

VESSEL SEGMENTATION
USING SHALLOW WATER EQUATIONS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

FATİH NAR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN
THE DEPARTMENT OF INFORMATION SYSTEMS

MARCH 2011

Approval of the Graduate School of Informatics

Prof.Dr.Nazife Baykal
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Doctor of Philosophy.

Prof.Dr. Yasemin Yardımcı
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Doctor of Philosophy.

Assist.Prof.Dr. Didem Gökçay
Supervisor

Examining Committee Members

Prof.Dr. Aydan Erkmen (METU, EE) _____

Assist.Prof.Dr. Didem Gökçay (METU, II) _____

Assist.Prof.Dr. Alptekin Temizel (METU, II) _____

Assist.Prof.Dr. Gözde Ünal (SABANCI UNIVERSITY, EE) _____

Prof.Dr. Yasemin Yardımcı (METU, II) _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Fatih Nar

Signature : _____

ABSTRACT

VESSEL SEGMENTATION USING SHALLOW WATER EQUATIONS

Nar, Fatih

Ph.D., Department of Information Systems

Supervisor: Assist.Prof. Dr. Didem Gökçay

March 2011, 118 pages

This thesis investigates the feasibility of using fluid flow as a deformable model for segmenting vessels in 2D and 3D medical images. Exploiting fluid flow in vessel segmentation is biologically plausible since vessels naturally provide the medium for blood transportation. Fluid flow can be used as a basis for powerful vessel segmentation because streaming fluid regions can merge and split providing topological adaptivity. In addition, the fluid can also flow through small gaps formed by imaging artifacts building connections between disconnected areas. In our study, due to their simplicity, parallelism, and low computational cost compared to other fluid simulation methods, linearized shallow water equations (LSWE) are used. The method developed herein is validated using synthetic data sets, two clinical datasets, and publicly available simulated datasets which contain Magnetic Resonance Angiography (MRA) images, Magnetic Resonance Venography (MRV) images and retinal angiography images. Depending on image size, one to two order of magnitude speed ups are obtained with developed parallel implementation using Nvidia Compute Unified Device Architecture (CUDA) compared to single-core and multi-core CPU implementation.

Keywords: Cerebral arterial system, Parallel Computing using GPU, Shallow Water Equations, Computational Fluid Dynamics, Vessel Segmentation

ÖZ

SIĞ SU DENKLEMLERİ KULLANARAK DAMAR BÖLÜTLEMESİ

Nar, Fatih
Doktora, Bilişim Sistemleri Bölümü
Supervisor: Yard. Doç. Dr. Didem Gökçay

Mart 2011, 118 sayfa

Bu tez sıvı akışını deforme olabilen bir model olarak kullanarak 2 boyutlu ve 3 boyutlu tıbbi görüntülerdeki damarları bölütleme yaklaşımını incelemektedir. Sıvı akışı kullanarak tıbbi görüntülerden damar bölütlemek damarların zaten kan naklini sağlayan dokular olmaları sebebiyle biyolojik olarak da akla yatkındır. Sıvı akışı ayrılabilir ve birleşebilir olması sebebi ile topolojik yapıya intibak edebilmektedir bu da sıvı akışını güçlü bir damar bölütleme yönteminin temeli olarak kullanmayı mümkün kılmaktadır. Buna ek olarak, sıvı akışı görüntüleme sınırlamaları ve hatalarından kaynaklı damar boşluklarını aşarak bağlantısı kopmuş bölgeleri birleştirebilmektedir. Sadeliği, paralel olarak hesaplanabilir yapısı ve diğer sıvı benzetim yöntemlerine kıyasla daha düşük olan hesaplama karmaşıklığı yüzünden sıvı akışı benzetimi için doğrusallaştırılmış sığ su denklemleri (LSWE) kullandık. Geliştirilen yöntem içinde manyetik tınlama anjiyografi (MRA), manyetik tınlama venografi (MRV), ve retina göz anjiyo görüntüler bulunan sentetik görüntüler, klinik görüntüler ve halka açık görüntüler kullanılarak doğrulanmıştır. Yöntem Nvidia Compute Unified Device Architecture (CUDA) üzerinde paralel olarak da geliştirilmiş ve görüntü büyüklüğüne bağlı olarak tek çekirdekli ve çok çekirdekli CPU ya kıyaslan on ile yüz kat arasında hızlanmalar sağlanmıştır.

Anahtar Kelimeler: Beyin Damar Sistemi, GPU ile Paralel Hesaplama, Sığ Su Denklemleri, Hesaplamalı Sıvı Akışkanları, Damar Bölütleme

This thesis is dedicated to my parents, brother,
and my wife Serap for their love, endless
support and encouragement.

ACKNOWLEDGMENTS

The author is profoundly thankful to his supervisor, Asst. Prof. Dr. Didem Gökçay who was extremely helpful and offered endless support and guidance with unabated patience for carrying out this thesis. Her momentum to the development of this study is ineffable. Her experience and supervision is the foundation block of this thesis.

Special thanks to Selim Nar, Gencer Berkman, Kemal Erbudak, Alireza Mazloumi Gavgani, Aydın Tahmasebifar, Murat Saran, Can Akkaya, Çağrı Hammuşoğlu and Saliha Hammuşoğlu for willing to be scanned by MRI scanner. The author feels in debt to many friends for their support; Selim Nar for his support, Devrim Toslak for retinal angiography images, Mazloumi Gavgani for proof reading, Erdal Yılmaz for his quick start to Nvidia CUDA, Murat Saran and Nurdan Saran for their collective efforts for learning Nvidia CUDA. The author is also grateful to Atilla Özgür, Nigar Şen, Habil Kalkan, Savaş Erişen, Yunus Çoban, Mesut Mert, Barbaros Can, Erol Özçelik, Cem Sermen, Emre Şener, Emre Kale, Erman Okman, Can Demirkesen, Nusrettin Güleç for their well-directed advices. Coming from a non-engineering background, author is also thankful to Yasemin Yardımcı, Fatoş Yarman Vural, Erol Şahin, Volkan Atalay, Ruşen Geçit, Aydan Erkmen, Kemal Leblebicioğlu, Mustafa Kuzuoğlu, Aydın Alatan, İlkay Ulusoy, and Müjdat Çetin for helping him create a sufficient theoretical background with their teachings and guidance.

The author expresses his boundless love and wholehearted appreciation to his wife Serap for her understanding and continual support. In the most painful moments of this study she gave necessary impetus which makes possible this study to be completed.

TABLE OF CONTENTS

ABSTRACT	iv
ÖZ	v
DEDICATION	vi
ACKNOWLEDGMENTS	vii
TABLE OF CONTENTS	viii
LIST OF TABLES	x
LIST OF FIGURES	xi
CHAPTER	1
1. INTRODUCTION	1
2. RELATED WORK AND BACKGROUND	5
2.1 Vessel Segmentation Methods	5
2.2 Fluid Flow Based Segmentation Methods	18
2.2.1 Charged Fluid Method	18
2.2.2 Segmentation Using Capillary Active Contour	19
2.2.3 Segmentation Using Water Flow	20
3. PROPOSED METHOD	22
3.1 Fluid Flow Based Segmentation Approach	22
3.2 Overview of Fluid Dynamics	23
3.2.1 Navier-Stokes Equations (NSE)	24
3.2.2 Fluid Simulation Methods	27
3.2.3 Lagrangian Approach versus Eulerian Approach	29
3.3 Shallow Water Equations	31
3.4 Stability of the Method	39
3.5 Proposed Vessel Segmentation Algorithm	40
3.6 Disconnected Vessel Systems	57
3.7 Calibration of Parameters	58
4. CPU AND GPU IMPLEMENTATIONS	61
4.1 CPU Implementation	61
4.2 CUDA Implementation	63
4.2.1 CUDA Overview	63

4.2.2 Jacobi Approximation to LSWE Solver	65
4.2.3 CPU versus CUDA Comparisons	68
5. RESULTS AND VALIDATION	69
5.1 Analysis of the Method	69
5.1.1 Vessel Shapes	69
5.1.2 Rough Segmentation	71
5.1.3 Water Shallowness	72
5.1.4 Time Step	72
5.1.5 Viscosity	75
5.1.6 Vaporization	78
5.1.7 Gaps in Vessels	79
5.1.8 RF Inhomogeneity	80
5.2 Segmentation Results	81
5.2.1 Brain MRI: BrainWeb Database	81
5.2.2 Brain MRA: 10 Volunteers / Clinical Datasets	84
5.2.3 Brain MRV: 10 Volunteers / Clinical Datasets	86
5.2.4 Retinal Angiography: DRIVE Database	88
6. CONCLUSIONS AND FUTURE WORK	89
BIBLIOGRAPHY	92
APPENDICES	104
A. LAGRANGIAN APPROACH	104
B. EULERIAN APPROACH	108
C. LATTICE-BOLTZMANN METHOD (LBM)	112
CURRICULUM VITAE	116
VITA	118

LIST OF TABLES

Table 1: Comparison of 3 fluid segmentation methods.	21
Table 2: Comparison of Lagrangian approach and Eulerian approach.....	31
Table 3: List of simulation parameters	48
Table 4: List of algorithm parameters.....	58
Table 5: Algorithm sub steps and implementation status.	66
Table 6: List of CUDA kernels used in proposed method.....	66
Table 7: Integration kernels thread structure and shared memory usage.....	67
Table 8: Example iteration times for i5 CPU (3.2 GHz) and GPU (GTX 465).....	68
Table 9: Segmentation successes for synthetic vessel structures.....	70
Table 10: Segmentation successes (Jaccard) for noisy synthetic vessel structures ...	70
Table 11: Segmentation successes (Jaccard) for different viscosity levels.....	76
Table 12: Segmentation successes (Jaccard) for different noise and gap size.....	79
Table 13: LSWE versus level set comparison on BrainWeb database	82
Table 14: Segmentation parameters for MRA volumes.....	84
Table 15: Statistics for observer study of 11 MRA volumes.....	85
Table 16: Segmentation parameters for MRV volumes.....	86
Table 17: Statistics for observer study of 11 MRV volumes.....	87
Table 18: LSWE results on DRIVE database	88

LIST OF FIGURES

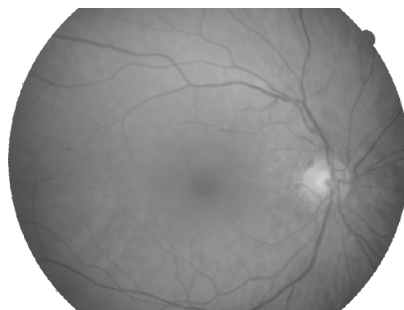
Figure 1: Example 2D and 3D medical images	2
Figure 2: Evolving level set curves.....	7
Figure 3: MIP view of the vessels and the filtered vessels	10
Figure 4: CTA image and zoomed parts of some vessel structures	11
Figure 5: Envelop of spheres that model a tubular surface.....	13
Figure 6: Region growing example.....	14
Figure 7: Inter-operator difference example	15
Figure 8: Retinal image with lesions and segmentation results	16
Figure 9: Propagating front of charged fluid.....	18
Figure 10: Segmentation results using irregularly shaped synthetic images	18
Figure 11: Capillary action.....	19
Figure 12: Water flow segmentation of low resolution retinal images.....	21
Figure 13: Example fluid simulation methods.....	28
Figure 14: Eulerian grid versus Lagrangian particles.	30
Figure 15: 2D height field representation	32
Figure 16: Shallow water representation.....	32
Figure 17: 1D SWE fluid evolution example.....	36
Figure 18: Importance of order of integration in 2D SWE fluid.....	37
Figure 19: 2D SWE fluid evolution example.....	38
Figure 20: Flowchart of proposed vessel segmentation algorithm.	40
Figure 21: 2D and 3D MRA examples.	41
Figure 22: 2D and 3D MRA example visualization.	41
Figure 23: Rough segmentation using Otsu method.....	42
Figure 24: Rough segmentation steps for 2D MIP image.....	43
Figure 25: Rough segmentation steps for a MRA volume.....	43
Figure 26: Rough segmentation steps for a retinal angiography image.....	44
Figure 27: LSWE segmentation example.	44
Figure 28: Water source detection using scale-space approach.....	45
Figure 29: Water source detection using filter-based approach.....	46
Figure 30: Slowly varying ground assumption and its affects.....	47
Figure 31: 2D image, water source, and fluid evolutions.	48
Figure 32: ADI method on 2D image with TDMA as solver	48
Figure 33: Effect of viscosity and vaporization schema.	49
Figure 34: Analysis of viscosity and vaporization schema.....	50
Figure 35: Fluid passing through a large gap.....	51

Figure 36: Error graph for 2D segmentation.....	52
Figure 37: Error graph for single water source using scale space approach.....	53
Figure 38: Error graph for water sources using filter-based approach.....	53
Figure 39: Error graph with two local minimums.....	53
Figure 40: Premature convergence example.....	55
Figure 41: Example error graphs.....	55
Figure 42: Post-processing example.....	56
Figure 43: Simulation steps.....	56
Figure 44: Multiple, four, execution with different water sources.....	57
Figure 45: Dynamic tracking of computational domain.....	62
Figure 46: 1-4 threads and CPU times during fluid simulation.....	62
Figure 47: Exponential performance growth of CUDA versus CPU.....	64
Figure 48: Error graph for TDMA and one step Jacobi.....	64
Figure 49: Shared memory usage.....	67
Figure 50: Various synthetic vessel structures.....	69
Figure 51: Rough segmentation and its effect on result.....	71
Figure 52: Segmentation success for different Δh values on BrainWeb database.....	72
Figure 53: Segmentation success for different Δt values on BrainWeb database.....	73
Figure 54: Synthetic 2D vessel structure on non-flat ground.....	73
Figure 55: Effect of large time steps.....	74
Figure 56: Vessel segmentation from retinal angiography image.....	74
Figure 57: Segmentation success for different viscosities on BrainWeb database.....	75
Figure 58: Segmentation results for different viscosity values.....	76
Figure 59: Image and its variance map.....	77
Figure 60: 2D image segmented with fixed and spatially varying viscosity.....	78
Figure 61: Segmentation success for vaporization schema on BrainWeb database.....	78
Figure 62: Synthetic vessels with; top row: small, bottom row: medium sized gaps.....	79
Figure 63: BrainWeb images with RF inhomogeneties.....	80
Figure 64: Segmentation success for different RF levels on BrainWeb database.....	80
Figure 65: Created simulated brain using BrainWeb vessel classes.....	81
Figure 66: Segmentation results from BrainWeb database.....	82
Figure 67: BrainWeb ground truth, LSWE and level set result – 3D rendered.....	83
Figure 68: BrainWeb ground truth, LSWE and level set result – 2D slices.....	83
Figure 69: LSWE segmentation results of 9 MRAs.....	84
Figure 70: Chan-Vese level set segmentation results of 3 MRAs.....	85
Figure 71: LSWE segmentation results of 9 MRVs.....	86
Figure 72: Chan-Vese level set segmentation results of 6 MRVs.....	87
Figure 73: Segmentation results from DRIVE retinal angiography database.....	88
Figure 74: MAC Staggered Grid.....	108
Figure 75: Flowchart of the simulation algorithm.....	110
Figure 76: Detailed steps of the simulation algorithm.....	111
Figure 77: D3Q19 and D2Q9 LBM models and corresponding velocities.....	113
Figure 78: DFs of a grid cell before and after streaming.....	114

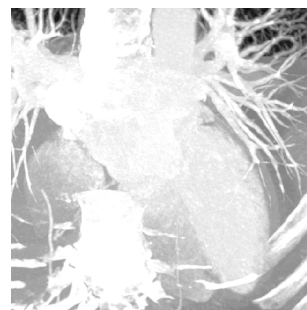
CHAPTER 1

INTRODUCTION

Segmenting vessel structures from medical images such as MRA, MRV, CTA, and retinal angiography is useful for extracting information such as vessel width, vessel branching, and making atlas comparisons, which in turn, can be used for reliable diagnosis and treatment of many diseases. Automating this segmentation process is useful especially for 3-Dimensional (3D) volumetric images or 2-Dimensional (2D) images with large number and sheer size. However, automatic and precise segmentation is a very challenging task due to several reasons such as: anatomic complexity and variability, low contrast between vessels and other tissue, existence of thin vessel structures, and discontinuities in vessel boundaries caused by imaging artifacts such as intensity inhomogeneities and imaging noise. These medical images can be 2-Dimensional (Figure 1.a) or 3-Dimensional (Figure 1.b, Figure 1.c, Figure 1.d), in various resolutions, where vessels are mostly defined as bright regions (Figure 1.b-d, MIP: maximum intensity projection of volumes). Vessel intensities can be negated into dark regions just by inverting the intensities.



(a) 2D retinal angiography



(b) MIP view of coronary CTA

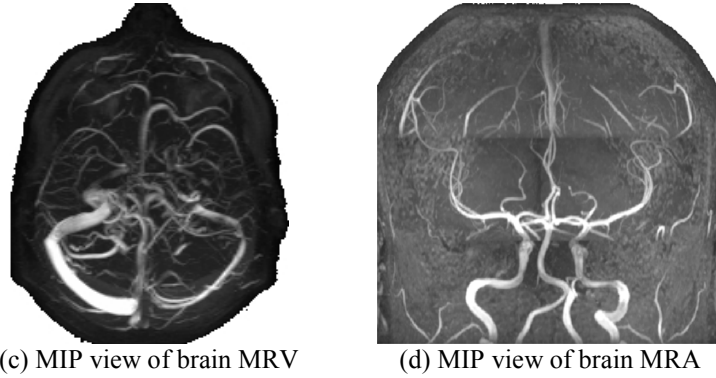


Figure 1: Example 2D and 3D medical images

Segmentation of vessel structures in medical images is important for reliable diagnosis, and treatment planning of many diseases. Therefore there is a vast literature on vessel segmentation dealing with various parts of the human body using different imaging modalities. In this study, a deformable model-based segmentation approach is developed to simulate fluid flow. Using fluid flow for segmenting vessels in medical images is biologically plausible since vessels are already being utilized for transporting blood. The natural flow process can merge gaps and split across branches achieving topological adaptivity. Small gaps formed by imaging artifacts or other data acquisition limitations can be eliminated while preserving connectivity. Hence our main motivation in developing this automatic vessel segmentation was based on the utilization of the biological foundations within which vessels operate.

In order to implement fluid flow, we used linearized shallow water equations (LSWE) (Kass and Miller 1990). Our main reasoning for using LSWE for fluid simulation is its simplicity, computational efficiency and convenience for parallel implementation. LSWE simulations can be used to segment medical images such as retinal angiography, MRV, MRA, and CTA. Within the scope of this thesis, a turn-key approach to vessel segmentation is presented, including pre-processing, initialization steps, fluid flow using LSWE, convergence, and post-processing. Validation and comparison with other methods are shown using synthetic, clinic,

BrainWeb, DRIVE, and Rotterdam Coronary Artery databases containing retinal angiography, MRV, MRA, and CTA images. Computational efficiency and parallel implementation using Nvidia CUDA with necessary numerical approximations are discussed. Parameters of the developed method and their usage are specified.

In the literature, there exist vessel segmentation methods using deformable physical models like fluid flow (Chang and Valentino 2006, Yan and Kassim 2006, Liu and Nixon 2007). However, we did not encounter any research utilizing Navier-Stokes Equations (NSE) to simulate fluid flow for image segmentation purposes. LSWE is preferred as a realistic approximation to NSEs to simulate blood flow for segmentation. LSWE can capture most of the fluid effects valid for the 3d closed space of the vessels, while it is computationally much efficient compared to other fluid simulation methods. Although LSWE is just an approximation, it still possesses many powerful aspects of NSEs such as topological adaptivity, control of smoothness of fluid flow using artificial viscosity, and physical realism for shallow water systems. Unlike NSEs, LSWE is relatively easy to implement, computationally more efficient, and allows parallel implementation.

Our contributions can be summarized as follows:

1. The LSWE approach is implemented for segmentation of 3D volume data for the first time. 3D LSWE is not intuitive and hard to visualize, but mathematical foundations allow for an extension to 3D as presented in Chapter 3.
2. The numerical solver of LSWE is parallelized using the Jacobi method. While a single CPU implementation takes around 1-3 minutes to segment 2D images, and 1-2 hours to segment 3D volumes, speed can further be improved up to one to two orders of magnitude using Nvidia CUDA technology.
3. An efficient data processing pipeline for automatic segmentation is generated, including pre-processing, fluid evolution, convergence, and post-processing steps for achieving best segmentation performance in shortest time.

4. Applications and validation on retinal vessels, brain vessels are provided using DRIVE, BrainWeb, and clinical databases. Optimal parameter settings for these applications are obtained.

The outline of this thesis is as follows: Chapter 2 starts with the literature survey of existing vessel segmentation methods and general issues in vessel segmentation. This is followed by a brief overview of fluid based segmentation methods. Chapter 3 explains the proposed method and its sub steps and chapter 4 shows the results of proposed method using various case scenarios along with validation through standard medical image databases. Conclusions, as well as potential future works, and possible research directions are provided in Chapter 5.

CHAPTER 2

RELATED WORK AND BACKGROUND

This chapter presents a brief overview of the vessel segmentation methods as well as the level set method, and vessel enhancement.

2.1 Vessel Segmentation Methods

The segmentation of vascular structures in various parts of the human body has been the focus of scientific research because of their importance for human health. Diagnosis, surgery and treatment planning can be counted as the main applications of vessel detection (Lesage et al. 2009). Some example applications include detection of coronary artery disease which is a leading cause of death according to World Health Organization (WHO 2008). In cancer treatments, it is important to know vascular structure of the targeted area and plan the treatment accordingly (Esneault 2010). Hepatocellular Carcinoma occurs in liver tissues and it is the third most common cause of death from cancer (Melanie 2005). The radiofrequency treatment of the disease involves the heating of the target area while the nearby blood vessels act as coolers. Segmented vessel structures can also be useful to address the patient-specific modeling of vessel structures and corresponding blood flow to investigate vascular disease characterization and assessment (Antiga 2003). The detection of nodules in the lung has clinical importance and their segmentation is required in various medical imaging applications (Agam et al. 2005).

Vessel segmentation is very important yet an ill-posed problem which is still an active research area. Hundreds of research papers are reviewed in Kirbas and Quek (2003), Lesage et al. (2009), and Radaelli et al. (2010).

The modalities in which vascular networks can be observed have increased in number to include three-dimensional rotational angiography (3DRA), computerized tomography angiography (CTA), magnetic resonance angiography (MRA), magnetic resonance venography (MRV) and ultrasound imaging (Radaelli et al. 2010). Segmenting vessel structures from medical images is useful for extracting information such as vessel shape, vessel width, vessel branching, and performing atlas comparisons which can be useful for reliable diagnosing and treatment of many diseases (Chung 2006). With the advances made in medical imaging, it is possible to obtain high resolution volumetric data of the parts of the human body with these imaging modalities. However, intensive human labor is required for the manual segmentation of these large amounts of data. The automatic and semi-automatic methods and tools aid the operator in this regard. However automatic and precise segmentation is a very challenging task due to anatomic complexity and variability and discontinuities in vessel boundaries caused by imaging artifacts such as intensity inhomogeneities and imaging noise (Lesage et al. 2009).

The vessel segmentation methods proposed in several studies can be categorized into: pattern recognition techniques, model-based approaches, tracking-based approaches, artificial intelligence-based approaches, neural network-based approaches, miscellaneous tube-like object detection approaches, and hybrid methods (Kirbas and Quek 2003). For example region growing methods aim to start with a seed point and spread to other pixels according to a determined criterion and skeleton-based methods try to extract the centerline of the vessels to obtain a network or to initiate segmentation by other means (Kirbas and Quek 2003). Active contours have also been used widely (Lesage et al. 2009). Active contour methods aim to fit deformable curves according to internal and external forces that are usually based on

image gradient (Radaelli et al. 2010). Level set methods are deformable models which evolve a curve or surface (Figure 2) driven by a partial differential equation hence considered under the active contour heading. The curve is embedded as a zero level set of a higher dimensional surface (Osher and Sethian 1988).

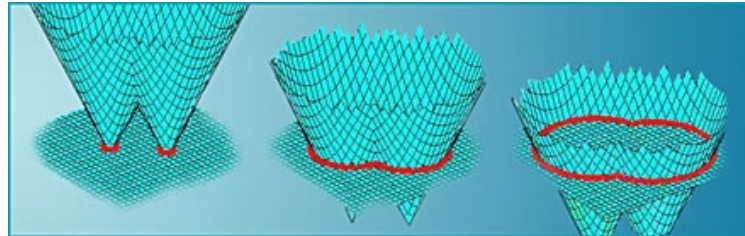


Figure 2: Evolving level set curves (Osher and Sethian 1985).

Level set is front propagation method which is first proposed by Osher and Sethian (Osher and Sethian 1988) where the entire surface (front) can be evolved into complex topologies of objects. A computationally efficient solver for the level set curve evolution is proposed by Sethian which is known as fast marching methods (Sethian 1996). Level set methods are used in computer vision, machine intelligence, computer graphics, computational physics and other areas (Malladi 1995, Osher and Fedkiw 2001). Chan-Vese model is introduced as active contours without edges (Chan and Vese 2001). Strzelecki et. al. (2009) analyzed level set segmentation methods in simulated vessel structures with different noise levels and concluded that segmentation results are good even for noisy images (Strzelecki 2009).

Active contours are also among the most popular approaches for vascular segmentation (Lesage et al. 2009). Holtzman Gazit et al. (2006) used a variational method which employs a functional with an edge-based term, a minimal variance term and a geodesic active contour term. The geodesic active contour term is implemented for regularization. Level set formulation is used for surface evolution. The study also included a hierarchical approach in order to extract multiple objects. This is achieved by quantizing the image with tree structured vector quantization. Geodesic active contour (GAC) is a front propagation model that depends on curvature and it is used in other studies as well. Gooya et al. (2008) introduced a

level set based geometric regularization method which is based on GACs. They proposed a modification to the CURVES regularization. The modification aims to preserve narrow structures in retinal angiogram and MRA images. Another regularization scheme named flux maximizing geometric flow is also modified in the mentioned study. Large vascular structures like arteries are in close proximity of smaller vessels. Their separation was posed as a problem and a level set based solution is suggested (Van Bemmelen et al. 2003). The central axes of the vessels are labeled with minimal user interaction and the vessels are enhanced prior to a level set based segmentation. The method propagates artery and vessel front simultaneously and decide pixel labeling by the arrival times of each front. Hao and Shen (2006) used watershed algorithm to segment vessel boundaries in order to use them as the initialization of the level set method. This was done to reduce computational complexity and achieve close to real-time interaction.

Pre-processing and noise removal are initial steps in many vessel segmentation techniques because sensor noise is unavoidable in medical imaging and it results in low image quality and artifacts (Radaelli 2010). Echevarria et al. (2004) proposed to use level set method in the noise removal process. Afterwards, they performed segmentation using *fast marching* and *narrow band* level set methods. The seed points required for Fast Marching are selected by double thresholding. A study based on level sets introduced a local term which uses a locally contrast enhanced image (Sum and Cheung 2008). This approach aims to suppress the effects of uneven illumination and is claimed to decrease disconnectivity in low contrast areas. Another problem with medical imaging is that different imaging systems even with similar modalities result in images with different intensity profiles. This problem is also present for images taken from the same system with varying parameters. Studies were made to compensate for this fact. A semi-automatic segmentation scheme in which the user selects the start and end points of the vessel is proposed (Koning et al. 2003). Waves are propagated from the start point with fast marching level set method. In order to account for the intensity level changes in different images, the

speed of the propagation is obtained by an iterative process. The vessel line is then extracted by backtracking on the wave propagation results. The vessel is segmented by model matching based on NURBS¹ surfaces as a final step. In another application of the level set methods Deschamps et al. (2004) simulated the blood flow in the vessels as well as segmenting them. The segmentation is initialized with the Fast Marching method and finalized by level set.

Pre-processing of the images before applying a segmentation method is a frequently employed practice in vascular extraction. In a recent study, vessels are pre-estimated by blurring the original image and subtracting the blurred image from the original one (Montes et al. 2008). The pre-estimation mask is then obtained by thresholding the resulting image. The initial vessel region estimation is completed by inverting and eroding this mask. Then edge detection applied to the initial vessel region estimation and the Sobel kernel applied to the original image are combined with weighted summation. This process is performed to obtain an estimation of the exterior of the vessels. Active contour is used on this estimation to complete the segmentation of the vascular structure.

Enhancing the vascular images for easier segmentation is undertaken by various other studies as well. Down-sampling, quantization, image improvement by filtering, anisotropic diffusion to decrease noise and preserve vessels and level sets are among the methods employed to obtain an image with better vascular structure (Lesage et al. 2009). Manniesing et al. (2006) proposed to enhance the vessels in images by using an anisotropic diffusion scheme which uses a tensor designed to increase isotropic diffusion for non-vessel structures and maximize diffusion in the vessel direction for vessel structures (Figure 3).

¹ Non-uniform rational B-spline

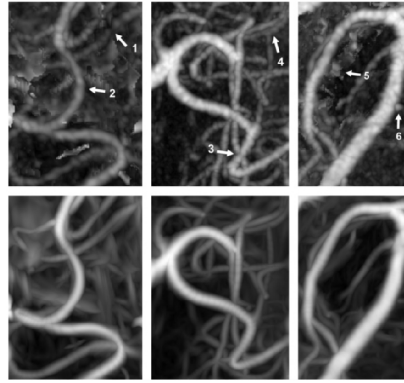


Figure 3: MIP view of the vessels (top row) and the filtered vessels (bottom row) (Manniesing et al. 2006).

Zonoobi et al. (2009) aimed to draw attention to the fact that thin vessels appear darker than thick vessels in images and the gradient magnitudes around them are lower. Thin and small structures are also subject to partial volume effect to higher degree. In order to gain more understanding of the intensity levels of thin structures they explored the boundary gradient versus vessel thickness relationship. The relationship was found to be an exponential one. It is proposed to use an equalizer function which compensates for the exponential. Therefore an equalizer function which compensates for the exponential relationship is suggested. This enables methods which use gradients for vessel segmentation to receive reasonably high values even if the actual gradient image is low. Wavelets are also considered for vessel enhancement. A 2-D Gabor wavelet and sharpening filter in conjunction enhances the blood vessels in retinal images (Akram et al. 2009). The vessels then can be extracted from the enhanced images by Canny edge detector. The segmentation results are fine-tuned by applying morphological dilation operator. Bauer and Bischof (2008) proposed to use an anisotropic diffusion process when calculating the gradients in the detection of tubular structures. Gradient Vector Flow is used which is robust to noise and preserves weak structures. Gradient Vector Flow is combined with Frangi's vesselness measure. The result of the combination is then turned into vessel centerlines by thresholding. The noise in the image and the large intensity changes along vessels are shown in zoomed regions of a CT angiography image taken from this study in Figure 4.

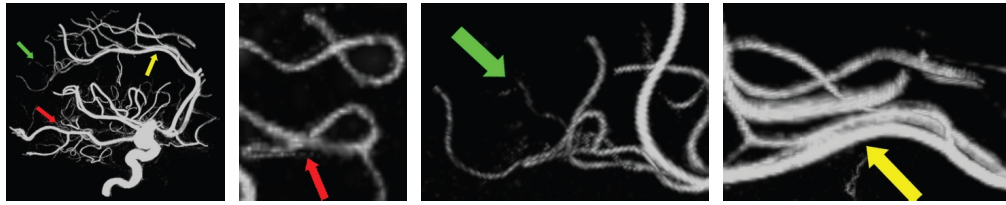


Figure 4: CTA image and zoomed parts of some vessel structures (Frangi 1998).

Extensive research has been performed to extract the centerlines of the vessels to obtain a vascular network or to use as an aid for segmentation. Centerline tracking is a semi-automatic approach that aims to track the centerline along the vessel starting from a manually entered seed point. Staal et al. (2004) extracted vessel centerline by Hessian matrix based ridge detection in retinal images. The centerlines are obtained as line elements and the remaining pixels in the image are partitioned according to their distances to the line elements. The line elements have their own local frames and features for the pixels are calculated according the frame to which they belong. The features are classified with a kNN classifier and the labeling of the image is completed. Centerline tracking is a semi-automatic approach that aims to track the centerline along the vessel starting from a manually entered seed point. Friman et al. (2008) suggested defining a vessel template function and fitting it to the image. The vessels are then detected with multiple hypotheses tracking. Mendonça et al. (2006) preprocessed retinal images by subtracting an estimate of the background from the monochromatic representation of the image. Candidate points for vessel centerlines are selected from the preprocessed image by applying multiple first-order derivative filters oriented in different directions. The seed points are then selected from the candidate points according to the filter responses. Region growing is applied at the seed points to connect them and form the centerlines. As the final step, vessels are enhanced and reconstructed by using a morphological operator. Vessel segmentation is also performed on ultrasound images (Guerrero et al. 2007). From a manually entered seed point and with the use of Kalman filter and Star algorithm, vessel edges are detected. The seed point is tracked in successive images in time with a temporal Kalman filter.

Another approach employed in vessel segmentation is fitting *tubular or line structures* to the image in order to locate vessels. Esneault and Lafon (2010) studied on liver vessel segmentation through graph-cuts. The graph-cut segmentation is originally a semi-automatic method in which a user is required to supply seed points inside and outside the regions to be segmented. But Esneault and Lafon (2010) introduced a term to the algorithm using a tubular structure detection process for full automation.

It is also possible to use line detectors to find vessel-like structures (Ricci and Perfetti 2007). The intensities are evaluated along twelve different orientations. When aligned with a vessel, this oriented line detector yields larger results. The line strength obtained by this method and the pixel intensity are used as features for a linear support vector machine with supervised training. Rohr and Wörz (2006) tested three intensity models to extract tubular structures from images. 3D Gaussian and phi lines² are tested along with the 3D cylinder model which is the most complex among these three models. Wörz and Rohr (2007) proposed to use Gaussian smoothed 3D cylinder in another study: 2D cross section is not modeled well enough with Gaussians because vessels are plateau-like, prohibiting an analytical expression through a cylinder. Therefore approximations are used. The approximate cylinder model is fit to the image intensities with an incremental process based on a Kalman filter. Li and Yezzi (2007) claimed that global minimal path methods which find the global minimum curve of a contour dependent energy function between two points yield a path which is not always the vessel center and that the tubular structure fitting to images only yield the vessel centerlines. They proposed to represent 3D vessel surfaces as a 4D curve in order to benefit from the minimal path algorithm in 4D. Tubular structures are defined as an envelope of a one parameter curve of spheres with different centers and radii. The added dimension of the sphere radii helps to find the vessel boundaries on top of vessel centerlines. A sample envelop of family of spheres is shown in Figure 5 (Li and Yezzi 2007).

² Two geometrical models for tubular structures

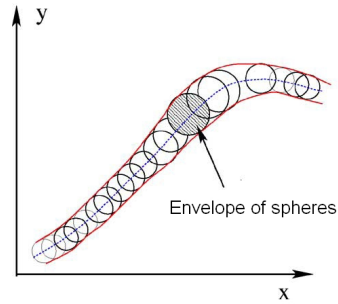


Figure 5: Envelop of spheres that model a tubular surface (Li and Yezzi 2007).

