discuss the impact of cross modality based contrast enhancement for tumor segmentation.

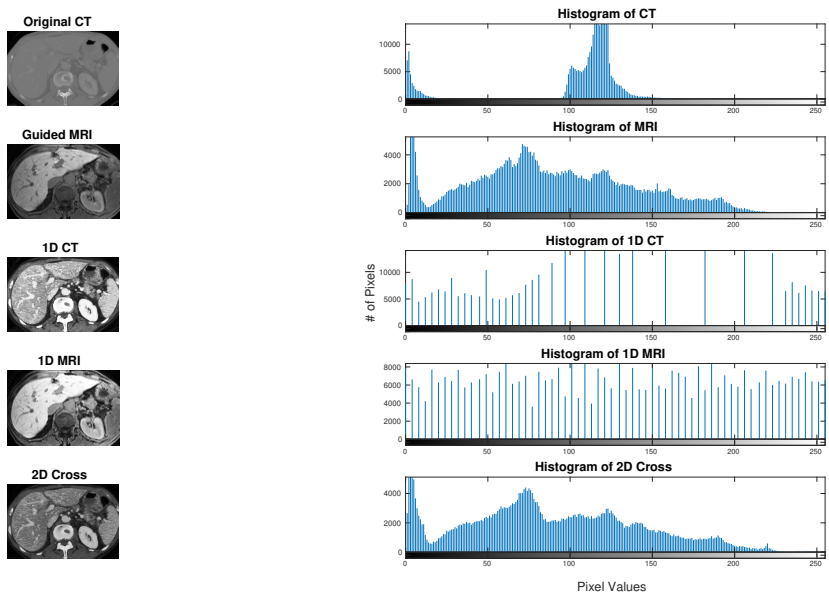


Figure 6.8
Comparison between 2D Cross Modality and 1D Histogram Approach

6.6.2 *Tumor Segmentation*

We propose fast parallel gradient based dynamic SRG for tumor segmentation. Our proposed parallel SRG is implemented on GPU. It does not involve

Table 6.1
Tumor Segmentation Analysis on Five Slices

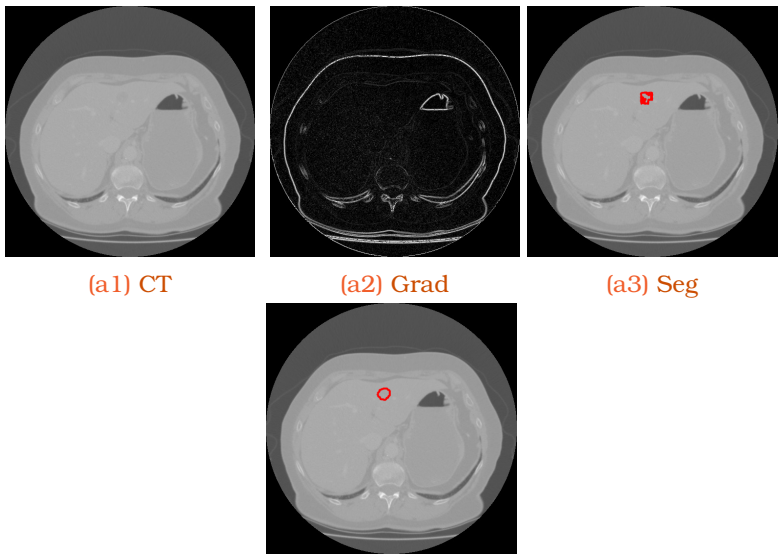
Tumor	without any Enhancement		with Enhancement		Time-Enh+SRG(s)		Speedup
Slice #	Sensitivity, Specificity	Accuracy	Sensitivity, Specificity	Accuracy	CPU	GPU	
1	0.55	0.99899	0.82	0.99906	272.07	2.48	109.706
2	0.38	0.99918	0.81	0.99898	265.98	2.41	110.365
3	0.47	0.99769	0.58	0.9968	167.81	1.68	99.887
4	0.83	0.87091	0.50	0.99765	162.03	1.61	100.64
5	0.47	0.99786	0.74	0.99823	172.52	1.70	101.482
Average	0.54	0.973	0.69	0.998	208.082s	1.976s	104.416
Std. Dev.	0.173	0.057	0.143	0.001	55.799s	0.43s	5.166

transfer of data between CPU and GPU. The data for the research work have been acquired from The Intervention Center, University of Oslo, Norway [5]. The ground truths for tumor segmentation are provided by the clinician. We present the visual comparison of tumor segmentation on both enhanced and original CT liver images. The results in Figures 6.9, 6.10, and 6.11 show the tumor segmentation from original and enhanced liver images. Figure 8.4a1 represents the original CT liver image. The gradient of input CT image is shown in Figure 6.9a2. The tumor segmentation (Seg) and the ground truth (GT) for the original CT liver slice are shown in Figures 8.4a2 and 8.4a3 respectively.

Table 6.2
Tumor Segmentation Analysis on Ten Different Datasets

Dataset #	Size of each Slice (wxh)	Total # of Slices	# of Tumor Slices	without any Enh (Average)		with Enh - Average (Avg.)		Enh+SRG Avg. Time (s)		Avg. Speedup
				Sensi, Speci	Model Accuracy	Sensi, Speci	Model Accuracy	CPU	GPU	
1	406x299	73	10	0.28	0.99132	0.36	0.99517	141.07	1.41	100.054
2	512x512	139	7	0.41	0.99213	0.52	0.99796	252.22	2.29	109.901
3	381x304	67	10	0.48	0.99412	0.65	0.99689	131.89	1.32	99.916
4	405x346	87	8	0.39	0.99325	0.47	0.99717	158.56	1.56	101.641
5	462x321	59	14	0.32	0.99173	0.50	0.99823	167.01	1.63	102.460
6	380x512	58	9	0.49	0.99112	0.64	0.99421	202.02	1.89	106.89
7	443x437	63	6	0.51	0.99201	0.71	0.99501	193.17	1.83	105.55
8	361x249	63	7	0.37	0.99312	0.57	0.99427	126.60	1.26	100.47
9	483x386	80	6	0.31	0.99415	0.59	0.99612	185.78	1.80	103.21
10	456x400	216	30	0.42	0.99178	0.62	0.99324	189.93	1.82	104.35

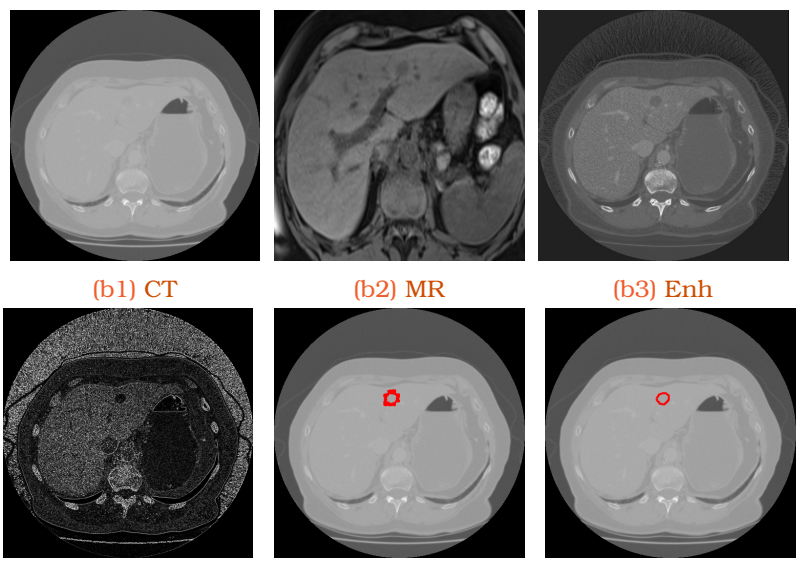
We enhance original CT liver image (Figure 8.4a1) using cross modality based liver enhancement and the enhanced image (Enh_CT) is shown in Figure 8.4b3. The tumor segmentation is performed on the enhanced CT liver image (Figure 8.4b3) and segmented tumor from enhanced CT image is shown in Figure 8.4b4. The quality of tumor segmentation is validated in our clinical validation section using Table 6.1. Tumor segmentation for other CT liver slices are shown in Figures 6.10, and 6.11 and the segmentation quality is improved when the image is enhanced. Hence the cross modality based contrast enhancement on CT liver images improves the quality of tumor segmentation and it is faster. The proposed fast parallel liver enhancement based tumor segmentation is 104.416 $\pm$ 5.166 times faster compared to the sequential implementation. We include Table 6.2 showing experimental evaluation on 10 different datasets (including 107 tumor slices) obtained from The Intervention Centre, Oslo University Hospital, Oslo, Norway. It can be observed from the table that the cross modality based liver enhancement helps in improving the sensitivity, specificity (denoted by 'Sensi' and 'Speci' respectively in Table 6.2) and accuracy of tumor segmentation and GPU implementation of proposed approach is around 100 times faster



(a1) CT (a2) Grad (a3) Seg

(a4) GT

(a) Original Liver Image 1



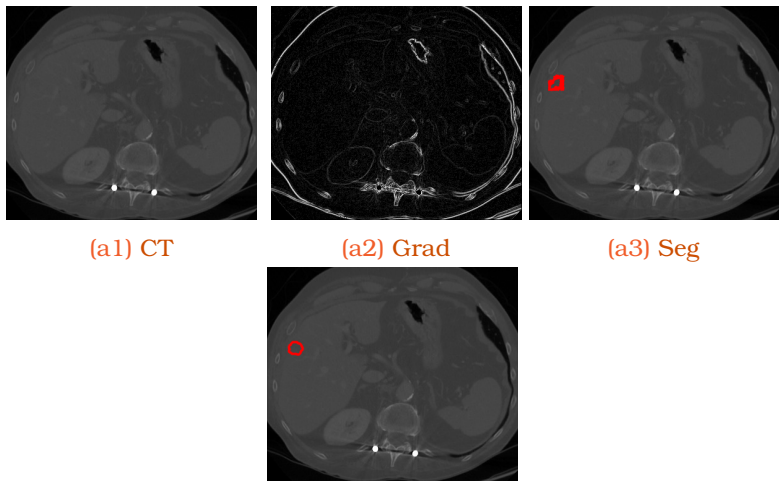
(b1) CT (b2) MR (b3) Enh

(b4) Grad (b5) Seg (b6) GT

(b) Liver Enhancement and Tumor Segmentation from Enhanced CT Image 1

Figure 6.9
Tumor Segmentation from Original and Enhanced CT Image 1

compared to the CPU implementation. P value from ANOVA (analysis of variance) for the ten datasets is 3.31×10^{-14} which is less than 0.05. We reject the null hypothesis and conclude that not all means are equal which confirms the means are statistically significant for the concerned experiments.



(a1) CT (a2) Grad (a3) Seg

(a4) GT

(a) Original Liver Image 2

(b1) CT (b2) MR (b3) Enh

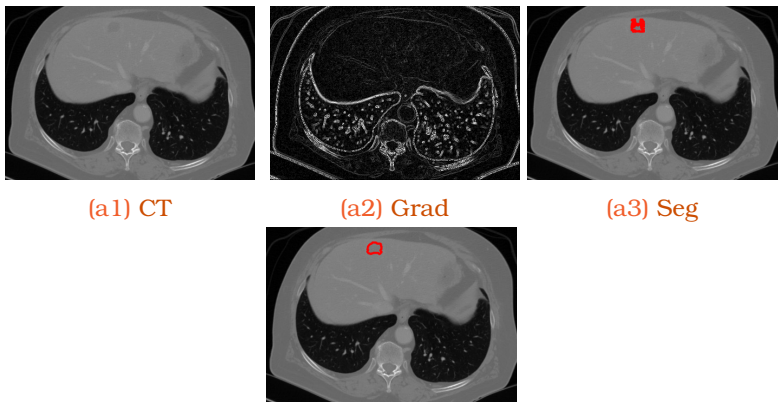
(b4) Grad (b5) Seg (b6) GT

(b) Liver Enhancement and Tumor Segmentation from Enhanced CT Image 2

Figure 6.10
Tumor Segmentation Original and Enhanced CT Image 2

6.6.3 *Clinical Validation*

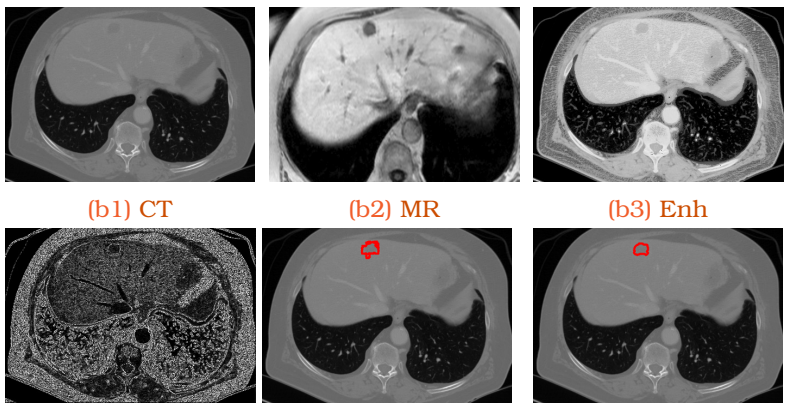
Table 6.1 and 6.2 show the analysis of tumor segmentation before and after enhancement of CT liver images. Table 6.1 includes 5 liver slices with tumors from different datasets and Table 6.2 shows performance evaluation on 10 different datasets including 107 tumor slices. We chose sensitivity (true positive rate or recall) and specificity (true negative rate) as performance metrics for the evaluation of tumor segmentation [26, 28]. It is observed that, the sensitivity and specificity are increased when the accuracy is nearly 1 on the enhanced image. This implies that when the tumor is actually present, then it is predicted more accurately when the image is enhanced.



(a1) CT (a2) Grad (a3) Seg

(a4) GT

(a) Original Liver Image 3



(b1) CT (b2) MR (b3) Enh

(b4) Grad (b5) Seg (b6) GT

(b) Liver Enhancement and Tumor Segmentation from Enhanced CT Image 3

Figure 6.11
Tumor Segmentation from Original and Enhanced CT Image 3

6.6.4 *Discussion*

In this paper, we propose fast parallel cross modality based contrast enhancement for CT liver images. Further GPU performs dynamic RoI based tumor segmentation on enhanced CT liver image. These fast parallel implementations are based on persistence, grid-stride loop and IBS. The process of cross modality based contrast enhancement is computationally expensive and hence time consuming. This involves 2D histogram calculation, equalization and histogram matching [3]. They require several light weight tasks. The performance on GPU is improved compared to the CPU by dividing the tasks on several active threads.

The second part of the process is tumor segmentation. We propose gradient and dynamic RoI based SRG inspired from the works of Rai and Nair [8]. Initially, the process needs small part of the region to be accessed instead of whole image (as implemented previously on GPU). As soon as region grows, RoI

should be increased to access more neighbouring elements. GPU implementation of SRG involves kernel termination and relaunch continuously from CPU. This is time consuming. We avoid this by using persistence and grid-stride loop and obtain the significant speedup i.e. 104.416 ± 5.166 times compared to the sequential implementation of liver enhancement and tumor segmentation.

6.7 Conclusion

In this paper, we discuss cross modality based contrast enhancement for CT liver images, application to tumor segmentation and their fast parallel implementation on GPU. Cross modality based liver enhancement includes CT liver image as an input and MRI liver image as a guided image. CT and corresponding MRI images need not be co-registered. Pairwise 2D histogram implementation and histogram equalization spreads the intensity values across the image producing contrast enhanced CT image. We propose persistence and grid-stride loop based fast parallel implementation on GPU. The enhanced image then used for segmentation of tumors from enhanced CT liver images effectively. We propose gradient and dynamic RoI based seeded region growing for tumor segmentation. The parallel approach for liver enhancement and tumor segmentation is 104.416 ± 5.166 times faster compared to the CPU implementation.

Acknowledgements

The work is supported by the project High Performance soft tissue Navigation (HiPerNav). This project has received funding from the European Union Horizon 2020 research and innovation program under grant agreement No. 722068. We thank The Intervention Centre, Oslo University Hospital, Oslo, Norway for providing images with ground truths for the clinical validation of tumor segmentation.

REFERENCES

[1] Satish S Bhairannawar: *Efficient medical image enhancement technique using transform hsv space and adaptive histogram equalization*. In *Soft Computing Based Medical Image Analysis*, pages 51–60. Elsevier, 2018.

[2] Turgay Celik: *Two-dimensional histogram equalization and contrast enhancement*. Pattern Recognition, 45(10):3810 – 3824, 2012, ISSN 0031-3203. http://www.sciencedirect.com/science/article/pii/S0031320312001525.

[3] Hu Chen, Yi Zhang, Mannudeep K Kalra, Feng Lin, Yang Chen, Peixi Liao, Jiliu Zhou, and Ge Wang: *Low-dose ct with a residual encoder-decoder convolutional neural network*. IEEE Transactions on Medical Imaging, 36(12):2524–2535, 2017.

[4] Konstantinos K. Delibasis, Aristides Kechriniotis, and I. Maglogiannis: *A novel tool for segmenting 3d medical images based on generalized cylinders and active surfaces*. Computer Methods and Programs in Biomedicine, 111(1):148 – 165, 2013, ISSN 0169-2607. http://www.sciencedirect.com/science/article/pii/S0169260713000989.

[5] Åsmund Avdem Fretland, Vegar Johansen Dagenborg, Gudrun Maria Waaler Bjørnelv, Airazat M Kazaryan, Ronny Kristiansen, Morten Wang Fagerland, John Hausken, Tor Inge Tønnessen, Andreas Abildgaard, Leonid Barkhatov, *et al.*: *Laparoscopic versus open resection for colorectal liver metastases*. Annals of surgery, 267(2):199–207, 2018.

[6] Shujun Fu, Ming Zhang, Chengpo Mu, and Xiaohong Shen: *Advancements of medical image enhancement in healthcare applications*. Journal of Healthcare Engineering, 2018, 2018.

[7] Kshitij Gupta, Jeff A Stuart, and John D Owens: *A study of persistent threads style gpu programming for gpgpu workloads*. In *Innovative Parallel Computing-Foundations & Applications of GPU, Manycore, and Heterogeneous Systems (INPAR 2012)*, pages 1–14. IEEE, 2012.

[8] GN Harikrishna Rai and TR Gopalakrishnan Nair: *Gradient based seeded region grow method for ct angiographic image segmentation*. arXiv preprint arXiv:1001.3735, 2010.

[9] Hamid Hassanpour, Najmeh Samadiani, and SM Mahdi Salehi: *Using morphological transforms to enhance the contrast of medical images*. The Egyptian Journal of Radiology and Nuclear Medicine, 46(2):481–489, 2015.

[10] Kaiming He, Jian Sun, and Xiaoou Tang: *Guided image filtering*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(6):1397–1409, 2012.

[11] VL Jaya and R Gopikakumari: *Iem: a new image enhancement metric for contrast and sharpness measurements*. International Journal of Computer Applications, 79(9), 2013.

[12] Seung Won Jung: *Two-dimensional histogram specification using two-dimensional cumulative distribution function*. Electronics Letters, 50(12):872–874, 2014.

[13] Yukihiro Komura and Yutaka Okabe: *Gpu-based single-cluster algorithm for the simulation of the ising model*. Journal of Computational Physics, 231(4):1209–1215, 2012.

[14] Yijun Li, Jia Bin Huang, Narendra Ahuja, and Ming Hsuan Yang: *Deep joint image filtering*. In *European Conference on Computer Vision, 2016*, pages 154–169. Springer, 2016.

[15] Rafael Palomar, Faouzi A Cheikh, Bjørn Edwin, Åsmund Fretland, Azeddine Beghdadi, and Ole J Elle: *A novel method for planning liver resections using deformable bézier surfaces and distance maps*. Computer Methods and Programs in Biomedicine, 144:135–145, 2017.

[16] Guo Qi, Shen Shu-Ting, and Ren Ping-Chuan: *Medical image enhancement algorithm based on improved contourlet*. Journal of Medical Imaging and Health Informatics, 7(5):962–967, 2017.