Region growing schemes have been popular in vessel segmentation although it has been noted in some studies that a global region growing is not effective in vascular segmentation due to the variety of structures to be segmented (Yi and Ra 2003, Lesage et al. 2009). In Tizon and Smedby (2002) the seeds for region growing are placed manually by a user. A fuzzy connectedness map according to the seeds is calculated and the image is divided up to regions of interest according to the seed points. Each voxel is then marked with the maximum connectedness among all seeds. Perez et al. (2010) aimed to parallelize an algorithm based on feature extraction and region growing. Features such as gradient magnitude, maximum eigenvalues are calculated at each pixel. The region growing parameters are obtained from the extracted features and seed points are placed automatically using a maximum eigenvalues criterion. The parallelization for feature extraction is straight forward although it is not for region growing. In order to parallelize the region growing process, it is performed along the horizontal and vertical directions separately and then combined to find the segmentation. They have reported that the parallelization speeds up the process by 8 to 10 times. Atlas based approaches have also been used in combination with region growing (Passat et al. 2005). An atlas of the head is constructed with 12 regions with different local vascular properties. The image to be segmented is registered to the image of a healthy patient that has an atlas attached to it. Each region uses different thresholds for the region growing process. Yi and Ra (2003) performed segmentation in a local cube based on a selected seed point. The segmentation in the cube is achieved by a locally adaptive region growing scheme. The vessel exit points from the local cubes are used for the selection of the

successive cubes that are to be segmented. The segmentation of the whole vascular structure is achieved by dividing the global space into cubes and connecting them through the vessel exit points. The main idea behind this scheme is to have robustness under large intensity changes along vessels. Sekiguchi et al. (2005) also proposed using a branch-based local region growing scheme. When multiple branches are reached during the segmentation, region growing is allowed to continue only towards a single branch at one time. The growing conditions are dynamic and they are modified according to the intensity of the local area. Comparative segmentation results obtained by using a conventional region growing method and the branch-based local region growing scheme are given in Figure 6.

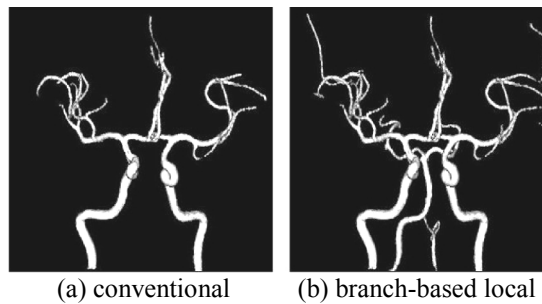


Figure 6: Region growing example (Sekiguchi et al. 2005)

The unsegmented thin vessels in the conventional method show the limitation of the conventional methods. The evaluation of the segmentation is slightly different than most studies and it is performed by comparing manually segmented MIP image and the projected 3D segmentation into 2D. The evaluation of the segmentation algorithms are usually done by running the method on synthetic images where the exact labeling can be known or comparing the results to manually segmented images. In order to have a more objective comparison scheme, databases such as DRIVE (Staal et al 2004) and STARE (Hoover et al 2002) are built by manually segmenting several vascular images. With this attempt, it is aimed that researchers are able to compare their segmentation results. However, having multiple modalities and multiple body regions with vascular structures, this comparison becomes problematic, because it is almost impossible to have a generic vessel extraction method applicable to every possible scenario (Kirbas and Quek 2003). It should also

be noted that even manual segmentations of vascular structures (i.e Figure 7) are not 100% accurate as they usually have inter-operator differences. Even same operator can produce different segmentation of the same image at different times which is called as intra-operator differences.

Several types of *model matching* are employed in the vascular segmentation such as superellipsoid models for the detection of vessels (Tyrrell 2007). The parameters of the superellipsoid model include information such as boundary and centerline. The parameters are approximated by using a maximum likelihood approach. The local ellipsoid model is also used for tracking on the vessel network.

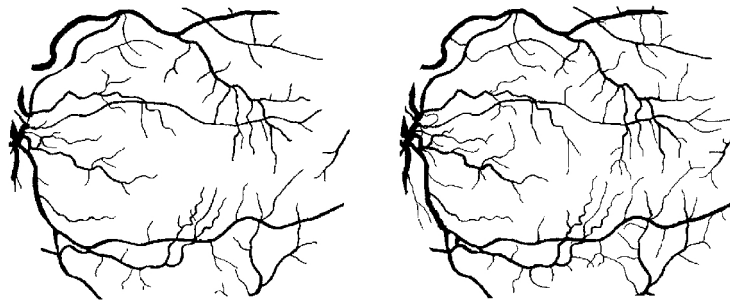


Figure 7: Inter-operator difference example (Hoover et al. 2000)

Mahadevan et al. (2004) investigated three methods for vessel segmentation. They are Huber's censored likelihood ratio test, a rank based algorithm and robust model selection algorithm. They concluded that the selection of the algorithm depends on the tolerance to missing a vessel and the required minimum level of false positives.

Bullitt et al. (2003) studied the properties of different vessel populations. The vessel structures located in different parts of the head such as anterior cerebral, right and left middle cerebral and posterior cerebral circulation sub systems. Features like vessel number, average radius of the vessel, how frequent branching occurs are extracted for these regions. It is concluded that these features show varying results for these vessel groups and they could be helpful in the local segmentation of the regions. Another study that uses local properties is presented by Hoover et al. (2002). Retinal images are convolved with twelve 16x16 kernels and a matched filter is selected as the value for the highest scoring kernel. The matched filter response

image is then thresholded to yield the connected pixels. Lupaşcu et al. (2010) extracted a 41 dimensional feature vector from retinal images in order to classify pixels as vessel or non-vessel. The features include region and boundary-related features, model-based vessel likelihood, Frangi vesselness, Lindeberg ridges, Staal ridges, Gabor wavelet transform and second-order detectors. They trained an AdaBoost classifier with the extracted features from several images.

Lam and Yan (2008) studied the segmentation of pathological retinal images. Normalized Gradient Vector Field is used in order to locate vessel centerlines and to detect blood vessel-like objects. Falsely marked pixels as vessels are cleared according to their distances from the centerlines. This cleaning process is performed to remove false vessel detection near pathologies. Lam et al. (2010) stated that it is harder to extract vessels from retinal images in the presence of bright and dark lesions. They claim that most methods do not perform well when pathology is involved in the retinal images. As a solution they proposed a line-shaped concavity measure to remove dark lesions. This measure is locally normalized to offset the effects of the uneven distribution of noise. A retinal image with dark lesions and their negative effects on segmentation methods are shown in Figure 8.

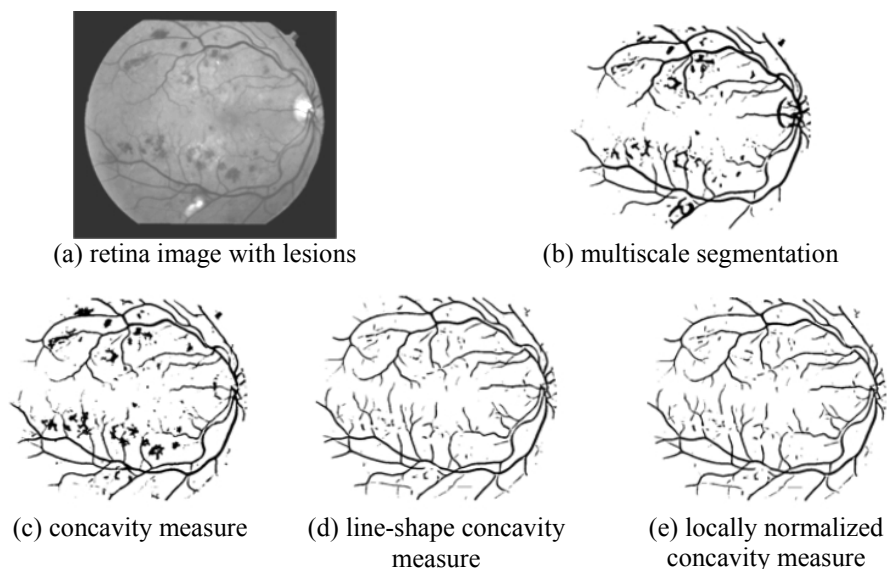


Figure 8: Retinal image with lesions and segmentation results (Lam et al. 2010)

All these vessel segmentation algorithms can be categorized using below criteria:

- Target image modalities that method works on (retinal angiography, MRA etc.)
- Dimensionality of image (2D image or 3D volumetric image)
- Ability to find thick and smooth structures
- Ability to find thin and low contrast structures
- Ability to deal with images containing pathology
- Robustness to imaging artifacts such as noise or gaps
- Topological adaptivity
- Ability to impose prior information
- Reproducibility
- Need of pre-processing
- Need of post-processing
- Number of parameters affecting the solution and ease of parameter tuning
- Amount of user interaction (manual, semi-automatic, automatic)
- Flexibility of method to adapt for different problems

In addition, these algorithms can be evaluated with respect to the following performance criteria:

- Computational complexity of the method
- Convenience to parallel implementation
- Memory requirements
- Simplicity of method (mathematical, algorithmic, and implementation aspects)

Since the method proposed in this thesis is centered on fluid flow, we focused our attention on the few methods that attempt to perform vessel segmentation using fluid flow. In general, these methods use simple approximations for modeling fluid flow or derive fluid analogies without utilizing a realistic physical model. In the following section, we provide a brief description of these methods, as well as their categorization according to the criteria listed above.

2.2 Fluid Flow Based Segmentation Methods

2.2.1 Charged Fluid Method

Chang and Valentino used charged fluid method for image segmentation (Chang and Valentino 2006). In this method charged elements which apply repelling electric forces to each other form a charged fluid (Figure 9). Since charged fluid flows through and around obstacles Chang and Valentino assumes that it is acting like liquid even though their method ignores various fluid properties such as pressure, surface tension and viscosity.

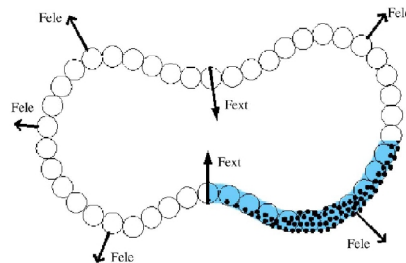


Figure 9: Propagating front of charged fluid (Chang and Valentino 2006).

Propagating front of charged fluid forms the segmented object boundary at the high values of image gradients. Charged fluid evolves in two steps according to the solution of Poisson's equation. They used finite-size particle method and fast Fourier transform (FFT) to solve Poisson's equation efficiently. In the first step, elements of charged fluid within the front distributed until reaching to electrostatic equilibrium state. In second step front is propagated in the conformance with image potential and equilibrium electric potential. These two steps are repeated until the propagating front converges to the final segmentation interface.

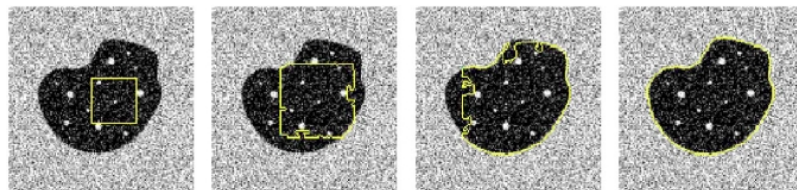


Figure 10: Segmentation results using irregularly shaped synthetic images (Chang and Valentino 2006).

Charged fluid method is limited to closed boundary segmentation problems. Due to its fluid like nature initial point must be inside the object but more than one initial point can be placed (Figure 10). Method does not support splitting but merge of regions are possible hence method is semi-topologically-adaptive. If there is no strong edge in the image over-segmentations can occur. Chang and Valentino did not mention the performance of method in thin and low contrast structures.

2.2.2 Segmentation Using Capillary Active Contour

Yan and Kassim proposed the use of capillary action to segment thin vessel structures especially having low contrast (Yan and Kassim 2006).

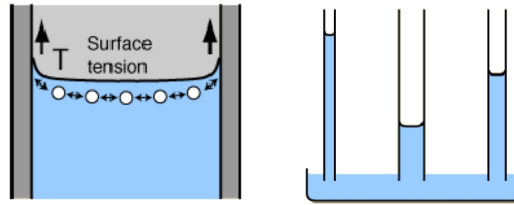


Figure 11: Capillary action (Yan and Kassim 2006).

Adhesion and surface tension causes the capillary action. Dissimilar molecules are attracted to each other being potentially in direct contact which is called adhesion. Adhesion causes fluid to make an upward motion in the vessel for the vessel having small radius. Surface of the fluid resists to the external force which is called surface tension. Adhesion causes movement in edges but surface tension causes surface to be smooth and intact so the whole liquid surface is pulled upward (Figure 11).

Yan and Kassim defined capillary active contour method through the minimization of surface energy function. They used variational calculus with various assumptions to find overall equation describing the energy of wetting surfaces as S_f .

$$\vec{S}_f = g(\kappa + c)\vec{\eta} - (\nabla g \cdot \vec{\eta})\vec{\eta} + \alpha(1 + \lambda\hat{\kappa}_2)(\vec{\eta} - \frac{\nabla g}{|\nabla g|} \cos \theta) \quad (\text{Equation 1})$$

where α is the relative adhesion coefficient, the constants c and λ control the forces which enable evolving surfaces similar to balloon to attach solid boundaries, $\hat{\kappa}_2$ and $\vec{\eta}$ are the Euclidean curvature and the unit normal vector of the contact line in 2D, ∇g is image gradient and θ is direction of contour gradient. (Equation 1) is solved using level set methods where the surface of the fluid is defined as the zero level set. Employed level set method is able to adapt topological changes while handling sharp corners and cusps. Yan and Kassim claim that capillary active contour is able to evolve to the topologically complex vasculature (Yan and Kassim 2006).

2.2.3 Segmentation Using Water Flow

Liu and Nixon proposed to ignore turbulence and viscosity via force field generation from image and using flow attributes such as pressure, surface tension and capillary actions. Such image forces are mainly edge forces and also region-based forces which has more capture range with high accuracy. Liu and Nixon claim that such model will be adaptable to topological changes (Liu and Nixon 2007). Total driving force and resistance of the flow determines the flow velocity.

$$v = F_D / (A \cdot R) \quad \text{(Equation 2)}$$

where v is the velocity, R is the flow resistance, F is the total driving force, and A is the area where F is applied to. Liu and Nixon used Mumford-Shah functional to adapt (Equation 2) with corresponding driving forces and flow resistance. In their formulation one pixel in the image is defined as basic water element. Water sources are placed adaptively hence water flows in the system until water sources are closed. Authors claim that water flow performs better comparing to level set for images having severe level of noise. Robustness to noise is very critical especially for medical image segmentation and water flow seems have this feature.

Retinal angiography images contain irregular and complex vessel structures. Water flow method seems immune to noise and works well for thick and smooth structures

but likely to discard thin and low contrast structures (Figure 12). Water flow method also requires multiple initializations and post-processing for gap-linking.

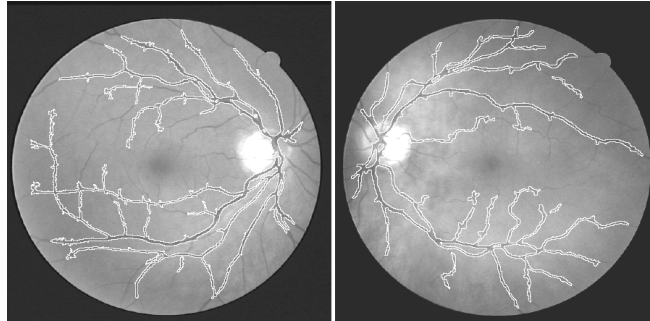


Figure 12: Water flow segmentation of low resolution retinal images (Liu and Nixon 2007).

Based on several criteria, the success, applicability and performance of these methods differ. These are listed below, for comparison purposes.

Table 1: Comparison of 3 fluid segmentation methods.

	Charged fluid method	Capillary active contour	Water flow
Image modalities	All	All	All
Dimensionality	Not specified	2D / 3D	2D / 3D
Smooth structures	Yes	Moderate	Moderate
Thin structures	Weak	Good	Weak
Noise immunity	Moderate	Weak	Good
Topological adaptivity	Limited	Yes	Yes
Ability to use prior information	Yes	Yes	Difficult
Reproducibility	Yes	Yes	Yes
Pre-processing	None	None	None
Post-processing	None	None	Gap-linking
Ease of parameter tuning	Moderate	Moderate	Moderate
User interaction	Initialization	None	Initialization
Flexibility	Flexible	Flexible	Moderate
Computational complexity	Moderate	Low	Low
Parallelism	Low	Low	Low
Memory requirements	Moderate	Low	Low
Simplicity of model	Complex	Moderate	Moderate
Ease of implementation	Difficult	Moderate	Moderate

CHAPTER 3

PROPOSED METHOD

This chapter explains the proposed method and its sub-steps such as initialization, pre-processing, rough segmentation, water source detection, fluid evolution, convergence criteria, and post-processing.

3.1 Fluid Flow Based Segmentation Approach

Using fluid flow for segmentation of vessel is biologically and physically plausible since vessels transport the necessary nutrients to tissues through the bloodstream.

Fluid flow can be used as a basis of powerful vessel segmentation algorithm because;

- a. Fluid flow can capture both sharp and smooth features of images such as thick and thin vessels with low contrast,
- b. Fluid flow is topologically adaptive therefore regions can merge or split,
- c. Smoothness of curve evolution can be controlled by viscosity parameter,
- d. Fluid flow can pass through small gaps in images (e.g. disconnected vessels due to imaging artifacts),
- e. Visualization of fluid has a practical merit (Chang and Valentino 2006, Yan and Kassim 2006, Liu and Nixon 2007).

Although various methods can be utilized to simulate fluid flow, we preferred Linearized Shallow Water Equations (LSWE) for our vessel segmentation method because (Shinbrot 1970, Kass and Miller 1990, Beneš 2007, Thürey et al 2007);

1. Solving LSWE numerically is computationally efficient,
2. Parallel implementation is possible using multiple thread CPU (e.g. OpenMP, Parallel Computing Toolbox of MATLAB) or streaming processors such as GPU (e.g. CUDA, OpenCL),
3. LSWE is relatively easy to understand and implement.

SWE and its variation LSWE cannot capture all fluid effects (Kass and Miller 1990) but these effects are not crucial for image segmentation purposes. For example, there cannot be any splashing in a circulatory system within vessels. Vessels are deformable tissues up to some limits but since we are trying to find vessels in a scanned image we can assume vessels as solid objects. Therefore complex interface tracking and interaction between blood flow and deformable vessels can be ignored. Blood flows turbulently so that all nutrients in blood can be transported to vessel boundary and tissues afterwards. Since our purpose is to simulate fluid flow for segmentation purposes, turbulent flow can be ignored.

3.2 Overview of Fluid Dynamics

Fluids, liquids and gases are materials which continually deform when *shear stress* is applied regardless of how small the applied stress is. Force per unit area is defined as *stress* which is a measure of the strength of internal forces formed as a reaction to externally applied forces and body forces (e.g. gravity). Stress is formed by its *shear* tangential and *normal* pieces where each has unparalleled physical significance. In short, stress is the force per unit area on a material that causes deformation called as *strain* (e.g. elastic solid, fluid). Fluids are either resistant to deformation, or show slight resistance according to their *viscosity* where high viscosity means more

resistance, (e.g. water has low viscosity – similar to low friction in solids). They can also have the form of their container due to their ability to flow. Unlike gases, which move freely in their container, liquids form a free surface unrelated to their container. Fluids and solids can be discriminated by determining the viscosity of the substance, yet it may not be completely obvious. For example *viscoelastic* materials show both fluid and solid elastic properties like toothpaste. Ideal fluids can only be subjected to *pressure* (force per unit area) which is normal, compressive stress. Real fluids can be subjected to shear stress, in low levels, due to their viscosity.

Fluid can be categorized as Newtonian where strain rate and stress has direct proportionality or Non-Newtonian where viscosity is not well defined (Fox et al 2008). Navier-Stokes Equations (NSE) describes the behavior of fluids with a set of PDE based on;

1. Continuity : conservation of mass
2. Conservation of linear momentum and angular momentum
3. Conservation of energy

The NSEs assume the continuity of fluid being studied, as continuum. The *continuum* term is usually used for gradual (smooth) transition of properties from one state, to a different state, with no allowed discontinuities. The concept of a continuum is the basis of classical fluid mechanics (Tan et al 2009).

3.2.1 Navier-Stokes Equations (NSE)

Governing equations for fluid flow are nonlinear partial differential equations called NSEs. The nonlinear nature of NSEs introduces a hardship to the solution process. Only for very simple or special cases analytical solutions exist. Most often, solution is found using state of the art numerical methods. There are many different forms of

NSEs, for example widely used Newtonian formulation for incompressible fluid is defined as

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) = 0 \quad \text{velocity should conserve mass (continuity equation)} \quad \text{(Equation 3)}$$

$$\frac{\partial \vec{u}}{\partial t} = -(\vec{u} \cdot \nabla) \vec{u} + \nabla \cdot (v \nabla \vec{u}) + -\frac{1}{\rho} \nabla p + \vec{f} \quad \text{(Equation 4)}$$

$$\frac{\partial \rho}{\partial t} = -(\vec{u} \cdot \nabla) \rho + \kappa \nabla^2 \vec{u} + f_e \quad \text{(Equation 5)}$$

over a
convection
diffusion
pressure
force
time step
(advect)
(diffuse)

sources

In the equations above

$\vec{u}(x, y, z)$ is 3D velocity vector field (similarly 1D or 2D)

v : Kinematic viscosity which measures the fluid viscosity, $v = \mu / \rho$

μ : Viscosity \vec{f} : Body forces such as gravity

ρ : Density of the fluid κ : Heat conduction coefficient,

p : Pressure f_e : External force per unit mass

In the equations above “.” denotes dot product between vectors, and $\nabla = [\partial / \partial x, \partial / \partial y, \partial / \partial z]^T$ indicates the spatial-partial derivatives of a vector.

Briefly, (Equation 3) denotes the continuity of fluid which is based on incompressibility of fluid (divergence free). In (Equation 4) velocity changes due to the forces which assume conservation of momentum. (Equation 5) describes density change and assumes the conservation of energy. Such density changes can be caused by heat flows which are produced by external source(s) (e.g. electromagnetic fields) or internal chemical reactions and mostly neglected.

If we assume kinematic viscosity as constant, then $\nabla \cdot (v \nabla \vec{u})$ term turns into $v \nabla^2 \vec{u}$ where $\nabla^2 = \Delta = \partial^2 / \partial x^2 + \partial^2 / \partial y^2 + \partial^2 / \partial z^2$ denotes the Laplacian operator. For further simplification, we can also assume constant density such as 1 kg/m^3 , therefore

$(1/\rho)\nabla p$ term turns into ∇p and $\partial\rho/\partial t = 0$ consequently (Equation 5) can be dropped out. After these simplifications we have new version of NSE as

$$\frac{\partial \vec{u}}{\partial t} = -(\vec{u} \cdot \nabla)\vec{u} + \nu \nabla^2 \vec{u} - \nabla p + \vec{f} \quad \text{with} \quad \nabla \cdot \vec{u} = 0 \quad (\text{Equation 6})$$

In (Equation 4), the first term models the direction of the pushing imposed by the surrounding fluid on the current fluid region. This term is what makes the NSEs nonlinear.

Imaginary fluid with zero viscosity, having also a zero kinematic viscosity, is called as *inviscid fluid* (or ideal fluid). This assumption is unrealistic because there is no material with zero kinematic viscosity; for fluid to have zero viscosity is similar as a solid material having zero friction. However, it can be a good approximation for fluid having very little viscosity (e.g. water and petrol). If we also assume density as a constant 1 kg/m^3 then NSE for inviscid (Euler equations) and incompressible fluid with constant density becomes even much simpler.

$$\frac{\partial \vec{u}}{\partial t} = -(\vec{u} \cdot \nabla)\vec{u} - \nabla p + \vec{f} \quad \text{and} \quad \nabla \cdot \vec{u} = 0 \quad (\text{Equation 7})$$

Even after these simplifications nonlinearity still exists due to the convection term. To guarantee incompressibility, divergence of velocity is taken as zero (isochoric flow). The main purpose of incompressibility assumption is to simplify analysis. On the contrary, little or much all the materials are compressible to some degree. This means with the occurrence of certain circumstances even compressible materials can flow in an incompressible manner. However, governing fluid flow equations can be considerably simplified by assuming the incompressibility of material. Another possible simplification is setting all the change in fluid properties to zero with respect to time. This assumption makes nonlinear convection term time independent. Various problems can be modeled with this simplified flow which is called as steady flow (Nealen et al 2006, Fox et al 2008). In steady flow, time dimension is dropped-out from NSE.

3.2.2 Fluid Simulation Methods

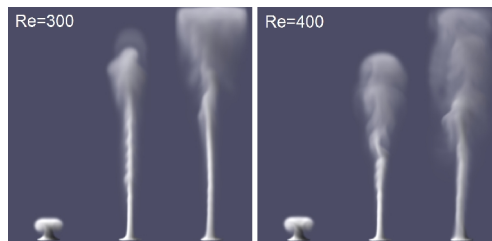
Simulating fluids is a challenging problem in aeronautics, mechanical engineering, computer graphics, and medical image analysis, because implementing a good numerical solver for the NSEs is a difficult undertaking. There exists a complicated relationship between causes and effects due to the nonlinear nature of NSEs.

Many phenomena such as water flow, blood flow, high viscosity, viscoelasticity, wave, turbulence, splashes, bubbles, foams, surface tension, erosion on materials due to fluid flow, motion of gases, clouds, melting, burning, explosions, shock waves, and motion can be simulated using various analytical forms of NSEs with different numerical approaches (e.g. Lagrangian, Eulerian, Lagrangian-Eulerian coupling etc).

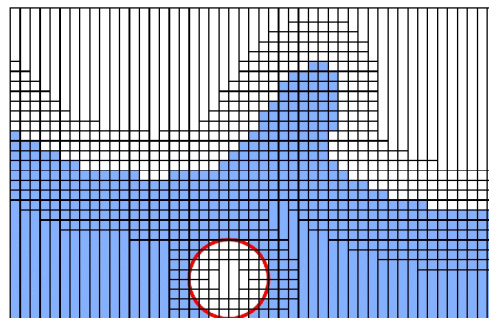
Some important factors in realistic fluid simulation are as follows (Kass and Miller 1990, Foster et al 1996, Stam 1999, Stora et al 1999, Foster et al 2001, Carlson et al 2002, Müller et al 2004, Goktekin et al 2004, Keiser et al 2005, Losasso et al 2006, Yüksel et al 2007):

- Geometrical complexity of environment that fluid(s) simulated,
- Number of interacting fluids,
- Interaction type (e.g. one-way or two way coupling),
 - fluid-solid (e.g. elastic, fixed/moving solid) interaction,
 - fluid-fluid (e.g. multiphase flow) interaction,
- Heat transfer,
- Phase-transformation (e.g. burning, melting),
- Characteristic of the fluid (e.g. viscosity, laminar or turbulent etc.),
- Behavior of simulated fluid in large-scale as well as small-scale,
- Surface tension,
- Buoyant effects

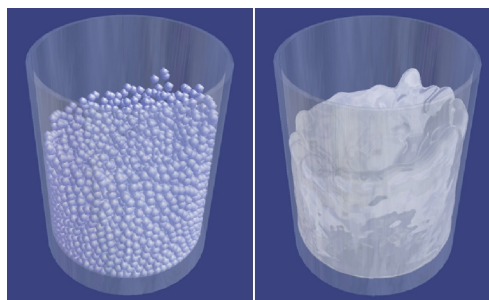
In order to address each problem, different approximations and assumption are used for simplification. For example, to simulate incompressible laminar fluid with constant density, 2D Shallow Water Equation (SWE) is used to solve fluid flow. Visualization is achieved by 3D fluid flow using pressure as height of water, ignoring viscosity, assuming Laminar flow, and assuming periodic boundaries like fluid flow in torus. Problems containing motion of hot and turbulent gases are difficult to solve due to their chaotic behavior. Problems having high Reynolds number such as inviscid and more turbulent fluid are difficult to simulate whereas problems having small Reynolds number such as viscous and laminar fluid are relatively easy to simulate.



(a) Lattice Boltzmann Method with different Reynold numbers (Alim et al 2009)



(b) Adaptive MAC grid (Irving et al 2006)



(c) Lagrangian particle method using SPH (Matthias et al 2003)

Figure 13: Example fluid simulation methods.

A variety of methods exist for physically based animation such as mass-spring, Finite Element / Difference / Volume methods, MAC-grid methods, particle methods, and Lattice Boltzmann methods (Tan et al 2009). In order to speed up the simulation and to decrease memory usage oct-tree and adaptive grid based-methods are proposed. Large volumes of water are efficiently simulated by coupling two and three dimensional techniques which employ a vertically adaptive grid (Figure 13.b). Recently mesh-free methods (e.g. (Figure 13.a,c) became popular especially for simulating fluids due to their capability of topological adaptivity without the remeshing obligation. Advected Radial Basis Function (RBF), control particles, gradient-based nonlinear optimization via adjoint method are proposed in production environments to handle fine-scale details through simulation (Foster et al 1997, McNamara et al 2004, Pighin et al 2004, Thürey et al 2009). Adjoint method based fluid control linearizes each step of simulation and solves the resulting inverse problem (McNamara et al 2004).

Speed of implementation is a very important aspect since it can be prohibiting. Implementation can be either done on Central Processing Unit (CPU) or Graphics Processing Unit (GPU) or combination of both. Implementations can be optimized considering the specifications of the hardware (e.g. on CPU try to reduce cache-miss during an array access, loop unrolling etc.). Optimized CPU implementations can become order of magnitude faster compared to un-optimized versions. GPU became an affordable commodity providing huge scientific processing power, providing scalability and very good parallelism due to their streaming processing nature; hence they became important alternatives to speedup fluid simulations.

3.2.3 Lagrangian Approach versus Eulerian Approach

For tracking the motion of a continuum such as fluid or deformable solid Lagrangian and Eulerian approaches are used (Fox et al 2008).

Lagrangian approach is mostly preferred for problems involving solids. In Lagrangian approach, the continuum is treated as a particle system. Each point in the material is determined by a distinct particle and its properties (e.g. position, mass, velocity etc.). As the particle moves, its properties also move with it, like a falling ball. Observing the properties (e.g. velocity etc.) of this ball as it falls is dictated by the Lagrangian approach. Predefined neighborhood model or connectivity within a mesh can be defined to create relationships between these discrete set of particles. This is a familiar approach from basic mechanics where objects are treated as separate entities and their motions are discussed in a Newtonian perspective (Tan et al 2009). Interested reader can read appendix A for further details.

Unlike the Lagrangian approach which keeps track of each particle, *Eulerian approach* measures how the material (fluid) quantities (e.g. velocity, density etc.) change in time for the predefined stationary points in space. Due to its convenience, Eulerian approach is mostly used for fluids (Gingold et al 1977). Interested reader can read appendix B for further details.

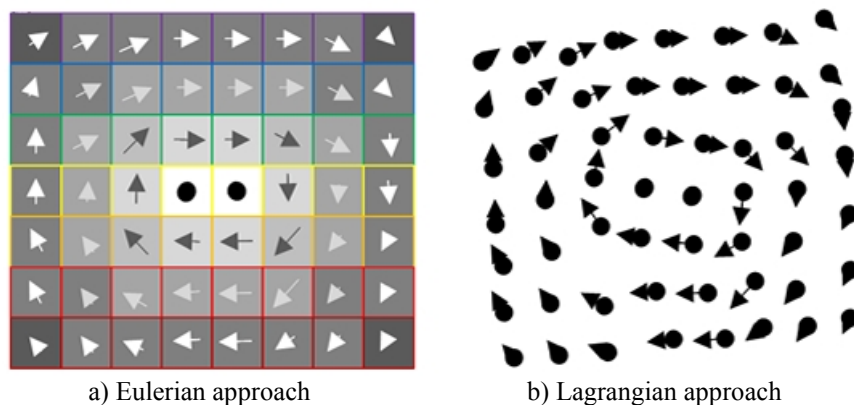


Figure 14: Eulerian grid versus Lagrangian particles.

Within fluid flow, fluid properties such as temperature, density, pressure, and velocity change in a certain point in space within a fluid flow. For example as we increase the temperature in environment, we will see a gradual temperature increase in the fluid for any chosen fixed point due to the second law of thermodynamics. For

this case, Eulerian approach imposes observing the change of temperature in a fixed point in the space where the material is placed (Monaghan 2005, Losasso et al 2006). Numerically, the Lagrangian approach is equivalent to a particle system either connected or not (Figure 14.b), and the Eulerian approach is equivalent to the employment of a fixed grid in space where the fluid flows through (Figure 14.a). Brief comparison of both models is discussed in the Table 2.

Table 2: Comparison of Lagrangian approach and Eulerian approach

Lagrangian	Eulerian
Easier to understand and implement	More complex
Computationally expensive models are needed to model pressure, density, and derivatives	Easy to analytically model pressure, density etc. and necessary spatial derivatives
Finding neighbor particles are very computational (remedy: spatial indexing)	Each grid cell contains neighborhood information → very efficient computation: $O(1)$ time
Efficient memory consumption (necessary amount of particles)	Considerable amount of memory consumption (grid covering the all fluids during simulation)
Does not scale well (due to huge number of particles)	Does not scale well (due to huge grid size)
Finding fluid interface is rather difficult (e.g. during splashes)	Finding fluid interface is relatively easy
Model does not contain incompressibility. Enforcing incompressibility is optional	Model contains incompressibility constraint. Enforcing incompressibility is a must
Since each particle has its own, mass there is no mass-loss problem	Mass loss problem must be carefully handled via interface tracking methods (e.g. level set)
Not successful with large deformations	Works better for large deformations
Preferred for problems involving solids	Preferred for problems involving fluids

3.3 Shallow Water Equations

The SWEs model the propagation of disturbances in water and other incompressible fluids. The main assumption is that the depth of the fluid is small compared to the wave length of the disturbance, like in the ocean. Thereby complex 3D description of a fluid can be reduced to a 2D height field using SWE (Figure 15).

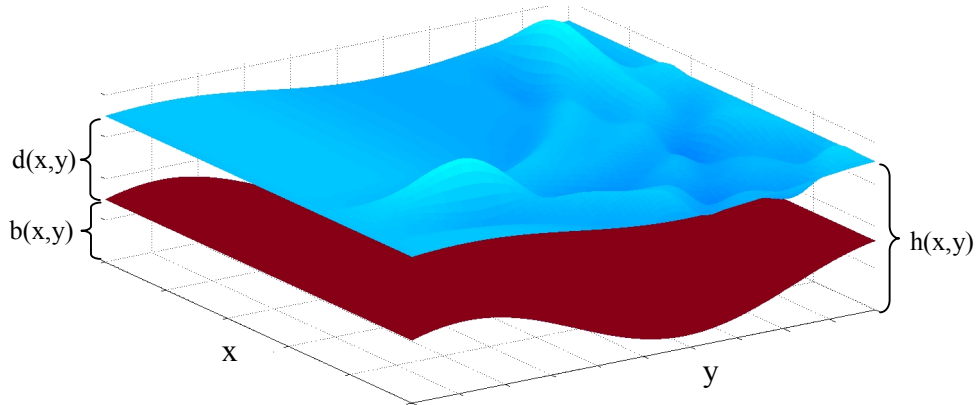


Figure 15: 2D height field representation: $b(x,y)$ is ground height, $d(x,y)$ is water height, $h(x,y) = b(x,y) + d(x,y)$ is surface height

It can be assumed that velocity does not vary significantly along the z axis, and pressure gradient is constant from the water bottom to the surface hence vertical component of velocity can be ignored. Therefore one can look for solutions in which the horizontal flow itself is independent of water column height. This is the key simplification that underlies the shallow water model. Since the horizontal flow is independent of height, incompressibility implies that the vertical velocity is linear in z . By taking the vertical velocity and variations throughout the depth of the fluid to be exactly zero in the NSEs and after depth-integrating, the SWEs are derived. In addition if we limit our simulation to inviscid fluids (having zero viscosity) such as water we can work with the simpler Euler equations. Based on all these assumptions and approximations 1D shallow water definition is given as water depth above a ground depth (Figure 16) (Shinbrot 1970, Kass and Miller 1990).