[17] Xiaoyong Shen, Chao Zhou, Li Xu, and Jiaya Jia: *Mutual-structure for joint filtering*. In *Proceedings of the International Conference on Computer Vision, 2015*, pages 3406–3414. IEEE, 2015.

[18] Erik Smistad: *Seeded region growing*. https://github.com/smistad/-FAST/tree/master/source, 2015.

[19] Erik Smistad, Mohammadmehdi Bozorgi, and Frank Lindseth: *Fast: framework for heterogeneous medical image computing and visualization*. International Journal of Computer Assisted Radiology and Surgery, 10(11):1811–1822, 2015.

[20] Erik Smistad, Anne C Elster, and Frank Lindseth: *Gpu accelerated segmentation and centerline extraction of tubular structures from medical images*. International Journal of Computer Assisted Radiology and Surgery, 9(4):561–575, 2014.

[21] Tyler Sorensen, Hugues Evrard, and Alastair F Donaldson: *Cooperative kernels: Gpu multitasking for blocking algorithms*. In *Proceedings of the 11th Joint Meeting on Foundations of Software Engineering, 2017*, pages 431–441. ACM, 2017.

[22] Bharath Subramani and Magudeeswaran Veluchamy: *Mri brain image enhancement using brightness preserving adaptive fuzzy histogram equalization*. International Journal of Imaging Systems and Technology, 28(3):217–222, 2018.

[23] Jelmer M Wolterink, Tim Leiner, Max A Viergever, and Ivana Išgum: *Generative adversarial networks for noise reduction in low-dose ct*. IEEE Transactions on Medical Imaging, 36(12):2536–2545, 2017.

[24] S. Xiao and W. C. Feng: *Inter-block gpu communication via fast barrier synchronization*. In *IEEE International Symposium on Parallel Distributed Processing (IPDPS), 2010*, pages 1–12, April 2010.

[25] Qiong Yan, Xiaoyong Shen, Li Xu, Shaojie Zhuo, Xiaopeng Zhang, Liang Shen, and Jiaya Jia: *Cross-field joint image restoration via scale map*. In *ICCV, 2013*, pages 1537–1544. IEEE, 2013.

[26] Z. Yan, X. Yang, and K. Cheng: *A three-stage deep learning model for accurate retinal vessel segmentation*. IEEE Journal of Biomedical and Health Informatics, 23(4):1427–1436, July 2019.

[27] Qingsong Yang, Pingkun Yan, Yanbo Zhang, Hengyong Yu, Yongyi Shi, Xuanqin Mou, Mannudeep K Kalra, Yi Zhang, Ling Sun, and Ge Wang: *Low-dose ct image denoising using a generative adversarial network with wasserstein distance and perceptual loss*. IEEE Transactions on Medical Imaging, 37(6):1348–1357, 2018.

[28] M. H. Yap, G. Pons, J. Martí, S. Ganau, M. Sentís, R. Zwiggelaar, A. K. Davison, and R. Martí: *Automated breast ultrasound lesions detection using convolutional neural networks*. IEEE Journal of Biomedical and Health Informatics, 22(4):1218–1226, July 2018.

[29] Y. Zhao, H. Li, S. Wan, A. Sekuboyina, X. Hu, G. Tetteh, M. Piraud, and B. Menze: *Knowledge-aided convolutional neural network for small organ segmentation*. IEEE Journal of Biomedical and Health Informatics, 23(4):1363–1373, July 2019.

[30] Wenzhong Zhu, Huanlong Jiang, Erli Wang, Yani Hou, Lidong Xian, and Joyati Debnath: *X-ray image global enhancement algorithm in medical image classification*. Discrete & Continuous Dynamical Systems-Series S, 12, 2019.

[31] Shaojie Zhuo, Xiaopeng Zhang, Xiaoping Miao, and Terence Sim: *Enhancing low light images using near infrared flash images*. In *International Conference on Image Processing 2010*, pages 2537–2540. IEEE, 2010.

7

ARTICLE 2: FAST PARALLEL VESSEL SEGMENTATION

INDEX

Fast Parallel Vessel Segmentation

Nitin Satpute, R. Naseem, R. Palomar, O. Zachariadis, J. Gomez-Luna, F. A. Cheikh, J. Olivares, Fast Parallel Vessel Segmentation, Computer Methods and Programs in Biomedicine (2020) 105430, DOI: https://doi.org/10.1016/j.cmpb.2020.105430.

7.1 Abstract

Background and Objective:
Accurate and fast vessel segmentation from liver slices remain challenging and important tasks for clinicians. The algorithms from the literature are slow and less accurate. We propose fast parallel gradient based seeded region growing for vessel segmentation. Seeded region growing is tedious when the inter connectivity between the elements is unavoidable. Parallelizing region growing algorithms are essential towards achieving real time performance for the overall process of accurate vessel segmentation.
Methods:
The parallel implementation of seeded region growing for vessel segmentation is iterative and hence time consuming process. Seeded region growing is implemented as kernel termination and relaunch on GPU due to its iterative mechanism. The iterative or recursive process in region growing is time consuming due to intermediate memory transfers between CPU and GPU. We propose persistent and grid-stride loop based parallel approach for region growing on GPU. We analyze static region of interest of tiles on GPU for the acceleration of seeded region growing.
Results:
We aim fast parallel gradient based seeded region growing for vessel segmentation from CT liver slices. The proposed parallel approach is 1.9x faster compared to the state-of-the-art.
Conclusion:
We discuss gradient based seeded region growing and its parallel implementation on GPU. The proposed parallel seeded region growing is fast compared to kernel termination and relaunch and accurate in comparison to Chan-Vese and Snake model for vessel segmentation.

7.2 Introduction

In medical imaging, vessel segmentation from liver slices is one of the challenging tasks. Seeded region growing (SRG) is a widely used approach for semi automatic segmentation [13, 18]. Delibasis et. al. [2] have proposed a tool based on a modified version of SRG algorithm, combined with a priori knowledge of the required shape. SRG starts with a set of pixels called seeds and grows a uniform, connected region from each seed. Key steps to SRG are to define seed(s) and a classifying criterion that relies on the image properties and user interaction [23]. SRG starts from a seed and finds the similar neighboring points based on the threshold criteria using 4 or 8 connectivity. Region is grown if the threshold criteria is satisfied. Similar neighbors are new seed points for the next iteration. This process is repeated until the region can not be grown further. In practice, it demands high computational cost to the large amount of dependent data to be processed in SRG especially in the medical image analysis and still requires efficient solutions [30].

SRG is an iterative process. SRG is invoked continuously until region can not be grown further. Iterative process in SRG, when implemented on GPU requires terminating kernel and relaunching from CPU (Kernel Termination and Relaunch (KTRL)) and data transfers between CPU and GPU [18, 23]. So our main objective is to reduce these data transfers using different inter block GPU

Table 7.1
List of Abbreviations with Full Forms

List	Full Forms
SRG	Seeded Region Growing
GPU	Graphics Processing Unit
CPU	Central Processing Unit
RoI	Region of Interest
KTRL	Kernel Termination and Relaunch
IBS	Inter Block GPU Synchronization
CT	Computed Tomography
PT	Persistent Threads
SM	Streaming Multiprocessor
CUDA	Compute Unified Device Architecture
DS	Dice Score

synchronization (IBS) methods resulting in an efficient parallel implementation of SRG. IBS provides flexibility to move all the computations on GPU by providing visibility to updated intermediate data without any intervention from CPU.

In this paper, we propose persistent, grid-stride loop and IBS based GPU approach for SRG to avoid intermediate memory transfers between CPU and GPU. This also reduces processing over unnecessary image voxels providing significant speedup. Persistent thread block (PT) approach is basically dependent on number of active thread blocks and grid-stride loop becomes essential when the number of threads in the grid are not enough to process the image voxels independently [1, 5].

We implement parallel image gradient using grid-stride loop and propose gradient and shared memory based fast parallel SRG implemented entirely on GPU without any intermediate transfers between CPU and GPU. This is inspired by parallel processing on static region of interest (RoI) of tiles on GPU. We compare the proposed persistent based parallel SRG with KTRL for accurate vessel segmentation. The gradient based fast parallel SRG for 2D vessel segmentation is 1.9x faster compared to the state-of-the-art.

The rest of the paper is structured as follows. Section 8.3 briefs relevant works and state-of-the-art with respect to SRG. Section 7.4 explains GPU approaches (KTRL and Static) for SRG implementation using persistence and grid-stride loop. The application of parallel SRG to vessel segmentation is discussed in the Section 7.5. Performance results and comparison of persistent and grid-stride loop based parallel SRG for vessel segmentation are mentioned in the Section 8.5. Section 8.6 concludes summarizing the main conclusions of this paper and indicating future directions. List of abbreviations with explanations are mentioned in Table 7.1.

7.3 Background and Motivation

There are many works done on image segmentation recently which are based on snake based model [17], gradient vector flow [22, 35], and level set based Chan-Vese model [19]. Scientists have explored the snake model for segmentation. Snakes are defined as a set of points around a contour [17]. But the problem with the snake model is that the contour never sees the strong edges that are far away and the snake gets hung up due

to many small noises in the image [17]. Hence researchers came up with the solution called gradient vector flow (GVF). In GVF, instead of using image gradient, a new vector field is created over image plane [22, 35]. Cost of GVF includes smoothness and edge map but it requires keeping track of the number of points and point distribution. Hence researchers came up with another solution called as level sets based Chan-Vese model for image segmentation [19]. In the absence of strong edges, a region based formulation for image segmentation is proposed by Chan-Vese model. Chan-Vese model for active contours is a powerful and flexible method which is able to segment many types of images. But amongst all, SRG is the simplest algorithm and plays a vital role in medical image segmentation [11, 18].

Smistad et. al. [20, 21] have discussed parallel SRG for image segmentation. The reference implementation is shown in Figure 7.1. Medical image dataset is cropped before processing. Then the CPU allocates the memory equivalent to the cropped size to copy the data to the cropped image on the GPU. Further SRG is performed for image segmentation. This is the simplistic representation of the work by Smistad et. al. [21]. We have not considered pre-processing stage in this work assuming the images are pre-processed. Smistad et. al. [21] have proposed non persistent thread (non-PT) approach for SRG based vessel segmentation.

Smistad et al. [23] have proposed parallel region growing with double buffering algorithm based on the parallel breadth first search algorithm by Harish and Narayanan [6]. They have suggested a dynamic queue for SRG and mentioned that changing the number of threads (due to border expansion of the region) typically involves restarting the kernel, and this requires reading all the values from global memory again. But they have not recommended probable solution for this problem. Smistad et al. [21] have presented a data parallel version of the SRG based Inverse Gradient Flow Tracking Segmentation algorithm using KTRL. Zhang et al. [33] have implemented bidirectional region growing where they have used a dynamic queue (stack). Jiang et al. [9] have proposed improved branch based region growing vessel segmentation algorithm using stack.

GPU based implementation of SRG needs a dynamic queue (stack). CPUs provide hardware support for stacks but GPUs do not [1]. Any queuing system has a large number of pieces of work to do and a fixed number of workers corresponding to the fixed number of computing units. Pieces are then assigned dynamically to the workers. The problem is deciding the maximum number of pieces of work in the queuing system. If decided, persistent blocks iterate through these pieces of work in the queuing system.

GPU implementation of a stack requires continuous changes in memory allocations which in turn requires iterative GPU kernel invocation from CPU in other words kernel termination and relaunch as discussed in the algorithms IVM backtracking and work stealing phase by Pessoa et al. [15]. Task-parallel run-time system, called TREES, that is designed for high performance on CPU/GPU platforms by Hechtman et al. [8] have shown the invocation of GPU kernels from CPU iteratively for updating task mask stack (TMS) in TREES execution. The loop involved while implementing data flow through the stream kernels of the rendering system (involving stack) on GPU controlled by CPU (that is KTRL) is proposed by Ernst et al. [4].

Nevertheless, there is an alternate GPU implementation of queuing system (stack) using dynamic kernel launching. Chen et. al. [2015] have proposed

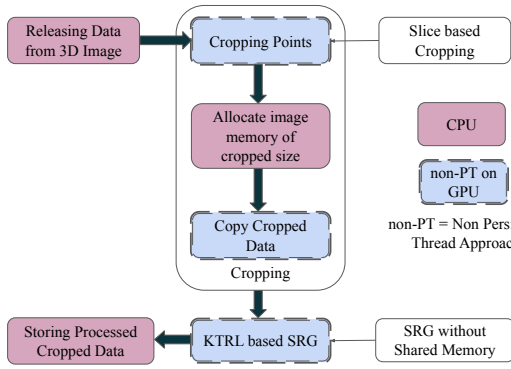


Figure 7.1
Reference Approach

free launch based dynamic kernel launches through thread reuse technique [1]. This technique requires no hardware extensions, immediately deployable on existing GPUs. By turning subkernel launch into a programming feature independent of hardware support, free launch provides alternate approach for subkernel launch which can be used beneficially on GPUs.

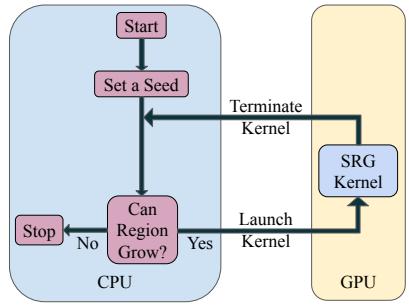
KTRL includes terminating a GPU kernel and invoking it from the CPU if the region can be grown further [21, 23]. GPU kernel SRG is called from CPU. Region grows from a seed based on the threshold criteria. SRG kernel is terminated and relaunched from CPU if region is not grown completely. This process continues until region can not be grown further. The process involves transfer of data to and fro from CPU and GPU. In KTRL, SRG kernel operates on each voxel of whole image data in all the iterations. It includes redundant memory transfers and unnecessary computations over complete image. Hence the main contributions of this paper are the implementation of persistence based approaches to improve the performance of SRG by reducing unwanted computations and avoiding intermediate memory transfers between CPU and GPU. Memory on the GPU is limited and may not be enough for processing large medical datasets. However, most medical datasets contain a lot of data that is not part of the RoI.

The process of KTRL which involves iterative calling of the kernel is not efficient when implemented on GPU. Hence, as an optimized solution to KTRL, we propose persistent and grid-stride loop based GPU approaches. These approaches are based on processing over static RoI of tiles and dynamic RoI of tiles. We discuss the further details in the upcoming sections.

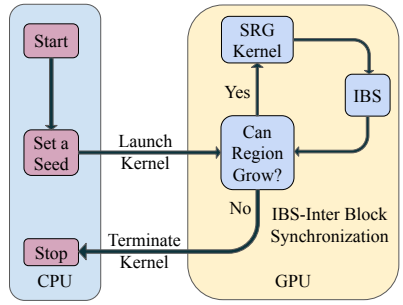
7.4 Parallel SRG

GPU is a grid of block of threads. Thread is the smallest computational unit mapped on the cores and block of threads are mapped on the streaming multiprocessors (SMs). Each SM can occupy more than one block. The threads from independent blocks can access data via shared memory in the SM [29]. In order to communicate valid data between the blocks, these persistent blocks need to be synchronized via IBS through device memory. Persistence implies maximum number thread blocks that can be active at the time of computation depending upon the GPU resources available [5, 31].

We use PT and shared memory based approaches for SRG implementations. Shared memory and grid-stride loop based SRG reduces total memory transfers and computations. Grid-stride is inspired when the grid is not large enough to occupy all the data elements [7, 25]. Rather than assuming that the thread grid is large enough to cover the entire image elements, the kernel loops over the image one grid-size at a time. The stride of the loop is the total number of threads on the grid [7]. These threads (or block of threads) iterate over the image until the process of SRG terminates.



(a) using Kernel Termination and Relaunch [20, 21]



(b) using Proposed Grid-Stride Loop

Figure 7.2
GPU Implementations of Seeded Region Growing (SRG)

For each thread in parallel on GPU, SRG starts from the seed thread and finds similar neighbours surrounding it Region is grown by making similar neighbouring elements as new seeds. The process of SRG is repeated until similar neighbours can not be found. Normally SRG can be implemented on the GPU as a recursive or iterative kernel calling (KTRL) as shown in Figure 7.2a. Kernel calling involves invocation of a grid of block of threads. The blocks are executed on streaming multiprocessors and threads are executed on cores. Park et el. [14] and Smistad et al. [23] have given brief introduction about CUDA (Compute Unified Device Architecture) architecture and GPU computing. They have detailed the information on grid, blocks, threads and memory hierarchy of CUDA architecture.

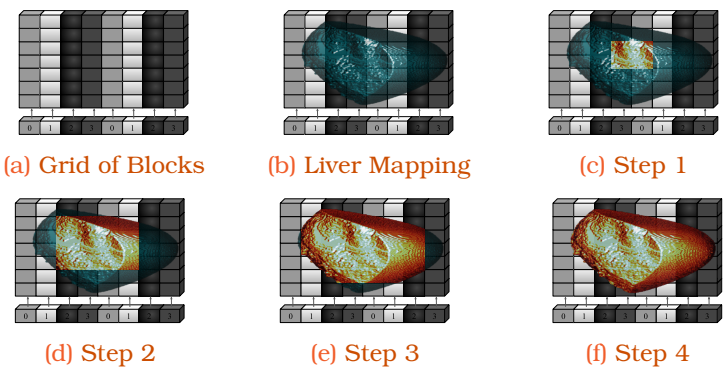
SRG can be recursive or iterative process. Recursive kernel calling can not utilize GPU cores efficiently due to hardware limitations [27]. Iterative GPU kernel call from CPU is costlier due to memory transfers between CPU and GPU and it involves all the image elements to be considered in each step of SRG. GPU implementation of SRG using KTRL is shown in the Figure 7.2a. It

shows that, the kernel SRG is called on GPU continuously from the host CPU until the region can not be grown further. It starts from the seed, finds similar neighbours and grows the region. This process continues until the region can not be grown further. The process of the KTRL causes unnecessary image elements to be part of computations and intermediate memory transfers between CPU and GPU.

Hence in order to avoid these problems, we propose grid-stride loop through complete image based GPU approach as shown in the Figure 7.2b. SRG starts from the seed and the control goes to GPU. The SRG kernel is launched if the region is not grown completely. IBS is needed in order to transfer valid data in between the active thread blocks. The number of active thread blocks on SMs are limited due to resource constraints. These maximum number of active blocks are persistent blocks [5, 29, 31]. The looping i.e. grid-stride loop terminate when the region can not be grown further and control returns to the host CPU as shown in the Figure 7.2b. We have discussed KTRL based GPU approach for SRG implementation and its disadvantages. Now, we are going to analyze PT based GPU approaches for high performance SRG implementation. Proposed approaches exploit parallelism using persistence and IBS as detailed in the static and dynamic approaches.

7.4.1 *Static Approach*

In the proposed approach, we apply grid-stride loop through static RoI (complete image) using persistence and IBS [5, 31]. The complete liver image is mapped on the GPU as grid of block of threads as shown in the Figure 7.3b. CPU invokes SRG kernel on GPU. Persistent blocks iterate through complete image and grow region from the seed in each and every iteration on GPU. This iteration of persistent blocks over the tiles of the image and the grid-stride loop based SRG is shown in Figure 7.2b. Steps of SRG in Figures 7.3c - 7.3f show the grown region of the liver. SRG kernel terminates when the region is grown completely. We copy the data from the device memory to the shared memory. This data is shared by all the threads inside the blocks. This is necessary to share the neighbouring elements between different voxels of the image. For each parallel thread in the block, if seed is found and is not the boundary element of the block, we calculate similar neighbouring elements. Region is grown by making similar neighbouring elements as new seeds.



(a) Grid of Blocks (b) Liver Mapping (c) Step 1

(d) Step 2 (e) Step 3 (f) Step 4

Figure 7.3
SRG using Persistence and Grid-Stride Loop through Complete Image

There are four persistent blocks shown in Figure 7.3. These four persistent blocks are iterated through liver elements. Tiles with the same color are iterated by same persistent block. In KTRL, these tiles are processed by the thread blocks randomly. Grid-stride loop by persistent blocks is applied on the tiles over complete liver image (Step 1 in Figure 7.3c). Region grows around the seed containing similar elements. IBS is applied to communicate valid data in between the blocks for the next step of SRG as shown in Figure 7.2b.

Algorithm 6 Grid-stride Loop through Complete Image

```
1: unfinished=1;
2: while unfinished==1 do
3:     unfinished=0;
4:     for int i=blockIdx.x; i <= width/(blockDim.x − 2); i=i+gridDim.x do
5:         for int j=blockIdx.y; j <= height/(blockDim.y − 2); j=j+gridDim.y do
6:             for int k=blockIdx.z; k <= depth/(blockDim.z − 2); k=k+gridDim.z do
7:                 Region_Growing(arguments, unfinished);
8:             end for
9:         end for
10:    end for
11:    Inter_Block_GPU_Sync();
12: end while
```

Persistent blocks iterate over the liver image and the region is grown again in step 2 as shown in Figure 7.3d. IBS is applied and valid data is communicated in between the blocks so that the region can be grown further as shown in Figures 7.3e and 7.3f. After step 4 in Figure 7.3f, SRG stops as region can not be grown further. Each step contain many iterations where region starts growing when persistent blocks iterate through tiles of the image. This iterative process continues until region can not be grown further. Code snapshot of the complete process is provided in the Algorithm 6.

Global variable "unfinished" is 1 if region has to be grown further else it is 0. Persistent blocks in x, y and z directions iterate through complete image. Two is subtracted from block dimensions to avoid computations around boundary voxels (from left and right in each dimensions) from shared memory as region can not be grown further in the block. After each step of SRG, when the processing on complete image is done then all the persistent blocks are globally synchronized via "Inter_Block_GPU_Sync()" barrier. This ensures that valid data is communicated for the next step of SRG. This barrier can be Atomic(), Quasi(), LockFree() or can be implemented using NVIDIA CUDA API Cooperative-groups [10, 24, 31]. We use quasi based IBS because of its efficicient implementation [10].

The difference between static and dynamic approach by Nitin et. al. [18] can be explained in terms of static and dynamic RoI of tiles. In static approach, RoI remains constant and SRG happens within the constant RoI until the region can not be grown further. Whereas in dynamic approach, SRG starts within

the initial RoI. RoI increases and includes more elements uniformly in all the directions for the next step of SRG. SRG takes place, RoI increases and the region is grown further. Hence RoI changes in each step of SRG until the region can not be grown further in a dynamic approach. In the next section, we present 2D vessel segmentation as an application to static RoI based SRG.

7.5 Application to 2D Vessel Segmentation

The 2D segmentation algorithm is inspired by the gradient based SRG algorithm developed by Rai and Nair [16]. We proposed the fast parallel SRG based segmentation algorithm on GPU for vessel segmentation. We discuss the two important modules i.e. image gradient and SRG for the fast parallel 2D segmentation of vessels from CT liver images.

7.5.1 *Parallel Image Gradient*

Rai and Nair [16] have presented homogeneity criterion selection and its impact on the quality of segmentation using SRG. In general, the threshold criteria include object contrast, region boundary, homogeneity of the region, intensities values and texture features like shape and color. But we include cost functions mainly based on intensity values and their gradient direction and magnitude.

Algorithm 7 Parallel Image Gradient using Grid-stride Loop

```
1:  voxel.x = blockIdx.x * blockDim.x + threadIdx.x;
2:  voxel.y = blockIdx.y * blockDim.y + threadIdx.y;
3:  stridex = blockDim.x * gridDim.x;
4:  stridey = blockDim.y * gridDim.y;
5:  for int k=voxel.x; k < rows; k=k+stridex do
6:      for int l=voxel.y; l < cols; l=l+stridey do
7:          candidate.x = k + 1; candidate.y = l + 1;
8:          check if neighbour candidate is within image dimensions;
9:          gx = 0.5*(data[candidate.x*cols + l] - data[k*cols + l]);
10:         gy = 0.5*(data[k*cols + candidate.y] - data[k*cols + l]);
11:         g = sqrt(gx*gx + gy*gy);
12:         data_g[k*cols + l]=g;
13:         if(max_g < g) atomicMax(&max_g, g);
14:         if(min_g > g) atomicMin(&min_g, g);
15:     end for
16: end for
```

The cost function exploits certain features of the image around the seed. Gradient based cost function requires gradient of the image, largest gradient mag-

nitude (max_g) and minimum gradient (min_g) present in the image. The cost functions are:

$$cost1 = g/(k * max\_g) \qquad 0 < cost1 < 1 \tag{7.1}$$

$$cost2 = (max\_g - g)/(max\_g - min\_g) \qquad 0 < cost2 < 1 \tag{7.2}$$

where g is gradient magnitude of the pixel under consideration and k is the constant parameter which controls the region growth. The pixel under consideration is added in the growing region if it matches with the seed elements i.e. cost functions specified by Equations 7.1 and 7.2 are satisfied otherwise it is excluded from consideration.

We propose grid-stride loop based parallel image gradient method in Algorithm 7. For each pixel in parallel, we calculate its gradient magnitude (g) with respect to neighboring element. Horizontal and vertical gradient components are given by gx and gy. The magnitude of maximum and minimum gradients are updated simultaneously. The gradient of the image is desired input for SRG based segmentation along with the seed. This is discussed in the next section.

7.5.2 *Parallel Vessel Segmentation*

We propose fast parallel vessel segmentation as shown in Figure 7.4. The algorithm is inspired from gradient based segmentation algorithm by Rai and Nair [16]. Figure 7.4 shows parallel implementation of vessel segmentation where the user selects seed(s). These seed(s) along with the image are transferred to the GPU. Device kernel calculates the image gradient in parallel as discussed in the earlier section. The IBS is necessary to reflect the updated image gradients in the device memory.

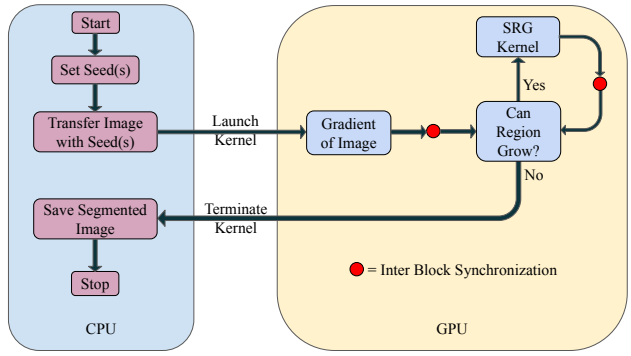


Figure 7.4
Proposed Parallel Vessel Segmentation

Further we apply SRG algorithm. The cost functions based on gradient are shown in Equations 7.1 and 7.2. For each pixel in parallel, the pixel under consideration invokes SRG kernel if it satisfies the cost functions. The seeds are updated after IBS and the gradient based cost functions are verified again for new pixels. This process continues until no new seeds are formed i.e. no new pixels are added to the growing region.

The kernel is terminated and the control returns to the CPU when the region is grown completely. The segmented image is transferred to the CPU. The process of segmentation stops. This GPU implementation avoids iterative call of SRG kernel from CPU. We use gradient and persistent based parallel SRG for vessel segmentation.

7.6 Performance Evaluation

We propose persistent and grid-stride based GPU approaches for fast parallel 2D vessel segmentation. The performance results are obtained from KTRL and proposed persistent based GPU approach. We compare proposed approaches with KTRL. We use Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz RAM 24 GB, NVIDIA GPU 1050 (RAM 4GB), OpenCL 1.2 (ref. [26]) and CUDA Toolkit 10.1 for the implementation.

7.6.1 *Liver Dataset and Ground Truth*

Liver data for the research work has been acquired from The Intervention Center, University of Oslo, Norway [3]. The ground truths for vessel segmentation are provided by the clinician. The modality used is Computed tomography (CT). For the ground truth, images are pre-processed through locally developed application with 3D Slicer to enhance vessels [12]. In some cases, the same application is used for vessel segmentation and separation of portal and hepatic vessels although another possibility is to use active contour tool using ITK-SNAP and manual correction [3, 18]. Table 8.1 shows information about images of different sizes including total number of vessel slices used for experimentation from particular volume.

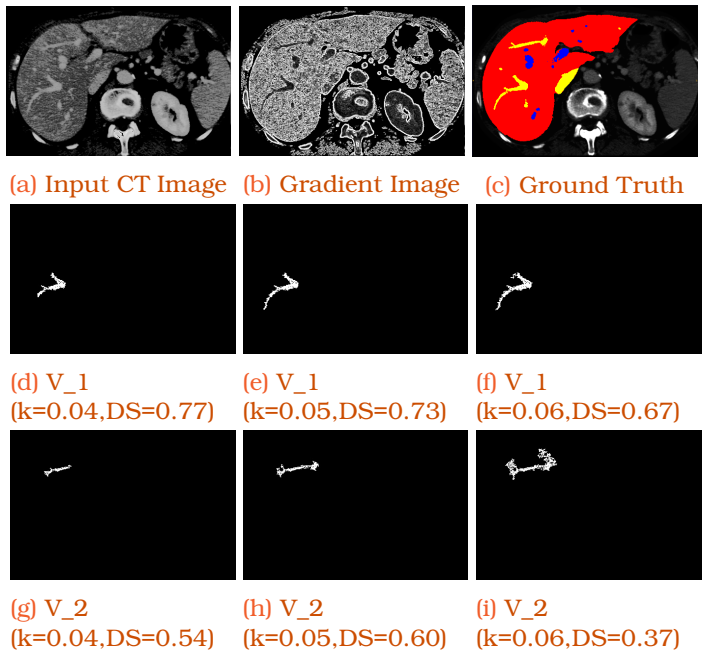
7.6.2 *Parallel 2D Vessel Segmentation*

We propose persistent and grid-stride based GPU approaches for fast parallel 2D vessel segmentation. Variations in vessel segmentation with parameter k using parallel SRG is shown in Figure 7.5. Input to parallel SRG is CT liver slice as shown in Figure 7.5a. Gradient of input CT image and the ground truth for the segmentation are shown in Figures 7.5b and 7.5c respectively. Dice similarity coefficient (DS) and Precision [28, 34] are used to assess the quality of vessel segmentation. Dice similarity coefficient measures the similarity between ground truth and the segmented output. If they are identical (i.e. they contain the same elements), the coefficient is equal to 1.0, while if they have no

Table 7.2
Liver Dataset with Vessels

Volume #	Total # of Slices	Image Size (wxh)	# of Slices with Vessels
10504	59	460x306	7
18152	139	512x512	5
23186	87	405x346	6
28059	59	462x321	6

elements in common, it is equal to 0.0. Otherwise it is somewhere in between 0 to 1. Precision describes the number of positive detections with respect to the ground truth. Of all of the elements that are segmented in a given liver vessel image, the number of these elements actually had a matching ground truth annotation can be called as precision.



(a) Input CT Image (b) Gradient Image (c) Ground Truth

(d) V_1 (k=0.04,DS=0.77) (e) V_1 (k=0.05,DS=0.73) (f) V_1 (k=0.06,DS=0.67)

(g) V_2 (k=0.04,DS=0.54) (h) V_2 (k=0.05,DS=0.60) (i) V_2 (k=0.06,DS=0.37)

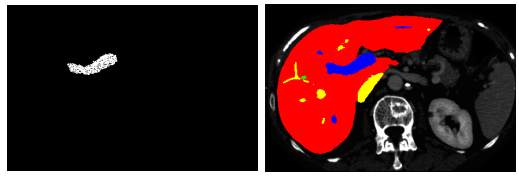
Figure 7.5
Variations in Fast Parallel Vessel Segmentation with Constant Parameter 'k' using Parallel SRG on First Liver Slice

We show the two segmented vessels with change in parameter k i.e. 0.04, 0.05, and 0.06. The first segmented vessel as shown in Figures 7.5d, 7.5e, 7.5f is accurate at 0.04 with high dice score i.e. DS=0.77. Similarly we show the segmentation of second vessel from the same slice. The variations in the quality segmentation due to k are shown in Figures 7.5g, 7.5h, 7.5i. The more accurate segmentation is obtained at 0.05 as the dice similarity coefficient value is higher i.e. DS=0.60.

Further we show the quality of segmentation on another CT Slice as shown in Figure 7.6a and calculate the gradient (Figure 7.6b) of the input CT image. GPU computes parallel SRG using gradient based thresholding criteria giving more accurate results with high dice score at k=0.05 for two vessels inside the CT slice as shown in Figures 7.6c and 7.6d. The ground truth for the segmentation is shown in Figure 7.6e. We analyze that the vessels are more accurately segmented due to better value of dice similarity coefficient when the parameter k takes the value 0.05 as shown in Figure 7.6.

The speedup obtained by proposed parallel static approach over KTRL on first two CT liver slices are shown in the Table 7.3. The maximum speedup for vessel segmentation by proposed parallel static SRG is 1.67x in comparison to KTRL on the first liver slice. But the average speedup obtained by proposed parallel static approach for all the vessels (in 6 slices tested) is 1.9x compared to KTRL. We evaluate the speedup of the vessel segmentation when the vessel segmentation is more accurate with better Dice score value (Figure 7.5).

(a) Input CT Image (b) Gradient Image (c) V_1, DS=0.88

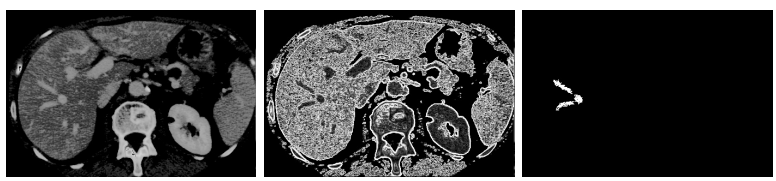


(d) V_2, DS=0.89 (e) Ground Truth

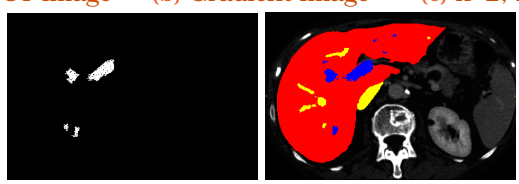
Figure 7.6
Vessel Segmentation (for k=0.05) using Parallel SRG on Second Liver Slice

Table 7.3
Time and Speedup for Vessel Segmentation

Data →	Vessel Segmentation	
GPU Approaches → Metrics ↓	KTRL	Static (Speedup)
Time in ms for kernel SRG - 1st Slice (k=0.05) 1st vessel	5.7	3.4 (1.67x)
Time in ms for kernel SRG - 1st Slice (k=0.05) 2nd vessel	2.1	1.5 (1.4x)
Time in ms for kernel SRG - 2nd Slice (k=0.05) 1st vessel	1.5	1 (1.5x)
Time in ms for kernel SRG - 2nd Slice (k=0.05) 2nd vessel	3.5	2.4 (1.45x)



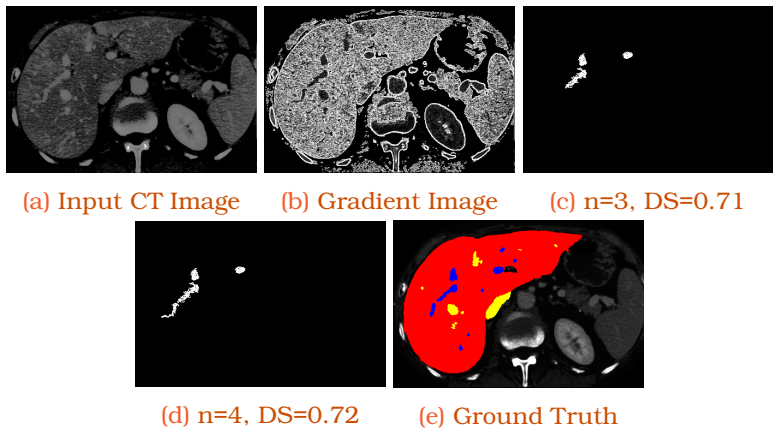
(a) Input CT Image (b) Gradient Image (c) n=2, DS=0.70



(d) n=4, DS=0.88 (e) Ground Truth

Figure 7.7
Fast Parallel Vessel Segmentation using Parallel SRG on 3rd Liver Slice using Multiple Seeds (n)

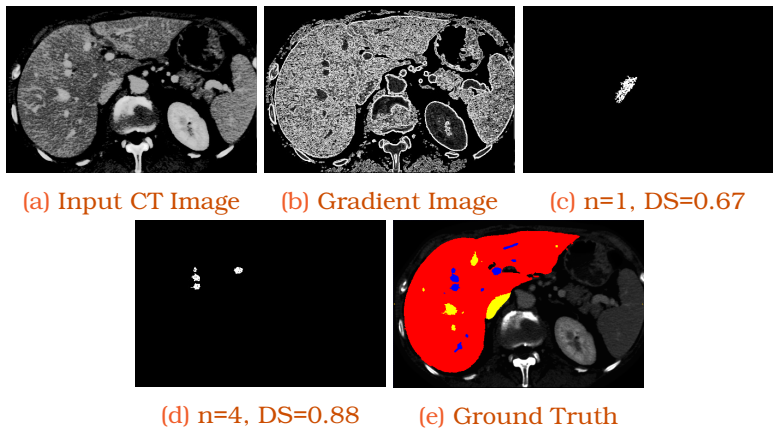
Further we analyze the effect of parallel SRG on different slices for multiple vessel segmentation using multiple seeds as shown in Figures 7.7, 7.8, 7.9, and



(a) Input CT Image (b) Gradient Image (c) n=3, DS=0.71

(d) n=4, DS=0.72 (e) Ground Truth

Figure 7.8
Fast Parallel Vessel Segmentation using Parallel SRG on 4th Liver Slice using Multiple Seeds (n)



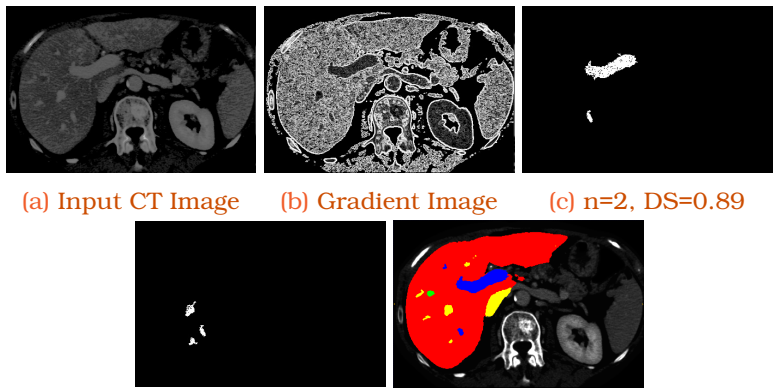
(a) Input CT Image (b) Gradient Image (c) n=1, DS=0.67

(d) n=4, DS=0.88 (e) Ground Truth

Figure 7.9
Fast Parallel Vessel Segmentation using Parallel SRG on 5th Liver Slice using Multiple Seeds (n)

7.10. Segmentation of the long vessel as shown in Figure 7.8d is slightly extended compared to the ground truth shown in Figure 7.8e. It can be seen from input CT image and gradient image (Figures 7.8a and 7.8b), long vessel has extension which is not shown in the ground truth. We show the thick vessel segmentation in Figures 7.7d, 7.9c, 7.8c and thin vessel segmentation in Figures 7.7c, 7.8c and 7.8d. The results of segmentation in terms of Dice Score and Precision are provided in Table 7.4. The highest and lowest value of precision for these slices are 0.94 and 0.79 respectively. This implies the number of positive detections in the segmented images are higher. It is possible to use multiple seeds for the same vessel in the proposed vessel segmentation approach. We have the flexibility to provide two seeds on the same vessel and then the proposed approach can create a curve or a line of initial seeds (if needed) as an input for SRG. It is useful in order to increase the quality of vessel segmentation.