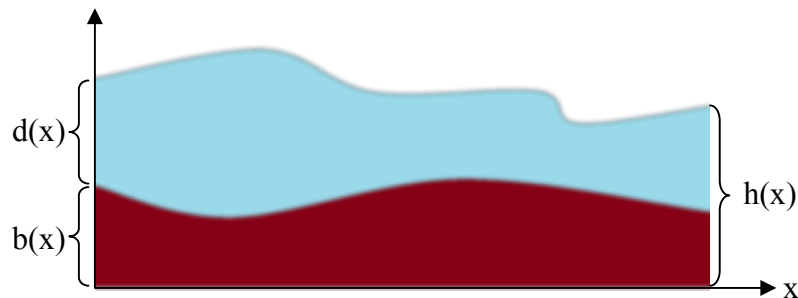


Figure 16: Shallow water representation. $b(x)$ is ground height, $d(x)$ is water height, $h(x) = b(x) + d(x)$ is surface height

Given above 1D shallow water definition and explained simplifications 1D shallow water equations can be defined as follows:

$$\frac{\partial d}{\partial t} + \frac{\partial}{\partial x}(ud) = 0 \quad (\text{Equation 8})$$

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + g \frac{\partial h}{\partial x} = 0 \quad (\text{Equation 9})$$

where g is the gravitational acceleration and $u(x)$ is the horizontal velocity of a vertical column of water. Since external force term in NSE is dropped out in SWE and only remaining external force becomes gravity. (Equation 8) express volume conservation constraint, while (Equation 9) express Newton's law $F=ma$. After applying simplifications based on the assumption of water is being shallow, (Equation 9) still stays non-linear because of $u(\partial u/\partial x)$ term, which is responsible for convection / advection of properties within fluid including velocity itself.

In this study we prefer to use LSWE formulation proposed by Kass and Miller. Kass and Miller further simplified (Equation 9) by ignoring second term $u(\partial u/\partial x)$ and linearize the equation around a constant value of h . This simplification is reasonable with the fluid having **small velocity** placed on the **slowly varying ground** (Kass and Miller 1990). The resulting equations are then

$$\frac{\partial h}{\partial t} + d \frac{\partial u}{\partial x} = 0 \quad (\text{Equation 10})$$

$$\frac{\partial u}{\partial t} + g \frac{\partial h}{\partial x} = 0 \quad (\text{Equation 11})$$

After doing some mathematical manipulations two equations can be combined as

$$\frac{\partial^2 h}{\partial t^2} = gd \frac{\partial^2 h}{\partial x^2} \quad (\text{Equation 12})$$

which is the 1D wave equation with wave velocity \sqrt{gd} , where initial h values within simulation boundaries ($0 < x < L$) can be specified by user as initial condition,

pressure at water surface being zero is a Dirichlet boundary condition, and fluid motion being zero ($\partial h / \partial t = 0$) at the simulation boundaries ($x=0, x=L$) is the Neumann boundary condition which makes wave to reflect at simulation boundaries. Due to simplifications made LSWE becomes unsuitable for many engineering purposes but it is still usable for image processing or computer graphics or similar tasks. In order to solve (Equation 12) we need to discretize it using a suitable finite-difference technique (Kass and Miller 1990). One of the other discretization alternatives is the finite-element technique where the continuous functions are represented as the linear sum of a continuous basis functions (Peaceman et al 1955).

$$\frac{\partial h_i}{\partial t} = \left(\frac{d_{i-1} + d_i}{2\Delta x} \right) u_{i-1} - \left(\frac{d_i + d_{i+1}}{2\Delta x} \right) u_i \quad (\text{Equation 13})$$

$$\frac{\partial u_i}{\partial t} = -g \left(\frac{h_{i+1} - h_i}{\Delta x} \right) \quad (\text{Equation 14})$$

where Δx is the separation between samples along the x direction. Putting the above two equations together we can come up with below equation.

$$\frac{\partial^2 h_i}{\partial t^2} = -g \left(\frac{d_{i-1} + d_i}{2(\Delta x)^2} \right) (h_i - h_{i-1}) + g \left(\frac{d_i + d_{i+1}}{2(\Delta x)^2} \right) (h_{i+1} - h_i) \quad (\text{Equation 15})$$

This is a discrete approximation to (Equation 12). (Equation 12), which is a Partial Differential Equation (PDE), is converted into (Equation 15), which is an Ordinary Differential Equation (ODE), hence it can be solved using a suitable ODE integrator. Kass and Miller preferred first-order implicit integrator (Kass and Miller 1990).

$$\frac{h^{(n)} - h^{(n-1)}}{\Delta t} = \dot{h}^{(n)} \quad (\text{Equation 16})$$

$$\frac{\dot{h}^{(n)} - \dot{h}^{(n-1)}}{\Delta t} = \ddot{h}^{(n)} \quad (\text{Equation 17})$$

where $h^{(n)}$ is n^{th} and $h^{(n-1)}$ is the prior iteration. Rearranging above two equations

$$h^{(n)} = 2h^{(n-1)} - h^{(n-2)} + (\Delta t)^2 \ddot{h}^{(n)} \quad (\text{Equation 18})$$

below corresponding discretized system is obtained.

$$h_i^{(n)} = 2h_i^{(n-1)} - h_i^{(n-2)} + g(\Delta t)^2 \left[\left(\frac{d_i + d_{i+1}}{2(\Delta x)^2} \right) (h_{i+1}^{(n)} - h_i^{(n)}) - \left(\frac{d_{i-1} + d_i}{2(\Delta x)^2} \right) (h_i^{(n)} - h_{i-1}^{(n)}) \right] \quad (\text{Equation 19})$$

(Equation 19) can be represented as a linear system with the symmetric tridiagonal matrix A as shown in (Equation 20).

$$Ah^{(n)} = 2h^{(n-1)} - h^{(n-2)} = y \quad (\text{Equation 20})$$

where the matrix A and its elements e and f are given by

$$A = \begin{bmatrix} e_0 & f_0 & 0 & 0 & 0 & 0 & 0 \\ f_0 & e_1 & f_1 & 0 & 0 & 0 & 0 \\ 0 & f_1 & e_2 & \ddots & 0 & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & 0 & 0 \\ 0 & 0 & 0 & \ddots & e_{w-3} & f_{w-3} & 0 \\ 0 & 0 & 0 & 0 & f_{w-3} & e_{w-2} & f_{w-2} \\ 0 & 0 & 0 & 0 & 0 & f_{w-2} & e_{w-1} \end{bmatrix} \quad \begin{cases} e_i = 1 + k \begin{cases} d_0 + d_1 & i = 0 \\ d_{i-1} + 2d_i + d_{i+1} & 0 < i < w-1 \\ d_{w-2} + d_{w-1} & i = w-1 \end{cases} \\ f_i = -k(d_i + d_{i+1}) \end{cases} \quad \text{and} \\ k = g(\Delta t)^2 / (2(\Delta x)^2)$$

Artificial viscosity can be established by changing the (Equation 20) as

$$Ah^{(n)} = h^{(n-1)} + (1 - \tau)(h^{(n-1)} - h^{(n-2)}) = y \quad (\text{Equation 21})$$

where $0 \leq \tau \leq 1$ is the damping coefficient and $\tau = 0.01$ usually provides reasonable damping effects. When damping coefficient is chosen as zero (inviscid fluid) (Equation 21) turns back into (Equation 20). As viscosity is increased (damping coefficient $\tau = 0.1$, fluid with higher viscosity) fluid flow will be slower with smoother boundary. It is possible to convert the viscosity term into a higher order polynomial for introducing smoother effects. $h^{(0)}$ corresponds to the initial water column height in shallow water system which is the initial values for (Equation 20) and (Equation 21). Water column height vector h has size of w and first and last elements in the h correspond to the boundaries of water system. Boundary reflection conditions for these boundaries can be seen in the tridiagonal matrix A as a top left corner and bottom right corner. One can use tridiagonal matrix algorithm (TDMA), Thomas algorithm, to solve the linear system presented in (Equation 20) and (Equation 21) in $O(n)$ operations (Kass and Miller 1990).

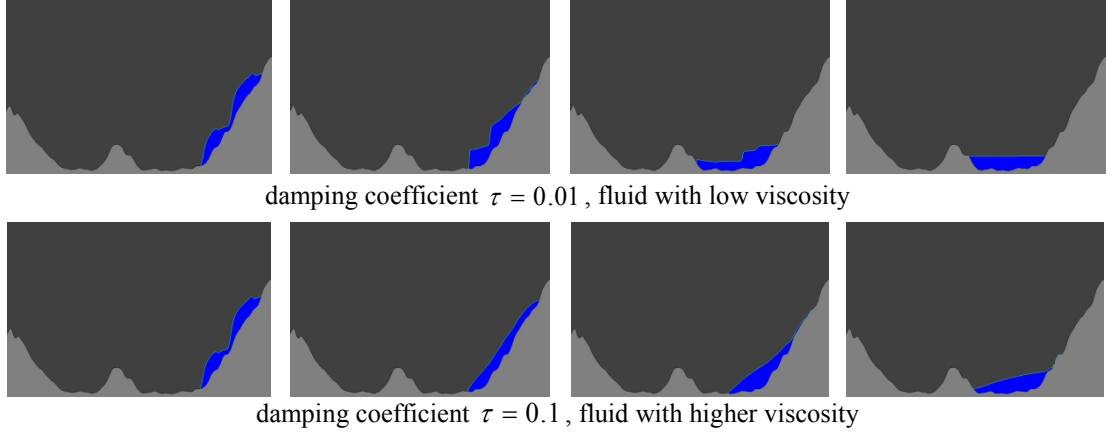


Figure 17: 1D SWE fluid evolution example.

2D and 3D LSWE are defined by introducing spatial terms containing y and z axes as

$$2D \rightarrow \frac{\partial^2 h}{\partial t^2} = gd \left(\frac{\partial^2 h}{\partial x^2} + \frac{\partial^2 h}{\partial y^2} \right) = gd \nabla^2 h \quad (\text{Equation 22})$$

$$3D \rightarrow \frac{\partial^2 h}{\partial t^2} = gd \left(\frac{\partial^2 h}{\partial x^2} + \frac{\partial^2 h}{\partial y^2} + \frac{\partial^2 h}{\partial z^2} \right) = gd \nabla^2 h \quad (\text{Equation 23})$$

(Equation 22) and (Equation 23) can be approximated by series of one dimensional LSWEs, which is (Equation 12). In order to solve the (Equation 22) and (Equation 23) Alternating Direction Implicit (ADI) method is used (Douglas et al 1993). ADI method suggests splitting the right hand side of equation into two terms in 2 dimensions and into three terms in 3 dimensions so that variables in right hand side become independent to each other. Then iteration is divided into 2 sub iterations for the (Equation 22) and 3 sub iterations for (Equation 23). Δt must be divided to the number of sub iterations respectively, $\Delta t/2$ for 2D case and $\Delta t/3$ for 3D case.

The advantage of ADI method is that, the required linear systems remain tridiagonal while using this method. Thus computational cost per iteration is proportional to the number of pixels in the image and voxels in the volume. For LSWE approach computational complexity is $O(n)$ where n is the number of pixels in image (2D) and number of voxels in volume (3D) as demonstrated bellow.

$$\begin{aligned}
\text{1D signal } w & \rightarrow O(w) = O(n) \\
\text{2D image } w \times h & \rightarrow hO(w) + wO(h) = O(2wh) = 2O(wh) = O(n) \\
\text{3D volume } w \times h \times c & \rightarrow hwO(W) + wcO(H) + whO(C) = O(3whc) = O(n)
\end{aligned}$$

where w is width, h is height, and c is number of slices in volume. In LSWE each pixel in 2D image or each voxel in a 3D volume corresponds to a water column in shallow water system. In short, LSWE can be solved in $O(n)$ hence linear time in a very computationally efficient manner whereas other nonlinear fluid methods require complex interface tracking methods such as level set, computational volume preservation method such as Poisson solvers (Appendix B) or restriction of using very small time steps such as in LBM (Appendix C). LBM is parallel in nature but other nonlinear methods are hard to parallelize. The parallel nature of LSWE is advantageous to acquire reasonable computational times. Computational efficiency of the fluid solver is crucial in clinical use for reasons listed in the following:

- Proposed segmentation method uses fluid flow as deformable model.
- There is huge number of water columns in a 3D volume (typically millions).
- Fluid simulation must iterate in the order of thousand during simulation.

2D implementation of LSWE is trivial since (Equation 22) can be separated using the ADI method. One can just integrate image columns as 1D LSWEs then rows as 1D LSWEs based on the result of the column integration. Although this is theoretically correct, same ordering favors fluid to move in one direction due to discretization (Figure 18.a). On way to fix this problem is alternating the integration order during iterations (Figure 18.b) (Kass and Miller 1990).

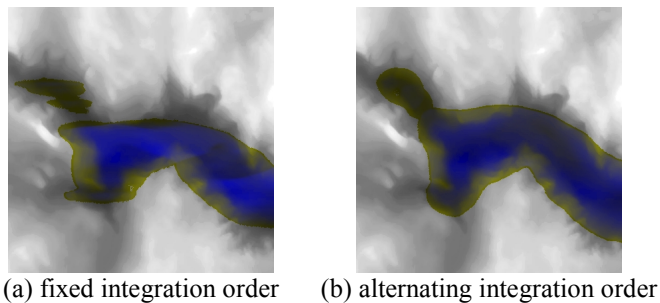


Figure 18: Importance of order of integration in 2D SWE fluid.

LSWE can capture most of the fluid motion which seems sufficient for our segmentation purposes. Image in Figure 19 top left is used as a height map and high amount of fluid is placed into the regions where ground height is low (bottom left). In this example, the fluid column moves to the regions where neighbor columns are lower in height, neighbor columns can contain fluid or they can be dry. LSWE is the main building block of the proposed vessel segmentation algorithm which will run on 2D images and 3D volumes. In Figure 19 $h(x)$ represents height of the fluid surface which represents z-direction of fluid to obtain 3D visualization of 2D SWE.

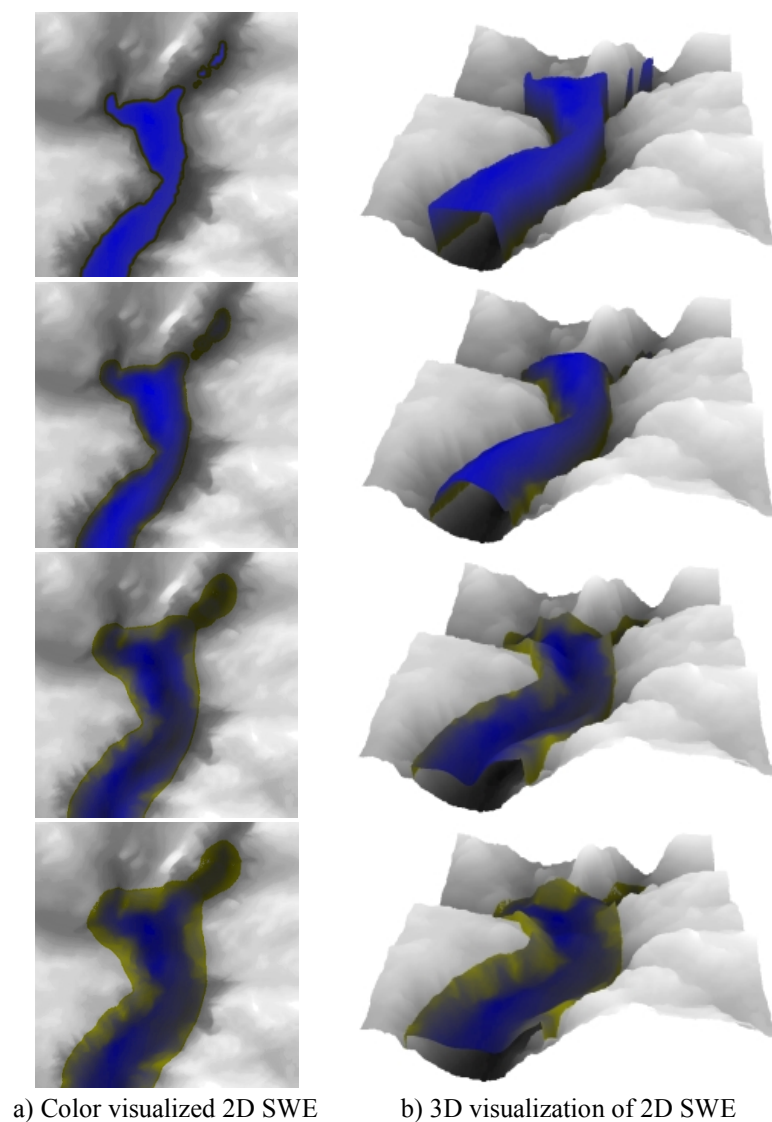


Figure 19: 2D SWE fluid evolution example.

3.4 Stability of the Method

Numerical schema used in LSWE is stable (Kass and Miller 1990, Moran and Shapiro 2007). In order to demonstrate the stability of the linear system given by (Equation 20), the growth of error has to be explored to ensure that the error is not amplified each iteration. The inverse of matrix A governs the stability of the scheme as it either amplifies or dampens the error from the previous time steps. It is assumed that the maximum water depth is d and the discretized d_i are equal to this depth when calculating A . Using the *matrix method*, the matrix A can be written as

$$A = \begin{bmatrix} 1+2kd & -2kd & & & 0 \\ -2kd & 1+4kd & -2kd & & \\ & \ddots & \ddots & \ddots & \\ & & -2kd & 1+4kd & -2kd \\ 0 & & & -2kd & 1+2kd \end{bmatrix} \quad (\text{Equation 24})$$

The eigenvalues of this tridiagonal matrix can be calculated as

$$\lambda_m = 1 + 4kd + 4kd \cos \frac{m\pi}{n} \quad (\text{Equation 25})$$

where n is the size of the matrix and $m = 1, 2, 3, \dots, n$ (Yueh 2005). The error relation between two time steps are given by

$$A \zeta^{j+1} = \zeta^j \quad \text{or} \quad \zeta^{j+1} = A^{-1} \zeta^j \quad (\text{Equation 26})$$

where ζ^j is the error at j^{th} time step. For stability, all eigenvalues of A^{-1} thus μ_m should be equal to or less than unity (Hochstadt 1964) (Nayfeh and Mook 1979).

$$\mu_m = \frac{1}{\lambda_m} \leq 1 \quad \text{and} \quad \lambda_m = 1 + 4kd + 4kd \cos \frac{m\pi}{n} \geq 1 \quad (\text{Equation 27})$$

This leads to

$$4kd \left(1 + \cos \frac{m\pi}{n} \right) \geq 0 \quad (\text{Equation 28})$$

Since $-1 \leq \cos \frac{m\pi}{n} \leq 1$, $k \geq 0$, $d \geq 0$, the inequality (Equation 28) will always be valid which means that the scheme is stable.

3.5 Proposed Vessel Segmentation Algorithm

Proposed algorithm is exactly the same for the 2D images and the 3D volumes other than subtle differences in simulation parameters. In that sense proposed method is generic and supports different type of vessels and image modalities (2D: retinal angiography, 3D: MRA, MRV, and CTA). Flowchart of the proposed algorithm is given in Figure 20.

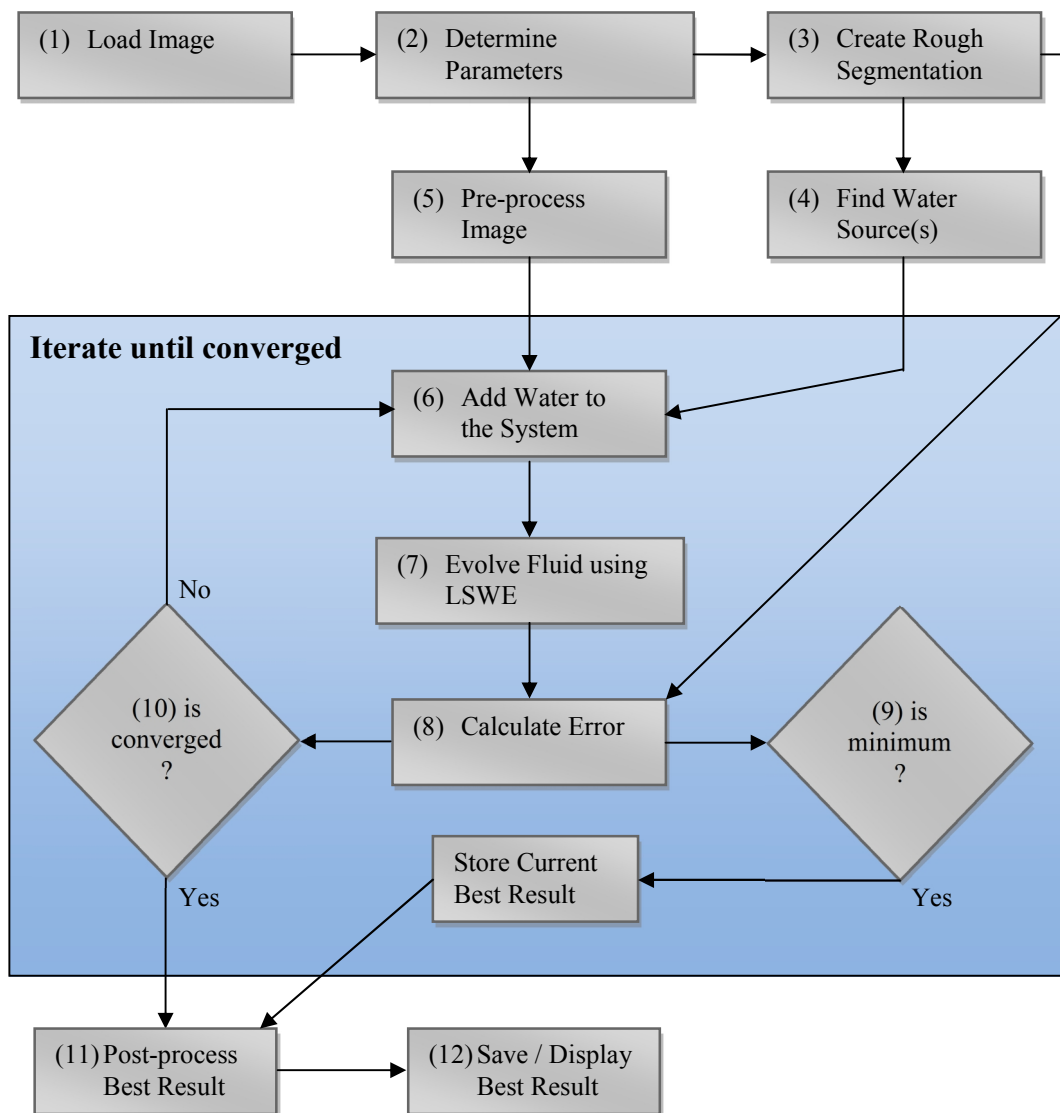


Figure 20: Flowchart of proposed vessel segmentation algorithm.

Step 1. Loading image

In MRA / MRV vessels are represented with bright intensity values (Figure 21.a and Figure 21.c). This intensity values must be inverted so that bright intensity values become dark intensity values (Figure 21.b and Figure 21.d). Once vessels are dark in image, these are interpreted as valleys and LSWE can be used to fill those valleys with fluid as in the proposed method (Figure 22). In Figure 21.b 2D image and Figure 21.d 3D volume is shown which are ready for the further segmentation steps.

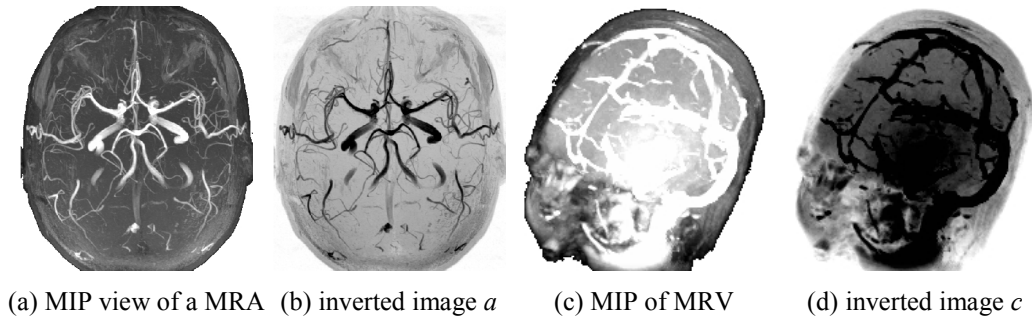


Figure 21: 2D and 3D MRA examples.

Step 2. Determining parameters

Although image acquisition is done in real world coordinates (e.g. 1 mm slice thickness in z-direction, 0.5 mm in-plane resolution in x, y directions), within our LSWE simulation, each pixel/voxel is assumed to be in meters. Formulation of LSWE supports both isotropic and anisotropic data but our implementation assumes isotropy of data. If image or volume is not isotropic then it must be resized so that it becomes isotropic. Pixel intensity values correspond to ground depth and have to be normalized so that the shallowness assumption holds for vessels³ (Figure 22). 10^{-8} is added to ground height to prevent possible division by zeros.

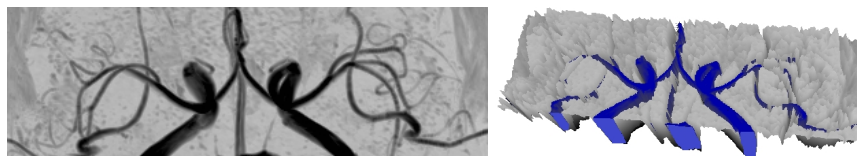


Figure 22: 2D and 3D MRA example visualization.

³ As an example, for volume data of size 180x220x180 with 1mm isotropic voxels, we may set maximum ground depth as 6 m. and allow maximum vessel depth to be 1.5 m, and normalize all intensities accordingly.

Step 3. Creating rough segmentation

Rough segmentation is an estimation of vessels to be used for determining water sources and convergence time. Rough segmentation can be found using a fast method, such as Otsu thresholding (1979) but it may fail if the image contains inhomogeneity artifacts. Otsu thresholding based rough segmentation (Figure 23.c) may contain non vessel components and may lack especially low contrast vessels. For obtaining a robust rough segmentation we developed a filtering procedure which uses a local approach instead of Otsu's global filtering approach. In the proposed local filtering approach each voxel intensity value is subtracted from the clamped maximum intensity value in a given window where clamping value is average intensity value within a larger window. This approach is inspired from the study of Alonso and Vilari (2008) where vessels are pre-estimated by blurring the original image and subtracting the blurred image from the original one.

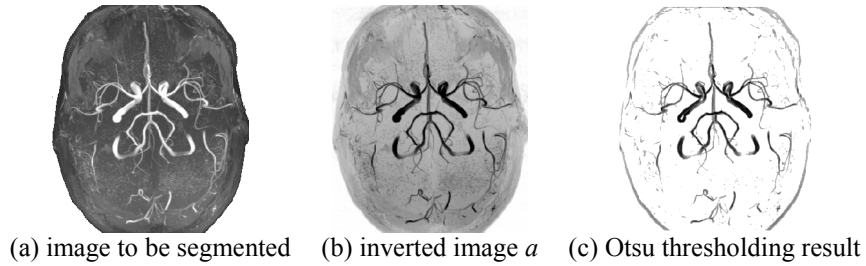


Figure 23: Rough segmentation using Otsu method.

$$R_{x,y,z} = \begin{cases} \hat{R}_{x,y,z} & \text{if } \hat{R}_{x,y,z} > T_{rs} \\ 0 & \text{if } \hat{R}_{x,y,z} \leq T_{rs} \end{cases} \quad (\text{Equation 29})$$

where $R_{x,y,z}$ is rough segmentation, T_{rs} is the noise threshold, $\hat{I}_{x,y,z}$ is the low-pass filtered version of the original image pixel $I_{x,y,z}$, r_v is estimated maximum vessel radius for given image,

$$\begin{aligned} \hat{R}_{x,y,z} &= \max(\min(I_{x,y,z}^{(m)}, I_{x,y,z}^{(c)}) - \hat{I}_{x,y,z}, 0) \text{ is clamped maximum intensity value,} \\ I_{x,y,z}^{(m)} &= \max_{-r_v \leq \Delta x, \Delta y, \Delta z \leq r_v} I_{x+\Delta x, y+\Delta y, z+\Delta z} \text{ is maximum intensity in window 1,} \\ I_{x,y,z}^{(c)} &= \frac{1}{(8r_v + 1)^3} \sum_{\Delta x=-4r_v}^{4r_v} \sum_{\Delta y=-4r_v}^{4r_v} \sum_{\Delta z=-4r_v}^{4r_v} I_{x+\Delta x, y+\Delta y, z+\Delta z} \text{ is average (clamping) value in window 2,} \end{aligned}$$

In Figure 24 rough segmentation steps in (Equation 29) is shown on a maximum intensity projection image of a MRA volume.

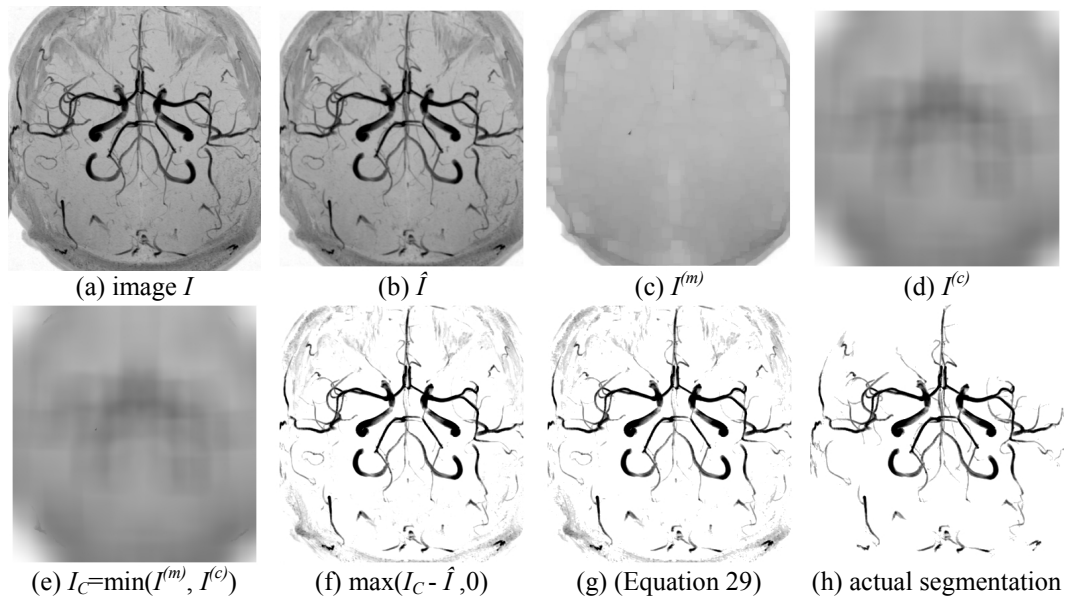


Figure 24: Rough segmentation steps for 2D MIP image.

In Figure 25 rough segmentation steps in (Equation 29) is shown on a MRA volume. As seen in Figure 25.a-e volume is divided into 3 regions due to an imaging artifact yet LSWE able to segment vessels successfully.

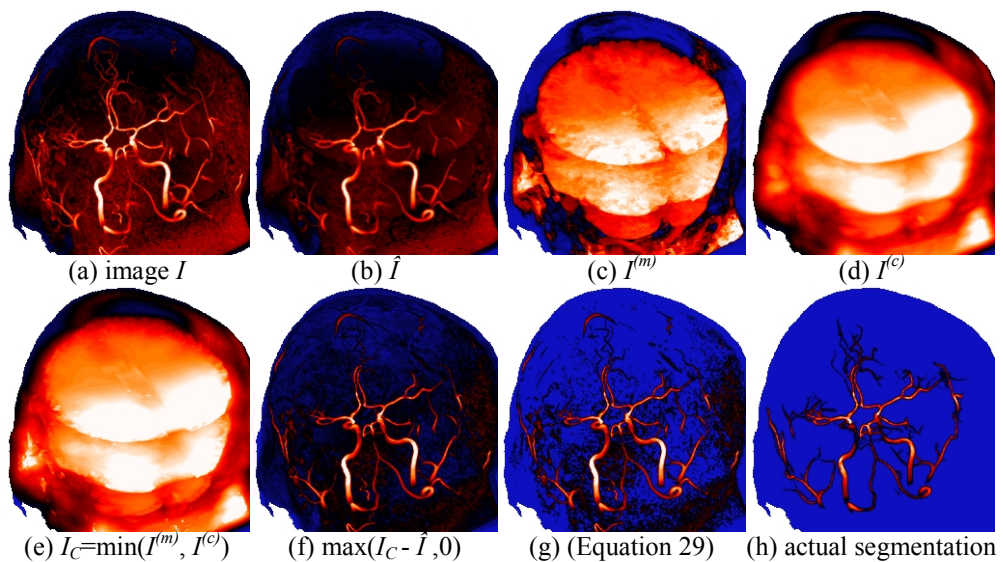


Figure 25: Rough segmentation steps for a MRA volume.

In Figure 26 rough segmentation steps in (Equation 29) is shown on a retinal angiography image. As seen in Figure 26 clamping operation wipes out the very high intensity values such as iris center in retinal angiography images.

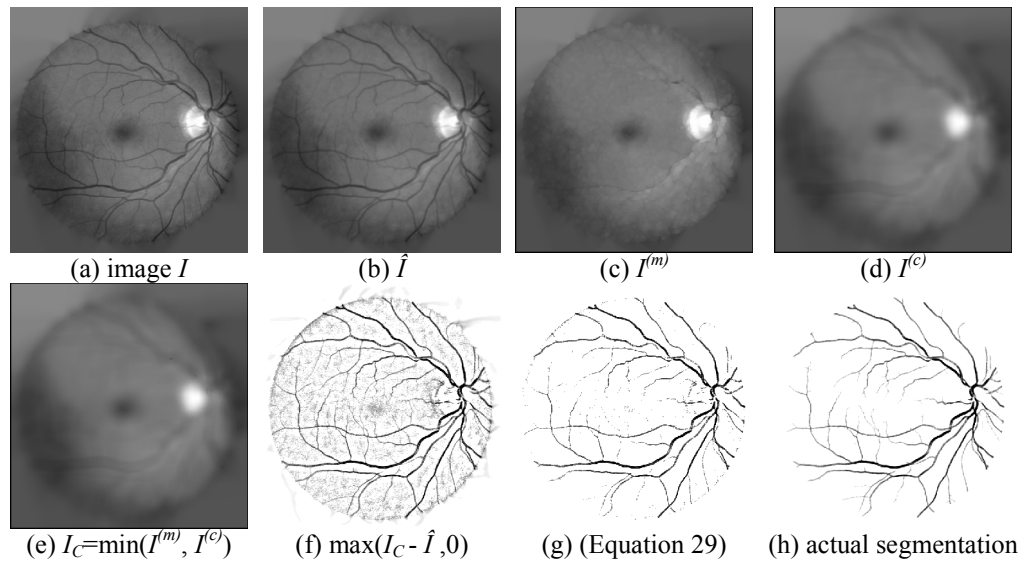


Figure 26: Rough segmentation steps for a retinal angiography image.

In Figure 24, Figure 25, and Figure 26 rough segmentation has got problems such as having noise, lack of thin and low contrast vessel structures, and existence of disconnected vessel like structures whereas actual segmentation eliminates such problems. One can think “rough segmentation” as overall rough view of vessel structures and LWSE as a detailed view of vessel structures. Actual segmentation is obtained when the fluid heights became most similar to the “rough segmentation”.

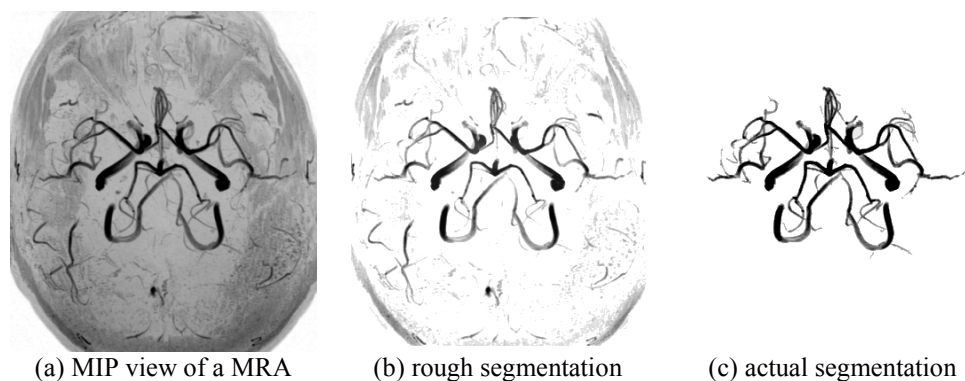


Figure 27: LSWE segmentation example.

Step 4. Finding the water source

a) Finding single water source using scale space approach

Water source must be placed in a region where intensities are low and potentially other low intensity regions are nearby. Such region(s) can be determined by the user or automatically. If an automatic initialization algorithm is used this algorithm must be robust to noise so it can escape from low intensity regions caused by noise. To find such a water source in a computationally efficient manner a multi scale approach is developed. Lower resolution image at level L is generated using the below filter

$$I_{x,y}^{(L+1)} = \sum_{m=1}^2 \sum_{n=1}^2 \left(\begin{cases} I_{2x+m,2y+n}^{(L)} & \text{if } I_{2x+m,2y+n}^{(L)} < \mu^{(L)} \\ \mu^{(L)} & \text{otherwise} \end{cases} \right) \quad (\text{Equation 30})$$

where $I^{(L)}$ is image at level L , $I^{(0)}$ is the rough segmentation, and $\mu^{(L)}$ is the average intensity value in $I^{(L)}$. As L increases image size is halved and normalized to 1 so that maximum intensity is always the brightest intensity value. Low resolution image must contain enough information about darkest region so it must be larger than $\phi \times \phi$ where $\phi = 16$. In the lowest resolution image, position of darkest pixel position (P) is found and in upper resolution image darkest pixel in 2 by 2 window is searched and procedure is repeated up to the original image $I^{(0)}$ (Figure 28). Water source $W_{x,y}^{(source)}$ is the circular region with radius r , and center point P . One can extend (Equation 30) for 3D volume by adding one more summation term for the z axis.

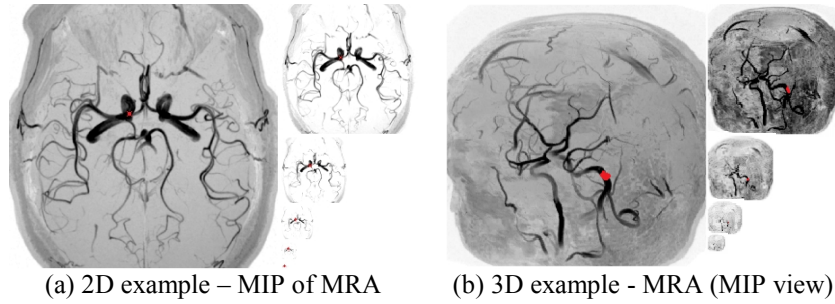


Figure 28: Water source detection using scale-space approach.

Computational complexity of the above algorithm is $O(2^{2k+1}n + 4k + \phi^2) \rightarrow O(n)$ hence linear, where n is the number of pixels in the image $I^{(0)}$ and k is the number of images in scale space.

b) Finding thick vessels as water sources using filter-based approach

A single run of LSWE with single water source segmentation is limited to capture a single vessel system. In addition, it causes longer convergence times. Fluid can pass through small or medium sized gaps so that segmented vessel system can contain small or medium sized gaps. But if there is a large gap or there are two separate vessel systems then either more than one water source is needed or single source method must be executed multiple times. Thick vessels can be detected as multiple water sources (Figure 29.c, Figure 29.f) to remedy this issue using the below filter on normalized rough segmentation

$$W_{x,y,z}^{(source)} = \begin{cases} 0 & \text{if } \tilde{R}_{x,y,z}^{(thinSupressed)} \leq T_{ws} \\ \tilde{R}_{x,y,z}^{(thinSupressed)} & \text{otherwise} \end{cases} \quad (\text{Equation 31})$$

where \tilde{R} is the normalized rough segmentation: $\tilde{R} = R / \max(R)$, T_{ws} is the threshold value, r_w is kernel radius, $2r_w + 1$ is the width and height of a square window,

$$\tilde{R}_{x,y,z}^{(thinSupressed)} = [\prod_{\Delta x=-r_w}^{r_w} \prod_{\Delta y=-r_w}^{r_w} \prod_{\Delta z=-r_w}^{r_w} (\tilde{R}_{x+\Delta x, y+\Delta y, z+\Delta z} + 1)] - 1$$

is used for suppressing thin vessels.

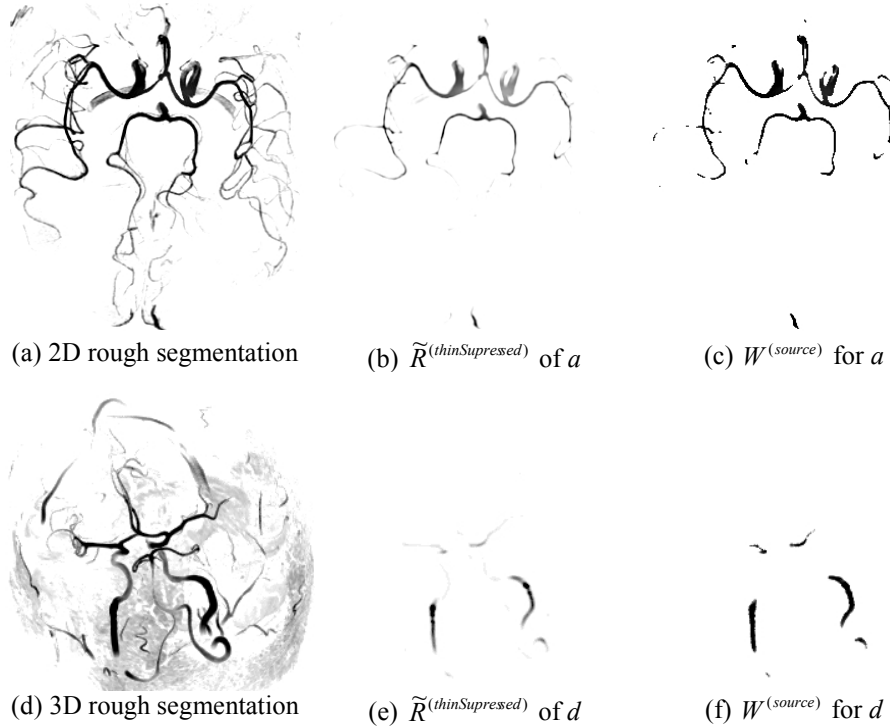


Figure 29: Water source detection using filter-based approach.

Step 5. Preprocessing the image

To linearize SWE, Kass and Miller (1990) ignored convection / advection term assuming the fluid with small velocity is placed on a smoothly varying ground. If there is a high amount of noise or sharp changes in image then smoothly varying grounds assumption does not hold. In a retinal image, the background is dark. Hence this background is changed into bright values (Figure 30.a). But there are still sharp changes in intensity, against the smoothly varying ground assumption, which in turn fails the segmentation process (Figure 30.b). In Figure 30.c these sharp changes are smoothed out therefore segmentation result is successful (Figure 30.d). If there is high amount of noise in the image / volume then these sharp changes must be smoothed only locally. Pre-processing may differ from modality to modality, for example MRA or MRV images mostly do not require any pre-processing.

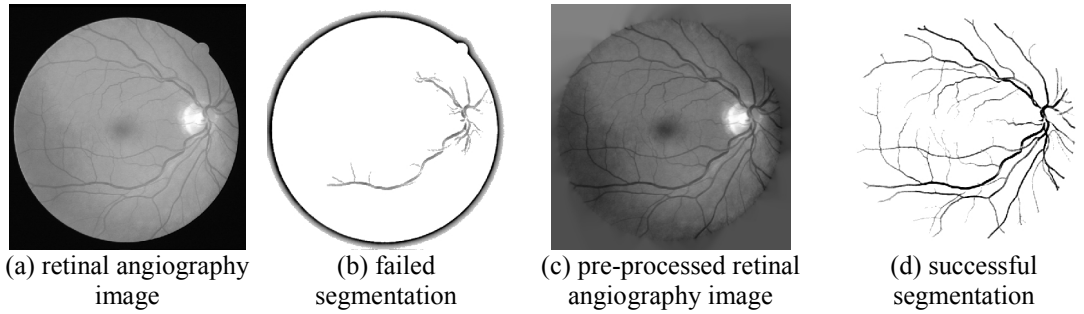


Figure 30: Slowly varying ground assumption and its affects.

Step 6. Pumping water from water source(s)

Water is added to the system at each time step using the equation below

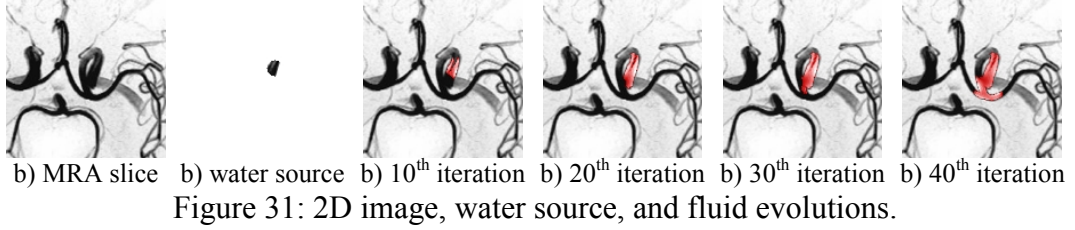
$$d_{x,y,z} = (d_{x,y,z} + W_{x,y,z}^{(source)}) / 2 \quad (32)$$

where $d_{x,y,z}$ is the water depth at voxel (x,y,z) . This schema assures pumping reasonable amount of water in to the vessel system hence slow velocity assumption of LSWE holds. Water source is also adaptively modified using below equation so that convergence becomes faster.

$$W_{x,y,z}^{(source)} = \max(W_{x,y,z}^{(source)}, d_{x,y,z}) / 2, \quad \text{for } d_{x,y,z} > W_{AVD} \quad (33)$$

where W_{AVD} is average vessel depth in rough estimation.

Water given into the system is limited with the height of vessel radius estimated in rough segmentation. This schema prevents water to flood non-vessel regions in large amount but small amount of water flood is always possible and must be controlled using viscosity and vaporization schema. This approach assures existence of reasonable amount of water in the system for homogenous water flow (Figure 31)..



Step 7. Calculating the LSWE time step and recalculating surface

ADI method is used for integration of (Equation 22) and (Equation 23) for 2D and 3D images respectively. Integration of rows on the initial 2D image and integration of columns on the resulting 2D image using ADI method and a tridiagonal matrix solver, such as TDMA algorithm, are shown in Figure 32. For both 2D and 3D images we need to solve quite a few tridiagonal systems.

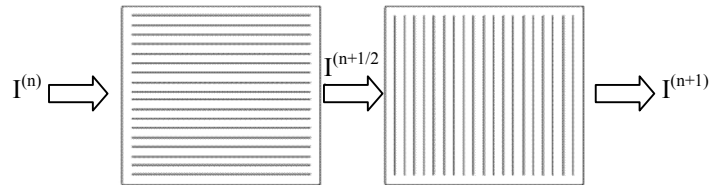


Figure 32: ADI method on 2D image with TDMA as solver

A list of the simulation parameters are given below in Table 3.

Table 3: List of simulation parameters

Parameter	Unit	Description	Default
\mathcal{Z}_{2D}		Integration order for 2D ($x-y, y-x$)	
\mathcal{Z}_{3D}		Integration order for 3D ($z-x-y, y-x-z, x-y-z, z-y-x, y-z-x, x-z-y$)	
Δt	Second	Time step for simulation ($0 < \Delta t \leq 0.1$)	$=10^{-2}$
Δx	Meter	Separation between samples along the x-direction ($\Delta x > 0$)	$=10^{-3}$
Δy	Meter	Separation between samples along the y-direction ($\Delta y > 0$)	$=10^{-3}$
Δz	Meter	Separation between samples along the z-direction ($\Delta z > 0$)	$=10^{-3}$
Δh	Meter	Height of each pixel / voxel ($\Delta h > 0$) [for 8 bit images]	$\approx 10^{-4}$
τ		Artificial viscosity (water damping coefficient) ($0 \leq \tau \leq 1$)	
V_d	Meter	Vaporization depth ($0 \leq V_d \leq 1$)	
V_r		Vaporization ratio ($0 \leq V_r \leq 1$)	

Time step (Δt) must be small enough so that the finite difference assumption $\Delta t \rightarrow 0$ holds. If time step is too small, the numerical solution is unstable and simulation times are too long. Experiments show that time step values between 10^{-3} and 10^{-2} are reasonably small and provide satisfactory numerical accuracy with acceptable simulation times. Δx , Δy , and Δz values must be small so that finite difference assumption holds. Δh must be small enough compared to Δx , Δy , and Δz values so that shallow water assumption holds.

When artificial viscosity is low, tiny amount of water may emerge in disconnected areas (Figure 33.c, gray areas). In order to circumvent this problem, we devised a ‘vaporization scheme’ (Figure 33.b). When water depth (d_i) is less than a previously defined vaporization depth (V_d) the depth is reduced by a vaporization ratio (V_r):

$$d_i = d_i - \begin{cases} d_i V_r & \text{if } d_i \leq V_d \\ 0 & \text{otherwise} \end{cases} \quad (34)$$

If V_r is equal to 1 then vaporization reduces to a simple thresholding operation. Entire water height (d_i) at n^{th} iteration is defined as $H(n)$. Vaporization may not be required if the noise in image is inconsiderable and contrast in image is.



(a) $\tau = 0.50$ no vaporization (b) $\tau = 0.25$ vaporization (c) $\tau = 0.25$ no vaporization

Figure 33: Effect of viscosity and vaporization schema.

There are two advantages of using viscosity value less than *average viscosity* with vaporization schema; the first advantage is that fluid evolution becomes faster and LSWE able to capture thin vessels with low viscosity values. This is one of the powerful aspects of LSWE segmentation since viscosity parameter and vaporization schema helps to control what will be the thinnest vessel in final segmentation (Figure 34). Viscosity and vaporization can be used together for severe level of noise.



vaporization is used, τ is, (b) 0.50, (c), 0.75, and (d) 1.00



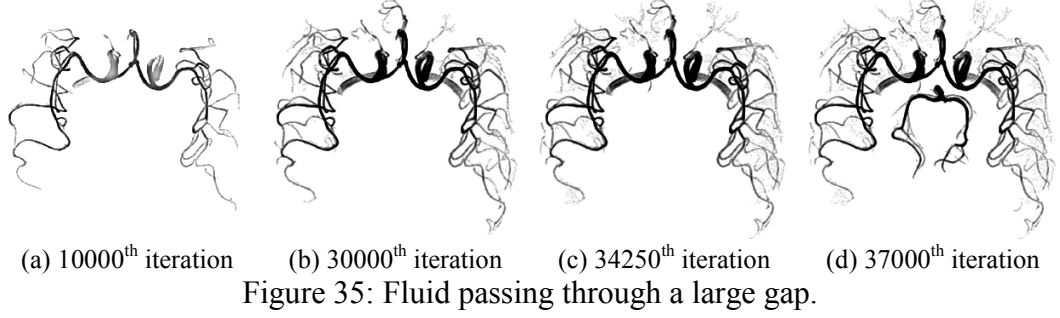
vaporization is not used, τ is (d) 0.25, (e) 0.50, (f), 0.75, and (g) 1.00

Figure 34: Analysis of viscosity and vaporization schema.

As seen in Figure 34, if viscosity is less than *average* viscosity (0.5), use of vaporization effects final segmentation considerably. While viscosity values are greater than or equal to average viscosity, use of vaporization differs less. Otherwise, vaporization acts as regularization. As viscosity and vaporization increase, LSWE fails to pass through gaps. If vessels contain gaps and we want the algorithm to pass through gaps then we would need to work with lower viscosity values along with vaporization to prevent tiny spills of water.

Occasionally, due to numerical round off errors, the surface of water can be lower than ground height which may cause negative volumes. Kass and Miller suggested clamping these negative volumes to zero and then recalculating the surface heights and normalizing total volume so that no volume loss occurs. This simple approach causes water transportation problem which can be prevented by tracking each fluid region and the associated volume separately. In our case, this is not a problem, rather a desired situation which helps fluid to fill regions beyond the gaps after it passes through it. In Figure 35.b upper arterial system is filled with fluid and after a certain time, the area does not expand anymore, instead height of fluid increases. When fluid

finds a channel to pass through bottom vessel system (Figure 35.c) even if this channel is very thin in size, fluid is transported very rapidly because of water transportation phenomena (Figure 35.d). Below examples are given for 2D images to illustrate experiments better but all these notions are also valid for 3D volumes.



Step 8. Calculating the error

In each iteration, smooth water height $\tilde{H}^{(n)}$ is found as follows:

$$\tilde{H}^{(n)} = (\tilde{H}^{(n-1)} + H^{(n)})/2 \quad (\text{Equation 35})$$

where $H^{(n)}$ is water height calculated using LSWE integration in step 7, $\tilde{H}^{(n-1)}$ is blended water height in prior iteration, and $\tilde{H}^{(0)} = 0$. Error is the mean absolute difference between blended water height ($\tilde{H}^{(n)}$) and rough segmentation (R).

$$E_{2D}^{(n)} = \frac{\frac{1}{n} \sum |\tilde{H}_{x,y}^{(n)} - R_{x,y}|}{E_{2D}^{(0)}} \quad (\text{Equation 36})$$

$$E_{3D}^{(n)} = \frac{\frac{1}{n} \sum |\tilde{H}_{x,y,z}^{(n)} - R_{x,y,z}|}{E_{3D}^{(0)}} \quad (\text{Equation 37})$$

where R is the rough segmentation. In first iteration there is no water in the system so error has its highest value (Figure 36.e). Both $E_{2D}^{(n)}$ and $E_{3D}^{(n)}$ normalized to 1 by dividing the error with the error of the first iteration (Figure 36). As fluid evolves error decreases (Figure 36b-c) and when error become global minimum best segmentation is obtained (Figure 36.d).

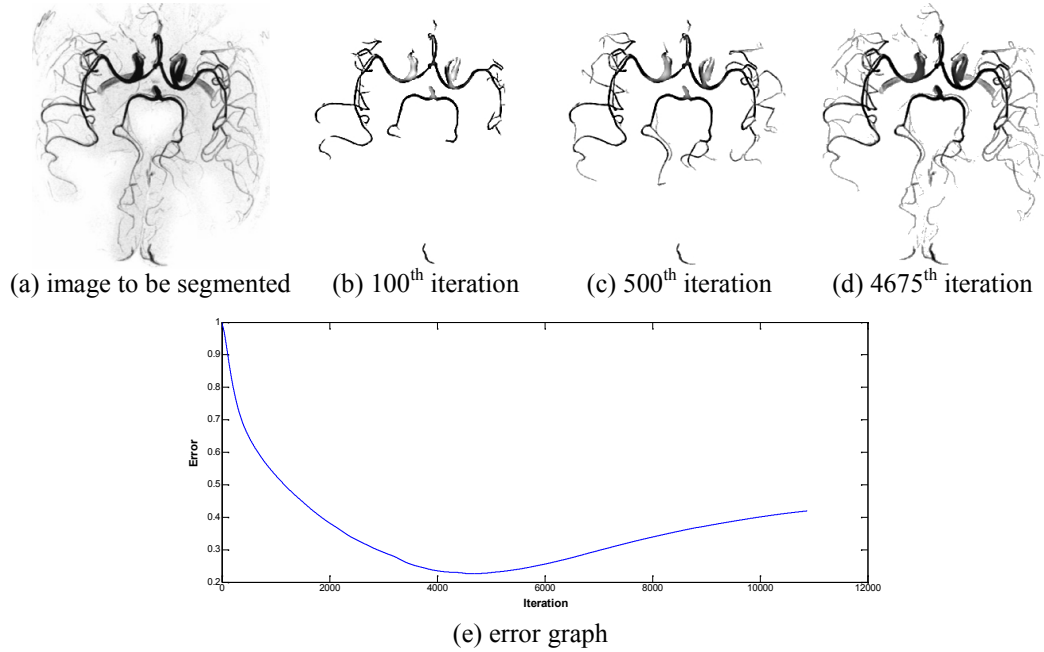


Figure 36: Error graph for 2D segmentation.

Error graph may change considerably if rough segmentation, water source or simulation parameters are changed. In Figure 37 segmentation using single water source is shown where convergence took 45000 iterations and there are 3 minimums. In Figure 38 thick vessels as water source is shown where convergence took 30000 iterations and there is only a global minimum.

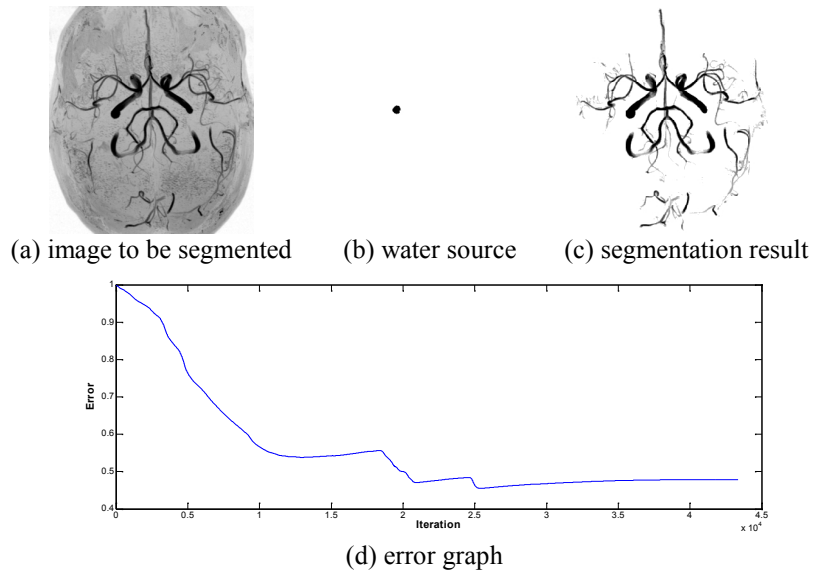


Figure 37: Error graph for single water source using scale space approach.

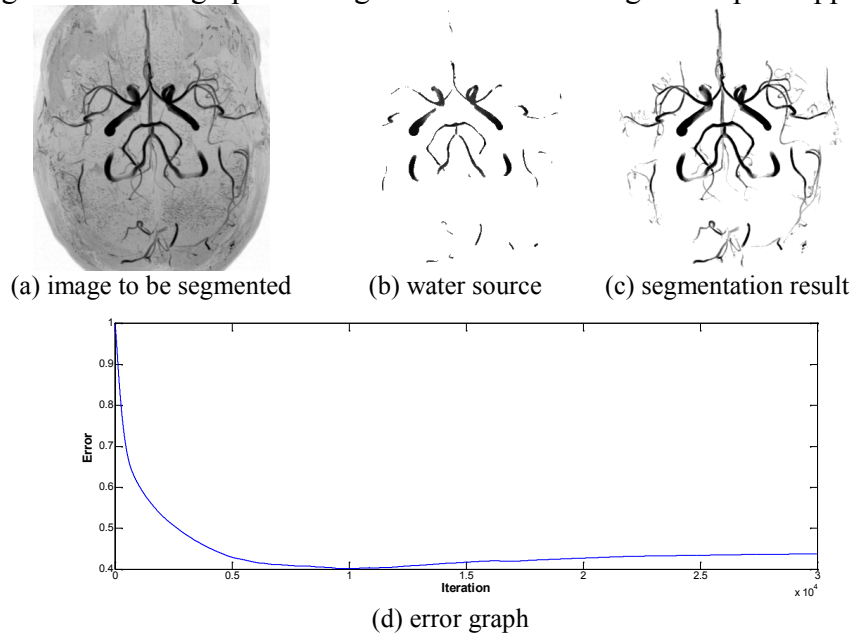


Figure 38: Error graph for water sources using filter-based approach.

Step 9. Detecting possible minimums

An error term is used to detect convergence of the algorithm, but at the convergence point error may be far from being the minimum (Figure 39).

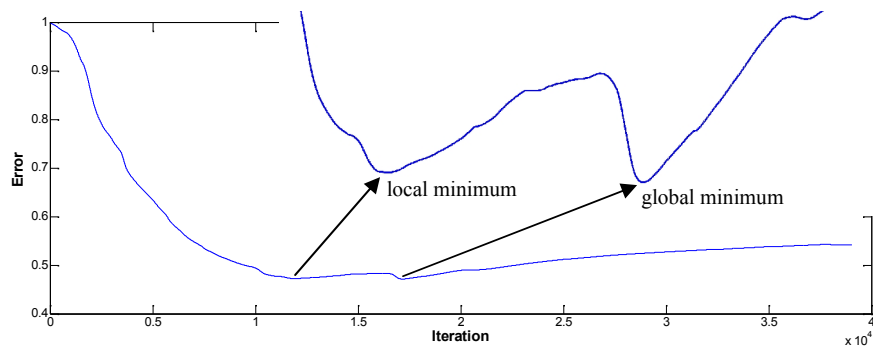


Figure 39: Error graph with two local minimums.

So during iterations if an error becomes a minimum it must be detected and result must be stored as the best result for that time instance. However we can only detect a minimum after the error increase, which is detected at the next iteration, after the real minimum point is already passed. Since there may be tiny oscillations in the error

this simple approach will lead to detection of many false possible minimums. At each possible minimum, the current result needs to be copied as the best result this will lead to large number of memory copy operations. This issue can be addressed by allowing copying only after checking a tolerance value. If the error is in an upward direction, which means there is already a detected minimum error, (Equation 38) can be used to detect a possible minimum

$$\frac{E^{(n)} - E_{\min}^{(m)}}{E^{(0)} - E_{\min}^{(m)}} > M_{tolerance} \quad (\text{Equation 38})$$

where $M_{tolerance} = 10^{-2}$ is the tolerance value for detection of possible minimums, $E^{(n)}$ is the error at n^{th} iteration, $E^{(0)}$ is the maximum error, and $E_{\min}^{(m)}$ is the minimum error found at the m^{th} iteration. This method does not give the minimum point but it gives a point in a few steps ahead which causes an ignorable difference in segmentation result. (Equation 38) will become true only for once if the error graph is smooth and contains only a global minimum. Error may contain small oscillations so (Equation 38) is satisfied 1-5 times which is still reasonably a small amount. If the tolerance value is increased then the number of possible minimum detection will be more immune to oscillations but the difference between the real minimum and possible minimum will also increase. If error becomes stationary (Equation 38) will always produce a false value and algorithm will never able to converge (Figure 38, after 25000th iteration). This issue can be resolved by detecting the stationary error via

- keeping the last K (e.g. 1024) errors (efficiency is obtained using modulo operator),
- dividing these errors into 4 groups and calculating average error for each group,
- concluding stationary error and also a possible minimum if error average between the groups is less than a tolerance value (e.g. 10^{-5}).

Step 10. Testing for the convergence

Error may contain more than one local minimums hence there may be local minimums (Figure 40.b) before the global minimum is reached (Figure 40.c). Finding the global minimum is not guaranteed if the algorithm stops as it reaches a

minimum point. There may be also many small local minimums due to small oscillations in the error. To alleviate such problems a tolerance value is used to decide the convergence in an efficient way using a convergence criterion given as

$$\frac{E^{(n)} - E_{\min}^{(m)}}{E^{(0)} - E_{\min}^{(m)}} > C_{tolerance} \quad (\text{Equation 39})$$

where $C_{tolerance}$ is the convergence tolerance ($0 < C_{tolerance} \leq 1$), $E^{(n)}$ is the error at n^{th} iteration, $E^{(0)}$ is the initial error (maximum error), and $E_{\min}^{(m)}$ the minimum error found until the m^{th} iteration. Best segmentation occurs when error decreases to global minimum. In Figure 39, there are 2 local minimums where last local minimum is also a global minimum. In Figure 36 there is only one local minimum which is also a global minimum.

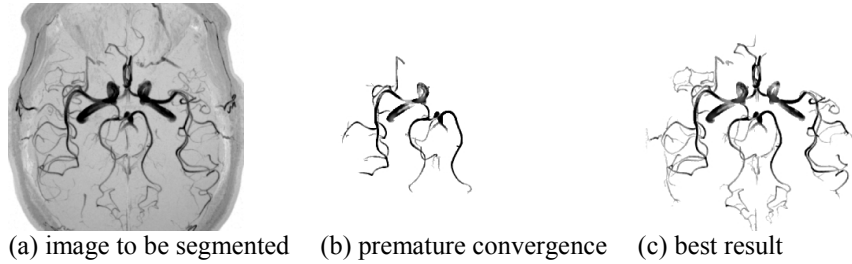


Figure 40: Premature convergence example.

Tolerance value must be greater than certain level to guarantee not being trapped in local minimums or small oscillations in the error. Tolerance value of 0.25 is sufficient to overcome most of the gaps while keeping the number of iterations in a reasonable number. In Figure 41 segmentation results and corresponding error graphs are shown (water source via filter-based approach and tolerance value of 0.25).

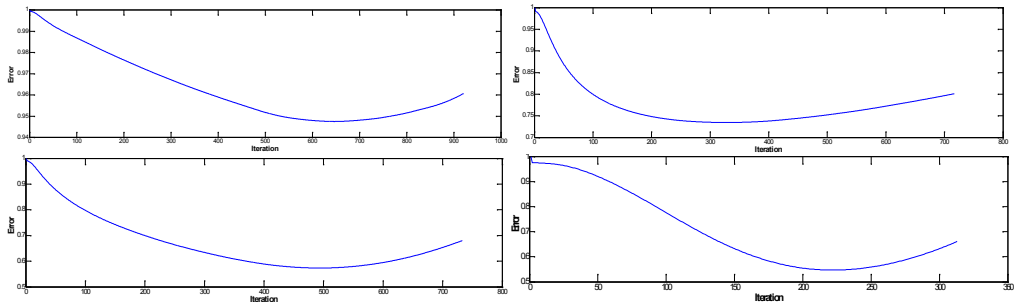


Figure 41: Example error graphs.

Step 11. Post processing the best result

Resulting segmentation may contain tiny fluid columns which can also interpreted as background noise. Especially with low viscosity values and without using vaporization, segmentation result is more prone to deteriorated by such noises. To eliminate such noises and enhance vessels, the approach shown below can be used:

- Small fluid columns are thresholded using T_p parameter (Figure 42.a).
- Gaps are recovered using coherence-enhancing diffusion filter (Figure 42.b).
- Smoothed regions (due to coherence-enhancing diffusion) are thresholded.
- Disconnected and isolated regions which are small in size are removed.

Although above procedure contains many steps “step a” is enough for most of the modalities if noise level in the image is not very high.



Figure 42: Post-processing example.

Step 12. Saving / displaying the best result

LSWE best segmentation result (at global minimum), sub-optimum segmentation results, thresholded result, coherence-enhancing diffusion filtered result, and its binarized version are stored in the disk for later analysis or display purposes.

Simulation steps for BrainWeb volume are briefly shown in Figure 43.

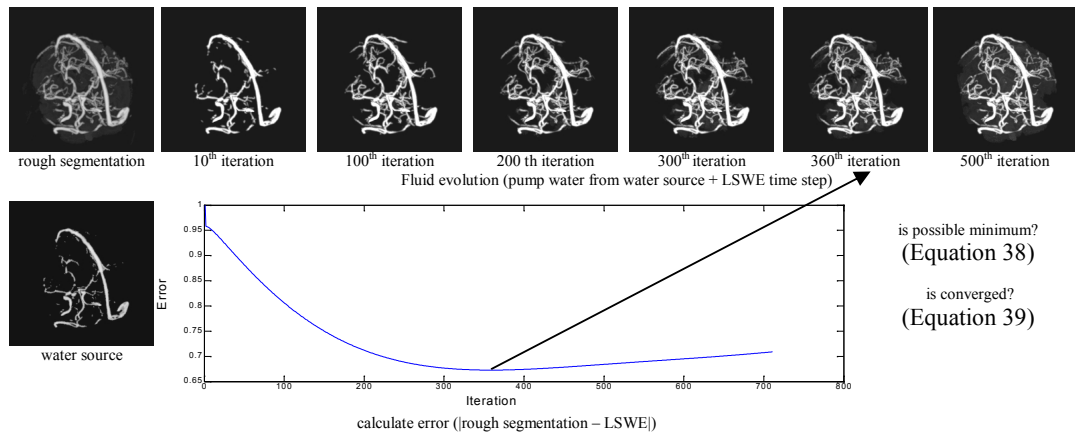


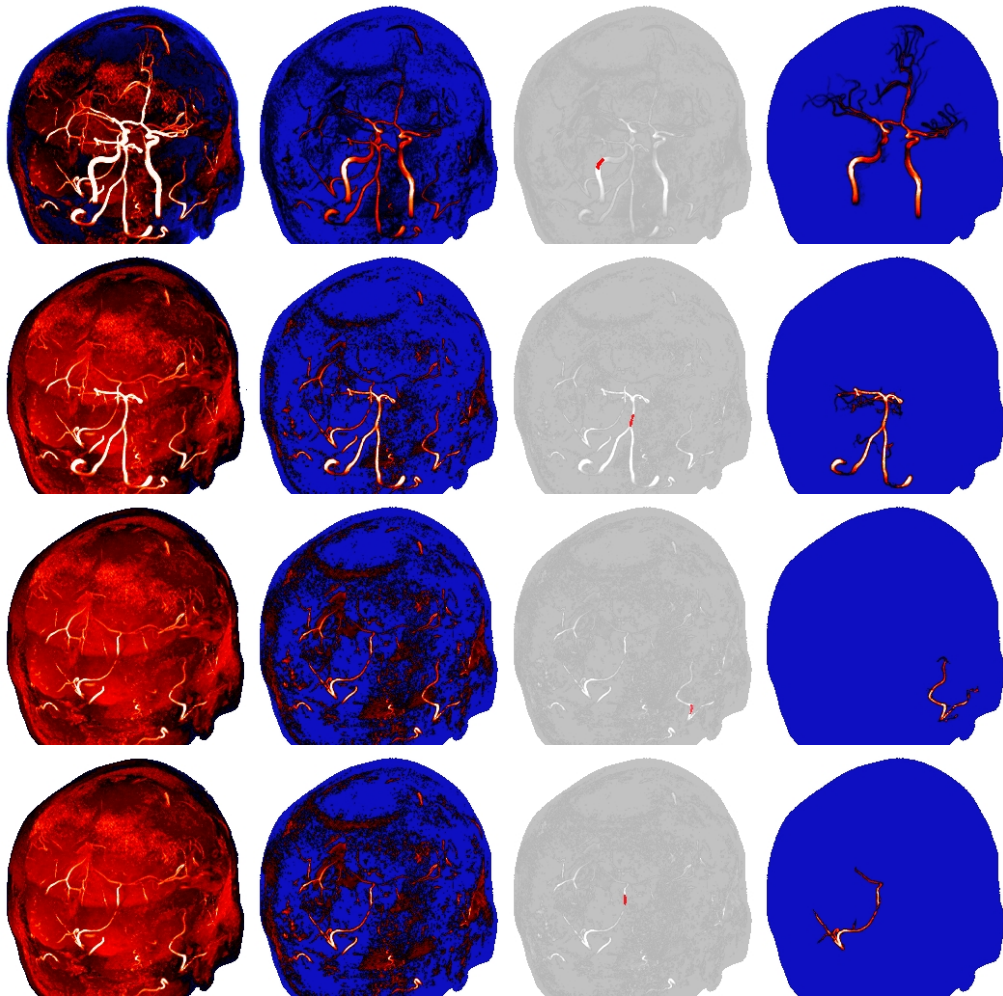
Figure 43: Simulation steps.