There are many works done on image segmentation recently which are based on snake based model [17], gradient vector flow [22, 35], and level set based Chan-Vese model [19]. We validate the performance on 72 vessels from 24 ves-



(a) Input CT Image (b) Gradient Image (c) n=2, DS=0.89

(d) n=3, DS=0.67 (e) Ground Truth

Figure 7.10
Fast Parallel Vessel Segmentation using Parallel SRG on 6th Liver Slice using Multiple seeds (n)

Table 7.4
Quality of Vessel Segmentation in Terms of Dice Score and Precision for Figures 7.5 - 7.10

Sr. No.	Ground Truth Figure	Segmented Image Figure	Value of 'k'	Dice Score	Precision
1	7.5c	7.5d	0.04	0.77	0.86
2		7.5e	0.05	0.73	0.94
3		7.5f	0.06	0.67	0.94
4	7.5c	7.5g	0.04	0.54	0.86
5		7.5h	0.05	0.60	0.83
6		7.5i	0.06	0.37	0.86
7	7.6e	7.6c	0.05	0.88	0.86
8		7.6d	0.05	0.89	0.85
9	7.7e	7.7c	0.05	0.70	0.91
10		7.7d	0.05	0.88	0.91
11	7.8e	7.8c	0.05	0.71	0.79
12		7.8d	0.05	0.72	0.90
13	7.9e	7.9c	0.05	0.67	0.82
14		7.9d	0.05	0.88	0.88
15	7.10e	7.10c	0.05	0.89	0.94
16		7.10d	0.05	0.67	0.91

sel slices obtained from 4 different volumes in Table 7.5. Table shows the comparison of the vessel segmentation accuracy between three different models i.e. Snake model [17], Chan-Vese [19], and proposed SRG in terms of dice score and precision. Average dice score value and precision obtained by proposed SRG is outperforming the Chan-Vese and Snake based vessel segmentation.

Our proposed parallel implementations of SRG is not only fast but also accurate for vessel segmentation. The accuracy of the segmentation depends on the parameter 'k'. Clinicians get the flexibility to decide which segmentation is more accurate. The process takes very less time (few ms). Hence this reduces the overall time for segmentation for various values of parameter 'k' if the clinician wants to have more accuracy. The advantage of k is, by adjusting the value of k from the threshold criteria, clinicians have the flexibility to fine tune the accuracy of the vessel segmentation. But, vessels vary in shapes, sizes, texture features etc not only in different CT slices but also in the same CT slice. Even if the authors propose k=0.05 provides better accuracy for the provided CT images, sometimes finding the same value of k for different vessels in the

Table 7.5
Segmentation Accuracy Comparison for 4 Volumes, 24 Vessel Slices and 72 Vessels

Volume #	Image Size (wxh)	# of Vessel Slices	Total # of Vessels	Chan-Vese [19]		Snake model [17]		Proposed SRG	
				Average Dice	Average Precision	Average Dice	Average Precision	Average Dice	Average Precision
10504	460x306	7	21	0.78	0.82	0.77	0.81	0.85	0.83
18152	512x512	5	15	0.75	0.85	0.78	0.83	0.82	0.87
23186	405x346	6	18	0.77	0.83	0.81	0.82	0.84	0.86
28059	462x321	6	18	0.72	0.81	0.76	0.78	0.81	0.84

same CT slice becomes difficult. The range of k lies between 0.03 to 0.12 for the better accuracy of vessel segmentation.

7.6.3 *Discussion*

In this paper, we propose persistence and grid-stride loop based SRG implementation. In order to obtain significant speedup, we need to exploit parallelism by using persistence and IBS. It involves change in the large body of SRG algorithm. We want algorithms that require as less synchronization as possible. In general if algorithm requires IBS, it is probably not going to be particularly fast. The fastest algorithms on GPUs are ones that fit nicely into the GPU programming model, where blocks are independent from each other and do not require synchronization [32].

But the problem arises when iterative calling of the kernel can not be avoided. It incurs memory transfers from CPU to GPU when KTRL is used for global synchronizations. Hence it has to go through synchronizations as the next step of SRG which is dependent on the current step. Terminating a kernel and relaunching incurs data transfers from CPU to GPU and vice versa. It is time consuming.

If we use IBS method along with persistence, then we can map whole algorithm on GPU with synchronization. Control comes back to CPU only if the kernel task is over. CPU launches a kernel on GPU, GPU executes it and final results are copied to CPU. No intermediate data communication occurs in the proposed approach (unlikely in KTRL).

7.7 Conclusion

In this paper, we discuss SRG based vessel segmentation and its parallel implementation on GPU. We propose persistence and grid-stride loop based GPU approach for SRG providing significant speedup. Normally recursion/iterative calling of a kernel is generally a bad idea on GPUs. We use persistence and grid-stride approach as an alternate implementation for KTRL. We compare proposed GPU optimization strategy for SRG implementation. The proposed persistent and gradient based parallel SRG for 2D vessel segmentation is accurate with high dice scores and 1.9x faster compared to the KTRL.

Acknowledgements

The work is supported by the project High Performance soft tissue Navigation (HiPerNav). This project has received funding from the European Union Horizon 2020 research and innovation program under grant agreement No. 722068. We thank The Intervention Centre, Oslo University Hospital, Oslo, Norway for providing the CT images with ground truths for the clinical validation of vessel segmentation.

REFERENCES

[1] Guoyang Chen and Xipeng Shen: *Free launch: optimizing gpu dynamic kernel launches through thread reuse*. In *Proceedings of the 48th International Symposium on Microarchitecture*, pages 407–419. ACM, 2015.

[2] Konstantinos K. Delibasis, Aristides Kechriniotis, and I. Maglogiannis: *A novel tool for segmenting 3d medical images based on generalized cylinders and active surfaces*. Computer Methods and Programs in Biomedicine, 111(1):148 – 165, 2013, ISSN 0169-2607. http://www.sciencedirect.com/science/article/pii/S0169260713000989.

[3] Åsmund Avdem Fretland, Vegar Johansen Dagenborg, Gudrun Maria Waaler Bjørnelv, Airazat M Kazaryan, Ronny Kristiansen, Morten Wang Fagerland, John Hausken, Tor Inge Tønnessen, Andreas Abildgaard, Leonid Barkhatov, *et al.*: *Laparoscopic versus open resection for colorectal liver metastases*. Annals of surgery, 267(2):199–207, 2018.

[4] MECVG Greiner: *Stack implementation on programmable graphics hardware*. Vision, Modeling, and Visualization 2004: Proceedings, page 255, 2004.

[5] Kshitij Gupta, Jeff A Stuart, and John D Owens: *A study of persistent threads style gpu programming for gpgpu workloads*. In *Innovative Parallel Computing-Foundations & Applications of GPU, Manycore, and Heterogeneous Systems (INPAR 2012)*, pages 1–14. IEEE, 2012.

[6] Pawan Harish and P. J. Narayanan: *Accelerating large graph algorithms on the gpu using cuda*. In Srinivas Aluru, Manish Parashar, Ramamurthy Badrinath, and Viktor K. Prasanna (editors): *High Performance Computing – HiPC 2007*, pages 197–208, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg, ISBN 978-3-540-77220-0.

[7] Mark Harris: *Cuda pro tip:write flexible kernels with grid-stride loops*, 2015. http://goo.gl/b8Vmkh.

[8] Blake A. Hechtman, Andrew D. Hilton, and Daniel J. Sorin: *TREES: A CPU/GPU task-parallel runtime with explicit epoch synchronization*. CoRR, abs/1608.00571:„ 2016. http://arxiv.org/abs/1608.00571.

[9] Huiyan Jiang, Baochun He, Di Fang, Zhiyuan Ma, Benqiang Yang, and Libo Zhang: *A region growing vessel segmentation algorithm based on spectrum information*. Computational and Mathematical Methods in Medicine, 2013, 2013.

[10] Yukihiro Komura and Yutaka Okabe: *Gpu-based single-cluster algorithm for the simulation of the ising model*. Journal of Computational Physics, 231(4):1209–1215, 2012.

[11] Rahul Prasanna Kumar, Fritz Albregtsen, Martin Reimers, Bjørn Edwin, Thomas Langø, and Ole Jakob Elle: *Three-dimensional blood vessel segmen-*

tation and centerline extraction based on two-dimensional cross-section analysis. Annals of biomedical engineering, 43(5):1223–1234, 2015.

[12] Rabia Naseem, Faouzi Alaya Cheikh, Azeddine Beghdadi, Ole Jakob Elle, and Frank Lindseth: *Cross modality guided liver image enhancement of ct using mri*. In *2019 8th European Workshop on Visual Information Processing (EUVIP)*, pages 46–51. IEEE, 2019.

[13] Rafael Palomar, Faouzi A Cheikh, Bjørn Edwin, Åsmund Fretland, Azeddine Beghdadi, and Ole J Elle: *A novel method for planning liver resections using deformable bézier surfaces and distance maps*. Computer Methods and Programs in Biomedicine, 144:135–145, 2017.

[14] Seongjin Park, Jeongjin Lee, Hyunna Lee, Juneseuk Shin, Jinwook Seo, Kyoung Ho Lee, Yeong Gil Shin, and Bohyoung Kim: *Parallelized seeded region growing using cuda*. Computational and Mathematical Methods in Medicine, 2014, 2014.

[15] Tiago Carneiro Pessoa, Jan Gmys, Nouredine Melab, Francisco Heron de Carvalho Junior, and Daniel Tuyttens: *A gpu-based backtracking algorithm for permutation combinatorial problems*. In Jesus Carretero, Javier Garcia-Blas, Ryan K.L. Ko, Peter Mueller, and Koji Nakano (editors): *Algorithms and Architectures for Parallel Processing*, pages 310–324, Cham, 2016. Springer International Publishing, ISBN 978-3-319-49583-5.

[16] GN Rai and TR Nair: *Gradient based seeded region grow method for ct angiographic image segmentation*. arXiv preprint arXiv:1001.3735, (2010), 2010.

[17] Sawrav Roy, Susanta Mukhopadhyay, and Manoj K Mishra: *Enhancement of morphological snake based segmentation by imparting image attachment through scale-space continuity*. Pattern Recognition, 48(7):2254–2268, 2015.

[18] Nitin Satpute, Rabia Naseem, Egidijus Pelanis, Juan Gomez-Luna, Faouzi Alaya Cheikh, Ole J Elle, and Joaquin Olivares: *Gpu acceleration of liver enhancement for tumor segmentation*. Computer Methods and Programs in Biomedicine, 184:105285, 2020, ISSN 0169-2607. http://www.sciencedirect.com/science/article/pii/S016926071931733X.

[19] Sangeeta K Siri and Mrityunjaya V Latte: *Combined endeavor of neutrosophic set and chan-vese model to extract accurate liver image from ct scan*. Computer Methods and Programs in Biomedicine, 151:101–109, 2017.

[20] Erik Smistad: *Seeded region growing*. https://github.com/smistad/FAST/tree/master/source/FAST/Algorithms/, 2015.

[21] Erik Smistad, Anne C Elster, and Frank Lindseth: *Gpu accelerated segmentation and centerline extraction of tubular structures from medical images*. International Journal of Computer Assisted Radiology and Surgery, 9(4):561–575, 2014.

[22] Erik Smistad, Anne C Elster, and Frank Lindseth: *Real-time gradient vector flow on gpus using opencl*. Journal of Real-Time Image Processing, 10(1):67–74, 2015.

[23] Erik Smistad, Thomas L Falch, Mohammadmehdi Bozorgi, Anne C Elster, and Frank Lindseth: *Medical image segmentation on gpus–a comprehensive review*. Medical Image Analysis, 20(1):1–18, 2015.

[24] Tyler Sorensen, Hugues Evrard, and Alastair F Donaldson: *Cooperative kernels: Gpu multitasking for blocking algorithms*. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*, pages 431–441. ACM, 2017.

[25] Mohammed Sourouri, Scott B Baden, and Xing Cai: *Panda: A compiler framework for concurrent cpu+gpu execution of 3d stencil computations on gpu-accelerated supercomputers*. International Journal of Parallel Programming, 45(3):711–729, 2017.

[26] John E. Stone, David Gohara, and Guochun Shi: *Opencl: A parallel programming standard for heterogeneous computing systems*. IEEE Des. Test, 12(3):66–73, May 2010, ISSN 0740-7475. http://dx.doi.org/10.1109/MCSE.2010.69.

[27] X. Tang, A. Pattnaik, H. Jiang, O. Kayiran, A. Jog, S. Pai, M. Ibrahim, M. T. Kandemir, and C. R. Das: *Controlled kernel launch for dynamic parallelism in gpus*. In *2017 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pages 649–660, Feb 2017.

[28] D. N. H. Thanh, D. Sergey, V. B. Surya Prasath, and N. H. Hai: *Blood Vessels Segmentation Method for Retinal Fundus Images based on Adaptive Principal Curvature and Image Derivative Operators*. ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLII-2/W12:211–218, may 2019. https://doi.org/10.5194%2Fisprs-archives-xlii-2-w12-211-2019.

[29] V. Vineet and P. J. Narayanan: *Cuda cuts: Fast graph cuts on the gpu*. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8, June 2008.

[30] Jan Wassenberg, Wolfgang Middelmann, and Peter Sanders: *An efficient parallel algorithm for graph-based image segmentation*. In *International Conference on Computer Analysis of Images and Patterns*, pages 1003–1010. Springer, 2009.

[31] S. Xiao and W. C. Feng: *Inter-block gpu communication via fast barrier synchronization*. In *2010 IEEE International Symposium on Parallel Distributed Processing (IPDPS)*, pages 1–12, April 2010.

[32] Orestis Zachariadis, Andrea Teatini, Nitin Satpute, Juan Gómez-Luna, Onur Mutlu, Ole Jakob Elle, and Joaquín Olivares: *Accelerating b-spline interpolation on gpus: Application to medical image registration*. Computer Methods and Programs in Biomedicine, 193:105431, 2020, ISSN 0169-2607.

[33] Xiaoli Zhang, Xiongfei Li, and Yuncong Feng: *A medical image segmentation algorithm based on bi-directional region growing*. Optik-International Journal for Light and Electron Optics, 126(20):2398–2404, 2015.

[34] Y. Zhao, H. Li, S. Wan, A. Sekuboyina, X. Hu, G. Tetteh, M. Piraud, and B. Menze: *Knowledge-aided convolutional neural network for small organ seg-*

mentation. IEEE Journal of Biomedical and Health Informatics, 23(4):1363–1373, July 2019.

[35] Huiyu Zhou, Xuelong Li, Gerald Schaefer, M Emre Celebi, and Paul Miller: *Mean shift based gradient vector flow for image segmentation*. Computer Vision and Image Understanding, 117(9):1004–1016, 2013.

8

ARTICLE 3: ACCELERATING CHAN-VESE MODEL WITH CROSS-MODALITY GUIDED CONTRAST ENHANCEMENT FOR LIVER SEGMENTATION

INDEX

Accelerating Chan-Vese Model with Cross-Modality Guided Contrast Enhancement for Liver Segmentation

Nitin Satpute, J. Gomez-Luna, J. Olivares, Accelerating Chan-Vese Model with Cross-Modality Guided Contrast Enhancement for Liver Segmentation, Computers in Biology and Medicine, 124 (2020) 103930. DOI: https://doi.org/10.1016/j.compbiomed.2020.103930.

8.1 Abstract

Accurate and fast liver segmentation remains a challenging and important task for clinicians. Segmentation algorithms are slow and inaccurate due to noise and low quality images in computed tomography (CT) abdominal scans. Chan-Vese is an active contour based powerful and flexible method for image segmentation due to superior noise robustness. However, it is quite slow due to time-consuming partial differential equations, especially for large medical datasets. This can pose a problem for a real-time implementation of liver segmentation and hence, an efficient parallel implementation is highly desirable. Another important aspect is the contrast of CT liver images. Liver slices are sometimes very low in contrast which reduces the overall quality of liver segmentation. Hence, we implement cross-modality guided liver contrast enhancement as a pre-processing step to liver segmentation. GPU implementation of Chan-Vese improves average speedup by 99.811 ($\pm$ 7.65) times and 14.647 ($\pm$ 1.155) times with and without enhancement respectively in comparison with the CPU. Average dice, sensitivity and accuracy of liver segmentation are 0.656, 0.816 and 0.822 respectively on the original liver images and 0.877, 0.964 and 0.956 respectively on the enhanced liver images improving the overall quality of liver segmentation.

8.2 Introduction

Image segmentation is a popular research topic in medical imaging; as it has a number of applications, such as tissue detection [11], segmentation [6, 23, 25, 29], reconstruction [22], registration [22, 41], etc. There are many methods proposed for image segmentation, such as region growing [34], thresholding [31], gradient approach [25], contour methods [13, 20], etc. They can be classified into edge or region based image segmentation methods. These methods can be further categorized based on histogram, spatial information of the image, convergence of active contours, etc [29, 38]. The active contour models are essential when the edge of the region of interest in the image is indistinct and diffused [29, 36]. The computed tomography (CT) scans sometimes provide poor quality images with indistinguishable liver boundaries which complicates the task of liver segmentation for clinicians for the treatment of patients.

The Chan-Vese algorithm is developed on active contour models using a level set approach [7, 29]. This operates on the initial contours, average intensity values inside and outside the curve and optimizes the energy based on the level set approach [1, 5]. The algorithm works on the principle of the energy minimization problem which relies on calculus and partial differential equations [14, 30]. It is one of the most influential and effective methods in order to optimize the Mumford-Shah function which includes energy terms defined in image and contour space [12, 38, 42]. Chan-Vese is flexible and robust at segmenting the CT liver image, which is difficult to segment using classical segmentation techniques [13, 32].

The study proposes a high performance Chan-Vese approach for liver segmentation by avoiding intermediate memory transfers between the CPU and GPU. However, the Chan-Vese approach alone is not sufficient for accurate liver segmentation as it can result in many false positives, lowering the sensitivity and

accuracy [39, 40] and degrading the quality of liver segmentation. Hence, we employ an enhancement module before the Chan-Vese approach for segmentation. This module provides the cross-modality guided liver contrast enhancement. This works on the target and guided image. We consider the CT liver image as the target image and the image from magnetic resonance (MR) imaging as the guided image. The cross-modality approach generates the histogram of target CT scan similar to guided MR image [2, 15]. The proposed parallel approach results in fast and accurate liver segmentation.

Our goal is to develop a fast parallel Chan-Vese approach for liver segmentation with and without liver contrast enhancement. The GPU implementation is faster compared to the CPU and the liver contrast enhancement improves the quality of liver segmentation by reducing the false positives and increasing the sensitivity, dice score and accuracy of the segmentation. The average dice score, sensitivity and accuracy of the liver segmentation are 0.877 ± 0.036, 0.964 ± 0.037 and 0.956 ± 0.022 respectively after liver contrast enhancement improving the quality of segmentation. GPU implementation of the Chan-Vese segmentation algorithm improves the average speedup by 99.811 ± 7.65 times and 14.647 ± 1.155 times with and without enhancement in comparison to the CPU.

The rest of the paper is structured as follows. Section 8.3 briefs the background and motivation with respect to the Chan-Vese approach for image segmentation. Section 8.4 explains the flow of the Chan-Vese approach and its parallel implementation on the GPU both with and without liver contrast enhancement. Performance evaluation based on the quality of liver segmentation and the speedup is analyzed in Section 8.5. Section 8.6 summarizes the results and main conclusions of the paper.

8.3 Background and Motivation

Image segmentation plays a vital role in medical image analysis. There are many methods developed for image segmentation [34]. Other researchers have investigated active contour models for image segmentation [7, 14, 29, 30]. We explain the background and motivation behind active contours and the benefits of Chan-Vese approach for image segmentation.