3.6 Disconnected Vessel Systems

If there are more than one disconnected vessel systems having large gaps between them, then a single water source may fail to segment all of the vessel structures. There are two possible alternatives for overcoming this problem

- a) using water source via filter-based approach.
- b) starting another water pump, when one area is completely segmented.

In Figure 44 MRA algorithm is executed four times on the same volume where each time extracted vessel structures are removed from MRA volume for next run.



Top to bottom: first execution, second execution, third execution, fourth execution
Left to right: MIP of MRA, rough segmentation, water source (red), segmentation result

Figure 44: Multiple, four, execution with different water sources.

3.7 Calibration of Parameters

There are 13 parameters (Table 4) which affect the final segmentation and each parameter must be tuned for each modality such as brain MRA, brain MRV, retinal angiography ... etc.

Table 4: List of algorithm parameters

Parameter	Belongs to	Description	Value range	Default value	Approach
r_v	Rough segmentation	Maximum vessel radius	$1 < r_v$	5	Manual
T_{rs}	Rough segmentation	Noise removal threshold	$0 \leq T_{rs} \leq 1$	0.1	Manual
r_w	Water source	Thin structure suppressing radius	$1 < r_w$	1	Manual
T_{ws}	Water source	Noise suppressing threshold	$0 \leq T_{ws} \leq 1$	10^{-2}	Manual
Δt	Fluid evolution	Time step for simulation	$0 < \Delta t \leq 0.1$	10^{-3}	User choice
$\Delta x, \Delta y, \Delta z$	Fluid evolution	Separation between samples	$\Delta x, \Delta y, \Delta z > 0$	10^{-3}	Fixed
Δh	Fluid evolution	Height of each pixel / voxel	$\Delta h > 0$	10^{-4}	Automatic
τ	Fluid evolution	Artificial viscosity	$0 \leq \tau \leq 1$	0.5	User choice
V_d	Fluid evolution	Vaporization depth	$0 \leq V_d \leq 1$	10^{-2}	Automatic
V_r	Fluid evolution	Vaporization ratio	$0 \leq V_r \leq 1$	0.5	Automatic
$M_{tolerance}$	Fluid evolution	Possible minimum tolerance	$0 < M_{tolerance} \leq 1$	10^{-2}	User choice
$C_{tolerance}$	Fluid evolution	Convergence tolerance	$0 < C_{tolerance} \leq 1$	0.25	User choice
T_p	Post-processing	Noise removal threshold	$0 \leq T_p \leq 1$	10^{-2}	Fixed

Optimizing these parameters for each modality is very difficult since search space is very large and nonlinear. There are 3 approaches for choosing optimum parameters:

- Manually minimizing segmentation error with respect to parameters
- Automatically minimizing segmentation error with respect to parameters
- Combining manual and automatic approaches

First we have to define an error metric such as Jaccard distance or weighted sensitivity and specificity or F-measure. Jaccard index is defined as

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (\text{Equation 40})$$

where A is the binary map of segmentation result, and B is the binary map of ground truth. When perfect segmentation is obtained A equals to B hence Jaccard index equals to 1. When segmentation is completely fails A intersection B is zero hence Jaccard index equals to 0. If we maximize Jaccard index with respect to the parameters in Table 4, optimum segmentation parameters can be obtained.

Sensitivity and specificity analyses can be used to understand the effect of the parameters on the algorithm. Sensitivity (recall) and specificity (precision) are calculated as

$$specificity = \frac{TN}{TN + FP} \quad (\text{Equation 41})$$

$$sensitivity = \frac{TP}{TP + FN} \quad (\text{Equation 42})$$

where TN is number of true negatives, TP is number of true positives, FN is number of false negatives, and FP is number of false positives. For two class problem the sensitivity and specificity values are dependent to algorithm parameters such as threshold. If specificity is 1 (100%) then algorithm is able to detect all non-vessel structures. If sensitivity is 1 (100%) then algorithm is able to detect all vessel structures. When sensitivity and specificity both equals to 1 then 100% segmentation success is obtained. If we know optimum parameters values then sensitivity and specificity analyses for each parameter around that optimum point can done in order to understand behavior of each parameter.

We can combine sensitivity and specificity as a correlation coefficient CC

$$CC = \frac{N(sensitivity) \times (specificity) - TP}{\sqrt{(N(sensitivity) - TP)(N(specificity) - TP)}} \quad (\text{Equation 43})$$

where N is the number of data ($TP+TN+FP+FN$), and sensitivity and specificity are different than zero (Baldi et al 2000). We can combine sensitivity and specificity as a performance measure as

$$F = 2 \times \frac{sensitivity \times specificity}{sensitivity + specificity} \quad (\text{Equation 44})$$

which is called as F-measure (Rijsbergen 1979). F-measure (Equation 43) can be maximized with respect to algorithm parameters in Table 4.

Parameters in Table 4 can be optimized using stochastic methods such as simulated annealing or genetic algorithm. In case of initial values for parameters being close to global optimum Powell method or simplex method can also be used.

To decrease the search space, the parameters in Table 4 are divided into 4 groups; manual, automatic, user choice, and fixed. Rough segmentation and water source parameters can be found using visual investigations on data and by trial and error. Vessel radius must be set as maximum vessel radius in the target anatomical place and target modality. For example $r_v = 3$ will be sufficient for brain MRA volumes. Noise must be decreased but not completely eliminated (Figure 25.f-g, Figure 26.f-g) in rough segmentation. Trying to threshold all noise values will cause lack of thin and low contrast structure in rough segmentation and also in segmentation error term since low contrast vessels hard to differentiate from noise. Water source must contain major vessel components and no non-vessel components at all. In rough segmentation there are low contrast and thin vessel structures mixed with noise. In water source detection, all noise must be eliminated so that low contrast and thin structures are also eliminated for the sake of that purpose (Figure 29). Time step (Δt) can be set between 10^{-2} and 10^{-3} since other values may cause very long simulation times or degraded numerical accuracy. Separation between samples can be fixed such as 1 meter. Water has to be shallow therefore fixing separation between samples and trying to optimize Δh parameter will be sufficient for that purpose. Viscosity can be selected high if thin vessel structures are not needed. To be able to capture thin and low contrast vessel structures low viscosity values can be preferred. But low viscosity may cause segmentation results to become noisy. As an initial guess, 0.5 is a proper value for viscosity. User can increase or decrease viscosity according to results of the segmentation and his or her own needs. Vaporization depth and vaporization ratio are critical since they determine how large the gap fluid can pass through and level of noise elimination. High vaporization decreases the noise level in segmentation result but at the cost of missing low contrast and thin structures and depleted ability of passing through gaps. Noise removal in post-processing must be minimal if low contrast and thin structures need to be preserved such as 10^{-3} . Explained approach allows us to optimize only 3 parameters automatically which shrinks the search space considerably.

CHAPTER 4

CPU AND GPU IMPLEMENTATIONS

4.1 CPU Implementation

All the implementations are done using C++ since it produces efficient and fast running executable. Using C++ *typedef* keyword new type definition for the floating variables (float, double) is created with the type name “number”. If “number” has type of float whole simulation runs in single precision and if “number” has type of double whole simulation runs in double precision. Experiments show that single precision is approximately 20% faster comparing to double precision with no observable extra round of error. For this reason and 50% memory saving we prefer to use single precision within our implementations. All the other performance optimizations are carefully implemented to achieve maximum possible execution speed such as efficient use of pointers, and minimizing cache misses.

Keeping track of computational domain (CD) and solving LSWEs within this domain is one of the most effective optimization we developed (Figure 45) which provides 1.5-2 times speedup. The computational domain is found dynamically by expanding the bounding box of wet region by 1 pixel / voxel. Since time step, as well as the amount of water pumped in to the system is small, fluid can only expand 1 pixel or voxel in each time step (Figure 45).

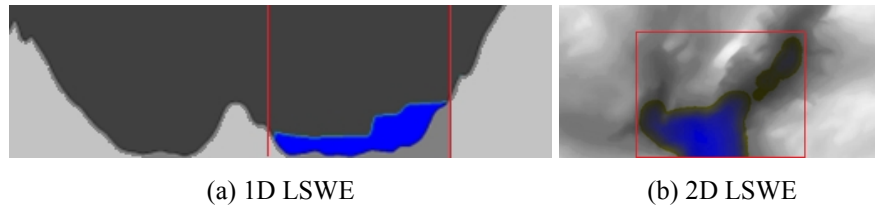
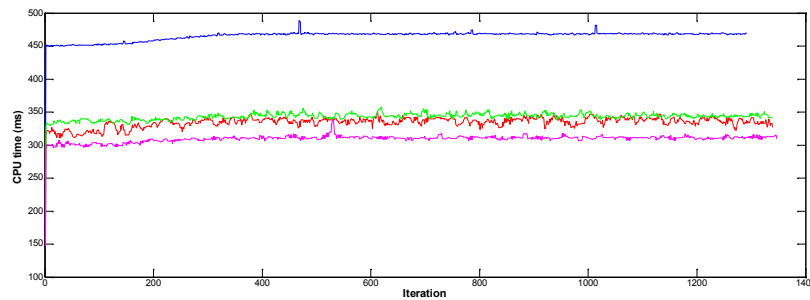


Figure 45: Dynamic tracking of computational domain (CD)

Multi-core implementations are realized using OpenMP which is specifically designed for scientific computation such as array processing. Figure 46 shows CPU times for Intel i5 3.2 GHz processor which has got 2 cores with 4 hardware threads. Speedup in 4 threads is 2 twice compared to single thread implementation. 2 and 3 threads provide speedups close to 4 threads. For Intel i7 processor with 8 threads speedup factor is 5 times comparing to single thread implementation. Although our algorithm can utilize hundreds of CPU threads, this is only possible through server categorized hardware which is an expensive solution.



4 threads (blue), 3 threads (green), 2 threads (red), single thread (magenta)

Figure 46: 1-4 threads and CPU times during fluid simulation.

CPU codes are efficiently implemented but simulation times are still high since

- Time step varies between 10^{-2} and 10^{-3} which causes thousand iterations.
- Large number of 1D tridiagonal linear systems must be solved at each time step:
 - 2D: $300 \times 400 \rightarrow 400$ tridiagonal system + 300 tridiagonal system resulting in 700 tridiagonal systems at each time step.
 - 3D: $200 \times 220 \times 150 \rightarrow 200 \times 220 + 200 \times 150 + 200 \times 150$ resulting in 104000 tridiagonal systems at each time step.

4.2 CUDA Implementation

Since CPU architectures are not parallel in nature, even multi-core CPUs are not able to scale for highly parallel algorithms. Since LSWE has linear computational complexity, simulation time increases proportional to the size of the image or volume. One way to increase speed is to use of parallel processing using multithread CPU implementation but a better way is using streaming processor implementation such as Nvidia Compute Unified Device Architecture (CUDA) technology.

4.2.1 CUDA Overview

CUDA is a parallel computing architecture which allows utilization of streaming processing power of Nvidia graphics cards using supported development platforms such as C/C++. C for CUDA defines minimal set of language extensions to C/C++ hence the learning curve is fast but writing an optimized CUDA code is not a trivial task. CUDA based applications run on a virtual instruction set and memory having parallel computational elements in CUDA supported GPUs. CUDA allows the developer to use CPU and GPU coding in the same code base by providing seamless integration of CPU and GPU source codes. CPUs are serial processing machines which provide few very fast threads whereas GPUs are parallel machines which execute many concurrent threads at the same time but with comparably slower speeds. CUDA performs especially well for problems which are parallel in nature such as algorithms running on 1D, 2D or 3D arrays. Processing power of CUDA supported devices increase exponentially each year (Figure 47) while they are 10 times cheaper compared to equivalent CPU based systems. Nvidia's latest technology Fermi already became mainstream as it provides enterprise computing features such as ECC support, parallel data cache technology, and better support for double precision operations.

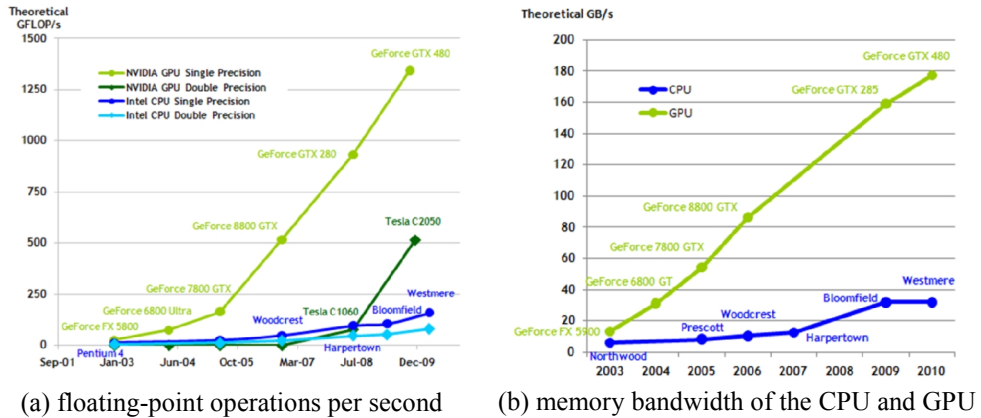


Figure 47: Exponential performance growth of CUDA versus CPU (CUDA Programming Guide).

Since LSWE is parallel in nature, CUDA is an appropriate solution to speed up the implementation. If we use TDMA for solving tridiagonal matrices, serial nature of TDMA will prevent full utilization of CUDA. We used one step Jacobi method as an approximation to solution of tridiagonal system so that all serial operations are converted into parallel operations. The Jacobi iteration may converge in one step, since simulation time step is very small and the change in fluid heights occur in a very small amounts. Experiments show that there may be small differences with TDMA and one step Jacobi for early iterations. However, in later iterations difference becomes very small and goes to zero as algorithms converges (Figure 48).

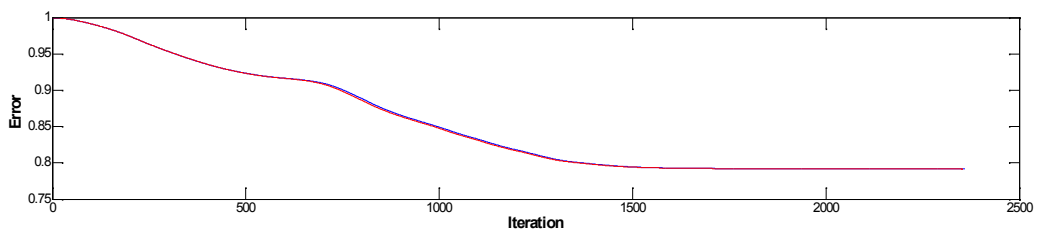


Figure 48: Error graph for TDMA and one step Jacobi.

On the other hand, one can use cyclic reduction tridiagonal solvers (Göddecke and Strzodka 2010) which are specifically developed for CUDA instead of TDMA so that parallelism can be achieved without loss of accuracy.

4.2.2 Jacobi Approximation to LSWE Solver

$Ah=y$ where y is a vector and A is a symmetric tridiagonal (positive definite and diagonally dominant) matrix which are defined in (Equation 20) and (Equation 21). The Jacobi method will always converge if the matrix A is strictly or irreducibly diagonally dominant⁴. We can define A as a sum of a diagonal entries and remainder as follows:

$$\begin{aligned}
 Ah &= (D + R)h = y \\
 \rightarrow h^{(n+1)} &= D^{-1}(y - Rh^{(n)})
 \end{aligned}
 \quad
 D = \begin{bmatrix} e_0 & & & & \\ & e_1 & & & \\ & & \ddots & & \\ & & & e_{w-1} & \\ & & & & e_{w-1} \end{bmatrix}, \quad
 R = \begin{bmatrix} 0 & f_0 & & & \\ f_0 & 0 & f_1 & & \\ & f_1 & 0 & \ddots & \\ & & & \ddots & f_{w-1} \\ & & & & f_{w-1} & 0 \end{bmatrix}$$

$$h^{(n+1)} = D^{-1}(y - Rh^{(n)}) = \begin{bmatrix} 1/e_0 & & & & \\ & 1/e_1 & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & 1/e_{w-1} \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{w-2} \\ y_{w-1} \end{bmatrix} - \begin{bmatrix} 0 & f_0 & & & \\ f_0 & 0 & f_1 & & \\ & f_1 & 0 & \ddots & \\ & & & \ddots & f_{w-1} \\ & & & & f_{w-1} & 0 \end{bmatrix} \begin{bmatrix} h_0^{(n)} \\ h_1^{(n)} \\ \vdots \\ h_{w-2}^{(n)} \\ h_{w-1}^{(n)} \end{bmatrix}$$

$$h_i^{(n+1)} = \frac{y_i - f_{i-1}h_{i-1}^{(n)} - f_i h_{i+1}^{(n)}}{e_i} \quad (\text{Equation 45})$$

where n is the iteration number, w is the number of water columns in 1D fluid, $i=0, 1, \dots, w$ is the water column index, $x_{-1}=x_w=f_{-1}=0$ due to boundary conditions, and $y_i = h_i^{(n-1)} + (1 - \tau)(h_i^{(n-1)} - h_i^{(n-2)})$ is the (Equation 21) applied on a single water column. e_i and f_i values are the ones used within (Equation 20) and (Equation 21).

In (Equation 45) each fluid column height can be calculated independently from other column heights. We only need to store $h^{(n)}$ in a separate array until calculation of all $h^{(n+1)}$ values are completed. For an arbitrary h vector an initial guess must be assigned to h and (Equation 45) must be iterated until no significant change occurs in h . For our LSWE solver initial guesses are the fluid heights which are calculated in the prior iteration. Since change in h is very small we can use (Equation 45) only one step as an approximation. (Equation 45) allows efficient use of threads and use of shared memory for coalescing memory access which is a major performance optimization (CUDA Best Practices).

⁴ Magnitude of the diagonal entry is larger than or equal to the sum of non-diagonal entries in a row

In Table 5 the computational sub steps of the algorithm and each step's implementation choice, execution count, and computational load are given.

Table 5: Algorithm sub steps and implementation status.

No	Sub step name	Implementation	Execution	Computation	Explanation
1	Load image/volume	CPU	Single	Low	Disk input/output
2	Coordinate transformation	CPU / OpenMP	Single	Low	Portable to CUDA
3	Rough segmentation	CPU / OpenMP	Single	High	Portable to CUDA
4	Water source detection	CPU / OpenMP	Single	High	Portable to CUDA
5	Pre-process	CPU / OpenMP	Single	Average	Portable to CUDA
6	Pump water	CUDA	Multiple	Average	10 CUDA kernels (Table 6)
7	LSWE integration	CUDA	Multiple	High	
8	Calculate error	CUDA	Multiple	Average	
9	Post-process	CPU / OpenMP	Single	Average	Portable to CUDA
10	Save / display result	CPU	Single	Low	Disk input/output

Table 6 shows the list of CUDA kernels used in parallel processing, such that Δh is height of a water column, Δx , Δy , Δz are pixel separation distances, Δt is LSWE integration time step, R is rough segmentation, W is water source, B is ground heights, D is water depths, D^* is blended water depths, $H^{(n)}$ is water surface heights at n^{th} iteration, H^* is temporary surface heights, N_r is volume normalization ratio, τ is viscosity, V_d is vaporization depth, and V_r is vaporization ratio.

Table 6: List of CUDA kernels used in proposed method.

Kernel name	Parameters	Explanation
intialize	$\Delta h, R, W, B, D, H^{(n)}, H^{(n-1)}, H^{(n-2)}$	Copies data from host to device
pumpWater	$\Delta t, D, W$	$D \leftarrow (1 - \Delta t)D + \Delta t W$
reduce6	S, T	Array sum with reduction: $T \leftarrow \Sigma S$
findSurfaceLevel	$B, D, H^{(n)}$	$H^{(n)} \leftarrow \max(H^{(n)}, B)$ $D \leftarrow H^{(n)} - B$
horizontalIntegration	$\tau, \Delta t, \Delta x, D, H^{(n)}, H^{(n-1)}, H^{(n-2)}, H^*$	Jacobi approximation with ADI
verticalIntegration	$\tau, \Delta t, \Delta y, D, H^{(n)}, H^{(n-1)}, H^{(n-2)}, H^*$	Jacobi approximation with ADI
depthIntegration	$\tau, \Delta t, \Delta z, D, H^{(n)}, H^{(n-1)}, H^{(n-2)}, H^*$	Jacobi approximation with ADI
prepareWaterHeights	$B, D, H^{(n)}, H^{(n-1)}, H^{(n-2)}$	$H^{(n-2)} \leftarrow H^{(n-1)}$ $H^{(n-1)} \leftarrow H^{(n)}$ $H^{(n)} \leftarrow B + D$
recalculateVolumes	$N_r, V_d, V_r, B, D, H^{(n)}, D^*$	$D \leftarrow N_r D$ volume preservation $D \leftarrow D - V_r D$ for water columns $D < V_d$ $H^{(n)} \leftarrow B + D$ $D^* \leftarrow (D^* + D) / 2$ $W \leftarrow \max(W, D^*)$
findError	D^*, R, E	$E \leftarrow R - D^* $ (sum using reduce6)

In Fermi GPUs maximum number of threads in a block is 1024 but choosing 256 or 512 threads in a block is common in use. Profiler data shows that 256 and 512 threads in a block size are good choice for horizontal, vertical, and depth integration kernels. For all other kernels number of threads in a block is 512. Thread structure and shared memory usage of integration kernels are given in Table 7.

Table 7: Integration kernels thread structure and shared memory usage.

Integration kernel	Block dimension	Thread count	Shared memory	Shared memory usage
Horizontal	32x8	256	[33x8]x3 float	3168 bytes
Vertical	32x16	512	[33x16] x3 float	6336 bytes
Depth	32x8	256	[33x8] x3 float	3168 bytes

Each integration kernel contains 5 integer parameters, 2 float parameters, 5 float pointer parameters, 5 local integer variables, 2 local float variables, 2 if statements, 4 conditional operators, 18 addition operations, 15 subtraction operations, 10 multiplication operations, 2 division operations, 1 modulo operation, and 2 synchronizations (via `__syncthreads()`).

In integration kernels, each water column is computed using (Equation 45). To compute (Equation 45) two neighbor water column heights (D) and water surface height (H) are needed. For horizontal integration and 32x8 block dimension 256 threads are created within each block. For coalescing memory accesses 3 shared memory arrays are created having size of 33x8, where last column is used as padding column for avoiding shared bank conflicts. New water surface heights are calculated for threads (1,0) – (30,7). For threads (0,0)-(0,7) and threads (31,0)-(31,7) meaning 6% of total threads no computation is done. In Figure 49 no computation is done for the threads corresponding to 0th column and 31th column. Last column is added for padding to avoid shared memory bank conflicts.

0.0	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0	11.0	12.0	13.0	14.0	15.0	16.0	17.0	18.0	19.0	20.0	21.0	22.0	23.0	24.0	25.0	26.0	27.0	28.0	29.0	30.0	31.0	32.0
0.1	1.1	2.1	3.1	4.1	5.1	6.1	7.1	8.1	9.1	10.1	11.1	12.1	13.1	14.1	15.1	16.1	17.1	18.1	19.1	20.1	21.1	22.1	23.1	24.1	25.1	26.1	27.1	28.1	29.1	30.1	31.1	32.1
0.2	1.2	2.2	3.2	4.2	5.2	6.2	7.2	8.2	9.2	10.2	11.2	12.2	13.2	14.2	15.2	16.2	17.2	18.2	19.2	20.2	21.2	22.2	23.2	24.2	25.2	26.2	27.2	28.2	29.2	30.2	31.2	32.2
0.3	1.3	2.3	3.3	4.3	5.3	6.3	7.3	8.3	9.3	10.3	11.3	12.3	13.3	14.3	15.3	16.3	17.3	18.3	19.3	20.3	21.3	22.3	23.3	24.3	25.3	26.3	27.3	28.3	29.3	30.3	31.3	32.3
0.4	1.4	2.4	3.4	4.4	5.4	6.4	7.4	8.4	9.4	10.4	11.4	12.4	13.4	14.4	15.4	16.4	17.4	18.4	19.4	20.4	21.4	22.4	23.4	24.4	25.4	26.4	27.4	28.4	29.4	30.4	31.4	32.4
0.5	1.5	2.5	3.5	4.5	5.5	6.5	7.5	8.5	9.5	10.5	11.5	12.5	13.5	14.5	15.5	16.5	17.5	18.5	19.5	20.5	21.5	22.5	23.5	24.5	25.5	26.5	27.5	28.5	29.5	30.5	31.5	32.5
0.6	1.6	2.6	3.6	4.6	5.6	6.6	7.6	8.6	9.6	10.6	11.6	12.6	13.6	14.6	15.6	16.6	17.6	18.6	19.6	20.6	21.6	22.6	23.6	24.6	25.6	26.6	27.6	28.6	29.6	30.6	31.6	32.6
0.7	1.7	2.7	3.7	4.7	5.7	6.7	7.7	8.7	9.7	10.7	11.7	12.7	13.7	14.7	15.7	16.7	17.7	18.7	19.7	20.7	21.7	22.7	23.7	24.7	25.7	26.7	27.7	28.7	29.7	30.7	31.7	32.7

Figure 49: Shared memory usage (f, x, d arrays).

4.2.3 CPU versus CUDA Comparisons

We tested our CPU and CUDA implementations using a workstation with 3.2 GHz i5 CPU (2-core, 4 hardware threads support), 4GB RAM, and Nvidia GTX 465 graphics card (GTX 590 has approximately three times more computational power). For 3D volumes, 2 threads CPU is 1.77 times faster than single thread CPU, CUDA is 45 times faster than single thread CPU and 25 times faster than 2 threads CPU. For 2D images, 4 threads CPU is 1.99 times faster than single thread CPU, CUDA is 12 times faster than single thread CPU and 6 times faster than 4 threads CPU (Table 8).

Table 8: Example iteration times for i5 CPU (3.2 GHz) and GPU (GTX 465)

Volumes	Data size (in bytes)	CPU 1 core (ms)	CPU 4 core (ms)	CPU-4 vs CPU-1	CUDA (ms)	CPU-1 vs CUDA	CPU-4 vs CUDA
224x256x220	12,615,680	2801.91	1420.71	1.97	47.12	59.46	30.15
256x256x176	11,534,336	2168.30	1406.99	1.54	43.78	49.53	32.14
256x256x155	10,158,080	1852.50	1058.14	1.75	38.43	48.20	27.53
282x281x127	10,028,047	1542.41	917.83	1.68	38.15	40.43	24.05
181x217x181	7,109,137	1203.04	630.91	1.91	27.61	43.57	22.85
240x240x120	6,912,000	970.11	614.76	1.58	26.66	36.39	23.06
132x166x137	3,001,944	498.74	256.23	1.95	12.36	40.35	20.73
				1.77		45.42	25.79
565x584	329,960	29.45	14.23	2.07	1.33	22.14	10.70
382x401	153,182	10.98	5.37	2.04	0.83	13.23	6.47
345x390	134,550	8.93	4.48	1.99	0.76	11.75	5.89
348x386	134,328	8.82	4.34	2.03	0.76	11.61	5.71
346x385	133,210	8.91	4.46	2.00	0.76	11.72	5.87
348x382	132,936	8.95	4.60	1.95	0.76	11.78	6.05
337x378	127,386	8.50	4.19	2.03	0.74	11.49	5.66
330x386	127,380	8.56	4.15	2.06	0.74	11.57	5.61
330x377	124,410	8.23	4.13	1.99	0.75	10.97	5.51
300x290	87,000	5.36	3.26	1.64	0.62	8.65	5.26
				1.98		12.49	6.27

In Table 8 iteration times is equal to (host to device copy + 100 iteration + device to host copy) / 100. Initializations such as rough segmentation, water source detection and CPU-GPU memory transfer times⁵ are negligible compared to integration cost.

⁵ 565x584 image \rightarrow (copy / integration) \rightarrow (508 + 203) / (1000 x 963) \rightarrow $7 \times 10^{-4} \rightarrow 0$

224x256x220 volume \rightarrow (copy / integration) \rightarrow (53494 + 21047) / (1000 x 37316) \rightarrow $2 \times 10^{-3} \rightarrow 0$

CHAPTER 5

RESULTS AND VALIDATION

Proposed method is analyzed and validated using synthetic data sets, clinical dataset, and public datasets containing MRA, MRV, and retinal angiography images.

5.1 Analysis of the Method

5.1.1 Vessel Shapes

To analyze the effect of vessel sizes nine 3D synthetic vessel structures divided into 3 groups are created (Figure 50). Differences between the vessels are in their thickness, curvature, and size difference in both ends. Artificial uniform background is added to all vessel structures in Figure 50.

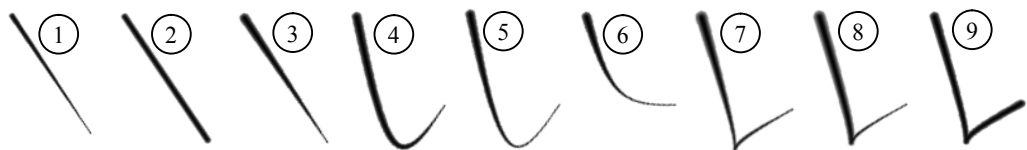


Figure 50: Various synthetic vessel structures

Segmentation result for the vessel structures in Figure 50 is analyzed under different amount of peak signal-to-noise ratio (PSNR) settings in logarithmic decibel (dB) scale are presented in Table 9 and Table 10.

Table 9: Segmentation successes for synthetic vessel structures.

	Noise free original image				PSNR=26 dB			
	Jaccard	Sensitivity	Specificity	F-measure	Jaccard	Sensitivity	Specificity	F-measure
Vessel 1	100.00	100.00	100.00	100.00	99.38	100.00	100.00	100.00
Vessel 2	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Vessel 3	100.00	100.00	100.00	100.00	99.80	100.00	100.00	100.00
Vessel 4	100.00	100.00	100.00	100.00	99.89	100.00	100.00	100.00
Vessel 5	99.93	100.00	100.00	100.00	99.70	99.93	100.00	99.96
Vessel 6	100.00	100.00	100.00	100.00	99.04	99.88	100.00	99.94
Vessel 7	90.37	90.37	100.00	94.94	73.85	73.89	100.00	84.98
Vessel 8	76.39	76.39	100.00	86.61	61.71	61.77	100.00	76.36
Vessel 9	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Average	96.30	96.31	100.00	97.95	92.60	92.83	100.00	95.69
Min	76.39	76.39	100.00	86.61	61.71	61.77	100.00	76.36
Max	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00

Table 10: Segmentation successes (Jaccard) for noisy synthetic vessel structures

	Original	PSNR=30	PSNR=24	PSNR=21	PSNR=18	PSNR=16	Average	Min	Max
Vessel 1	100.00	98.85	97.19	86.26	88.61	88.20	93.19	86.26	100.00
Vessel 2	100.00	100.00	98.05	92.86	78.20	81.63	91.79	78.20	100.00
Vessel 3	100.00	99.84	97.10	87.31	80.55	72.74	89.59	72.74	100.00
Vessel 4	100.00	99.84	97.81	74.78	74.98	72.33	86.62	72.33	100.00
Vessel 5	99.93	99.82	97.91	77.63	67.20	59.25	83.62	59.25	99.93
Vessel 6	100.00	99.70	97.16	80.28	80.79	76.77	89.12	76.77	100.00
Vessel 7	90.37	84.50	72.30	58.71	38.00	44.99	64.81	38.00	90.37
Vessel 8	76.39	72.03	62.48	47.90	37.55	38.07	55.74	37.55	76.39
Vessel 9	100.00	100.00	98.50	82.85	67.94	66.23	85.92	66.23	100.00
Average	96.30	94.95	90.94	76.51	68.20	66.69			
Min	76.39	72.03	62.47	47.90	37.55	38.07			
Max	100.00	100.00	98.50	92.86	88.61	88.20			

Some observations are as follows:

- Performance for vessel 2 is best (straight, constant vessel radius).
- Performance for vessel 1 and 3 is second best (straight, smooth shape change).
- Performance for vessel 4, 5, and 6 is well (low curvature, smooth shape change).
- Performance for vessel 9 is well (sharp curvature, constant vessel radius).
- Vessel 7 performs badly (sharp curvature).
- For vessel 8 performance become worse (sharp curvature, rapid shape change).

These observations imply that the success of the algorithm mostly depends on the amount of change in vessel radius, then the amount of noise, and least on the amount of vessel curvature. Although vessel structures in human body have low curvature and slow change in vessel radius, patients having anomalies may have unexpected vessel structures which may cause performance degradation in LSWE. The algorithm is also able to perform well up to certain amount of noise which is also sufficient for real world data. If the image contains higher amount of noise or some local sharp intensity variations, then it must be pre-processed to decrease the noise level.

5.1.2 Rough Segmentation

Rough segmentation is used for detecting water source and guiding convergence time. Therefore parameter selection is critical to obtain a good rough segmentation. Estimated vessel radius (V_r) and noise threshold ($0 \leq T_{rs} \leq 1$) determines the vessel structures and amount of noise mixed with thin vessels contained in the rough segmentation. Vessel radius can be determined by finding maximum vessel size from literature for the target anatomy. Best noise threshold can be determined by visually determining rough segmentation for few data sets to make them contain almost all thick vessels and some amount of thin vessels and noise. For the synthetic vessel structure presented in Figure 51 threshold values between 0.2 ± 0.1 produce correct segmentation results which cover the 20% of the threshold range. It shows that any T_{rs} value around optimum threshold value will yield the same segmentation result.

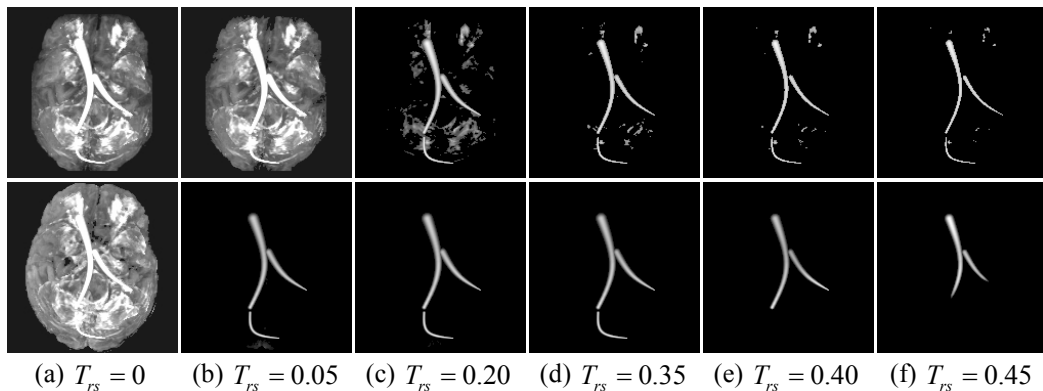


Figure 51: Rough segmentation (top row) and its effect on result (bottom row)

5.1.3 Water Shallowness

SWE assumes water columns to be shallow. However, when water columns are very shallow, then the depth of water becomes negligible compared to gravity. Therefore, maximum ground depth (Δh) must be chosen accordingly. In Figure 52, different maximum ground depth ($10^{-3} \leq \Delta h \leq 10^3$) values and corresponding average Jaccard index values for the first 5 patients (IDs: 4, 5, 6, 18, 20) in BrainWeb database is shown. As seen in Figure 52, for BrainWeb database best segmentation is obtained (Jaccard index = 88.06%) when Δh equals to 10^{-1} meter ($\log_{10}\Delta h = -1$). Segmentation performance decreases for lower and higher values than $\Delta h = 10^{-1}$.

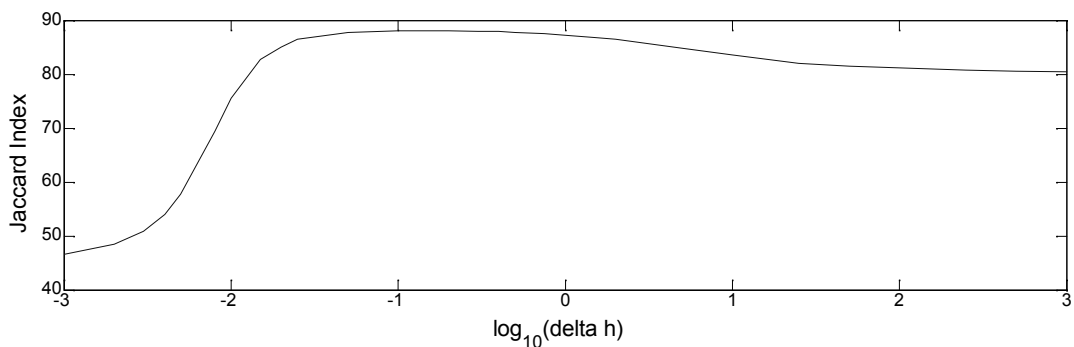


Figure 52: Segmentation success for different Δh values on BrainWeb database.

5.1.4 Time Step

Kass and Miller used an implicit method hence they claimed that numerical schema is always stable thus large time steps are allowed [27]. Within LSWE, simulation time step is divided by grid space and then squared hence two parameters lead to one parameter in essence. Small time step values result in long simulation times and premature convergence, while large time step values result in degraded numerical accuracy. Moreover, large time steps are not allowed since one step Jacobi method requires change in water heights to be small at the each iteration. In Figure 53, different time step ($10^{-2} \leq \Delta t \leq 10$) values and corresponding average Jaccard index values for the first 5 patients in BrainWeb database is shown. As seen in Figure 53 best segmentation is obtained when Δt equals to 1 second. Numerical accuracy and segmentation performance decreases for time steps deviating from 1 second.

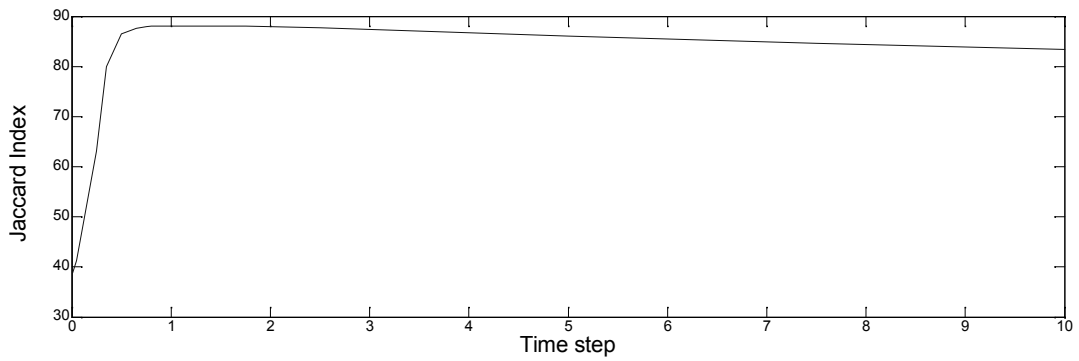


Figure 53: Segmentation success for different Δt values on BrainWeb database

An interesting phenomenon occurs when time step is greater than 10^{-2} such that fluid can flow in upward direction. To illustrate this phenomenon, a simulated vessel with smoothly changing ground height is created (Figure 54).

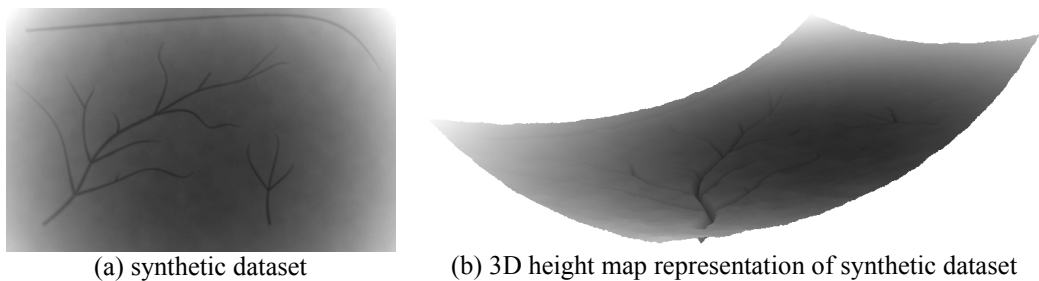


Figure 54: Synthetic 2D vessel structure on non-flat ground.

If we use small time step then water will accumulate in low regions as expected (Figure 55.a-c). But if we run the LSWE with larger time step, fluid flow will be able to move in upward direction unrealistically. This phenomenon seems very similar to surface tension which is caused by cohesive forces among the fluid molecules and result in capillary action. In cases where water source is placed in a thick vessel, fluid will be flowing through vessels even if the ground is upwards (Figure 55.d-f). But if we place the water source in a higher ground inside a thin vessel water may flood to the outside of vessel (Figure 55.g-i).

Main reason for this phenomenon is not intuitive but this observation can be still useful for vessel segmentation in retinal angiography and similar images having smoothly changing ground height (Figure 56).

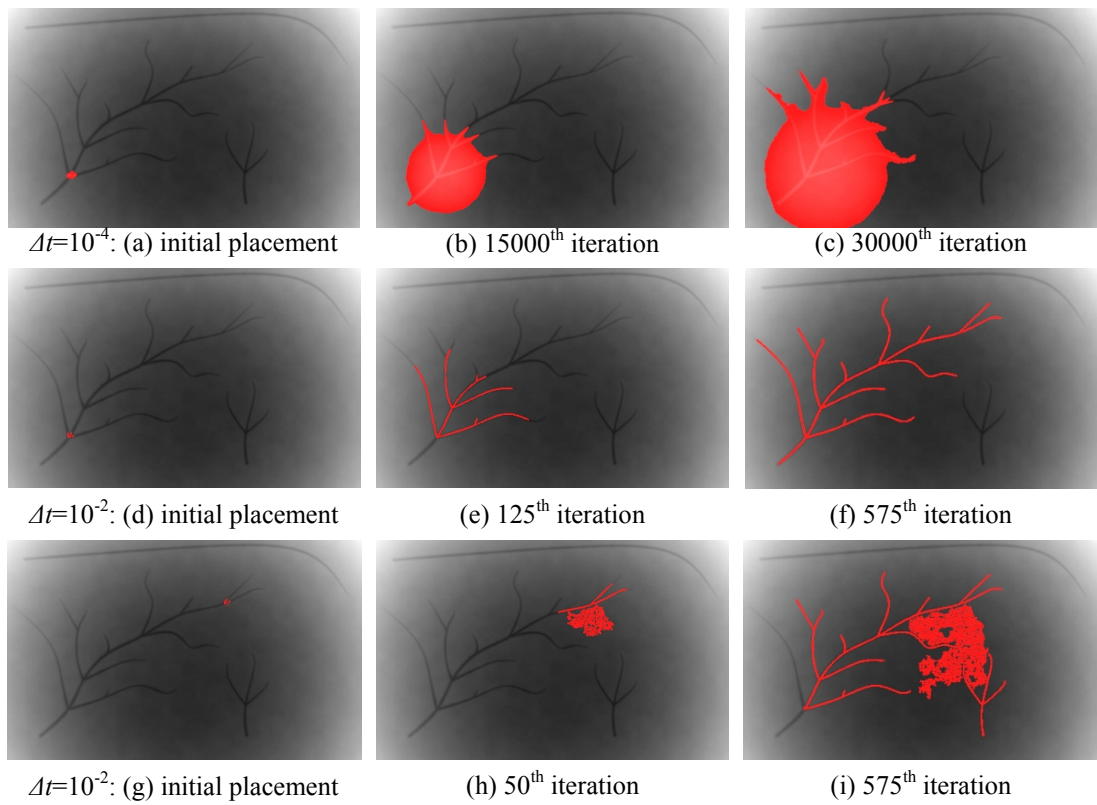


Figure 55: Effect of large time steps.

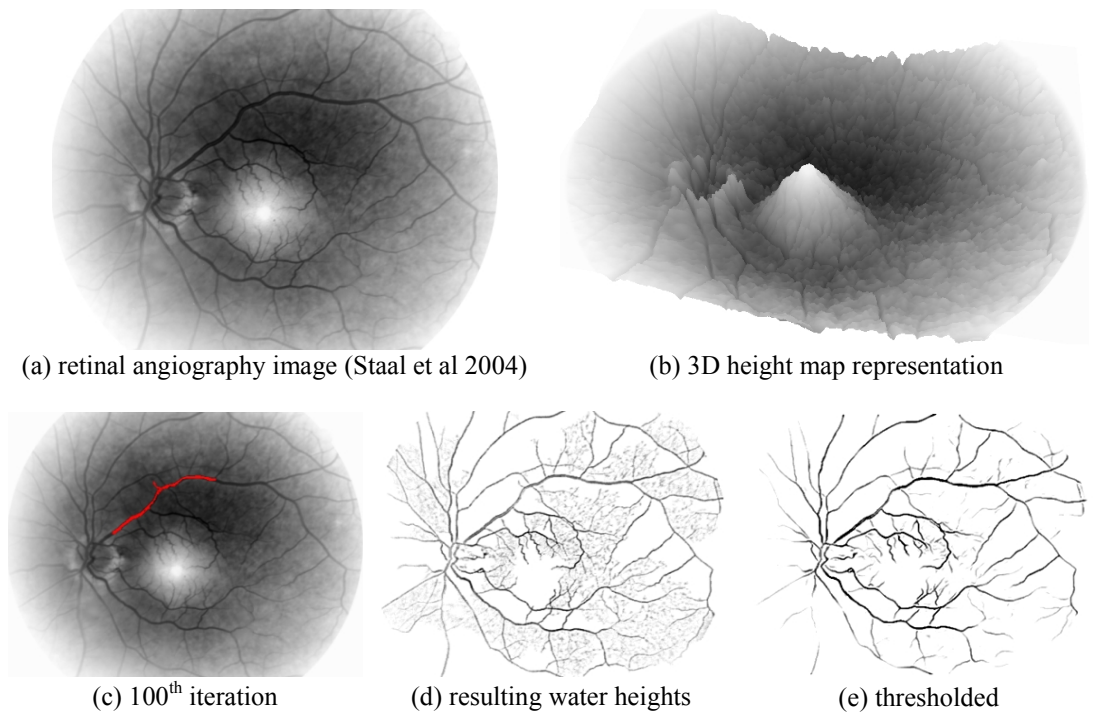


Figure 56: Vessel segmentation from retinal angiography image.

5.1.5 Viscosity

SWE and consequently LSWE models inviscid fluid flow hence (Equation 9) does not contain any term related to viscosity. Therefore (Equation 20) which is the matrix form of discretized (Equation 9) does not provide any viscous fluid flow behavior. Right hand side of (Equation 20) can be considered as extrapolation of the previous evolution's water surface. Using this fact, Kass and Miller (1990) slightly changed the (Equation 20) to create (Equation 21) by introducing damping in the extrapolation thereby they are able to add artificial viscosity into the LSWE. In Figure 57, different viscosity values and corresponding average Jaccard index values for the first 5 patients in BrainWeb database is shown. For noise free volumes, best segmentation results are obtained for $\tau=0.3$ (solid line), and for the same volumes with PSNR equals to 30 dB best segmentation results are obtained for $\tau=0.6$ (dotted line). Results for $\tau=0.3$ and $\tau=0.6$ are close to each other. For images containing noise, viscosity must be increased to create smoother fluid boundary which is less prone to noise. In that sense, viscosity acts as regularization term on fluid boundary. However, there are two down sides of using higher viscosity values; slower fluid evolution, and miss of thin structures due to smoothed boundary

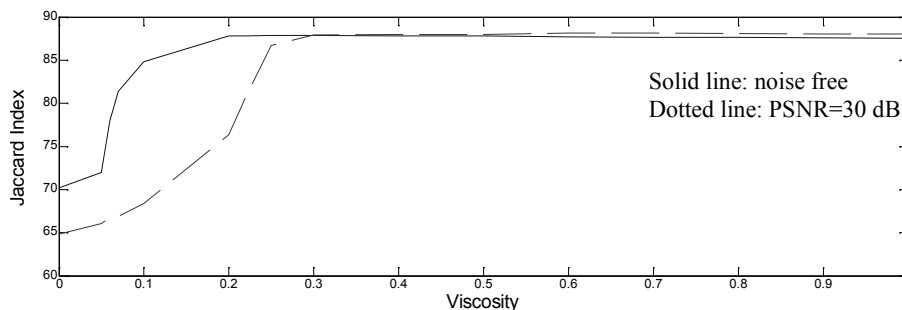


Figure 57: Segmentation success for different viscosities on BrainWeb database.

In Figure 58 segmentation results of MIP view of MRA for three different viscosity values are shown. Since MRA in Figure 58 contains few amount of noise change in viscosity parameter does not affect the segmentation result significantly. Thus, different viscosity values result in similar segmentation performance for nine 3D synthetic noiseless vein structures (Table 11).

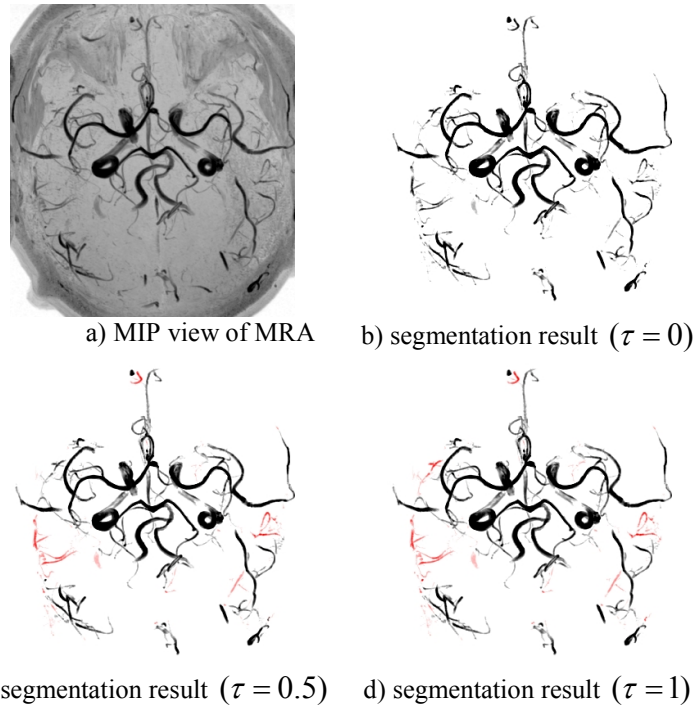


Figure 58: Segmentation results for different viscosity values.
 Black vessels: segmentation result, red vessels: difference with (b)

Table 11: Segmentation successes (Jaccard) for different viscosity levels

Viscosity	0.00	0.25	0.50	0.75	1.00	Average	Min	Max
Vessel 1	97.13	97.13	97.13	100.00	100.00	98.28	97.13	100.00
Vessel 2	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Vessel 3	99.01	99.20	99.80	100.00	100.00	99.60	99.01	100.00
Vessel 4	99.46	99.61	100.00	100.00	100.00	99.81	99.46	100.00
Vessel 5	98.47	99.05	99.34	99.93	99.93	99.34	98.47	99.93
Vessel 6	98.51	98.51	98.80	100.00	100.00	99.16	98.51	100.00
Vessel 7	91.16	92.96	92.42	90.53	90.48	91.51	90.48	92.96
Vessel 8	82.73	80.74	78.44	76.78	76.71	79.08	76.71	82.73
Vessel 9	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Average	96.27	96.36	96.21	96.36	96.35			
Min	82.73	80.74	78.44	76.78	76.71			
Max	100.00	100.00	100.00	100.00	100.00			

Viscosity values can vary spatially such that viscosity can have a linear relation with the variance of a pixel or radius of vessel or image gradient. We used variance to create a spatially varying viscosity map. For each pixel, variance within a window (e.g. 5x5) can be calculated. In edge pixels (vessel borders) and noisy pixels,

variance will be higher hence viscosity will be higher and for smooth regions viscosity will be lower. By this definition of viscosity, fluid flow will be faster in smoother regions and fluid flow will be slower in noisy regions or at vessel borders. In Figure 59.b variance map and corresponding viscosity map are shown.

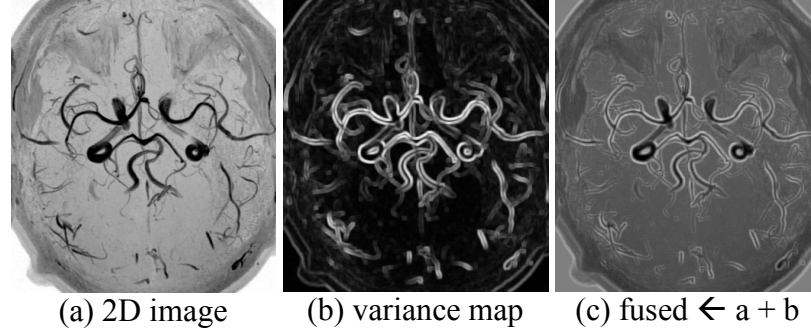


Figure 59: Image and its variance map.

In (Equation 21) we can only define a fixed viscosity value for 1D height map. Since we want viscosity value to vary spatially it must have different values for each pixel thus (Equation 21) is not suitable. In order to parallelize LSWE Jacobi approximation is used and pixel based height update formula is obtained in (Equation 45). In this equation y_i values containing independent viscosity values such that y_i is defined as

$$y_i = h_i^{(n-1)} + (1 - \tau_i)(h_i^{(n-1)} - h_i^{(n-2)}) \quad (\text{Equation 46})$$

where τ_i (alias for τ_x) corresponds to viscosity value for each water column in 1D height map. $\tau_{x,y}$ and $\tau_{x,y,z}$ is defined in 2D and 3D using ADI method applied on (Equation 45). For 2D volume $\tau_{x,y}$ is calculated using (Equation 49)

$$b_{x,y}^{(\mu)} = \frac{1}{(2k+1)^2} \sum_{\Delta x, \Delta y = -k}^k b_{x+\Delta x, y+\Delta y} \quad (\text{Equation 47})$$

$$b_{x,y}^{(\sigma)} = \frac{1}{(2k+1)^2} \sum_{\Delta x, \Delta y = -k}^k (b_{x+\Delta x, y+\Delta y} - b_{x,y}^{(\mu)})^2 \quad (\text{Equation 48})$$

$$\tau_{x,y} = \frac{(b_{x,y}^{(\sigma)} - \min b^{(\sigma)})(1 - \tau)}{\max b^{(\sigma)} - \min b^{(\sigma)}} + \tau \quad (\text{Equation 49})$$

where $2k+1$ is window size, $b_{x,y}^{(\mu)}$ is the mean of ground height at point $P(x,y)$, $b_{x,y}^{(\sigma)}$ is variance of ground height at $P(x,y)$, and τ is the minimum viscosity level. (Equation 49) maps ground height variance values between τ and 1.

There are just minor improvements (red vessels in Figure 60.c) in segmentation result with spatially varying viscosity. For image having high amount of noise, spatially varying viscosity schema can be used instead of a fixed high viscosity value for obtaining similar segmentation result in shorter time.

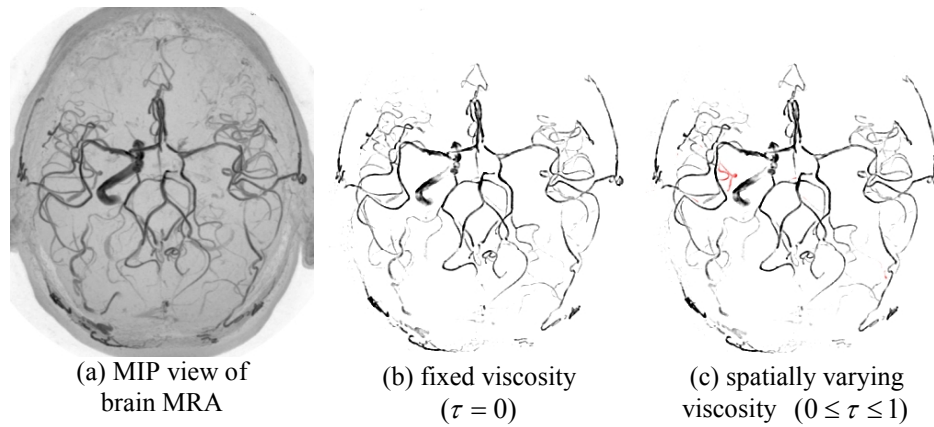


Figure 60: 2D image segmented with fixed and spatially varying viscosity.

5.1.6 Vaporization

Vaporization schema eliminates water columns which floods to non-vessel regions.

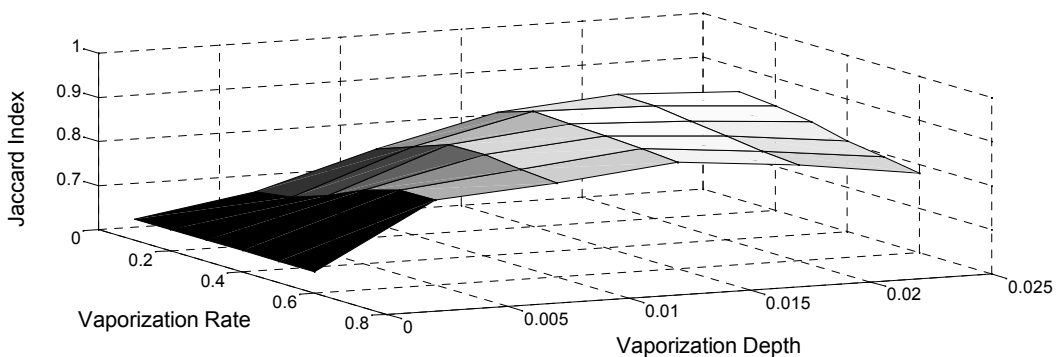


Figure 61: Segmentation success for vaporization schema on BrainWeb database.

In Figure 61, different vaporization depth, vaporization rate, and corresponding average Jaccard index values for the first 5 patients in BrainWeb database is shown. As seen in Figure 61 best segmentation is obtained (Jaccard index = 86.12%) when vaporization depth is 0.015 and vaporization rate is 0.20. Vaporization itself is not sufficient as a regularization schema but it can support viscosity to remedy high amount of noise in images.

5.1.7 Gaps in Vessels

The proposed method is able to pass through small and medium sized gaps in vessel structures due to physical flow. In Figure 62, nine 3D synthetic vessel structures with small and medium sized gaps are shown. Segmentation success for these synthetic vessel structures are shown in Table 12. Ability of passing through gaps decrease as viscosity or vaporization level is increased since fluid boundaries become smoother and fluid flows slower. For average amount of noise (PSNR = 30dB) algorithm able to produce similar performance result to zero noise (Table 12). In general, performance is good for vessels having gaps with natural curvature.

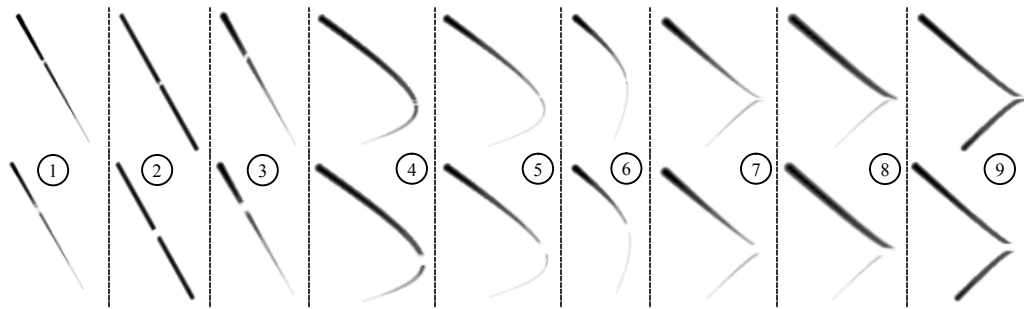


Figure 62: Synthetic vessels with; top row: small, bottom row: medium sized gaps.

There are three important parameters which controls the gap size that algorithm can pass through; a) vaporization depth and rate, b) water shallowness, and c) viscosity. Noise level in image, gap distance, vessel diameter also contributes to maximum passable gap size by LSWE. Orientation of vessel has no importance in LSWE since fluid flow is driven by surface height differences.

Table 12: Segmentation successes (Jaccard) for different noise and gap size.

	No gaps		Small gaps		Medium gaps	
	Original	PSNR=30 dB	Original	PSNR=30 dB	Original	PSNR=30 dB
Vessel 1	100.0	99.3	100.0	98.4	100.0	98.6
Vessel 2	100.0	100.0	99.9	100.0	99.9	100.0
Vessel 3	100.0	99.7	100.0	99.5	100.0	99.7
Vessel 4	100.0	99.8	100.0	99.7	100.0	99.8
Vessel 5	99.9	99.2	95.8	99.4	95.1	95.1
Vessel 6	100.0	99.6	94.6	98.8	94.5	94.6
Vessel 7	91.1	83.6	86.5	82.7	82.9	78.7
Vessel 8	79.5	72.7	75.4	71.7	76.5	74.8
Vessel 9	100.0	100.0	100.0	100.0	100.0	100.0
Average	96.7	94.9	94.7	94.5	94.3	93.5
Min	79.5	72.7	75.4	71.7	76.5	74.8
Max	100.0	100.0	100.0	100.0	100.0	100.0

5.1.8 RF Inhomogeneity

Inhomogeneities in RF fields are inevitable hence the acquired image contains a smoothly changing intensity field imposed onto the actual signal. RF field inhomogeneity causes performance degradation in almost all segmentation algorithms such as region growing, and Chan-Vese level set [32]. For a segmentation algorithm it is important to produce good segmentation results for images having RF inhomogeneity without RF field correction step. In Figure 63, BrainWeb image is fused with RF field such that, for 50% RF field new image intensities are calculated as: $I_{x,y,z} = \min(I_{x,y,z} + 0.50 \times RF_{x,y,z}, 1)$ where maximum value of $I_{x,y,z}$ and $RF_{x,y,z}$ are 1.

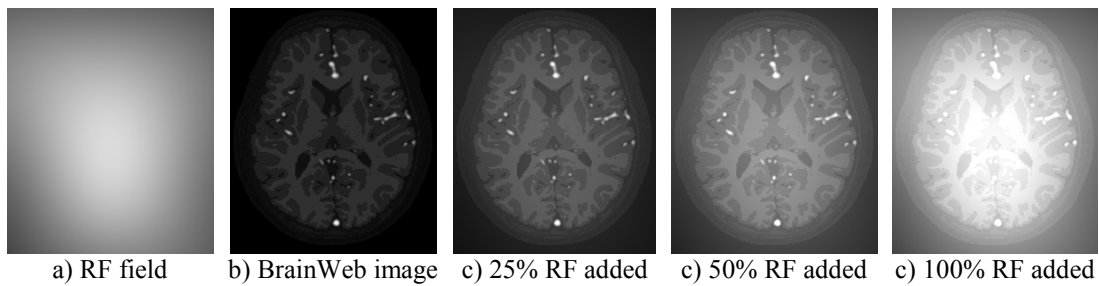


Figure 63: BrainWeb images with RF inhomogeneities.

Average Jaccard index values for 20 images in the BrainWeb database for various levels of inhomogeneity are shown in Figure 64. Even for 50% of inhomogeneity, the segmentation performance changes only slightly. For higher amounts of inhomogeneity, vessel and white matter and gray matter tissues become almost inseparable. In short, the proposed method is able to deal with smooth RF inhomogeneities without need of bias field correction as a pre-processing step.

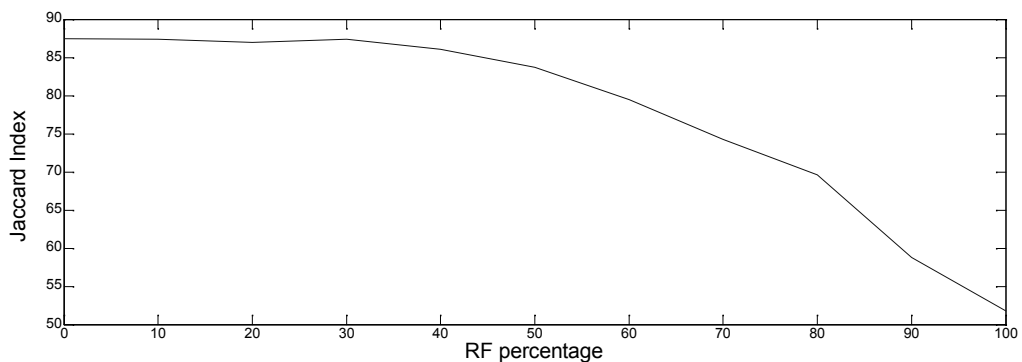


Figure 64: Segmentation success for different RF levels on BrainWeb database.

5.2 Segmentation Results

5.2.1 Brain MRI: BrainWeb Database

BrainWeb is a simulated brain database which contains 20 segmented brain images and their 12 tissue probabilities (Kwan et al 1999). We used class probabilities to reconstruct realistic MRI volumes (Figure 65) for validation and comparison studies. Vessel class contains vessel structures scanned by both MRA and MRV hence reconstructed MRI contains wide range of vessel structures. BrainWeb vessel probability map contains erroneous vessel structures which are placed outside the brain and CSF. We removed such vessels using automated and manual procedures.

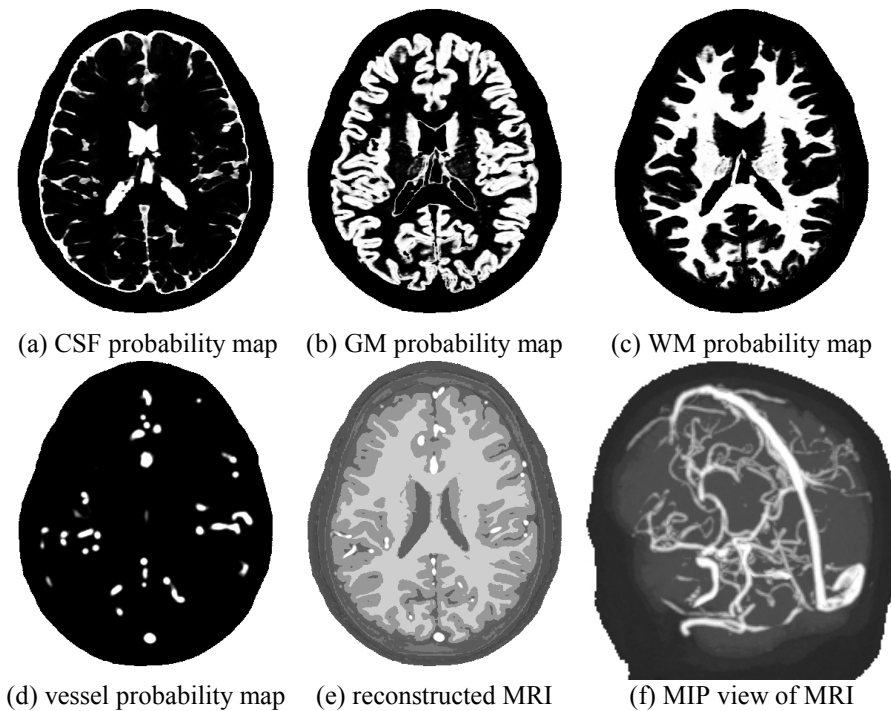


Figure 65: Created simulated brain using BrainWeb vessel classes.

For our problem we used vessel class of BrainWeb database as our ground truth data. Various amount of noise is added to those datasets and performance of LSWE is tested using this datasets and original vessel class as ground truth. We also compared our results with Chan-Vese level set (Chan and Vese 2001) on BrainWeb database using Jaccard index as performance measure (Table 13).