Scientists have explored the snake model for segmentation. Snakes are defined as a set of points around a contour [4, 18, 24]. The contour can be initialized inside the object forcing the snake to expand outside. This is the Balloon Force algorithm [17, 19]. The energy of the snake based model which provides a high quality segmentation can be defined as follows. Total energy of curve C

$$E(C) = E_{internal}(C) + E_{external}(C) \tag{8.1}$$

Equation 8.1 expresses the total energy where the curve repeatedly evolves to minimize energy E. $E_{internal}(C)$ and $E_{external}(C)$ depend on the shape of the snake curve and image intensities respectively.

$$E_{internal}(C) = \int_0^1 w_1 \|c'(s)\|^2 + w_2 \|c''(s)\|^2 ds \tag{8.2}$$

Equation 8.2 expresses the internal energy. Low c' means the curve is not too stretchy and it keeps the points on the curve together. Low c'' implies the

curve is not too bendy i.e. it is smooth and keeps the points on the curve from oscillating.

$$F(s) = -[(\frac{\partial I(X(s),Y(s))}{\partial X})^2 + (\frac{\partial I(X(s),Y(s))}{\partial Y})^2] \tag{8.3}$$

$$E_{external}(C) = \int_0^1 -\|\nabla I(c(s))\|^2 ds = \int_0^1 F(s) ds \tag{8.4}$$

If there is no edge then $\nabla I(c(s)) = 0$ and $F(s) = 0$ (from Equations 8.3 and 8.4). If there is a thick edge then $\|\nabla I(c(s))\|$ is large and $F(s)$ is more negative. It implies that the $E_{external}(C)$ is lowered. The aim is to minimize E(C) from Equation 8.1. However, the contour never sees the distant strong edges and the snake gets hung up due to many small noises in the image [17–19, 24]. Hence researchers devised a solution called a gradient vector flow (GVF). Instead of using an image gradient, they created a new vector field over the image plane [33, 44]. The mathematical representation of GVF [3, 43] is given by the following equations.

$$GVF1 = [(\frac{\partial V_x}{\partial x})^2 + (\frac{\partial V_x}{\partial y})^2 + (\frac{\partial V_y}{\partial x})^2 + (\frac{\partial V_y}{\partial y})^2] \tag{8.5}$$

$$GVF2 = \|\nabla e\|^2 \|V - \nabla e\|^2 \tag{8.6}$$

$$cost\_GVF = \int\int \mu(GVF1) + GVF2 \quad dxdy \tag{8.7}$$

GVF has two components defining smoothness (GVF1 from Equation 8.5) and edge map (GVF2 from Equation 8.6) as shown in Equation 8.7. If ∇e is high then the gradient is also high and V follows the edge gradient faithfully. If ∇e is low then the gradient is also low and V becomes as smooth as possible. μ is a tuning parameter to define the scaling of smoothness in comparison to the edge map. GVF2 from Equation 8.6 defines the characteristic of the image where ∇e is a magnitude of the edge map and $(V - \nabla e)$ shows similarity between V and ∇e. If the region has a thick edge (high ∇e) then $(V - \nabla e)$ should be low which implies V is pushed towards ∇e.

Nevertheless, there are problems with both the snake and GVF models [4, 18, 44]. They require the number of points and point distribution to be monitored. Snakes as defined can never wrap around multiple objects at once. They cannot determine the inner boundary of the region of interest. Hence researchers devised another solution called level sets [14, 30, 37]. The shape-intensity prior level set proposed by Wang et. al. [37] contains the atlases which are weighted in the selected training datasets by calculating the similarities between the atlases and the test image to dynamically generate a subject-specific probabilistic atlas for the test image.

The idea of level sets is derived from fluid dynamics. Instead of parameterizing the curve using a set of ordered points, discretize the image plane (x,y) and define a function f(x,y). $f(x,y) > 0$ implies the pixels are inside the curve and $f(x,y) < 0$ describes the pixels are outside the curve [1, 38].

In the absence of strong edges, we can use a region based formulation which is a Chan-Vese approach for segmentation [29, 38, 42].

$$SD_{inside} = \int_{inside} (I(x,y) - \mu_{inside})^2 dxdy \tag{8.8}$$

SD_{inside} from Equation 8.8 denotes the standard deviation of pixels inside the curve.

$$SD_{outside} = \int_{outside} (I(x,y) - \mu_{outside})^2 dxdy \tag{8.9}$$

$SD_{outside}$ from Equation 8.9 denotes the standard deviation of pixels outside the curve.

$$SD_{total} = \lambda_1 * SD_{inside} + \lambda_2 * SD_{outside} + \lambda_3 * LC + \lambda_4 * AUC \tag{8.10}$$

SD_{total} from Equation 8.10 represents the Chan-Vese algorithm where LC is the length of the curve, AUC is the area under the curve and $\lambda_1 > 0, \lambda_2 > 0, \lambda_3 \geqslant 0, \lambda_4 \geqslant 0$ are fixed parameters [1, 42]. The algorithm maximizes the difference in standard deviations of pixel distributions between inside and outside the curve.

The default value of λ is 0.1. It describes the relative weighting of curve smoothness. However, after experimentation, the authors found the following values (in Equation 8.10) suitable for convergence and accurate liver segmentation. The weight parameter of the term inside the level set is $\lambda_1 = 0.2$. The weight parameter of the term outside the level set is $\lambda_2 = 0.2$. The weight parameter of the length term is $\lambda_3 = 0.04 * width(image) * height(image)$ and $\lambda_4 = 0.0002 * width(image) * height(image)$ is the weight parameter of the area term.

In the next section, we discuss the CPU and GPU implementation of the Chan-Vese approach for liver segmentation.

8.4 Methodology

In this section, we discuss the proposed methodology based on the Chan-Vese approach and the impact of cross-modality guided contrast enhancement on liver segmentation. The sequential and parallel implementations of the Chan-Vese approach with and without liver contrast enhancement are explained in the following sections.

8.4.1 *CPU Implementation of Chan-Vese*

In this section, we discuss the flowchart for CPU implementation of the Chan-Vese approach for liver segmentation. The flowchart in Figure 8.1 represents the Chan-Vese approach for liver segmentation.

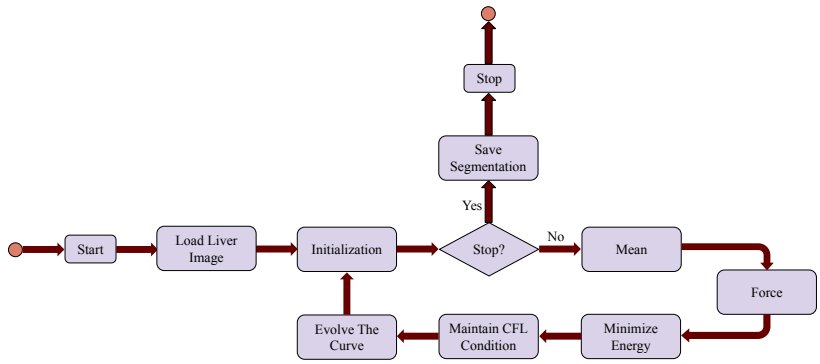


Figure 8.1
CPU implementation of Chan-Vese

- Initialization: The aim of this step is to create a mask or the initial contour in order to generate a signed distance function (SDF) [7, 42]. This consists of two regions i.e. the liver as the foreground and the non-liver region as the background. The mask should be similar to the liver area in order to increase the sensitivity of segmentation and reduce the computation time.

- Stopping Criteria: This step detects if the process of liver segmentation is complete or not. If the process is complete then the segmented image is stored and the process of liver segmentation stops, otherwise the model calculates the mean of the interior and exterior regions with respect to the initial mask.

- Mean: This step includes the computation of the SDF as the first step. This is computed from the initial mask on the liver using Euclidean distance. In this work, we choose ϕ as an image with real values in order to choose distances from the curve so that the distance function (SDF) is positive inside the curve and negative outside. However, the computation of the SDF is time-consuming. Hence we apply a narrow band approach to reduce the computation time by restricting the computation to a band of grid points near the level set (or mask). The SDF helps to find the average value of pixels inside and outside the curve [7, 12].

- Force: The calculation of the average value of the pixels inside and outside the curve is essential to compute the force from the Chan-Vese energy Equation 8.11.

$$E = SD_{in} + SD_{out} = \int_{in} (I(x,y) - \mu_{in})^2 dxdy + \int_{out} (I(x,y) - \mu_{out})^2 dxdy \tag{8.11}$$

Force is computed from the image using the average value of the pixels inside and outside the curve as shown in Equation 8.12.

$$F = \nabla E = (I(x,y) - \mu_{in})^2 + (I(x,y) - \mu_{out})^2 \tag{8.12}$$

Then the curvature is calculated using the kappa equation [1, 38] and the central difference approximation scheme is applied to approximate the derivatives of SDF with respect to x and y.

- Minimize Energy: The gradient descent algorithm helps to minimize the energy given by Equation 8.11. The curve is updated by the calculation of the SDF after a small time interval and is approximated by the first-order Taylor expansion.

- Maintain CFL Condition: The Courant, Friedrichs, Lewy (CFL) [1, 7, 12] condition is necessary for convergence while solving the partial differential equations in order to maintain the accuracy of the curve. The equation is given as

$$C = u\Delta t/\Delta x \leqslant Cmax \tag{8.13}$$

  Where C is the courant number, u is the dependent variable which is a magnitude of the velocity, Δt is the time interval and Δx is the space interval. The value of Cmax is typically 1 for the explicit methods. Equation 8.13 is a one dimensional case of the CFL condition. The courant number can be enlarged by increasing the time interval or decreasing the space interval. The courant number controls the stability and it is necessary to choose the space and time intervals precisely.

- Evolve The Curve: We calculate the Sussman function [1] to maintain the smoothness of the curve. Re-initialization of the curve takes place and the process of Chan-Vese based segmentation continues until the liver is segmented completely and the curve can not be evolved further.

8.4.2 *GPU Implementation of Chan-Vese*

Our objective is a fast parallel implementation of the Chan-Vese approach for liver segmentation. In this section, we discuss the GPU implementation of the Chan-Vese approach. Chan-Vese is an iterative algorithm. The flow of GPU implementation of Chan-Vese approach for liver segmentation is shown in Figure 8.2.

We load the liver image and send it to the GPU memory. The CPU calls the Chan-Vese kernel on the GPU. Each thread on the GPU in parallel performs the initialization of the curve. The stopping criteria is checked on the device memory to ensure the process is finished or not. If the process is finished then the control returns to the CPU storing the segmented image and the process stops.

Each thread in parallel is responsible for the calculation of the average value of the pixels inside and outside the curve. Inter block GPU synchronization (IBS) [9, 35] between stages is essential and ensures valid data is communicated between the blocks. Then the Chan-Vese kernel on the GPU calculates the force based on the image pixels and the curvature and the gradient descent algorithm to minimize the energy given by Equation 8.11. All the threads maintain the CFL condition for convergence and calculate the Sussman function to maintain the smoothness of the curve. The parallel threads reinitialize the curve and the process of liver segmentation using Chan-Vese continues until the curve cannot be evolved further. The process of segmentation stops and control returns to the CPU if the stopping criteria is satisfied.

These blocks communicate via IBS and the intermediate kernel calls are avoided using the proposed approach which helps to increase the performance. The

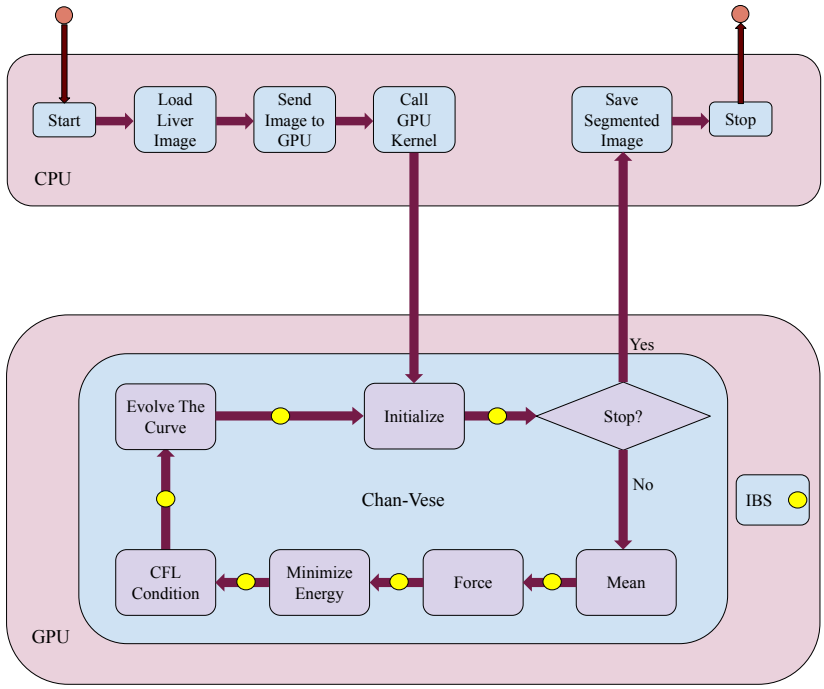


Figure 8.2
GPU implementation of Chan-Vese

kernel invokes enough blocks of threads to compute liver segmentation. The thread blocks on the GPU are the computational units launched in parallel to perform independent operations. The maximum number of active blocks are called persistent blocks [9, 25]. However, the application may require more blocks compared to the persistent blocks. We apply a grid-stride loop so that the persistent blocks are iterated to perform the task of the remaining blocks [10, 35].

Chan-Vese is a powerful approach for segmentation due to improved noise robustness although the quality of liver segmentation is questionable due to the low contrast of the liver images. Hence it is necessary to assess the impact of contrast enhancement on liver segmentation. We employ cross-modality guided liver contrast enhancement as a pre-processing step for liver segmentation. The parallel implementation of liver segmentation with contrast enhancement is discussed in the next section.

8.4.3 *GPU Implementation of Chan-Vese with Enhancement*

The Chan-Vese approach for liver segmentation results in false positives. In order to reduce the number of false positives and increase the sensitivity of liver segmentation, we enhance the CT liver image using cross-modality guided contrast enhancement. The flow of liver contrast enhancement and segmentation using Chan-Vese is shown in Figure 8.3.

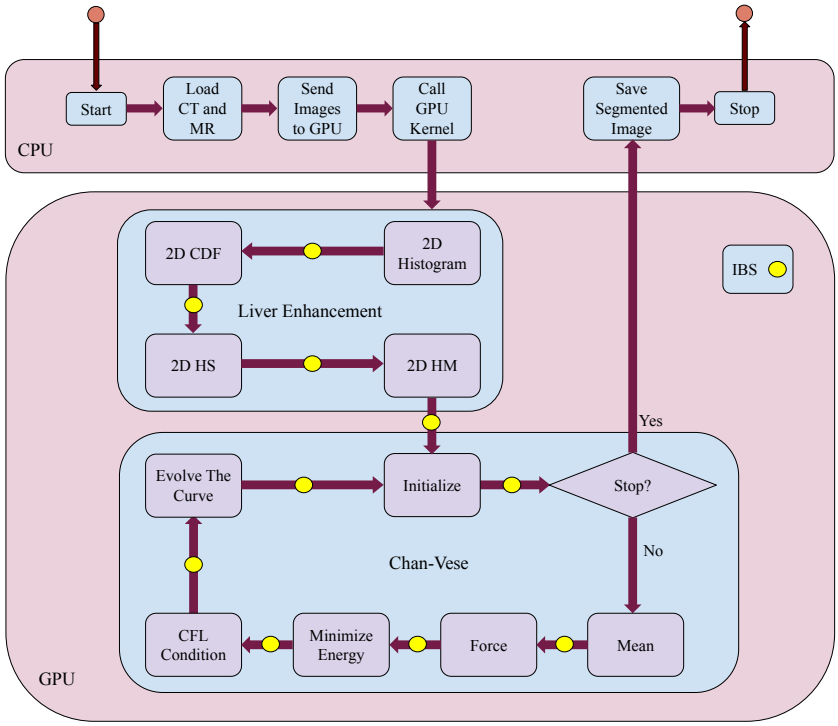


Figure 8.3
GPU implementation of Chan-Vese with Enhancement

We load CT and MR images and send them to the GPU. The CPU invokes a single kernel on GPU for liver enhancement and segmentation. Liver enhancement improves the contrast of the CT liver image considering the MR image as the guidance image. The parallel computing units on the GPU match the histogram of the CT image with the guided MR image before performing segmentation. The liver contrast enhancement consists of the following modules:

- 2D Histogram: The contrast enhancement module calculates the 2D histogram of both CT and MR images. A 2D histogram is a plot of a pixels and neighbouring pixels to discover the underlying 2D frequency distribution of the image. This involves calculating the frequency that how often the neighbouring pair of values in an image occurs instead of just considering the individual pixel values [15, 21, 26].

- 2D Cumulative Distributive Function (CDF): The 2D histograms help to find the 2D CDFs of the CT and MR images for contrast enhancement. 2D CDF calculates the probability of a possible pixel pair in the CT and MR images [2, 21].

- 2D Histogram Specification (HS): This is also called as histogram equalization. HS extends the most frequent intensity values improving the global contrast of the image [15, 21].

- 2D Histogram Matching (HM): The process of histogram equalization over the CT image provides the enhanced image by mapping the modi-

fied intensity values obtained from the 2D histogram equalization to the corresponding pixels [2, 15, 26].

The enhanced CT image obtained from 2D cross-modality is sent to the Chan-Vese approach for segmentation. The GPU performs the segmentation and the control returns to the CPU saving the segmented liver image. This parallel Chan-Vese implementation is described in the previous Section 8.4.2.

8.5 Performance Evaluation

In this section, we analyze and compare the performance of the Chan-Vese approach on CPU and GPU, the impact of enhancement on liver segmentation and the quality of liver segmentation using dice score, sensitivity and accuracy. We use Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz RAM 24 GB, NVIDIA GPU GeForce GTX 1050 (RAM 4GB) and CUDA Toolkit 10.1 for the implementation and we evaluate the performance of liver segmentation in the following section.

8.5.1 *Dataset*

Liver data for the research work has been acquired from The Intervention Center, University of Oslo, Norway [8, 25]. The ground truths for liver segmentation are provided by the clinician. Inter and intra observer errors exist while creating the ground truths for the input CT images. Intra observer error is when the same clinician creates the ground truth for the input CT image in different time stamps. Inter observer error is created when different clinicians create the ground truth for the same input CT image. Errors depend upon the registration of the input CT and MR images which are used for cross-modality guided contrast enhancement. Errors may also be introduced if clinicians use different registration techniques for the CT and MR slices. In this work, a 3D slicer is used for the registration.

The CT and MR volumes are loaded into the 3D Slicer and; then the region of interests (RoIs) are extracted from both volumes using the Surface Cut and Mask Volume options available in the Segment Editor tool. The RoI can also be extracted using the Threshold option in the Segment Editor. The RoIs can be registered using General Registration by selecting the appropriate Degree of Freedom) and Initialization Transform Mode. Note that the registration results depend on the organs whose CT and MR volumes are being registered. Liver CT and MR images are quite challenging to register.

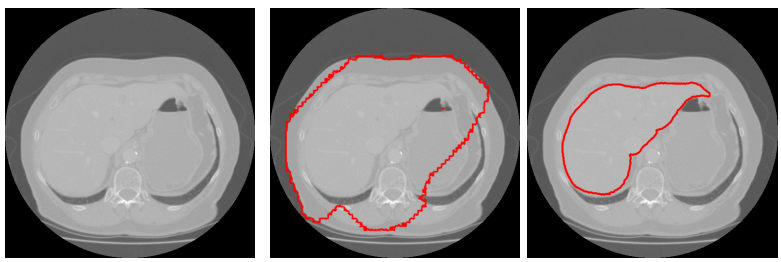
Table 8.1 shows information about images of different sizes including the total number of liver slices used for the performance analysis from a particular volume. We validate the performance on 24 liver slices obtained from 4 different registered volumes. For ground truths, images are pre-processed through locally developed applications with a 3D Slicer. In some cases, the same application is used for liver segmentation and separation of portal and hepatic vessels although another possibility is to employ the active contour tool using ITK-SNAP and manual correction [8].

Table 8.1
Liver Dataset

Volume #	Total # of Slices	Image Size (wxh)	# of Slices with Liver
28059	59	462x321	6
23186	87	405x346	6
18152	139	512x512	5
10504	59	460x306	7

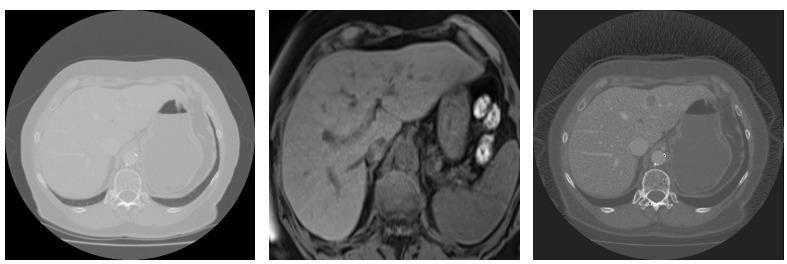
8.5.2 *Quality of Liver Segmentation*

We discuss the Chan-Vese approach for liver segmentation and the impact of cross-modality guided contrast enhancement on segmentation. The segmented results using Chan-Vese on the original and enhanced images are shown in Figures 8.4, 8.5, and 8.6.

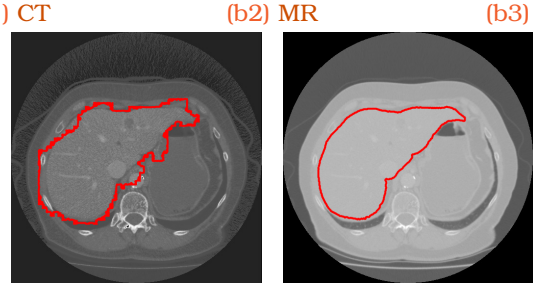


(a1) CT (a2) Seg (a3) GT

(a) Original Liver Image 1



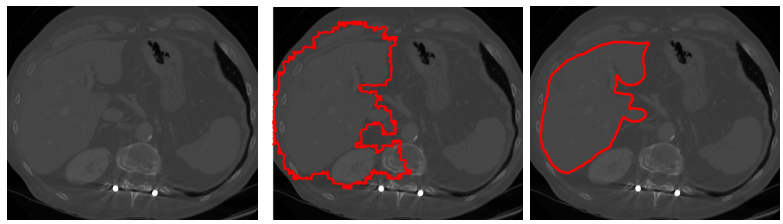
(b1) CT (b2) MR (b3) Enh



(b4) Seg (b5) GT

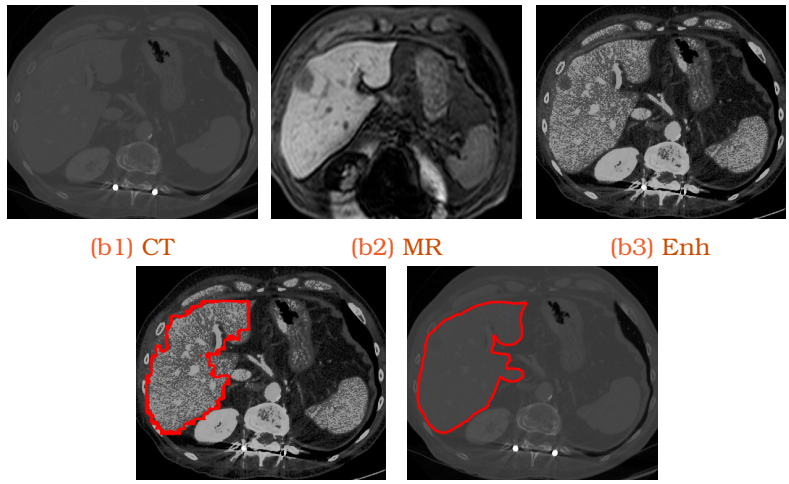
(b) Liver Enhancement and Segmentation from Enhanced CT Image 1

Figure 8.4
Liver Segmentation from Original and Enhanced CT Image 1



(a1) CT (a2) Seg (a3) GT

(a) Original Liver Image 2



(b1) CT (b2) MR (b3) Enh

(b4) Seg (b5) GT

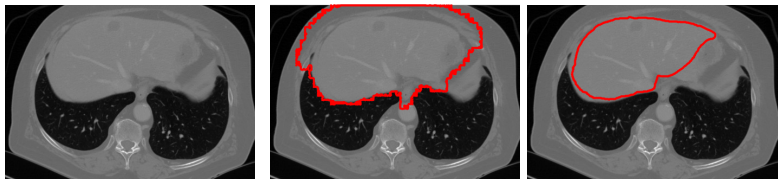
(b) Liver Enhancement and Segmentation from Enhanced CT Image 2

Figure 8.5
Liver Segmentation Original and Enhanced CT Image 2

Table 8.2
Liver Segmentation Quality Analysis

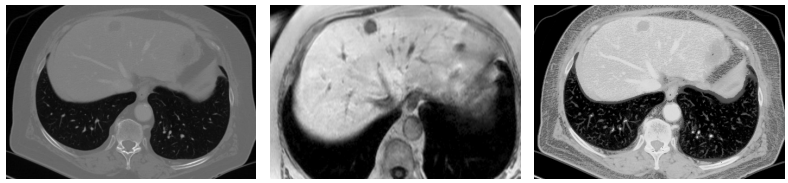
Liver	Chan-Vese without Enhancement			Chan-Vese with Enhancement		
Slice #	Dice	Sensitivity	Accuracy	Dice	Sensitivity	Accuracy
1	0.504	0.831	0.719	0.904	0.979	0.969
2	0.533	0.669	0.748	0.895	0.961	0.966
3	0.762	0.882	0.896	0.894	0.988	0.967
4	0.759	0.868	0.890	0.879	0.991	0.961
5	0.721	0.829	0.858	0.815	0.901	0.917
Average	0.656	0.816	0.822	0.877	0.964	0.956
Std. Dev.	0.126	0.085	0.082	0.036	0.037	0.022

Figures 8.4a, 8.5a, and 8.6a show the liver segmentation on original CT liver slices and Figures 8.4b, 8.5b, and 8.6b show the liver segmentation with enhancement. We analyze the Chan-Vese based segmentation of the original and enhanced images. Figures 8.4a2 and 8.4b4 show the Chan-Vese based liver segmentation of the original image (Figure 8.4a1) and the enhanced image (Figure 8.4b3) respectively and the ground truth is shown in Figure 8.4a3 (or 8.4b5). Figures 8.5a2 and 8.5b4 show liver segmentation from the original (Figure 8.5a1) and enhanced image (Figure 8.5b3) respectively with ground truth shown in Figure 8.6a3 or 8.5b5. Similarly, Figures 8.6a2 and 8.6b4 show liver segmenta-

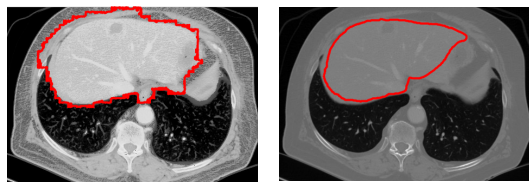


(a1) CT (a2) Seg (a3) GT

(a) Original Liver Image 3



(b1) CT (b2) MR (b3) Enh



(b4) Seg (b5) GT

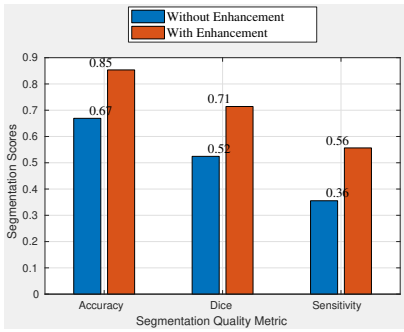
(b) Liver Enhancement and Segmentation from Enhanced CT Image 3

Figure 8.6
Liver Segmentation from Original and Enhanced CT Image 3

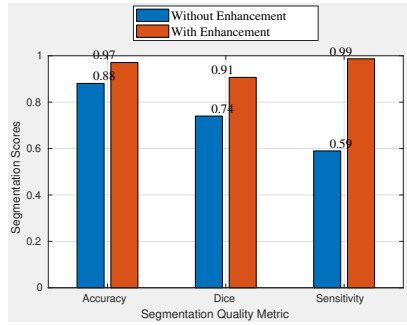
tion from the original image (Figure 8.6a1) and enhanced image (Figure 8.6b3) respectively with ground truth shown in Figure 6.11a4 or 8.6b5. The input CT images and ground truths are the same for Figures 8.4, 8.5, and 8.6. For example, the same input slices are shown in Figures 8.4a1 and 8.4b1, Figures 8.5a1 and 8.5b1, and Figures 8.6a1 and 8.6b1. Identical ground truth images are shown in Figures 8.4a3 and 8.4b5, Figures 8.6a3 and 8.5b5, and Figures 6.11a4 and 8.6b5. CT images (Figures 8.4b1, 8.5b1, and 8.6b1) and MR images (Figures 8.4b2, 8.5b2, and 8.6b2) are used to obtain the corresponding enhanced images (Figures 8.4b3, 8.5b3 and 8.6b3) using cross-modality guided liver contrast enhancement. We compare the segmented result with the ground truth. It can be seen from the quality assessment of liver segmentation (Table 8.2) that the segmented liver is more accurate when the contrast of the liver is enhanced.

Table 8.2 shows the quality assessment parameters i.e. dice, sensitivity and accuracy [39, 40]. The dice measure indicates the region of overlap. Sensitivity also called the true positive rate defines whether the method is sensitive to the liver elements. Accuracy explains the number of liver and non-liver elements segmented accurately. It can be seen from Table 8.2 that the average dice, sensitivity and accuracy of the liver segmentation are 0.656 ± 0.126, 0.816 ± 0.085 and 0.822 ± 0.082 respectively on the original liver images and 0.877 ± 0.036, 0.964 ± 0.037 and 0.956 ± 0.022 respectively on the enhanced liver images. It can be seen that the Chan-Vese approach improves the dice, sensitivity and accuracy of liver segmentation on the enhanced liver slices.

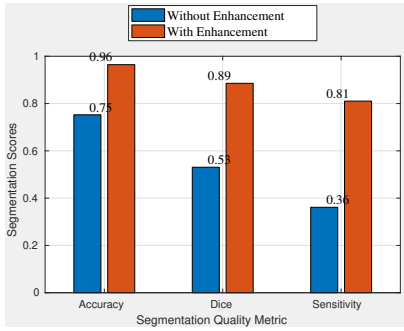
We further analyze the performance of liver segmentation on 24 liver slices from 4 different volumes as shown in Figure 8.7. Figures 8.7a, 8.7b, 8.7c and



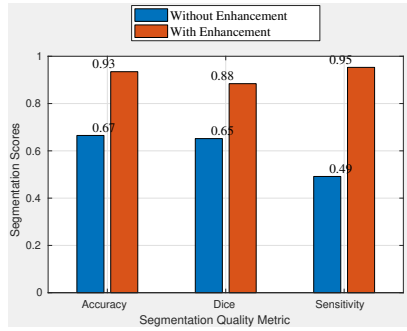
(a) Volume 28059



(b) Volume 23186



(c) Volume 18152



(d) Volume 10504

Figure 8.7
Segmentation Quality Assessment on Different Volumes

8.7d show the average values of accuracy, dice and sensitivity for the Chan-Vese approach both with and without enhancement. It can be seen from the performance results comparison (Figures 8.7a, 8.7b, 8.7c and 8.7d) that the cross-modality guided liver enhancement improves the quality of segmentation in terms of accuracy, dice and sensitivity using the proposed Chan-Vese approach for liver segmentation.

A shape-intensity prior level set proposed by Wang et. al. [37] used atlases which are weighted in the selected training datasets by calculating the similarities between the atlases and the test image to dynamically generate a subject-specific probabilistic atlas for the test image. The most likely liver region of the test image is further determined based on the generated atlas. A rough segmentation is obtained by a maximum a posteriori classification of the probability map, and the final liver segmentation is produced by a shape intensity prior level set in the most likely liver region. Thus the overall process is slow due to the training phase. The process also depends upon large datasets for training.

In the proposed work, we do not use training and hence do not need large datasets. In the absence of strong edges, region based formulation using Chan-Vese performs well for segmentation which can be seen from the performance results and the authors employ a cross-modality guided image enhancement as a pre-processing step which further improves the quality of segmentation. The proposed segmentation algorithm can delineate liver boundaries that have

levels of variability similar to those obtained manually. The proposed approach speeds up the overall process of liver segmentation by 100 times on the GPU compared to the CPU implementation.

MICCAI test data provided by the organizers of the "SLIVER07" contains clinical 3D computed tomography (CT) scans [27, 28]. The proposed method is based on a cross-modality approach in which we require MR scans for the guided enhancement. To the best of our knowledge, this is the first non-learning approach using cross-modality guided contrast enhancement for liver segmentation. The registered MR image is used to enhance the low quality CT image. This is done by using a non-learning approach of 2D histogram equalization and matching.

Kavur et. al. [16] reported that CT-MR liver segmentation is inferior to CT or MR image segmentation due to CT-MR visual difference. The study from CHAOS Challenge by Kavur et. al. [16], proposes a learning based approach for segmentation taking CT-MR images as training inputs in order to increase the training data and reveal common features of incorporated modalities for an organ. The deep model learns from the combined CT and MR datasets. The cross-modality (CT-MR) learning proved to be more challenging than individual training. Such complicated tasks could benefit from spatial priors, global topological, or shape-representations in their loss functions as employed by some of the deep learning models.

However, we show that the enhanced CT image using cross-modality approach provides better segmentation results in terms of dice , accuracy and sensitivity compared to the original CT image. In the next section, we discuss the speedup obtained by the proposed GPU implementation in comparison to the CPU.

8.5.3 *Speedup*

In this section, we discuss the speedup obtained by the GPU implementation of Chan-Vese compared to CPU implementation and analyze the impact of enhancement on speedup. The speedup analysis of liver segmentation on the CPU and the GPU is shown in Table 8.3.

Table 8.3
Liver Segmentation Speedup Analysis

Liver	Chan-Vese without Enhancement			Chan-Vese with Enhancement		
Slice #	CPU (s)	GPU (s)	Speedup	CPU (s)	GPU (s)	Speedup
1	4.324	0.276	15.667	276.15	2.78	99.335
2	4.117	0.256	16.082	270.098	2.44	110.696
3	2.679	0.195	13.738	173.93	1.95	89.195
4	2.857	0.211	13.54	165.03	1.65	100.018
5	3.112	0.219	14.21	175.82	1.73	101.63
Average	3.49425	0.2345	14.647	221.302	2.205	99.811
Std. Dev.	0.752	0.033	1.155	55.8	0.484	7.65

The computational complexity of the proposed Chan-Vese algorithm is O(N) where N is the number of elements in the CT image. So even for the large images, it is also very efficient. The average time taken by CPU implementa-

tions (with and without enhancement) are 221.302 $\pm$ 55.8 seconds and 3.49425 $\pm$ 0.752 seconds respectively and GPU implementations are 2.205 $\pm$ 0.484 seconds and 0.2345 $\pm$ 0.033 seconds respectively. Hence the GPU implementations (with and without enhancement) on an NVIDIA GPU GeForce GTX 1050 with RAM 4GB provide an average speedup of 99.811 $\pm$ 7.65 times and 14.647 $\pm$ 1.155 times in comparison to the CPU implementation on Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz RAM 24 GB. The reason behind the obtained speedup is the avoidance of intermediate kernel calls and exploiting high level parallelism present in liver contrast enhancement and Chan-Vese approach for image segmentation.

Liver contrast enhancement uses a 2D histogram technique which includes histogram of the pair of neighbouring elements in the CT and MR images. Hence the complexity increases due to pairwise histogram analysis of cumulative distributive function and histogram matching. The Chan-Vese approach includes time-consuming numerical calculations of partial differential equations. These tasks i.e. 2D histogram calculations and solutions to the partial differential equations have been implemented on an NVIDIA GPU in parallel for liver segmentation providing an average speedup of 99.811 $\pm$ 7.65 compared to the CPU implementation. The time is computed in GPU time and it is optimized by avoiding the intermediate memory transfers.

We perform the statistical treatment of results. P value from analysis of variance (ANOVA) for the datasets is $2.43 * 10^{-14}$ which is less than 0.0005 (0.05%). We reject the null hypothesis and conclude that not all means are equal which confirms the means are statistically significant for the experiments.

8.5.4 *Discussion*

The Chan-Vese algorithm is quite slow due to time-consuming computations of the partial differential equations, especially when dealing with large medical datasets. It can pose a problem for a real-time implementation and an efficient parallel approach is highly desirable. The Chan-Vese algorithm is a very powerful algorithm due to improved noise robustness . However, there are cases in which the liver segmentation is less accurate and sensitive. It is necessary to enhance the contrast of the liver for more accurate segmentation. Hence, we incorporate cross-modality guided image enhancement as a pre-processing step to improve the quality of liver segmentation. However, the cross-modality approach includes 2D histogram analysis which is time-consuming and includes repetitive tasks of pairwise histogram analysis of liver image elements. This is also applicable to numerical calculations of the partial differential equations in Chan-Vese. These repetitive tasks are implemented on an NVIDIA GPU using threads of blocks and the performance is improved significantly in comparison to the CPU implementation.

8.6 Conclusion

In this paper, we propose a fast parallel liver segmentation using the Chan-Vese approach and study the impact of contrast enhancement on liver segmentation. The proposed approach is fast, accurate and outperforms other approaches for low quality CT liver slices. The proposed segmentation algorithm can delineate

liver boundaries that have levels of variability similar to those obtained manually. GPU implementation of the proposed approach speeds up the overall process of liver segmentation by 100 times compared to the CPU implementation.

Chan-Vese approach for liver segmentation is less sensitive (0.816 ± 0.085 from Table 8.2) when applied to original CT liver images. Sensitivity should be increased for more accurate liver segmentation. Hence, we apply a cross-modality guided contrast enhancement on CT liver images and segment the liver using the proposed Chan-Vese approach for segmentation. The work compares CPU and GPU implementations with and without enhancement. The average dice, sensitivity and accuracy of the liver segmentation are 0.877 ± 0.036, 0.964 ± 0.037 and 0.956 ± 0.022 respectively on the enhanced liver images. The cross-modality guided contrast enhancement improves the quality of the results by decreasing the false positives. The proposed GPU implementation with enhancement improves the speedup by 99.811 ± 7.65 times over CPU implementation. Hence the parallel implementation of Chan-Vese approach for liver segmentation is faster when implemented on the GPU and more accurate when the contrast of the CT liver image is enhanced.

Acknowledgements

The work is supported by the project High Performance soft tissue Navigation (HiPerNav). This project has received funding from the European Union Horizon 2020 research and innovation program under grant agreement No. 722068. We thank The Intervention Centre, Oslo University Hospital, Oslo, Norway for providing the CT images with ground truths for the clinical validation of the liver segmentation.

REFERENCES

[1] Walid Aydi, Nouri Masmoudi, and Lotfi Kamoun: *Active contour without edges vs gvf active contour for accurate pupil segmentation*. International Journal of Computer Applications, Citeseer, 54 (4), 2012.

[2] Turgay Celik: *Two-dimensional histogram equalization and contrast enhancement*. Pattern Recognition, 45(10):3810 – 3824, 2012, ISSN 0031-3203.

[3] Bingyu Chen, Jianhui Zhao, Erqian Dong, Jun Chen, Yang Zhao, and Zhiyong Yuan: *An improved gvf snake model using magnetostatic theory*. In *Computer, Informatics, Cybernetics and Applications*, pages 431–440. Springer, 2012.

[4] Yuanzhi Cheng, Xin Hu, Ji Wang, Yadong Wang, and Shinichi Tamura: *Accurate vessel segmentation with constrained b-snake*. IEEE transactions on image processing : a publication of the IEEE Signal Processing Society, 24, April 2015.

[5] Rami Cohen: *The chan-vese algorithm*. CoRR, abs/1107.2782, 2011. http://arxiv.org/abs/1107.2782,(2011).

[6] Konstantinos K. Delibasis, Aristides Kechriniotis, and I. Maglogiannis: *A novel tool for segmenting 3d medical images based on generalized cylinders and active surfaces*. Computer Methods and Programs in Biomedicine, 111(1):148 – 165, 2013, ISSN 0169-2607.