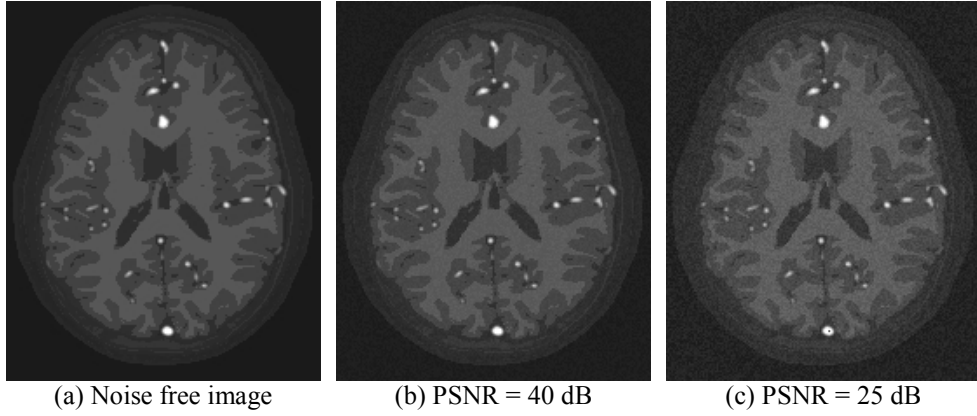
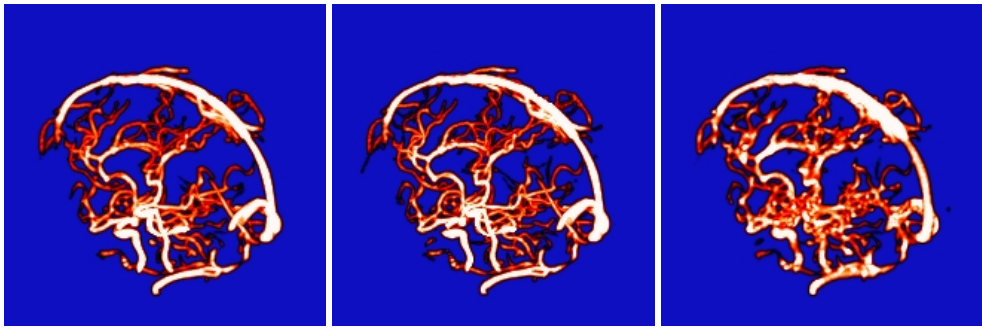


Figure 66: Segmentation results from BrainWeb database

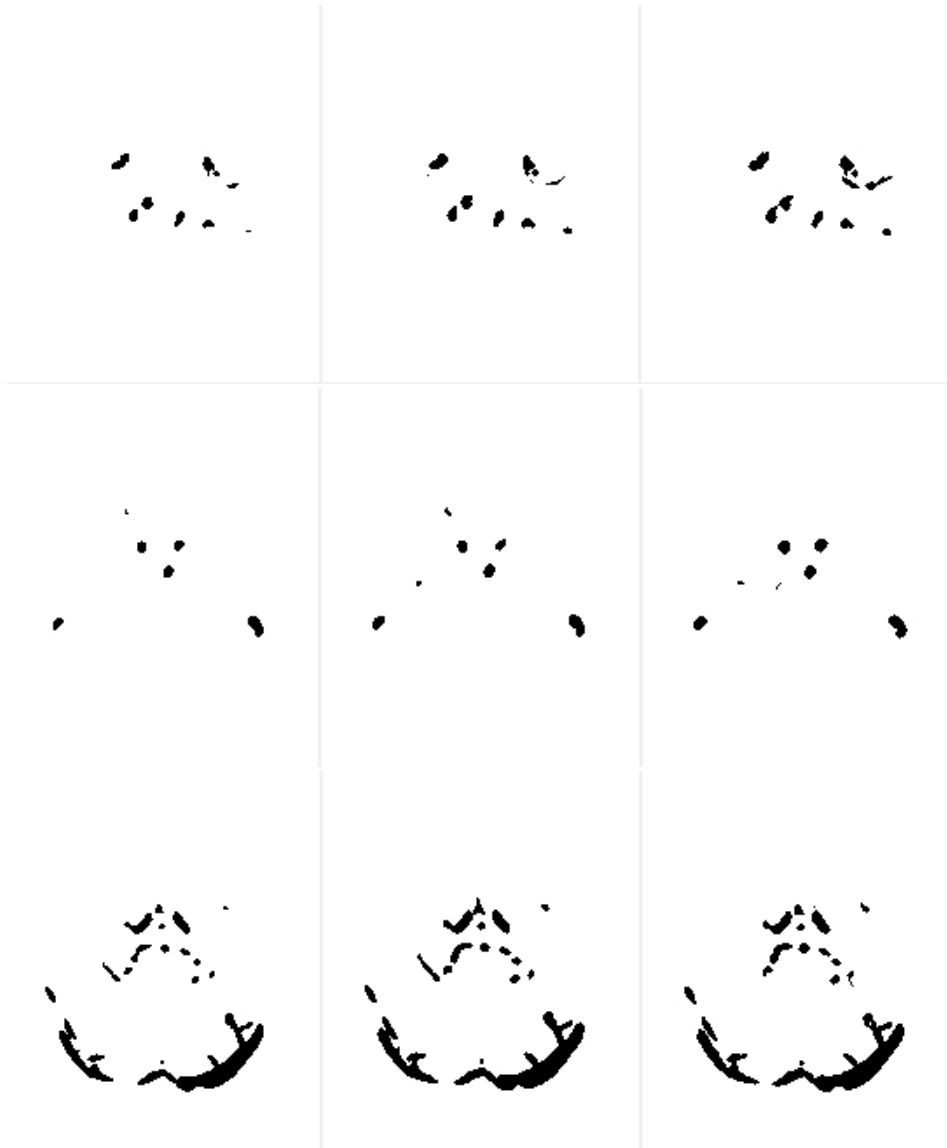
Table 13: LSWE versus level set comparison on BrainWeb database

Patient	Original image		PSNR=30 dB		PSNR=25 dB	
	LSWE	Chan-Vese	LSWE	Chan-Vese	LSWE	Chan-Vese
04	86,35	82,41	82,76	74,81	73,11	65,56
05	88,57	83,46	84,06	75,91	75,20	66,92
06	88,44	83,93	84,50	76,56	76,13	67,79
18	87,82	83,76	84,69	76,63	76,01	67,87
20	87,91	83,47	84,00	76,03	75,13	67,40
38	86,82	82,32	80,72	74,99	72,44	65,99
41	86,75	82,91	82,08	75,34	74,04	66,21
42	88,43	83,29	84,35	76,03	75,11	67,23
43	84,47	81,16	80,20	73,59	71,38	64,38
44	88,22	83,24	79,66	75,84	73,19	67,30
45	87,34	83,06	82,47	75,51	73,69	66,52
46	89,12	83,76	85,34	76,65	77,07	68,38
47	86,19	82,75	80,65	75,68	73,25	66,47
48	84,98	81,71	74,89	73,90	68,58	64,96
49	85,03	81,16	79,87	73,21	71,32	64,26
50	83,90	82,37	79,06	74,79	70,72	65,44
51	89,01	83,26	84,40	76,13	76,27	67,50
52	88,26	84,12	79,75	76,55	64,14	67,37
53	85,10	80,77	82,13	74,06	73,56	65,60
54	87,91	82,61	83,93	75,03	74,51	65,91
Average	87,03	82,78	81,98	75,36	73,24	66,45
Min	83,90	80,77	74,89	73,21	64,14	64,26
Max	89,12	84,12	85,34	76,65	77,07	68,38

Although performance values in Table 13 may be improved, currently both LSWE and Chan-Vese level set were able to find all vessel structures. LSWE tend to shrink the actual boundary (especially for thick vessels) in a small amount whereas Chan-Vese method tends to smooth and expand the boundary (Figure 67). As seen in Table 13 LSWE can deal with the imaging noise.



a) LSWE segmentation b) Ground truth c) Chan-Vese segmentation
 Figure 67: BrainWeb ground truth, LSWE and level set result – 3D rendered



a) LSWE segmentation, b) Ground truth, c) Chan-Vese segmentation
 Figure 68: BrainWeb ground truth, LSWE and level set result – 2D slices

5.2.2 Brain MRA: 10 Volunteers / Clinical Datasets

11 MRA volumes⁶ of size 224x256x220 with 0.7 mm (rescaled) isotropic voxels are segmented using LSWE method and Chan-Vese level set method. LSWE parameters are optimized via trial and error (Table 14).

Table 14: Segmentation parameters for MRA volumes.

Parameter	Value	Unit	Parameter	Value	Unit
Δt	1	second	r_v	4	voxel
$\Delta x, \Delta y, \Delta z$	1	meter	T_{rs}	0.15	%
Δh	0.5	meter	r_w	2	voxel
τ	0.1		T_{ws}	0.025	%
V_d	0.01	%	T_p	0.001	%
V_r	0.9	%	$C_{tolerance}$	0.25	%

LSWE segmentation results of first 9 MRAs of 11 volunteers are shown in Figure 69.

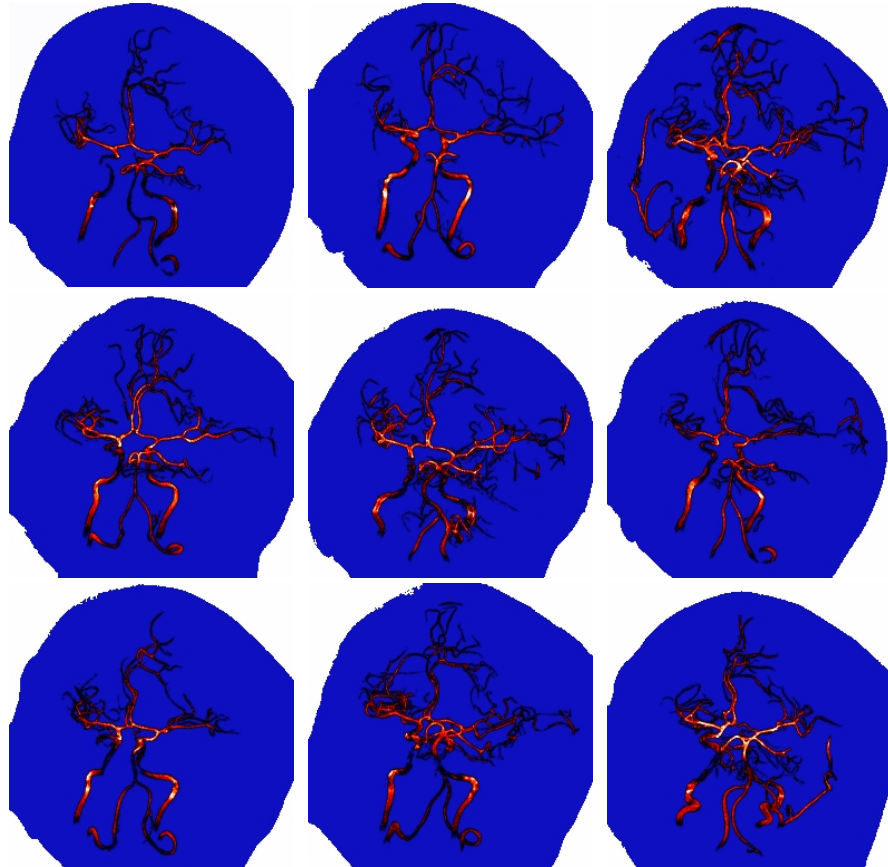


Figure 69: LSWE segmentation results of 9 MRAs

⁶ MRA parameters used within the 1.5T Siemens scanner are as follows: TR=23 ms, TE=7 ms, FA=25 degrees, 12 bit, 0.35mm x 0.35 mm x 0.70 mm

Level set segmentation results of top 3 MRAs from Figure 69 are given in Figure 70.

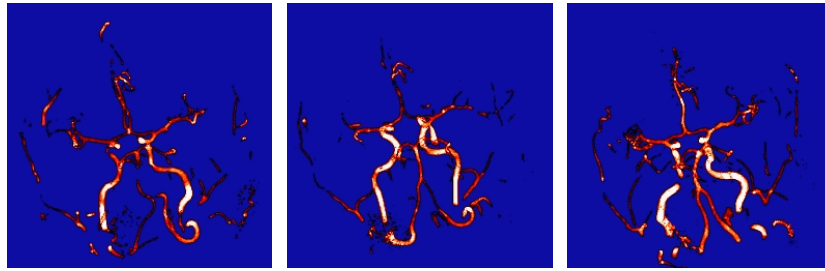


Figure 70: Chan-Vese level set segmentation results of 3 MRAs

LSWE segmentation took between 45-120 seconds for each MRA volume. A validation study blindfolded to the chosen segmentation method is done by the help of a radiologist where the results are evaluated with 10 pt Likert scale separately for thin vessels, thick vessels, number of false alarms, and overall quality. During rating, the original and segmented image slices are observed in sagittal, coronal, transverse planes, as well as 3D MIP render in any angle with different brightness and contrast settings. Each segmented image is displayed twice in randomized order to investigate the consistency of the blind observer. Chan-Vese segmentations for top 3 images in Figure 69 are shown in Figure 70.

Table 15: Statistics for observer study of 11 MRA volumes.

		Thin structures	Thick structures	False alarms with a low rate	Overall quality	Weighted Average
LSWE	Average	7.73	8.82	9.86	8.55	8.74
	Min	6.00	7.00	8.00	7.00	7.88
	Max	9.00	9.00	10.00	9.00	9.13
	Std.Dev.	0.70	0.50	0.47	0.60	0.33
Chan-Vese	Average	5.86	7.68	9.68	6.77	7.50
	Min	4.00	5.00	5.00	5.00	5.50
	Max	7.00	9.00	10.00	8.00	8.25
	Std.Dev.	0.83	1.25	1.13	0.92	0.84

We used paired t-test for validating the consistency of first and second observation for each segmentation method. The differences in these evaluations are found to be not statistically significant ($p=1.0$ for overall LSWE evaluations and $p=0.1$ for overall Chan-Vese evaluations). Therefore we merged these 2 evaluations under each segmentation method and used paired t-test to compare the evaluations across LSWE and Chan-Vese. The differences between these two methods are found to be statistically significant for thin structures ($p<10^{-4}$), thick structures ($p<0.0015$) and overall quality ($p<10^{-4}$).

5.2.3 Brain MRV: 10 Volunteers / Clinical Datasets

11 MRV volumes⁷ of size 256x256x176 with 1.0 mm (rescaled) isotropic voxels are segmented using LSWE method and Chan-Vese level set method. LSWE parameters are optimized via trial and error (Table 16).

Table 16: Segmentation parameters for MRV volumes.

Parameter	Value	Unit	Parameter	Value	Unit
Δt	1	second	r_v	3	voxel
$\Delta x, \Delta y, \Delta z$	1	meter	T_{rs}	0.05	%
Δh	6	meter	r_w	1	voxel
τ	0.5		T_{ws}	0.01	%
V_d	0.025	%	T_p	0.05	%
V_r	0.5	%	$C_{tolerance}$	0.25	%

LSWE segmentation results of first 9 MRVs of 11 volunteers are shown in Figure 71.

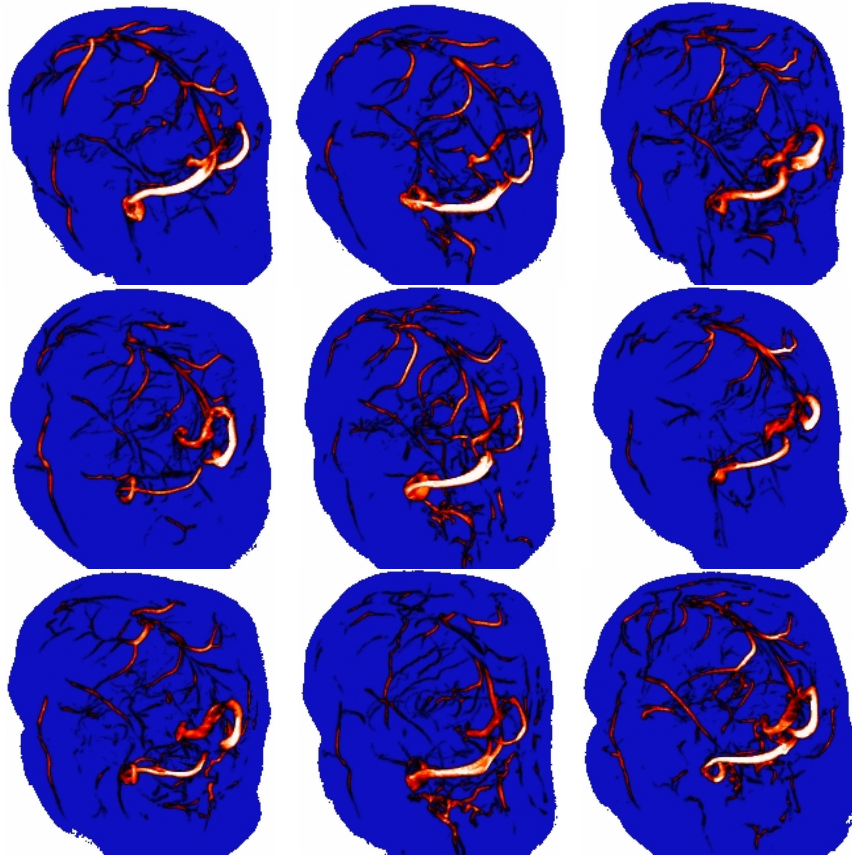


Figure 71: LSWE segmentation results of 9 MRVs

⁷ MRV parameters used within the 1.5T Siemens scanner are as follows: TR=25 ms, TE=7 ms, FA=60 degrees, 12 bit, 0.98mm x 0.98 mm x 1.34 mm

Level set segmentation results of top 3 MRVs from Figure 71 are given in Figure 72.

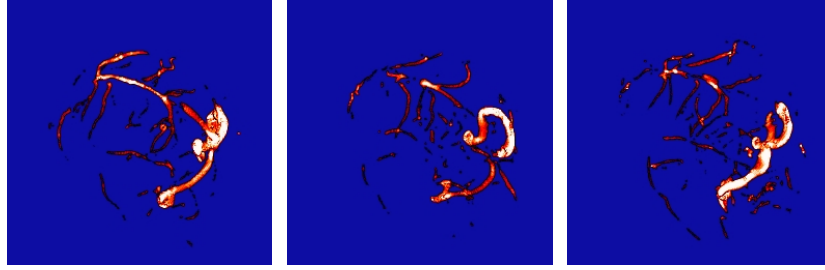


Figure 72: Chan-Vese level set segmentation results of 6 MRVs

LSWE segmentation took between 60-120 seconds for each MRV volume. A validation study blindfolded to the chosen segmentation method is done by the help of a radiologist where the results are evaluated with 10 pt Likert scale separately for thin vessels, thick vessels, number of false alarms, and overall quality. During rating, the original and segmented image slices are observed in sagittal, coronal, transverse planes, as well as 3D MIP render in any angle with different brightness and contrast settings. Each segmented image is displayed twice in randomized order to investigate the consistency of the blind observer. Chan-Vese segmentations for top 3 images in Figure 71 are shown in Figure 72.

Table 17: Statistics for observer study of 11 MRV volumes.

		Thin structures	Thick structures	False alarms with a low rate	Overall quality	Weighted Average
LSWE	Average	7.18	8.64	10.00	7.82	8.41
	Min	6.00	7.00	10.00	7.00	7.75
	Max	8.00	9.00	10.00	8.00	8.75
	Std.Dev.	0.66	0.66	0.00	0.39	0.29
CChan-Vese	Average	4.41	5.41	10.00	5.00	6.20
	Min	4.00	5.00	10.00	4.00	5.88
	Max	6.00	7.00	10.00	6.00	6.88
	Std.Dev.	0.59	0.59	0.00	0.44	0.29

We used paired t-test for validating the consistency of first and second observation for each MRV segmentation method. The differences in these evaluations are found to be not statistically significant ($p=1.0$ for overall LSWE evaluations and $p=1.0$ for overall Chan-Vase evaluations). So we merged these 2 evaluations under each segmentation method and used paired t-test to compare the evaluations across LSWE and Chan-Vese. The differences between these two methods are found to be statistically significant for thin structures ($p<10^{-4}$), thick structures ($p<10^{-4}$) and overall quality ($p<10^{-4}$).

5.2.4 Retinal Angiography: DRIVE Database

Retinal angiography images in DRIVE database (Staal et al 2004) are segmented using the proposed method (Table 16).

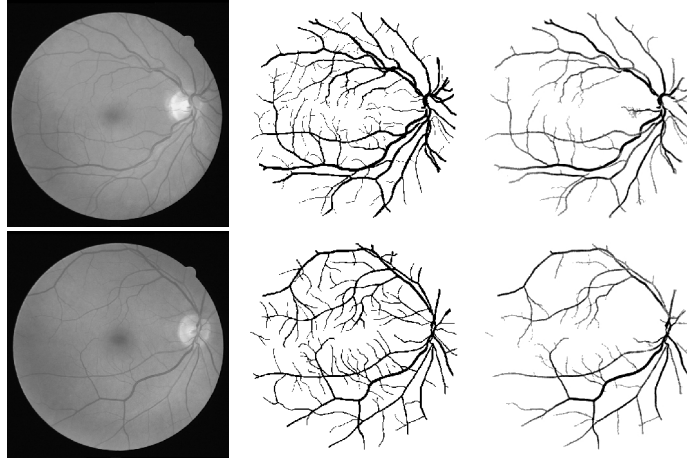


Figure 73: Segmentation results from DRIVE retinal angiography database.

Table 18: LSWE results on DRIVE database

Image no	Jaccard	F-measure	Image no	Jaccard	F-measure
1	61.81	88.11	20	51.57	72.14
2	69.08	88.56	21	61.61	82.06
3	56.37	81.59	22	57.84	77.43
4	55.93	73.29	23	53.05	85.53
5	59.82	77.98	24	61.08	79.54
6	56.19	74.24	25	43.76	64.97
7	57.69	77.61	26	53.28	77.15
8	47.96	69.52	27	55.42	75.42
9	53.64	75.83	28	60.88	82.81
10	60.86	83.41	29	56.77	75.53
11	61.27	83.07	30	47.86	69.88
12	60.86	83.67	31	53.18	77.08
13	60.55	83.20	32	54.64	73.19
14	59.52	88.37	33	62.99	83.38
15	61.98	85.17	34	38.31	62.92
16	62.59	82.46	35	59.33	78.32
17	52.72	74.37	36	43.13	61.42
18	57.05	78.47	37	61.81	85.61
19	67.19	88.38	38	56.62	83.01
20	51.57	88.11	39	56.58	82.90

Average Jaccard index is 56.78% and average F-measure is 78.73% which requires improvement in segmentation steps. In order to increase performance better pre-processing and post-processing is required for retinal angiography images.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

Using fluid flow for segmenting vessels in medical images is biologically plausible since vessels constitute the physical media for transporting blood. With this motivation, we developed a generic fluid flow simulation which is capable of segmenting vessels in 2D and 3D images. Fluid flow based methods are computationally intensive so rarely used in image processing, especially in image segmentation. However, the algorithm we utilized, LSWE, is simple, can operate on parallel processors, and has low computational cost compared to other fluid simulation methods.

In this study, a new turn-key approach is proposed for vessel segmentation. Basic operational principles of SWE and its relationship with NSE as well as the linearization procedure for obtaining a fast fluid-flow simulation are briefly outlined. Critical steps of the proposed algorithm such as rough segmentation, water source detection, and error calculation and convergence criteria for obtaining the global minimum of the defined error function are explained. A fast method is proposed for obtaining rough segmentation which is the main input of the water source detection algorithm and error function. Two kind of efficient water source detection algorithms are proposed for dealing with different kind of vessel systems. A novel convergence criterion for robust minimization of error function is proposed so that we know when to stop adding fluid into the vessel system. Due to the nature of fluid-flow and

convergence criteria, our method produces robust segmentation with capability to connect the gaps formed by imaging artifacts or limitations.

All steps of the proposed algorithm are either very fast or parallelizable. Integration speed of LSWE, which is the most computationally intensive part of the proposed method, is further parallelized using one step Jacobi solver as an approximation to tridiagonal matrix solver. Depending on image size, one to two order of magnitude speed ups are obtained with developed Nvidia CUDA parallel implementation comparing to single-core highly optimized CPU implementation. Using GPU based LSWE implementation, 2D images can be segmented in the order of seconds and 3D volumes can be segmented between 30 and 120 seconds.

Limitations of the method and the behavior of the simulation parameters are investigated on nine different 3D synthetic vein structures and the BrainWeb database under various noise levels. Simulation parameters versus error function exhibit graceful degradation characteristics, besides segmentation results are mostly robust to variations in parameter values. Segmentation performance of LSWE and Chan-Vese Level set are compared on the BrainWeb database where vessel probability class is used as ground truth. The proposed method is also validated on clinical MRA images and clinical MRV images via blind validation study. The proposed method outperformed Chan-Vese level set method for BrainWeb database, MRA and MRV images for the all applied tests.

Proposed algorithm is successful at segmenting vessels in 2D and 3D medical images under various imaging artifacts such as imaging noise, gaps in vessels, and RF inhomogeneity. Similar to various other methods, our method is able to segment thick structures very well. In addition, our method is also successful at segmenting thin and low contrast vessel structures even for challenging images considered as difficult to segment. Although our method contains many parameters, these parameters are easy to determine and the system is stable for subtle parameter

variances. Also our algorithm contains parallelizable building blocks allowing segmentation of large 3D volumes in reasonable times. Since the method is implemented using GPU, the speed of segmentation is high.

For future work, the proposed method can be applied to other types of data such as road networks in remote sensing images. Different fluid-flow simulation methods instead of LSWE can be used and their impact on segmentation results can be analyzed. In addition to the currently investigated vessel structures, segmentation performance on vessels having anomalies need to be analyzed. For eliminating the effect of noise and increasing the speed of the algorithm, use of scale-space approach can be investigated. Speed of the implementation can be further improved by porting rough segmentation and water source detection algorithms to GPU.

BIBLIOGRAPHY

Adams, B., Pauly, M., Keiser, R., and Guibas, L. J. (2007). Adaptively sampled particle fluids. In SIGGRAPH '07: ACM SIGGRAPH 2007 papers, New York, NY, USA. ACM.

Agam, G., & Wu, C. (2005). Probabilistic modeling based vessel enhancement in thoracic CT scans. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on (Vol. 2, pp. 649 - 654 vol. 2).

Agam, G., Armato, S. G., & Wu, C. (2005). Vessel tree reconstruction in thoracic CT scans with application to nodule detection. Medical Imaging, IEEE Transactions on, 24(4), 486 -499.

Akram, M. U., Atzaz, A., Aneeque, S. F., & Khan, S. A. (2009). Blood Vessel Enhancement and Segmentation Using Wavelet Transform. In Digital Image Processing, 2009 International Conference on (pp. 34 -38).

Al-Diri, B., Hunter, A., and Steel, D. (2009). An active contour model for segmenting and measuring retinal vessels. Medical Imaging, IEEE transactions on, 28(9): 1488-1497.

Alim, U.R., Entezari, A., Moller, T. (2009). The Lattice-Boltzmann Method on Optimal Sampling Lattices. Visualization and Computer Graphics, IEEE Transactions on, 15(4): 630-641.

Alonso-Montes, C., Vilari, D. L., Dudek, P., & Penedo, M. G. (2008). Fast retinal vessel tree extraction: A pixel parallel approach. Int. J. Circuit Theory Appl., 36: 641–651.

Antiga, L., Ene-Iordache, B., and Remuzzi, A. (2003). Computational geometry for patient-specific reconstruction and meshing of blood vessels from MR and CT angiography. Medical Imaging, IEEE Transactions on, 22(5): 674-684.

Baldi, P., Brunak, S., Chauvin, Y., Andersen, C. A. F., and Nielsen, H. (2000). Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, 16(5):412-424.

Bauer, C. Bischof H. (2008). A Novel Approach for Detection of Tubular Objects and Its Application to Medical Image Analysis. LNCS 5096: 163-172.

Beneš, B. (2007). Real-time erosion using shallow water simulation. Proc. of Workshop in Virtual Reality Interactions and Physical Simulation, VRIPHYS '07.

Bock, S., Kühnel, C., Boskamp, T., and Peitgen, H. (2008). Robust vessel segmentation. *SPIE Computer-Aided Diagnosis*, 6915, page 391-399.

Bullitt, E., Muller, K., Jung, I., Lin, W., And Aylward, S. (2005). Analyzing attributes of vessel populations. *Medical Image Analysis*, 9(1):39-49.

Carlson, M., Mucha, P. J., Van Horn, B. R., and Turk, G. (2002). Melting and flowing. In *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 167-174, New York, NY, USA. ACM Press.

Chan, T., and Vese, L. (2001). Active Contours Without Edges. *IEEE Transactions on Image Processing*, 12(2) : 266-277.

Chang, H. and Valentino, D.J. (2006). Image segmentation using a charged fluid method. *Journal of Electronic Imaging*. 15(2): 023011.

Chen, S. and Doolen, G. D. (1998). Lattice boltzmann method for fluid flows. *Annual Review of Fluid Mechanics*, 30(1):329-364.

Chung, A.C.S. (2006). Image segmentation methods for detecting blood vessels in angiography. *Control, Automation, Robotics and Vision*, ICARCV'06, pages 1-6.

de Koning, P. J., Schaap, J. A., Janssen, J. P., Westenberg, J. J., van der Geest, R. J., & Reiber, J. H. (2003). Automated segmentation and analysis of vascular structures in magnetic resonance angiographic images. *Magnetic Resonance in Medicine*, 50(6), 1189–1198.

Deschamps, T., Schwartz, P., Trebotich, D., Colella, P., Saloner, D., & Malladi, R. (2004). Vessel segmentation and blood flow simulation using Level-Sets and Embedded Boundary methods. *International Congress Series*, 1268: 75 - 80.

Descoteaux, M., Collins, L., and Siddiqi, K. (2004). Geometric flows for segmenting vasculature in MRI: Theory and validation. pages 500-507.

Douglas, J., Kellogg, R.B., and Varga, R.S. (1963). Alternating direction iteration methods for n space variables. *Mathematics of Computation*.

Duda, R. O., Hart, P. E., and Stork, D. G. (2000). *Pattern Classification* (2nd Edition). Wiley-Interscience, 2 edition.

El-Baz, A., Farag, A., Gimel'farb G., El-Ghar, G., and Eldiasty, T. (2006). Probabilistic modeling of blood vessels for segmenting MRA images. *International Conference on Pattern Recognition, ICPR'06*, 3:917-920.

Esneault, S., Lafon, C., & Dillenseger, J. -. (2010). Liver Vessels Segmentation Using a Hybrid Geometrical Moments/Graph Cuts Method. *Biomedical Engineering, IEEE Transactions on*, 57(2), 276 -283.

Farag, A.A., Hassan, H., Falk, R., Hushek, S.G. (2004). 3D volume segmentation of MRA data sets using Level sets. *Academic Journal of Radiology*, page 419-435.

Foster, N. and Fedkiw, R. (2001). Practical animation of liquids. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 23-30, New York, NY, USA. ACM.

Foster, N. and Metaxas, D. (1996). Realistic animation of liquids. *Graphical models and image processing: GMIP*, 58(5):471-483.

Foster, N. and Metaxas, D. (1997). Controlling fluid animation. In *CGI '97: Proceedings of the 1997 Conference on Computer Graphics International*, pages 178+, Washington, DC, USA. IEEE Computer Society.

Fox, R.W., Pritchard, P.J., and McDonald, A.T. (2008). *Introduction to Fluid Mechanics*. John Wiley.

Frangi, A. F., Niessen, W. J., Vincken, K. L., and Viergever, M. A. (1998). Multiscale vessel enhancement filtering. *Medical Image Computing and Computer-Assisted Intervention - MICCAI'98*, pp.130.

Friman, O. Hindennach, M. Peitgen, H.-O. (2008). Template-based multiple hypotheses tracking of small vessels. *5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 1047-1050.

Gingold, R. A. and Monaghan, J. J. (1977). Smoothed particle hydrodynamics - theory and application to non-spherical stars. *Mon. Not. Roy. Astron. Soc.*, 181:375-389.

Goktekin, T. G., Bargteil, A. W., and O'Brien, J. F. (2004). A method for animating viscoelastic fluids. *ACM Trans. Graph.*, 23(3):463-468.

Gooya, A., Liao, H., Matsumiya, K., Masamune, K., Masutani, Y., & Dohi, T. (2008). A Variational Method for Geometric Regularization of Vascular Segmentation in Medical Images. *Image Processing, IEEE Transactions on*, 17(8), 1295 -1312.

Göddecke, D. and Strzodka, R. (2010). Cyclic reduction tridiagonal solvers on GPUs applied to mixed precision multigrid. *Parallel and Distributed Systems, IEEE Transactions on*, preprint.

Guerrero, J., Salcudean, S. E., McEwen, J. A., Masri, B. A., and Nicolaou, S. (2007). Real-Time vessel segmentation and tracking for ultrasound imaging applications. *Medical Imaging, IEEE Transactions on*, 26(8):1079-1090.

Hao, J., Shen, Y., & Wang, Y. (2006). Segmentation for MRA Image: An Improved Level Set Approach. In *Instrumentation and Measurement Technology Conference, 2006. IMTC 2006. Proceedings of the IEEE* (pp. 382 -385).

Holtzman-Gazit, M., Kimmel, R., Peled, N., Goldsher, D. (2006). Segmentation of thin structures in volumetric medical images. *Image Processing, IEEE Transactions on*, 15(2): 354-363.

Hoover, A. D., Kouznetsova, V., and Goldbaum, M. (2002). Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response. *IEEE Transactions on Medical Imaging*, 19(3):203-210.

Hochstadt, H. (1964). *Differential Equations*. Dover Publications, New York.

Huang, K., Yan, M. (2006). A region based algorithm for vessel detection in retinal images. *MICCAI*, 9(1):645-653.

Irving, G., Guendelman, E., Losasso, F. and Fedkiw, R. (2006). Efficient Simulation of Large Bodies of Water by Coupling Two and Three Dimensional Techniques. *SIGGRAPH 2006*, 25:805-811.

Jia, F., Wang, S., Liu, L., and Li, Hua. (2004). Preprocessing based statistical segmentation of MRA dataset. *Virtual Reality Continuum and Its Applications*, pages 306-308.

Kass, M. and Miller, G. (1990). Rapid, stable fluid dynamics for computer graphics. *International Conference on Computer Graphics and Interactive Techniques*, page 49-57.

Kass, M., Witkin, A., and Terzopoulos, D. (1988). Snakes: active contour models. *International Journal of Computer Vision*, page 321-331.

Keiser, R., Adams, B., Gasser, D., Bazzi, P., Dutre, P., and Gross, M. (2005). A unified lagrangian approach to solid-fluid animation. *Eurographics Symposium on Point- Based Graphics*.

Kirbas, C. and Quek, F. K. H. (2004). A review of vessel extraction techniques and algorithms. *ACM Computing Surveys*, 36:81-121.

Kwan, R. K. S., Evans, A. C., and Pike, G. B. (1999). MRI simulation-based evaluation of image-processing and classification methods. *Medical Imaging, IEEE Transactions on*, 18(11):1085-1097.

Lacoste, C., Finet, G., & Magnin, I. E. (2006). Coronary tree extraction from X-ray angiograms using marked point processes. In *Biomedical Imaging: Nano to Macro*, 2006. 3rd IEEE International Symposium on (pp. 157 -160).

Lam, B. S. Y. and Yan, H. (2008). A novel vessel segmentation algorithm for pathological retina images based on the divergence of vector fields. *Medical Imaging, IEEE Transactions on*, 27(2):237-246.

Lam, B.S.Y., Gao, Y., Liew, A.W.C. (2010). General Retinal Vessel Segmentation Using Regularization-Based Multiconcavity Modeling. *IEEE Transactions on Medical Imaging*, 29(7): 1369-1381.

Läthén, G., Jonasson, J., and Borga, M. (2010). Blood vessel segmentation using multi-scale quadrature filtering. *Pattern Recognition Letters*, 31(8):762-767.

Lee, J., Beighley, P., Ritman, E., & Smith, N. (2007). Automatic segmentation of 3D micro-CT coronary vascular images. *Medical Image Analysis*, 11(6), 630 - 647.

Lee, R. and O'Sullivan, C. (2007). A fast and compact solver for the shallow water equations. *4th Workshop in Virtual Reality Interactions and Physical Simulation, VRIPHYS'07*, page 51-58.

Lesage, D., Angelini, E. D., Bloch, I., and Funka-Lea, G. (2009). A review of 3d vessel lumen segmentation techniques: Models, features and extraction schemes. *Medical Image Analysis*, 13(6):819-845.

Li, H. and Yezzi, A. (2007). Vessels as 4-D curves: Global minimal 4-D paths to extract 3-D tubular surfaces and centerlines. *IEEE Transactions on Medical Imaging*, 26(9):1213-1223.

Liu, X.U. and Nixon, M.S. (2007). Image and volume segmentation by water flow. *Lecture Notes in Computer Science*, 4842: 62-74.

Losasso, F., Shinar, T., Selle, A., and Fedkiw, R. (2006). Multiple interacting liquids. *ACM Transactions on Graphics*, 25(3): 812-819.

Lupascu, C.A., Tegolo, D., Trucco, E. (2010). FABC: Retinal Vessel Segmentation Using AdaBoost. *IEEE Transactions on Information Technology in Biomedicine*, 14(5): 1267-1274.

Mahadevan, V., Narasimha-Iyer, H., Roysam, B., Tanenbaum, H.L. (2004). Robust model-based vasculature detection in noisy biomedical images. *IEEE Transactions on Information Technology in Biomedicine*, 8(3): 360-376.

Malladi, R., Sethian, J. A., and Vemuri, B. C. (1995). Shape modeling with front propagation: a level set approach. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(2):158-175.

Manniesing, R., Viergever, M. A., & Niessen, W. J. (2006). Vessel enhancing diffusion: A scale space representation of vessel structures. *Medical Image Analysis*, 10(6), 815 - 825.

Mcinerney, T. and Terzopoulos, D. (1996). Deformable models in medical image analysis: a survey. *Medical Image Analysis*, 1(2):91-108.

Mcinerney, T. and Terzopoulos, D. (1997). Medical image segmentation using topologically adaptable surfaces. (CVRMED) *Computer Vision, Virtual Reality and Robotics in Medicine*, 1205: 23-32.

McNamara, A., Treuille, A., Popovic, Z., and Stam, J. (2004). Fluid control using the adjoint method. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 449-456, New York, NY, USA. ACM.

Melanie B. Thomas. Hepatocellular Carcinoma: The Need for Progress. *Journal of Clinical Oncology*, Vol 23, No 13 (May 1), 2005: pp. 2892-2899.

Mendonca, A. M. and Campilho, A. (2006). Segmentation of retinal blood vessels by combining the detection of centerlines and morphological reconstruction. *Medical Imaging, IEEE Transactions on*, 25(9):1200-1213.

Monaghan, J. J. (2005). Smoothed particle hydrodynamics. *Reports on Progress in Physics*, 68(8):1703-1759.

Moran, M.J., Shapiro, H.N. (2007). *Fundamentals of Engineering Thermodynamics*, 6th Ed, Wiley.