[7] Jinming Duan, Zhenkuan Pan, Xiangfeng Yin, Weibo Wei, and Guodong Wang: *Some fast projection methods based on chan-vese model for image segmentation*. EURASIP Journal on Image and Video Processing, 2014(1):7, Jan 2014, ISSN 1687-5281.

[8] Åsmund Avdem Fretland, Vegar Johansen Dagenborg, Gudrun Maria Waaler Bjørnelv, Airazat M Kazaryan, Ronny Kristiansen, Morten Wang Fagerland, John Hausken, Tor Inge Tønnessen, Andreas Abildgaard, Leonid Barkhatov, *et al.*: *Laparoscopic versus open resection for colorectal liver metastases*. Annals of surgery, 267(2):199–207, 2018.

[9] Kshitij Gupta, Jeff A Stuart, and John D Owens: *A study of persistent threads style gpu programming for gpgpu workloads*. In *Innovative Parallel Computing-Foundations & Applications of GPU, Manycore, and Heterogeneous Systems (INPAR 2012)*, pages 1–14. IEEE, 2012.

[10] Mark Harris: *Cuda pro tip:write flexible kernels with grid-stride loops*, 2015. http://goo.gl/b8Vmkh.

[11] Mohammad Farukh Hashmi, Satyarth Katiyar, Avinash G Keskar, Neeraj Dhanraj Bokde, and Zong Woo Geem: *Efficient pneumonia detection in chest xray images using deep transfer learning*. Diagnostics, 10(6), 2020, ISSN 2075-4418.

[12] Lin He and Stanley Osher: *Solving the chan-vese model by a multiphase level set algorithm based on the topological derivative*. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 777–788. Springer, 2007.

[13] RJ Hemalatha, TR Thamizhvani, A Josephin Arockia Dhivya, Josline Elsa Joseph, Bincy Babu, and R Chandrasekaran: *Active contour based segmentation techniques for medical image analysis*. Medical and Biological Image Analysis, page 17, 2018.

[14] Assaf Hoogi, Christopher F. Beaulieu, Guilherme M. Cunha, Elhamy Heba, Claude B. Sirlin, Sandy Napel, and Daniel L. Rubin: *Adaptive local window for level set segmentation of ct and mri liver lesions*. Medical Image Analysis, 37:46 – 55, 2017, ISSN 1361-8415.

[15] Seung Won Jung: *Two-dimensional histogram specification using two-dimensional cumulative distribution function*. Electronics Letters, 50(12):872–874, 2014.

[16] A Emre Kavur, N Sinem Gezer, Mustafa Barış, Pierre Henri Conze, Vladimir Groza, Duc Duy Pham, Soumick Chatterjee, Philipp Ernst, Savaş Özkan, Bora Baydar, *et al.*: *Chaos challenge–combined (ct-mr) healthy abdominal organ segmentation*. arXiv preprint arXiv:2001.06535, 2020.

[17] A. Khadidos, V. Sanchez, and C. Li: *Active contours based on weighted gradient vector flow and balloon forces for medical image segmentation*. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 902–906, Oct 2014.

[18] Thomas Kronfeld, David Brunner, and Guido Brunnett: *Snake-based segmentation of teeth from virtual dental casts*. Computer-Aided Design and Applications, 7(2):221–233, 2010.

[19] Zhong Bing Li, Xian Ze Xu, Yi Le, and Feng Qiu Xu: *An improved balloon snake for hifu image-guided system*. Journal of Medical Ultrasonics, 41(3):291–300, 2014.

[20] Xuesong Lu, Qinlan Xie, Yunfei Zha, and Defeng Wang: *Fully automatic liver segmentation combining multi-dimensional graph cut with shape information in 3d ct images*. Scientific Reports, 8(1):10700, 2018.

[21] Rabia Naseem, Faouzi Alaya Cheikh, Azeddine Beghdadi, Ole Jakob Elle, and Frank Lindseth: *Cross modality guided liver image enhancement of ct using mri*. In *2019 8th European Workshop on Visual Information Processing (EUVIP)*, pages 46–51. IEEE, 2019.

[22] Rafael Palomar, Juan Gómez-Luna, Faouzi A Cheikh, Joaquín Olivares-Bueno, and Ole J Elle: *High-performance computation of bézier surfaces on parallel and heterogeneous platforms*. International Journal of Parallel Programming, 46(6):1035–1062, 2018.

[23] A Prabhakara Rao, Neeraj Bokde, and Saugata Sinha: *Photoacoustic imaging for management of breast cancer: A literature review and future perspectives*. Applied Sciences, 10(3):767, 2020.

[24] Sawrav Roy, Susanta Mukhopadhyay, and Manoj K Mishra: *Enhancement of morphological snake based segmentation by imparting image attachment through scale-space continuity*. Pattern Recognition, 48(7):2254–2268, 2015.

[25] Nitin Satpute, Rabia Naseem, Rafael Palomar, Orestis Zachariadis, Juan Gómez-Luna, Faouzi Alaya Cheikh, and Joaquín Olivares: *Fast parallel vessel segmentation*. Computer Methods and Programs in Biomedicine, 192:105430, 2020, ISSN 0169-2607.

[26] Nitin Satpute, Rabia Naseem, Egidijus Pelanis, Juan Gomez-Luna, Faouzi Alaya Cheikh, Ole J Elle, and Joaquin Olivares: *Gpu acceleration of liver enhancement for tumor segmentation*. Computer Methods and Programs in Biomedicine, 184:105285, 2020, ISSN 0169-2607.

[27] Changfa Shi, Yuanzhi Cheng, Fei Liu, Yadong Wang, Jing Bai, and Shinichi Tamura: *A hierarchical local region-based sparse shape composition for liver segmentation in ct scans*. Pattern Recognition, 50:88–106, 2016.

[28] Changfa Shi, Yuanzhi Cheng, Jinke Wang, Yadong Wang, Kensaku Mori, and Shinichi Tamura: *Low-rank and sparse decomposition based shape model and probabilistic atlas for automatic pathological organ segmentation*. Medical image analysis, 38:30–49, 2017.

[29] Sangeeta K Siri and Mrityunjaya V Latte: *Combined endeavor of neutrosophic set and chan-vese model to extract accurate liver image from ct scan*. Computer Methods and Programs in Biomedicine, 151:101–109, 2017.

[30] Dirk Smeets, Dirk Loeckx, Bert Stijnen, Bart De Dobbelaer, Dirk Vandermeulen, and Paul Suetens: *Semi-automatic level set segmentation of liver tumors combining a spiral-scanning technique with supervised fuzzy pixel classification*. Medical Image Analysis, 14(1):13 – 20, 2010, ISSN 1361-8415.

[31] Erik Smistad, Mohammadmehdi Bozorgi, and Frank Lindseth: *Fast: framework for heterogeneous medical image computing and visualization*. International Journal of Computer Assisted Radiology and Surgery, 10(11):1811–1822, 2015.

[32] Erik Smistad, Anne C Elster, and Frank Lindseth: *Gpu accelerated segmentation and centerline extraction of tubular structures from medical images*. International Journal of Computer Assisted Radiology and Surgery, 9(4):561–575, 2014.

[33] Erik Smistad, Anne C Elster, and Frank Lindseth: *Real-time gradient vector flow on gpus using opencl*. Journal of Real-Time Image Processing, 10(1):67–74, 2015.

[34] Erik Smistad, Thomas L Falch, Mohammadmehdi Bozorgi, Anne C Elster, and Frank Lindseth: *Medical image segmentation on gpus–a comprehensive review*. Medical Image Analysis, 20(1):1–18, 2015.

[35] Mohammed Sourouri, Scott B Baden, and Xing Cai: *Panda: A compiler framework for concurrent cpu+gpu execution of 3d stencil computations on gpu-accelerated supercomputers*. International Journal of Parallel Programming, 45(3):711–729, 2017.

[36] Sho Tomoshige, Elco Oost, Akinobu Shimizu, Hidefumi Watanabe, and Shigeru Nawano: *A conditional statistical shape model with integrated error estimation of the conditions; application to liver segmentation in non-contrast ct images*. Medical Image Analysis, 18(1):130 – 143, 2014, ISSN 1361-8415.

[37] Jinke Wang, Yuanzhi Cheng, Changyong Guo, Yadong Wang, and Shinichi Tamura: *Shapeintensity prior level set combining probabilistic atlas and probability map constrains for automatic liver segmentation from abdominal ct images*. International Journal of Computer Assisted Radiology and Surgery, 11, December 2015.

[38] Xiao Feng Wang, De Shuang Huang, and Huan Xu: *An efficient local chan–vese model for image segmentation*. Pattern Recognition, 43(3):603–618, 2010.

[39] Z. Yan, X. Yang, and K. Cheng: *A three-stage deep learning model for accurate retinal vessel segmentation*. IEEE Journal of Biomedical and Health Informatics, 23(4):1427–1436, July 2019.

[40] M. H. Yap, G. Pons, J. Martí, S. Ganau, M. Sentís, R. Zwiggelaar, A. K. Davison, and R. Martí: *Automated breast ultrasound lesions detection using convolutional neural networks*. IEEE Journal of Biomedical and Health Informatics, 22(4):1218–1226, July 2018.

[41] Orestis Zachariadis, Andrea Teatini, Nitin Satpute, Juan Gómez-Luna, Onur Mutlu, Ole Jakob Elle, and Joaquín Olivares: *Accelerating b-spline interpolation on gpus: Application to medical image registration*. Computer Methods and Programs in Biomedicine, 193:105431, 2020, ISSN 0169-2607.

[42] Kaihua Zhang, Huihui Song, and Lei Zhang: *Active contours driven by local image fitting energy*. Pattern Recognition, 43(4):1199–1206, 2010.

[43] Jianhui Zhao, Bingyu Chen, Mingui Sun, Wenyan Jia, and Zhiyong Yuan: *Improved algorithm for gradient vector flow based active contour model using global and local information*. The Scientific World Journal, 2013, 2013.

[44] Huiyu Zhou, Xuelong Li, Gerald Schaefer, M Emre Celebi, and Paul Miller: *Mean shift based gradient vector flow for image segmentation*. Computer Vision and Image Understanding, 117(9):1004–1016, 2013.

Part III

DISCUSSION AND CONCLUSIONS

9 DISCUSSION

Current technological landscape is focused on the real time implementation of the liver enhancement and accurate segmentation. In response to this growing challenge, the GPU optimization of medical image enhancement and segmentation is a priority objective in medical image analysis. This doctoral thesis addresses this problem by using cross modality and semiautomatic approach. With this objective a methodological procedure for liver contrast enhancement using cross modality and semiautomatic segmentation approaches have been proposed to demonstrate, the result of which the articles addressed in the previous block have been published. This chapter discusses a discussion of the work done from the general perspective of the thesis approach.

The fast parallel cross modality based contrast enhancement for CT liver images has been proposed. Further GPU performs dynamic RoI based tumor segmentation on enhanced CT liver image. These fast parallel implementations are based on persistence, grid-stride loop and IBS. The process of cross modality based contrast enhancement is computationally expensive and hence time consuming. This involves 2D histogram calculation, equalization and histogram matching [9]. They require several light weight tasks. The performance on GPU is improved compared to the CPU by dividing the tasks on several active threads.

The second part of the process is tumor segmentation. The gradient and dynamic RoI based SRG inspired from the works of Rai and Nair [23] has been proposed and implemented on GPU [49, 50]. GPU places vital role to speedup the image guided surgery [49, 50, 72]. Initially, the process needs small part of the region to be accessed instead of whole image (as implemented previously on GPU). As soon as region grows, RoI should be increased to access more neighboring elements. GPU implementation of SRG involves kernel termination and relaunch continuously from CPU. This is time consuming. We avoid this by using persistence and grid-stride loop and obtain the significant speedup i.e. 104.416 ± 5.166 times compared to the sequential implementation of liver enhancement and tumor segmentation.

Chan-Vese algorithm is a very powerful algorithm due to better robustness for noise. But there are cases when false positives can not be avoided. It is necessary to enhance the contrast of liver to reduce false positives and increase the sensitivity of the liver segmentation. Hence we incorporate impact of enhancement on Chan-Vese model for improving the quality of the liver segmentation.

10

CONCLUSIONS

As discussed in Chapter 2, trends in the development of medical image computing focus on contrast enhancement and liver segmentation, this thesis contributes to the GPU implementation of cross modality based liver contrast enhancement and segmentation that evaluate the state of liver, tumor and vessel segmentation, and whose CPU implementations are computationally costlier. The proposals are obtained through the fast parallel contrast enhancement and seeded region growing and high performance Chan-Vese implementation for liver, vessel and tumor segmentation.

The developed work is presented by means of three publications presented in Chapter ii. The first one, with a methodological objective, presents a cross modality based liver contrast enhancement, suitable for tumor segmentation. The second article deals with the fast parallel vessel segmentation where gradient based static seeded region growing has been proposed. Based on the Chan-Vese model, the third publication presents GPU acceleration of Chan-Vese model and the impact of liver enhancement for liver segmentation, an algorithm that maximizes the quality of liver segmentation by minimizing the false positives simultaneously.

The following conclusions can be drawn from the work of this doctoral thesis:

- Liver enhancement, based on cross modality [26, 42] based contrast enhancement provides accurate tumor segmentation using dynamic seeded region growing compared to the state-of-the-art. The GPU implementation of the liver contrast enhancement for tumor segmentation is around 100 times faster compared to the CPU implementation [50].

- Fast parallel vessel segmentation has been proposed. Gradient based static seeded region growing provides accurate vessel segmentation compared to the state-of-the-art. Proposed persistence [21, 59] and grid-stride loop [24] based static seeded region growing is outperforming kernel termination and relaunch based vessel segmentation.

- GPU implementation of Chan-Vese model [10, 53, 63, 73] has been proposed for liver segmentation. The impact of liver enhancement on liver segmentation has been analyzed and the GPU implementation of Chan-Vese model with enhancement for liver segmentation is around 100 times faster compared to the CPU implementation.

In conclusion, the experiments illustrate the potential of the proposed methodology, whose GPU implementation is around 100 times faster compared to the CPU implementation. The results demonstrate the hypotheses of this thesis which, structuring the papers with a series of methodological objectives, propose the possibility of false positive reduction, measuring the effect of liver enhancement on the accuracy of the segmentation, and the use of this knowledge in a procedure that achieves increased accuracy of the liver, tumor or vessel segmentation.

11

FUTURE WORK

This chapter provides visions of future work and potential improvements of the work described in this doctoral thesis.

As for the liver segmentation of the liver, tumor or vessel is preprocessing stage. The quality of segmentation is largely dependent on how the images are preprocessed. Contrast of the CT images are sometimes low. It is necessary to enhance the low contrast CT liver images for better quality of liver segmentation. This aspect can be addressed by introducing a cross modality based contrast enhancement where guided image is provided to enhance the target image. On the other hand, an interesting aspect is to consider the possibility of the guided image from registered or unregistered CT and MRI images from same patient or different patients. This application requires rethinking for the selection of guided MRI image for contrast enhancement to incorporate the features of guided MRI into target CT image, considering the use of 2D histogram equalization and matching. This potential application opens the door to developing a framework to select a single generalized guided MR image for the contrast enhancement of the CT liver image which necessitates the optimized use of GPU resources due to heavy computations involved.

With respect to cross modality based liver contrast enhancement, the information provided by the proposed approach suggests the possibility of analyzing the effect of various guided MR images in a system to evaluate the quality of enhanced CT liver images. The analysis of these guided MR images can lead to a thorough knowledge of the behavior of the system for image quality assessment, opening the door to possible analytical methods that allow working with various MR images, rather than a bounded and predefined set. On the other hand, the cropping and denoising have a great influence on the accuracy of the liver segmentation, so it would be interesting to analyze the GPU optimization strategies for these preprocessing steps to improve the quality of segmentation.

In relation to the segmentation, the algorithm uses a semi-automatic method to segment liver, vessel or tumor. If the effects of the variation of the initial seed points or initial curves or threshold criteria are appropriately modeled, we could consider how to determine the optimal configuration, thus avoiding possible problems in scenarios with manual adjustment of parameters for improving the quality of segmentation. In addition, the proposed method is based on the preprocessing of the images. It would be interesting to explore the impact of cropping, enhancement, and denoising on semi automatic or automatic segmentation. On the other hand, semi-automatic segmentation using seeded region growing or Chan-Vese is an iterative process in which each iteration determines the ideal candidate node for the next step. GPU helps in speeding up the iterative process but the quality depends on the selection of the candidate pixels or voxels that need to be grown. This logic follows a heuristic of candidate selection initially and growing the region based on the threshold

criteria in each step, but deserves an in-depth analysis that contemplates the initial selection of seeds that can improve the quality of segmentation.

Finally, the techniques developed in this thesis have a place in applications that require to accurately detect and segment liver, vessel and tumors. The adaptation of the volume of images generated through scans to the imminence of a speedup and quality of segmentation is an interesting feature that enables a novel enhancement and segmentation works for increasing the accuracy and the speedup of the segmentation. This type of model aims to detect unusual patterns in the desired organs that can generally happen at any point during CT or MRI scans, progressively increasing the quality of results in those areas in which they are difficult to segment. Semi automatic methods can be a relevant concept in applications where we can not compromise with the accuracy of the results.

The current approaches are inclined towards multitask, multimodal and federated learning for medical image analysis in order to achieve real time implementation with better accuracy. Multitask learning is useful especially when the different regions of interests in the image share the similar features. For e.x., instead of 3 deep learning models, it will be beneficial to develop a single model for the segmentation of liver, tumor and vessels as they share the similar features inside the liver volume. The liver volumes from different scans (CT, MR and US) can be exploited at the same time for contrast enhancement and/or denoising in order to achieve better accuracy of the segmentation. And finally, federated learning plays a vital role in achieving the real time training of the patient specific data in local computing units instead of centralized servers. This helps in preserving the privacy of the patient specific data. The data is made available and processed at the same time when it is generated. It helps in achieving better accuracy and real time analysis of the patient data.

BIBLIOGRAPHY

[1] Mahmoud Al-Ayyoub, Shadi AlZubi, Yaser Jararweh, Mohammed A Shehab, and Brij B Gupta. Accelerating 3d medical volume segmentation using gpus. *Multimedia Tools and Applications*, 77(4):4939–4958, 2018. (Cited on page 14.)

[2] Andika Elok Amalia, Gregorius Airlangga, and Afandi Nur Aziz Thohari. Breast cancer image segmentation using k-means clustering based on gpu cuda parallel computing. *Jurnal Infotel*, 10(1):33–38, 2018. (Cited on page 13.)

[3] Walid Aydi, Nouri Masmoudi, and Lotfi Kamoun. Active contour without edges vs gvf active contour for accurate pupil segmentation. *International Journal of Computer Applications, Citeseer*, 54 (4), 2012. (Cited on page 13.)

[4] Jose Bernal, Kaisar Kushibar, Daniel S Asfaw, Sergi Valverde, Arnau Oliver, Robert Martí, and Xavier Lladó. Deep convolutional neural networks for brain image analysis on magnetic resonance imaging: a review. *Artificial intelligence in medicine*, 95:64–81, 2019. (Cited on pages 13 and 15.)

[5] Satish S Bhairannawar. Efficient medical image enhancement technique using transform hsv space and adaptive histogram equalization. In *Soft Computing Based Medical Image Analysis*, pages 51–60. Elsevier, 2018. (Cited on pages 11 and 26.)

[6] Turgay Celik. Two-dimensional histogram equalization and contrast enhancement. *Pattern Recognition*, 45(10):3810 – 3824, 2012. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2012.03.019. (Cited on page 28.)

[7] Chen Chen, Chen Qin, Huaqi Qiu, Giacomo Tarroni, Jinming Duan, Wenjia Bai, and Daniel Rueckert. Deep learning for cardiac image segmentation: A review. *Frontiers in Cardiovascular Medicine*, 7:25, 2020. (Cited on page 13.)

[8] Guoyang Chen and Xipeng Shen. Free launch: optimizing gpu dynamic kernel launches through thread reuse. In *Proceedings of the 48th International Symposium on Microarchitecture*, pages 407–419. ACM, 2015. (Cited on pages 4, 5, and 27.)

[9] Hu Chen, Yi Zhang, Mannudeep K Kalra, Feng Lin, Yang Chen, Peixi Liao, Jiliu Zhou, and Ge Wang. Low-dose ct with a residual encoder-decoder convolutional neural network. *IEEE Transactions on Medical Imaging*, 36 (12):2524–2535, 2017. (Cited on page 97.)

[10] Rami Cohen. The chan-vese algorithm. *CoRR*, abs/1107.2782, 2011. URL `http://arxiv.org/abs/1107.2782,(2011)`. (Cited on pages 4, 13, and 99.)

[11] Carlo N De Cecco, Giuseppe Muscogiuri, U Joseph Schoepf, Damiano Caruso, Julian L Wichmann, Paola M Cannaò, Christian Canstein,

Stephen R Fuller, Lauren Snider, Akos Varga-Szemes, et al. Virtual unenhanced imaging of the liver with third-generation dual-source dual-energy ct and advanced modeled iterative reconstruction. *European journal of radiology*, 85(7):1257–1264, 2016. (Cited on page 11.)

[12] Konstantinos K. Delibasis, Aristides Kechriniotis, and I. Maglogiannis. A novel tool for segmenting 3d medical images based on generalized cylinders and active surfaces. *Computer Methods and Programs in Biomedicine*, 111(1):148 – 165, 2013. ISSN 0169-2607. doi: https://doi.org/10.1016/j.cmpb.2013.03.009. (Cited on pages 4 and 11.)

[13] Nilanjan Dey and Amira S Ashour. Meta-heuristic algorithms in medical image segmentation: a review. In *Advancements in Applied Metaheuristic Computing*, pages 185–203. IGI Global, 2018. (Cited on page 14.)

[14] Jinming Duan, Zhenkuan Pan, Xiangfeng Yin, Weibo Wei, and Guodong Wang. Some fast projection methods based on chan-vese model for image segmentation. *EURASIP Journal on Image and Video Processing*, 2014(1):7, Jan 2014. ISSN 1687-5281. doi: 10.1186/1687-5281-2014-7. (Cited on page 13.)

[15] Yuncong Feng, Haiying Zhao, Xiongfei Li, Xiaoli Zhang, and Hongpeng Li. A multi-scale 3d otsu thresholding algorithm for medical image segmentation. *Digital Signal Processing*, 60:186–199, 2017. (Cited on page 12.)

[16] King-Sun Fu and JK Mui. A survey on image segmentation. *Pattern recognition*, 13(1):3–16, 1981. (Cited on page 11.)

[17] Shujun Fu, Ming Zhang, Chengpo Mu, and Xiaohong Shen. Advancements of medical image enhancement in healthcare applications. *Journal of Healthcare Engineering*, 2018, 2018. (Cited on pages 11 and 26.)

[18] MECVG Greiner. Stack implementation on programmable graphics hardware. *Vision, Modeling, and Visualization 2004: Proceedings*, page 255, 2004. (Cited on page 5.)

[19] I. Grinias and G. Tziritas. A semi-automatic seeded region growing algorithm for video object localization and tracking. *Signal Processing: Image Communication*, 16(10):977 – 986, 2001. ISSN 0923-5965. doi: http://doi.org/10.1016/S0923-5965(01)00014-5. (Cited on page 12.)

[20] Sakshi Gujral. Early diabetes detection using machine learning: a review. *Int. J. Innov. Res. Sci. Technol*, 3(10):57–62, 2017. (Cited on page 14.)

[21] Kshitij Gupta, Jeff A Stuart, and John D Owens. A study of persistent threads style gpu programming for gpgpu workloads. In *Innovative Parallel Computing-Foundations & Applications of GPU, Manycore, and Heterogeneous Systems (INPAR 2012)*, pages 1–14. IEEE, 2012. (Cited on pages 27 and 99.)

[22] Patrick Nigri Happ, Raul Queiroz Feitosa, Cristiana Bentes, and Ricardo Farias. A region growing segmentation algorithm for gpus. *Boletim de Ciências Geodésicas*, 19(2):208–226, 2013. (Cited on page 12.)

[23] GN Harikrishna Rai and TR Gopalakrishnan Nair. Gradient based seeded region grow method for ct angiographic image segmentation. *arXiv preprint arXiv:1001.3735*, 2010. (Cited on pages 4 and 97.)

[24] Mark Harris. Cuda pro tip:write flexible kernels with grid-stride loops, 2015. URL http://goo.gl/b8Vmkh. (Cited on page 99.)

[25] Hamid Hassanpour, Najmeh Samadiani, and SM Mahdi Salehi. Using morphological transforms to enhance the contrast of medical images. *The Egyptian Journal of Radiology and Nuclear Medicine*, 46(2):481–489, 2015. (Cited on pages 4 and 11.)

[26] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(6):1397–1409, 2012. (Cited on pages 11 and 99.)

[27] Lin He and Stanley Osher. Solving the chan-vese model by a multiphase level set algorithm based on the topological derivative. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 777–788. Springer, 2007. (Cited on page 13.)

[28] Blake A. Hechtman, Andrew D. Hilton, and Daniel J. Sorin. TREES: A CPU/GPU task-parallel runtime with explicit epoch synchronization. *CoRR*, abs/1608.00571:„ 2016. URL http://arxiv.org/abs/1608.00571. (Cited on page 5.)

[29] RJ Hemalatha, TR Thamizhvani, A Josephin Arockia Dhivya, Josline Elsa Joseph, Bincy Babu, and R Chandrasekaran. Active contour based segmentation techniques for medical image analysis. *Medical and Biological Image Analysis*, page 17, 2018. (Cited on pages 11 and 13.)

[30] Milan Jaroš, Petr Strakoš, Tomáš Karásek, Lubomír Říha, Alena Vašatová, Marta Jarošová, and Tomáš Kozubek. Implementation of k-means segmentation algorithm on intel xeon phi and gpu: Application in medical imaging. *Advances in Engineering Software*, 103:21–28, 2017. (Cited on page 14.)

[31] VL Jaya and R Gopikakumari. Iem: a new image enhancement metric for contrast and sharpness measurements. *International Journal of Computer Applications*, 79(9), 2013. (Cited on pages 4 and 11.)

[32] Seung-Won Jung. Two-dimensional histogram specification using two-dimensional cumulative distribution function. *Electronics Letters*, 50(12): 872–874, 2014. (Cited on page 28.)

[33] Seung-Won Jung. Two-dimensional histogram specification using two-dimensional cumulative distribution function. *Electronics Letters*, 50(12): 872–874, 2014. (Cited on pages 11 and 26.)

[34] T Kalaiselvi, P Sriramakrishnan, and K Somasundaram. Survey of using gpu cuda programming model in medical image analysis. *Informatics in Medicine Unlocked*, 9:133–144, 2017. (Cited on pages 13, 14, and 15.)

[35] Zubair Khan, Jianjun Ni, Xinnan Fan, and Pengfei Shi. An improved k-means clustering algorithm based on an adaptive initial parameter estima-

tion procedure for image segmentation. *Int. J. Innov. Comput. Inf. Control*, 13(5):1509–1525, 2017. (Cited on page 14.)

[36] Yijun Li, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep joint image filtering. In *European Conference on Computer Vision, 2016*, pages 154–169. Springer, 2016. (Cited on pages 4 and 11.)

[37] Xuesong Lu, Qinlan Xie, Yunfei Zha, and Defeng Wang. Fully automatic liver segmentation combining multi-dimensional graph cut with shape information in 3d ct images. *Scientific Reports*, 8(1):10700, 2018. (Cited on pages 4 and 13.)

[38] Subhamoy Mandal, Xosé Luís Deán-Ben, and Daniel Razansky. Visual quality enhancement in optoacoustic tomography using active contour segmentation priors. *IEEE transactions on medical imaging*, 35(10):2209–2217, 2016. (Cited on page 11.)

[39] Andrew Mehnert and Paul Jackway. An improved seeded region growing algorithm. *Pattern Recogn. Lett.*, 18(10):1065–1071, Oct 1997. ISSN 0167-8655. doi: 10.1016/S0167-8655(97)00131-1. (Cited on page 12.)

[40] Sara Moccia, Elena De Momi, Sara El Hadji, and Leonardo S Mattos. Blood vessel segmentation algorithmsreview of methods, datasets and evaluation metrics. *Computer methods and programs in biomedicine*, 158:71–91, 2018. (Cited on page 14.)

[41] Mehrdad Moghbel, Syamsiah Mashohor, Rozi Mahmud, and M Iqbal Bin Saripan. Review of liver segmentation and computer assisted detection/-diagnosis methods in computed tomography. *Artificial Intelligence Review*, 50(4):497–537, 2018. (Cited on page 14.)

[42] Rabia Naseem, Faouzi Alaya Cheikh, Azeddine Beghdadi, Ole Jakob Elle, and Frank Lindseth. Cross modality guided liver image enhancement of ct using mri. In *2019 8th European Workshop on Visual Information Processing (EUVIP)*, pages 46–51. IEEE, 2019. (Cited on pages 14 and 99.)

[43] Rafael Palomar, Faouzi A Cheikh, Bjørn Edwin, Åsmund Fretland, Azeddine Beghdadi, and Ole J Elle. A novel method for planning liver resections using deformable bézier surfaces and distance maps. *Computer Methods and Programs in Biomedicine*, 144:135–145, 2017. (Cited on pages 4, 11, and 13.)

[44] Tiago Carneiro Pessoa, Jan Gmys, Nouredine Melab, Francisco Heron de Carvalho Junior, and Daniel Tuyttens. A gpu-based backtracking algorithm for permutation combinatorial problems. In Jesus Carretero, Javier Garcia-Blas, Ryan K.L. Ko, Peter Mueller, and Koji Nakano, editors, *Algorithms and Architectures for Parallel Processing*, pages 310–324, Cham, 2016. Springer International Publishing. ISBN 978-3-319-49583-5. (Cited on page 5.)

[45] VB Prasath, Haneen Arafat Abu Alfeilat, Omar Lasassmeh, Ahmad Hassanat, and Ahmad S Tarawneh. Distance and similarity measures effect on the performance of k-nearest neighbor classifier–a review. *arXiv preprint arXiv:1708.04321*, 2017. (Cited on page 14.)

[46] Rangaraj Rangayyan. Parallel implementation of the adaptive neighborhood contrast enhancement technique using histogram-based image partitioning. *Journal of Electronic Imaging*, 10:804, 07 2001. doi: 10.1117/1.1382810. (Cited on page 14.)

[47] O. Ronneberger, P.Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, pages 234–241. Springer, 2015. URL http://lmb.informatik.uni-freiburg.de/Publications/2015/RFB15a. (available on arXiv:1505.04597 [cs.CV]). (Cited on pages 4 and 13.)

[48] Prasanna K Sahoo, SAKC Soltani, and Andrew KC Wong. A survey of thresholding techniques. *Computer vision, graphics, and image processing*, 41 (2):233–260, 1988. (Cited on page 11.)

[49] Nitin Satpute, Rabia Naseem, Rafael Palomar, Orestis Zachariadis, Juan Gómez-Luna, Faouzi Alaya Cheikh, and Joaquín Olivares. Fast parallel vessel segmentation. *Computer Methods and Programs in Biomedicine*, page 105430, 2020. ISSN 0169-2607. doi: https://doi.org/10.1016/j.cmpb.2020.105430. (Cited on page 97.)

[50] Nitin Satpute, Rabia Naseem, Egidijus Pelanis, Juan Gomez-Luna, Faouzi Alaya Cheikh, Ole J Elle, and Joaquin Olivares. Gpu acceleration of liver enhancement for tumor segmentation. *Computer Methods and Programs in Biomedicine*, 184:105285, 2020. ISSN 0169-2607. doi: https://doi.org/10.1016/j.cmpb.2019.105285. (Cited on pages 4, 10, 11, 13, 15, 27, 97, and 99.)

[51] Amruta K Savaashe and Nagaraj V Dharwadkar. A review on cardiac image segmentation. In *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, pages 545–550. IEEE, 2019. (Cited on page 13.)

[52] Xiaoyong Shen, Chao Zhou, Li Xu, and Jiaya Jia. Mutual-structure for joint filtering. In *Proceedings of the International Conference on Computer Vision, 2015*, pages 3406–3414. IEEE, 2015. (Cited on page 11.)

[53] Sangeeta K Siri and Mrityunjaya V Latte. Combined endeavor of neutrosophic set and chan-vese model to extract accurate liver image from ct scan. *Computer Methods and Programs in Biomedicine*, 151:101–109, 2017. (Cited on pages 4, 11, 13, 15, and 99.)

[54] Erik Smistad. Seeded region growing. https://github.com/smistad/-FAST/tree/master/source, 2015. (Cited on pages 4 and 10.)

[55] Erik Smistad, Anne C Elster, and Frank Lindseth. Gpu accelerated segmentation and centerline extraction of tubular structures from medical images. *International Journal of Computer Assisted Radiology and Surgery*, 9 (4):561–575, 2014. (Cited on page 13.)

[56] Erik Smistad, Anne C Elster, and Frank Lindseth. Gpu accelerated segmentation and centerline extraction of tubular structures from medical images. *International Journal of Computer Assisted Radiology and Surgery*, 9 (4):561–575, 2014. (Cited on pages 4, 10, 13, 14, and 15.)

[57] Erik Smistad, Mohammadmehdi Bozorgi, and Frank Lindseth. Fast: framework for heterogeneous medical image computing and visualization. *International Journal of Computer Assisted Radiology and Surgery*, 10 (11):1811–1822, 2015. (Cited on pages 11 and 14.)

[58] Erik Smistad, Thomas L Falch, Mohammadmehdi Bozorgi, Anne C Elster, and Frank Lindseth. Medical image segmentation on gpus–a comprehensive review. *Medical image analysis*, 20(1):1–18, 2015. (Cited on pages 4, 11, 15, and 27.)

[59] Tyler Sorensen, Hugues Evrard, and Alastair F Donaldson. Cooperative kernels: Gpu multitasking for blocking algorithms. In *Proceedings of the 11th Joint Meeting on Foundations of Software Engineering, 2017*, pages 431–441. ACM, 2017. (Cited on page 99.)

[60] P Sreeja and S Hariharan. An improved feature based image fusion technique for enhancement of liver lesions. *Biocybernetics and Biomedical Engineering*, 38(3):611–623, 2018. (Cited on page 11.)

[61] Bharath Subramani and Magudeeswaran Veluchamy. Mri brain image enhancement using brightness preserving adaptive fuzzy histogram equalization. *International Journal of Imaging Systems and Technology*, 28(3):217–222, 2018. (Cited on pages 11 and 26.)

[62] Nima Tajbakhsh, Laura Jeyaseelan, Qian Li, Jeffrey N Chiang, Zhihao Wu, and Xiaowei Ding. Embracing imperfect datasets: A review of deep learning solutions for medical image segmentation. *Medical Image Analysis*, page 101693, 2020. (Cited on page 13.)

[63] Xiao-Feng Wang, De-Shuang Huang, and Huan Xu. An efficient local chan–vese model for image segmentation. *Pattern Recognition*, 43(3):603–618, 2010. (Cited on pages 11, 13, and 99.)

[64] Jan Wassenberg, Wolfgang Middelmann, and Peter Sanders. An efficient parallel algorithm for graph-based image segmentation. In *International Conference on Computer Analysis of Images and Patterns*, pages 1003–1010. Springer, 2009. (Cited on page 12.)

[65] Jelmer M Wolterink, Tim Leiner, Max A Viergever, and Ivana Išgum. Generative adversarial networks for noise reduction in low-dose ct. *IEEE Transactions on Medical Imaging*, 36(12):2536–2545, 2017. (Cited on pages 11 and 26.)

[66] Jingjun Wu, Ailian Liu, Jingjing Cui, Anliang Chen, Qingwei Song, and Lizhi Xie. Radiomics-based classification of hepatocellular carcinoma and hepatic haemangioma on precontrast magnetic resonance images. *BMC medical imaging*, 19(1):23, 2019. (Cited on page 14.)

[67] Qiong Yan, Xiaoyong Shen, Li Xu, Shaojie Zhuo, Xiaopeng Zhang, Liang Shen, and Jiaya Jia. Cross-field joint image restoration via scale map. In *ICCV, 2013*, pages 1537–1544. IEEE, 2013. (Cited on pages 4 and 11.)

[68] Z. Yan, X. Yang, and K. Cheng. A three-stage deep learning model for accurate retinal vessel segmentation. *IEEE Journal of Biomedical and Health*

Informatics, 23(4):1427–1436, July 2019. doi: 10.1109/JBHI.2018.2872813. (Cited on pages 10, 11, and 28.)

[69] Dong Yang, Daguang Xu, S Kevin Zhou, Bogdan Georgescu, Mingqing Chen, Sasa Grbic, Dimitris Metaxas, and Dorin Comaniciu. Automatic liver segmentation using an adversarial image-to-image network. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 507–515. Springer, 2017. (Cited on pages 4 and 13.)

[70] Qingsong Yang, Pingkun Yan, Yanbo Zhang, Hengyong Yu, Yongyi Shi, Xuanqin Mou, Mannudeep K Kalra, Yi Zhang, Ling Sun, and Ge Wang. Low-dose ct image denoising using a generative adversarial network with wasserstein distance and perceptual loss. *IEEE Transactions on Medical Imaging*, 37(6):1348–1357, 2018. (Cited on pages 11 and 26.)

[71] M. H. Yap, G. Pons, J. Martí, S. Ganau, M. Sentís, R. Zwiggelaar, A. K. Davison, and R. Martí. Automated breast ultrasound lesions detection using convolutional neural networks. *IEEE Journal of Biomedical and Health Informatics*, 22(4):1218–1226, July 2018. doi: 10.1109/JBHI.2017.2731873. (Cited on pages 10, 11, and 28.)

[72] Orestis Zachariadis, Andrea Teatini, Nitin Satpute, Juan Gómez-Luna, Onur Mutlu, Ole Jakob Elle, and Joaquín Olivares. Accelerating b-spline interpolation on gpus: Application to medical image registration. *Computer Methods and Programs in Biomedicine*, page 105431, 2020. ISSN 0169-2607. doi: https://doi.org/10.1016/j.cmpb.2020.105431. (Cited on page 97.)

[73] Kaihua Zhang, Huihui Song, and Lei Zhang. Active contours driven by local image fitting energy. *Pattern Recognition*, 43(4):1199–1206, 2010. (Cited on pages 13 and 99.)

[74] Y. Zhao, H. Li, S. Wan, A. Sekuboyina, X. Hu, G. Tetteh, M. Piraud, and B. Menze. Knowledge-aided convolutional neural network for small organ segmentation. *IEEE Journal of Biomedical and Health Informatics*, 23(4): 1363–1373, July 2019. doi: 10.1109/JBHI.2019.2891526. (Cited on page 10.)

[75] Wenzhong Zhu, Huanlong Jiang, Erli Wang, Yani Hou, Lidong Xian, and Joyati Debnath. X-ray image global enhancement algorithm in medical image classification. *Discrete & Continuous Dynamical Systems-Series S*, 12, 2019. (Cited on pages 4 and 11.)

[76] Shaojie Zhuo, Xiaopeng Zhang, Xiaoping Miao, and Terence Sim. Enhancing low light images using near infrared flash images. In *International Conference on Image Processing 2010*, pages 2537–2540. IEEE, 2010. (Cited on page 11.)

DECLARATION

I declare that I have developed this Doctoral Thesis, under the supervision of the thesis directors and, therefore, I take the authorship of everything described in this document.

I also declare that the work performed is neither plagiarism nor total or partial of any other investigation carried out by other persons.

I affirm that all the data exposed in this investigation have not been falsified and that any error that could exist in the document has not been introduced consciously.

Córdoba, Sept 2020

Nitin Satpute