Müller, M., Charypar, D., and Gross, M. (2003). Particle-based fluid simulation for interactive applications. In *SCA '03: Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 154-159, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

Müller, M., Solenthaler, B., Keiser, R., and Gross, M. (2005). Particle-based fluid-fluid interaction. *ACM SIGGRAPH/Eurographics symposium on Computer animation*, page 237-244.

Müller, M., Schirm, S., and Teschner, M. (2004). Interactive blood simulation for virtual surgery based on smoothed particle hydrodynamics. *Technol. Health Care*, 12(1):25-31.

Müller, M., Schirm, S., Teschner, M., Heidelberger, B., and Gross, M. (2004). Interaction of fluids with deformable solids. *Computer Animation and Virtual Worlds*, 15(3-4): 159-171.

Nayfeh, A.H., and Mook, D.T. (1979). *Nonlinear Oscillations*. John Wiley & Sons, New York.

Nealen, A., Mueller, M., Keiser, R., Boxerman, E., and Carlson, M. (2006). Physically based deformable models in computer graphics. *Computer Graphics Forum*, 25(4):809-836.

NVIDIA CUDA Compute Unified Device Architecture – Best Practices Guide.

NVIDIA CUDA Compute Unified Device Architecture - Programming Guide. Retrieved September 7, 2010, from http://developer.download.nvidia.com/compute/cuda/3_1/toolkit/docs/NVIDIA_CUDA_C_ProgrammingGuide_3.1.pdf

Osher, S. and Fedkiw, R. P. (2001). Level set methods: An overview and some recent results. *Journal of Computational Physics*, 169(2):463-502.

Osher, S. and Sethian, J. (1988). Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79(1):12-49.

Otsu, N. (1979). A threshold selection method from gray-level histograms. *Systems, Man and Cybernetics, IEEE Transactions on*, 9(1):62-66.

Palomera-Perez, M. A., Martinez-Perez, M. E., Benitez-Perez, H., & Ortega-Arjona, J. L. (2010). Parallel Multiscale Feature Extraction and Region Growing: Application in Retinal Blood Vessel Detection. *Information Technology in Biomedicine, IEEE Transactions on*, 14(2): 500 -506.

Passat, N., Ronse, C., Baruthio, J., Armspach, J., Maillot, C., & Jahn, C. (2005). Region-growing segmentation of brain vessels: An atlas-based automatic approach. *Journal of Magnetic Resonance Imaging*, 21(6), 715–725.

Peaceman, D. W. and Rachford, H.H. (1955). The numerical solution of parabolic and elliptic differential equations. *Journal of the Society for Industrial and Applied Mathematics*, 3: 28-41.

Peng, Q., You, X., Zhou, L., Yiu-Ming, C. (2010). Retinal blood vessels segmentation using the radial projection and supervised classification. *International Conference on Pattern Recognition, ICPR'10*.

Pighin, F., Cohen, J.M., and Shah, M. (2004). Modeling and editing flows using advected radial basis functions. *Eurographics/ACM SIGGRAPH Symposium on Computer Animation*.

Premoze, S., Tasdizen, T., Bigler, J., Lefohn, A., Whitaker, R.T. (2003). Particle-based simulation of fluids. *Computer Graphics Forum*, 22(3): 401-410.

Radaelli, A. G., & Peiró, J. (2010). On the segmentation of vascular geometries from medical images. *International Journal for Numerical Methods in Biomedical Engineering*, 26(1), 3–34.

Retrieved September 7, 2010, from http://developer.download.nvidia.com/compute/cuda/3_1/toolkit/docs/NVIDIA_CUDA_C_BestPracticesGuide_3.1.pdf

Ricci, E. and Perfetti, R. (2007). Retinal blood vessel segmentation using line operators and support vector classification. *IEEE Transactions on Medical Imaging*, 26(10):1357-1365.

Rohr, K., and Worz, S. (2006). High-precision localization and quantification of 3D tubular structures. *IEEE International Symposium on Biomedical Imaging*, 1160-1163.

Sekiguchi, H., Sugimoto, N., Eiho, S., Hanakawa, T., & Urayama, S. (2005). Blood vessel segmentation for head MRA using branch-based region growing. *Syst. Comput. Japan*, 36: 80–88.

Sethian, J. A. (1996). A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences of the United States of America*, 93(4):1591-1595.

Sezgin, M. and Sankur, B. (2004). Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13(1):146-168.

Shinbrot, M. (1970). The shallow water equations. *Journal of Engineering Mathematics*, 4(4):293-304.

Staal, J., Abramoff, M. D., Niemeijer, M., Viergever, M. A., & van Ginneken, B. (2004). Ridge-based vessel segmentation in color images of the retina. *Medical Imaging, IEEE Transactions on*, 23(4), 501 -509.

Stam, J. (1999). Stable fluids. In *Proc. of SIGGRAPH*: 121-128.

Stora, D., Agliati, P., Cani, M., Neyret, F., and Gascuel, J. (1999). Animating lava flows. *Graphics Interface*, page 203-210.

Strzelecki, M., Szczypinski, P., Materka, A., Kocinski, M., and Sankowski, A. (2009). Level-set segmentation of noisy 3D images of numerically simulated blood vessels and vascular trees. *International Symposium on Image and Signal Processing and Analysis*, page 742-747.

Sum, K. W., & Cheung, P. Y. (2008). Vessel Extraction Under Non-Uniform Illumination: A Level Set Approach.,IEEE Transactions on Biomedical Engineering, 55(1): 358 -360.

Tan, J. and Yang, X. (2009). Physically-based fluid animation: A survey. Science in China Series F: Information Sciences, 52(5):723-740.

Thürey, N., Keiser, R., Pauly, M., and Rüdè, U. (2009). Detail-preserving fluid control. Graphical Models.

Thürey, N., Müller-Fischer, M., Schirm, S., and Gross, M. (2007). Real-time breaking waves for shallow water simulations. Pacific Conference on Computer Graphics and Applications.

Thürey, N., Sadlo, F., Schirm, S., Fischer, M. M., and Gross, M. (2007). Real-time simulations of bubbles and foam within a shallow water framework. ACM SIGGRAPH / Eurographics symposium on Computer animation, pages 191-198.

Tizon, X. and Smedby, O. (2002). Segmentation with gray-scale connectedness can separate arteries and veins in MRA. Journal of magnetic resonance imaging : JMRI, 15(4):438-445.

Tyrrell, J.A., di Tomaso, E., Fuja, D., Tong, R., Kozak, K., Jain, R.K., Roysam, B. (2007). Robust 3-D Modeling of Vasculature Imagery Using Superellipsoids. IEEE Transactions on Medical Imaging, 26(2): 223-237.

van Bommel, C. M., Spreeuwens, L. J., Viergever, M. A., & Niessen, W. J. (2003). Level-set-based artery-vein separation in blood pool agent CE-MR angiograms. Medical Imaging, IEEE Transactions on, 22(10): 1224 -1234.

van Rijsbergen, C. J. (1979). Information Retrieval. Butterworth-Heinemann.

Weickert, J. (1999). Coherence-enhancing diffusion filtering. International Journal of Computer Vision, 31(2/3): 111-127.

World Health Organization. (2008) The top ten causes of death – Fact sheet N310

Worz, S., and Rohr, K. (2007). Segmentation and Quantification of Human Vessels Using a 3-D Cylindrical Intensity Model. *IEEE Transactions on Image Processing*, 16(8): 1994-2004.

Yan, P. and Kassim, A.A. (2006). Segmentation of volumetric MRA images by using capillary active contour. *Medical Image Analysis*, 10(3):317-329.

Yi, J., & Ra, J. B. (2003). A locally adaptive region growing algorithm for vascular segmentation. *International Journal of Imaging Systems and Technology*, 13(4), 208-214.

Yueh, W. (2005). Eigenvalues of Several Tridiagonal Matrices. *Applied Mathematics E-Notes*, Vol(5): 66-74.

Yüksel, C., House, D.H., and Keyser, J. (2007). Wave particles, *International Conference on Computer Graphics and Interactive Techniques*, 26(3).

Zhao, Y. and Li, M. (2004). Isosurface-based Level set framework for MRA segmentation. *Image Analysis and Recognition*, 3212/2004: 116-123.

Zitkevicius, E., Grigaitis, D., and Navakauskas, D. (2007). On skeletonization of blood vessels in angiographic MRI images of human brain. *Information Technology and Control, Kaunas, Technologija*, Vol. 36(4): 372-376.

Zonoobi, D., Kassim, A. A., & Shen, W. (2009). Vasculature Segmentation in MRA Images Using Gradient Compensated Geodesic Active Contours. *J. Signal Process. Syst.*, 54, 171–181.

APPENDICES

APPENDIX A. LAGRANGIAN APPROACH

In Lagrangian approach particle properties such as mass, position, and velocity are advected with the particle itself. Conservation of mass is guaranteed since fix number of particles, with each having constant mass, is used hence the (Equation 3) can be omitted. However, since (Equation 3) is omitted incompressibility of fluid is not guaranteed anymore. The only remaining NSE is the second one. In this remaining (Equation 4) move advection term $-(\vec{u} \cdot \nabla)\vec{u}$ to the other side

$$\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla)\vec{u} = \nabla \cdot (v\nabla\vec{u}) - \frac{1}{\rho} \nabla p + \vec{f} \quad (\text{Equation 50})$$

multiply each side by density (ρ):

$$\rho \left(\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla)\vec{u} \right) = \nabla \cdot (v\rho\nabla\vec{u}) - \nabla p + \rho\vec{f} \quad (\text{Equation 51})$$

since $\mu = v\rho$ (viscosity = kinematic viscosity x density)

$$\rho \left(\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla)\vec{u} \right) = \nabla \cdot (\mu\nabla\vec{u}) - \nabla p + \rho\vec{f} \quad (\text{Equation 52})$$

and if viscosity is assume as constant;

$$\rho \left(\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla)\vec{u} \right) = \mu\nabla^2\vec{u} - \nabla p + \rho\vec{f} \quad (\text{Equation 53})$$

particles are moving with the fluid so convective term $(\vec{u} \cdot \nabla)\vec{u}$ is not needed

$$\rho \frac{\partial \vec{u}}{\partial t} = \mu\nabla^2\vec{u} - \nabla p + \rho\vec{f} = \vec{F} \quad (\text{Equation 54})$$

Consequently, (Equation 4) which describes the change of velocity transformed into change of momentum.

For i^{th} particle

$$\vec{a}_i = \frac{\partial \vec{u}_i}{\partial t} = \frac{\vec{F}_i}{\rho_i} \quad (\text{Equation 55})$$

and equivalently

$$\vec{a}_i = \frac{\mu \nabla^2 \vec{u}_i}{\rho_i} - \frac{\nabla p_i}{\rho_i} + \frac{\vec{f}_i}{\rho_i} \quad \text{where} \quad \frac{\vec{f}_i}{\rho_i} = \frac{\vec{f}_i^{\text{ext}}}{\rho_i} + \mathbf{g} \quad (\text{Equation 56})$$

Well known method for simulating fluids via Lagrangian approach is Smoothed Particle Hydrodynamics (SPH), which is a coupled particle system. SPH defines how to interpolate material (e.g. fluid) properties at any point in space using the given particles that are placed in discrete locations. Using SPH we can find acceleration of each particle (\vec{a}_i) in (Equation 56). SPH defines a smooth surface passing through discretely sampled particle properties using a so-called smoothing kernel $W(r)$. The kernel acts as a weighting function in the neighborhoods of the position \vec{x}_i (x, y, z) of i^{th} particle via $W(|\vec{x} - \vec{x}_i|)$. Kernel function W is symmetric around the particle since it only depends on the distance to the particle. Kernel function W must also be normalized thus

$$\int W(|\vec{x} - \vec{x}_i|) d\vec{x} = 1 \quad (\text{Equation 57})$$

A smooth density field can be found as

$$\rho(\vec{x}) = \sum_j m_j W(|\vec{x} - \vec{x}_j|) \quad (\text{Equation 58})$$

where $\rho_i = \rho(\vec{x}_i)$ is the density of i^{th} particle. One must also consider

$$\int \rho(\vec{x}) d\vec{x} = \int \left(\sum_j m_j W(|\vec{x} - \vec{x}_j|) \right) d\vec{x} = \sum_j \left(m_j \int W(|\vec{x} - \vec{x}_j|) d\vec{x} \right) = \sum_j m_j \quad (\text{Equation 59})$$

which shows that for the conservation of total mass, kernel W needs to be normalized. Properties of material can be computed at an arbitrary point \vec{x} as demonstrated in (Equation 60)

$$P_A(\vec{x}) = \sum_j m_j \frac{P_j}{\rho_j} W(|\vec{x} - \vec{x}_j|) \quad (\text{Equation 60})$$

where P_j corresponds to properties of j^{th} particle and P_A corresponds to the interpolated properties at point \vec{x} . Gradient of particle properties can be easily found by using the gradient of kernel function W

$$\nabla \bar{P}_A(\vec{x}) = \sum_j m_j \frac{\bar{P}_j}{\rho_j} \nabla W(|\vec{x} - \vec{x}_j|) \quad (\text{Equation 61})$$

For solving the (Equation 56) values of ρ_i , $-\nabla p_i$, and $\mu \nabla^2 \vec{u}_i$. \vec{f}_i are needed which are the known body forces that act on the i^{th} particle such as gravitational force, collision forces and applied user interaction forces. (Equation 58) can be used to find the density of i^{th} particle ($\rho_i = \rho(\vec{x})$). After calculating density for i^{th} particle, $-\nabla p_i$ and $\mu \nabla^2 \vec{u}_i$ can also be determined

$$-\frac{\nabla p(\vec{x}_i)}{\rho_i} = -\sum_j m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \nabla W(|\vec{x}_i - \vec{x}_j|) \quad \text{pressure} \quad (\text{Equation 62})$$

$$\frac{\mu \nabla^2 \vec{u}(\vec{x}_i)}{\rho_i} = \mu \sum_j m_j \frac{\vec{u}_j - \vec{u}_i}{\rho_i \rho_j} \nabla^2 W(|\vec{x}_i - \vec{x}_j|) \quad \text{viscosity} \quad (\text{Equation 63})$$

Note that, in (Equation 62) $p_i/\rho_i^2 + p_j/\rho_j^2$ is used instead of p_j/ρ_j and in (Equation 63) $(u_j - u_i)/(\rho_i \rho_j)$ is used instead of u_j/ρ_j so the i^{th} and j^{th} particles become symmetric to each other. The pressures at i^{th} and j^{th} particle positions are required to solve (Equation 62).

Particles contain only the position, mass, and velocity properties so that pressure at particle locations need to be computed. Ideal gas state equation can be used to compute pressure

$$p = c_s^2(\rho - \rho_0) \quad (\text{Equation 64})$$

where c_s is the speed of the sound and ρ_0 is the environmental pressure (e.g. atmosphere pressure).

New particle positions can be calculated using an integration schema such as Leap-Frog or implicit/explicit Euler integration using properly small time steps. Kernels that are claimed to provide good results while being computationally efficient are

for (Equation 58)	density	$W_{poly6}(r, h) = \begin{cases} \frac{315}{64\pi h^9}(h^2 - r^2)^3 & 0 \leq r \leq h \\ 0 & otherwise \end{cases} \quad (\text{Equation 65})$
for (Equation 62)	pressure	$W_{spiky}(r, h) = \begin{cases} \frac{15}{\pi h^6}(h - r)^3 & 0 \leq r \leq h \\ 0 & otherwise \end{cases} \quad (\text{Equation 66})$
for (Equation 63)	viscosity	$W_{viscosity}(r, h) = \begin{cases} \frac{15}{2\pi h^3} \left(\frac{-r^3}{2h^3} + \frac{r^2}{h^2} + \frac{h}{2r} - 1 \right) & 0 \leq r \leq h \\ 0 & otherwise \end{cases} \quad (\text{Equation 67})$ <p>whose Laplacian is definitely positive with the following expressions</p> $\nabla^2 W(r, h) = \frac{45}{\pi h^6}(h - r)$ $W(r = h, h) = 0$ $\nabla^2(r = h, h) = 0$

This neighbor search within the kernel radius h is very slow (most computational part) hence efficient spatial hashing mechanisms are used to speed up the simulation. One widely used way divides the simulation space into a grid and keeps the list of particles within each grid cell. Grid must be modified in each time step since particles may move to another grid cells (Gingold et al 1997, Premoze et al 2003, Monaghan 2005, Müller et al 2005, Adams et al 2007).

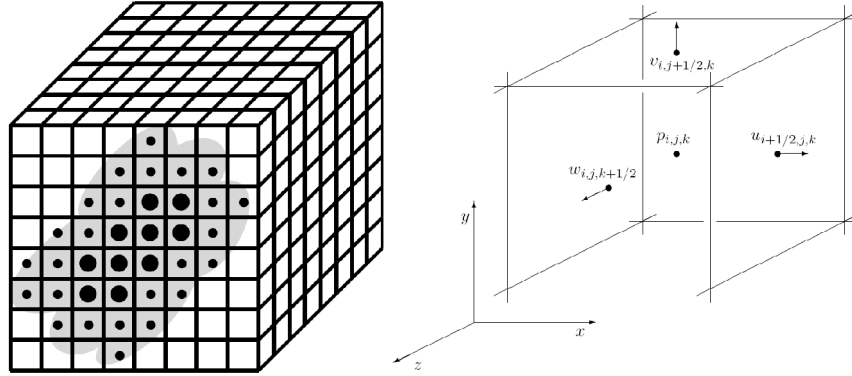
APPENDIX B. EULERIAN APPROACH

Incompressible fluid with constant density field and constant viscosity governed by so called Newtonian NSEs are given as

$$\nabla \cdot \vec{u} = 0 \quad (\text{divergence free, ensures incompressibility}) \quad (\text{Equation 68})$$

$$\frac{\partial \vec{u}}{\partial t} = -(\vec{u} \cdot \nabla)\vec{u} + \nu \nabla^2 \vec{u} - \frac{1}{\rho} \nabla p + \vec{f} \quad (\text{Equation 69})$$

MAC (staggered) grid is the most preferred method for solving incompressible NSEs. In MAC grid, density and pressure values are stored in the grid cell centers whereas velocity values are stored on the grid cell faces (Figure 74). MAC grid allows efficient and simple discretization of stored material properties.



(a) Fluid flow in a MAC grid, (b) density and velocity arrangements on a grid cell

Figure 74: MAC Staggered Grid.

The numerical simulation of (Equation 68) and (Equation 69) progresses by updating the value of \vec{u} at n^{th} time step, \vec{u}^n , to \vec{u}^{n+1} during a finite time step Δt . Using Chorin's projection method, (Equation 69) can be discretized as

$$\frac{\partial \bar{u}}{\partial t} = \frac{\Delta \bar{u}}{\Delta t} = \frac{\bar{u}^{n+1} - \bar{u}^n}{\Delta t} = -(\bar{u}^n \cdot \nabla) \bar{u}^n + \nu \nabla^2 \bar{u}^n - \frac{1}{\rho} \nabla p + \bar{f} \quad (\text{Equation 70})$$

where $\Delta t \rightarrow 0$

We can introduce \bar{u}^* which will be an intermediate velocity

$$\frac{\bar{u}^{n+1} - \bar{u}^*}{\Delta t} + \frac{\bar{u}^* - \bar{u}^n}{\Delta t} = \left[-\frac{1}{\rho} \nabla p \right] + \left[-(\bar{u}^n \cdot \nabla) \bar{u}^n + \nu \nabla^2 \bar{u}^n + \bar{f} \right] \quad (\text{Equation 71})$$

(Equation 71) is split into two equations

$$\frac{\bar{u}^* - \bar{u}^n}{\Delta t} = -(\bar{u}^n \cdot \nabla) \bar{u}^n + \nu \nabla^2 \bar{u}^n + \bar{f} \quad (\text{Equation 72})$$

$$\frac{\bar{u}^{n+1} - \bar{u}^*}{\Delta t} = -\frac{1}{\rho} \nabla p \quad (\text{Equation 73})$$

By rearranging (Equation 73) final velocity can be found as

$$\bar{u}^{n+1} = \bar{u}^* - \frac{\Delta t}{\rho} \nabla p \quad (\text{Equation 74})$$

(Equation 72) can be used to calculate intermediate velocity \bar{u}^* as

$$\bar{u}^* = \bar{u}^n + \Delta t \left(-(\bar{u}^n \cdot \nabla) \bar{u}^n + \nu \nabla^2 \bar{u}^n + \bar{f} \right) \quad (\text{Equation 75})$$

but this explicit integration requires time step Δt to be very small to produce stable results, especially for long running simulations. Instead, Stam proposed a semi-Lagrangian method which is unconditionally stable (Stam 1999). Divergence of the final velocity is ensured by setting $\nabla \cdot \bar{u} = 0$ by taking the divergence of (Equation 74)

$$\nabla \cdot \bar{u}^{n+1} = \nabla \cdot \bar{u}^* - \frac{\Delta t}{\rho} \nabla \cdot (\nabla p) = 0 \quad (\text{Equation 76})$$

Rearranging (Equation 76) gives a form of Poisson's equation

$$\nabla^2 p = \frac{\rho}{\Delta t} \nabla \cdot \bar{u}^* \quad (\text{Equation 77})$$

We must solve (Equation 77) for pressure (p) values, which will make the fluid incompressible by guaranteeing velocity to be divergence free. Stam proposed *Helmholtz-Hodge vector field decomposition* based projection method to achieve same divergence free velocity field. For both methods, the diffusion operator ∇^2 can be discretized and the resulting sparse, linear system can be solved using an implicit, iterative method such as Jacobi Method or Gauss-Seidel (Figure 75, step 5). This linear system is symmetric and positive definite (therefore it can be also solved with Conjugate Gradient) since there is a Dirichlet boundary condition for water-air that is set as $p = 1$ (or atmosphere pressure). Normal component of fluid velocity is zero, so no mass can move through the solid (Foster and Metaxas 1996, Foster and Fedkiw 2001, Goktekin et al 2004, Losasso et al 2006, Nealen et al 2006).

For viscous fluids, the tangential component of fluid velocity is zero as well (no-slip). The pressure level in the adjacent fluid cell is assigned to the pressure in the boundary cell thus acceleration among the boundary does not happen. Finite difference is also calculated using this assigned pressure in the boundary cell.

We can find the final divergence free velocity by using (Equation 77) with \vec{u}^* and p values that we found by solving Poisson's equation. To obtain \vec{u}^* from \vec{u}^n ,

- a) Find a velocity field \vec{u}^{***} by using forces $\Delta t \vec{f}$ on \vec{u}^n ,
- b) Find velocity field \vec{u}^{**} by computing the advection term, $-(\vec{u}^{***} \cdot \nabla) \vec{u}^{***}$, using a semi-Lagrangian (Figure 75) method proposed by Stam (Figure 76, step 3),
- c) Find velocity field \vec{u}^* by computing velocity diffusion term, $\nu \nabla^2 \vec{u}^{**}$, using a method proposed by Stam (Figure 76, step 4).

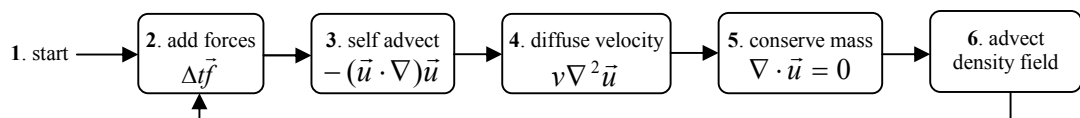


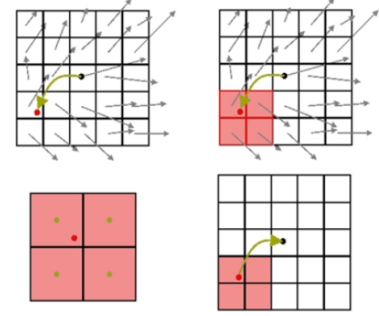
Figure 75: Flowchart of the simulation algorithm.

Step 1. Create initial fluid arrangement in MAC grid (e.g. initial density field etc.).

Step 2. Find \vec{u}^{***} , via external forces (e.g. gravity) by multiplying \vec{f} with Δt .

Step 3. Find \vec{u}^{**} by solving the advection term $-(\vec{u}^{***} \cdot \nabla)\vec{u}^{***}$, using the semi-Lagrangian advection technique proposed by Stam. This technique is based on *method of characteristics* which is used for solving partial differential equations.

In time $t + \Delta t$, velocities can be calculated by back tracking where they are advected from the previous time t . The position of velocities at time $t + \Delta t$ is on chosen grid faces, but tracked position of these velocities in time t may not be necessarily on the surface, hence velocity must be computed using an interpolation method (e.g. linear, cubic).



Semi-Lagrangian back-tracking

Advection can be seen as velocity field transporting itself. A disturbance in the velocity field propagates in accordance with the advection term $-(\vec{u} \cdot \nabla)\vec{u}$.

Step 4. Find \vec{u}^* by computing viscosity term $\nu \nabla^2 \vec{u}^{**}$, which generates diffusion and has a smoothing effect on the velocity field, having the form of a standard diffusion equation.

$$\frac{\partial \vec{u}^{**}}{\partial t} = \nu \nabla^2 \vec{u}^{**} \quad (\text{Equation 78})$$

Stam propose formulation of (Equation 78) as

$$(I - \nu \Delta t \nabla^2 \vec{u}^*) = \vec{u}^{**} \quad (\text{Equation 79})$$

Diffusion operator ∇^2 is discretized leading to a sparse system of linear equations which can be solved using an implicit, iterative method such as Jacobi Method, or Gauss-Seidel. The calculated result will be \vec{u}^* .

Step 5. Solve (Equation 77) to find pressures (p) which makes velocity divergent free. Use (Equation 74) with \vec{u}^* and p to find final divergent free velocity field.

Step 6. Use semi-Lagrangian tracing for density field advection. If the termination condition is not met go to step (2).

Figure 76: Detailed steps of the simulation algorithm.

APPENDIX C. LATTICE-BOLTZMANN METHOD (LBM)

LBM models the Boltzmann particle dynamics and becomes an important alternative to traditional CFD methods even though it is a relatively new approach. It is inspired from microscopic (molecular) physics approach to solve macroscopic continuum physics problems; hence it is between microscopic and macroscopic level aimed to capture the best of both. LBM models a simplified kinetic model that incorporate the essential physics of microscopic processes so that the macroscopic averaged properties obey the desired macroscopic NSEs. The LBM method is known for following advantages:

1. Unlike conventional numerical methods macroscopic continuum equations need not to be discretized.
2. Contains only local and linear computations, hence can be easily parallelized.
3. Fast computational speed and high accuracy.
4. Capability to handle complex moving geometries for flow-structure interactions.
5. Model interactions can be easily unified since thermodynamics is continually included in the interfacial dynamics (liquid-vapor or liquid-liquid interfaces).

Variation in the movement of microscopic particle flow between neighboring sites such as inter-particle interactions in an averaged manner can be expressed by the Boltzmann equation.

Likewise Jacobi iteration, Lattice formulations are formed by finite difference discretization and show similarities to cellular automata implementations. In cellular automation regular grid (in any finite dimension) of cells with each having finite number of states is used with discrete time steps. In each grid cell the particle

distributions that represent the probability of a particle existence with a given velocity are stored (Chen et al 1998).

In propagation phase, the particles move from one lattice node to the neighboring one in accordance with their discrete velocity. Then, in collision phase, the particles obtain a new velocity after their collision. This consequent particle propagations and collision forms the LBM simulation loop. The Bhatnager-Gross-Krook (BGK) model defines the evolution of the system toward equilibrium via statistical redistribution of momentum caused by collisions while conserving hydrodynamic moments (mass, momentum, and energy). The LBM is second-order accurate both in time and space. As time step and grid spacing goes to zero it also approaches to the NSEs for an incompressible fluid having small velocities compared to the speed of sound.

The time-averaged and space-averaged microscopic particles movements are modeled using Distribution Functions (DFs), which define the density and velocity at each lattice node using molecular populations. Specific particle interaction rules are defined so that NSEs are satisfied, for 3D, the D3Q19 model, Figure 77 with usually nineteen velocity directions, for 2D, D2Q9 model, with nine velocity directions.

During each time step (t) for a cell at located \vec{x} a DF $\tilde{f}_i(\vec{x}, t)$ has to be stored for corresponding velocity direction ($i = 1, 2, \dots, 19$), with the direction vectors \vec{e}_i as

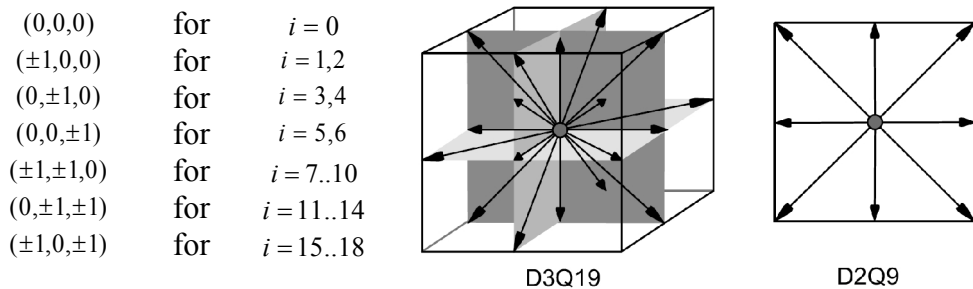


Figure 77: D3Q19 and D2Q9 LBM models and corresponding velocities.

Note that the particles at the rest state are represented by first DF $f_0(\vec{x}, t)$. The macroscopic values for density $\rho(\vec{x}, t)$ and velocity $\vec{u}(\vec{x}, t)$ can be easily computed using the moments of $f(\vec{x}, t)$ using these values.

$$\rho(\vec{x}, t) = \sum_{i=1}^{19} f_i(\vec{x}, t) \quad (\text{Equation 80})$$

$$\vec{u}(\vec{x}, t) = \frac{1}{\rho} \sum_{i=1}^{19} f_i(\vec{x}, t) \vec{e}_i \quad (\text{Equation 81})$$

The algorithm proceeds in following consequent steps;

- a) streaming
- b) collide

In streaming phase particles are moved according to their velocities $\vec{u}(\vec{x}, t)$. As shown in Figure 78, particles move in their velocity direction towards the neighboring cell since time step has a unit length ($\Delta t = 1$) due to normalization. After each streaming step, complete set of particle distribution functions are being formed for all the cells.

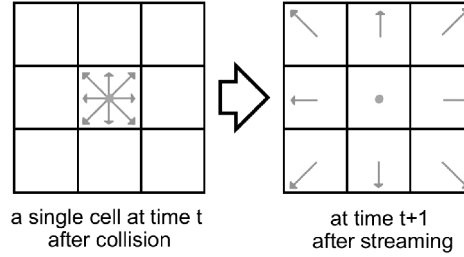


Figure 78: DFs of a grid cell before and after streaming.

Collisions between particles occur for real fluid flow which are handled in collide step in LBM. Incoming DFs in streaming step are relaxed with a set of equilibrium DFs (f_i^{eq}) where collisions are handled for each cell by calculating their density using (Equation 80) and velocity using (Equation 81). The equilibrium DFs are defined as

$$f_i^{eq}(\rho, \vec{u}) = w_i \rho \left[A + B \langle \vec{e}_i, \vec{u} \rangle + C \langle \vec{e}_i, \vec{u} \rangle^2 + D \langle \vec{u}, \vec{u} \rangle \right] \quad (\text{Equation 82})$$

where $A = 1, B = 3, C = 9/2, D = 3/2$

and w_i are weights with the relation to velocity vector lengths

$$\begin{aligned}
w_i &= 1/3 & \text{for } & i = 0 \\
w_i &= 1/18 & \text{for } & i = 1..6 \\
w_i &= 1/36 & \text{for } & i = 7..18
\end{aligned}$$

For the next time step, the value of the DF f'_i is calculated using a weight parameter determined by the viscosity of the fluid which is called as omega (ω).

$$f'_i(\vec{x} + \vec{e}_i \Delta t, t + \Delta t) = (1 - \omega)f_i(\vec{x}, t) + \omega f_i^{eq}(\rho, \vec{u}) \quad (\text{Equation 83})$$

where $\omega = 2/(6\nu + 1)$ and ν is the viscosity of the fluid within lattice units. The value of omega (ω) must be between 0 and 2 to guarantee stability. Stream and collide steps can be combined as

$$f'_i(\vec{x} + \vec{e}_i \Delta t, t + \Delta t) = f_i(\vec{x}, t) - \Delta t \omega (f_i(\vec{x}, t) - f_i^{eq}(\rho, \vec{u})) \quad (\text{Equation 84})$$

(Equation 84) approximates the particle collisions with a single parameter henceforth known as the BGK equation. Fluid can be accelerated in each cell before the computation of the equilibrium DFs to create external forces such as gravity. Immersed object and fluid interactions can be defined by momentum exchange at the contacting boundaries by alteration of the discrete packet distributions next to streaming step. This can further be used for deforming or moving the object at fluid boundary. Periodic, no-slip, free-slip, frictional slip, sliding walls, and open inlet and outlet boundary conditions can be implemented for simple (e.g., boundary aligned with the grid) and complex geometries (e.g., curve). For example, zero normal and tangential velocity can be obtained along the boundary by taking the opposing DF to reflect them during streaming which leads to no-slip boundary condition (Chen et al 1998, Tan et al 2009).

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Nar, Fatih
Nationality: Turkish (TC)
Date and Place of Birth: 2 July 1975, Adana
Marital Status: Married
Phone: +90 533 725 72 63
Email: dr.fatih.nar@gmail.com

EDUCATION

Degree	Institution	Year of Graduation
MS	METU, Informatics Institute	2003
BS	METU, Computer Education	1999

WORK EXPERIENCE

Year	Place	Enrollment
2010- Present	Space Defense Technology	Senior Software Engineer
2010 - 2010	Çankaya University	Instructor
2008 - 2009	Türksat	Database Administrator
2006 - 2007	Turkish Army	System Administrator
2004 - 2005	METU, Informatics Institute	Researcher
1999 - 2003	Elba Computer	Software Developer

FOREIGN LANGUAGES

Fluent English

PUBLICATIONS

Journal Papers:

[1] Mumcuoğlu, E., Nar, F., et al. (2006). Simultaneous Surface Registration of Ictal, Interictal SPECT and MR Images for Epilepsy Studies. Nuclear Medicine Communication.

Conference Papers:

[1] Nar, F., Yılmaz, E., (2010). Graphic Processor Unit Accelerated Simulated Annealing Framework. EURO XXIV.

[2] Mumcuoğlu, E., Nar, F., Yardımcı, Y., Koçak, U., et al. (2008). Image-based Retrieval System and Computer-aided Diagnosis System for Renal Cortical Scintigraphy Images. SPIE Computer-Aided Diagnosis, Vol. 6915.

[3] Mumcuoğlu, E., Nar, F., et al. (2005). Simultaneous Triple-Registration of Ictal SPECT, Interictal SPECT and MR Images for Epilepsy Studies: Method and Validation. EUSIPCO.

[4] Nar, F., Mumcuoğlu, E., et al. (2005). Simultaneous Registration of Ictal SPECT, Interictal SPECT and MR Images for Epilepsy Studies: Method and Simulations. IASTED, 458-115.

[5] Nar, F., Çetin, A., (2003). Generalization and Localization Based Style Imitation for Grayscale Images. ISCIS, 465-473

(National)

Conference Papers:

[1] Nar, F., Demikesen, C., Okman, O.E., Çetin, M. (2011). A Region Based Target Detection Method for SAR Images. IEEE Signal Processing and Communications Application.

[2] Nar, F., Mumcuoğlu, E., Koçak, U., et al. (2006). A Content Based Retrieval System for Renal Scintigraphy Images. IEEE Signal Processing and Communications Applications.

VITA

Fatih Nar was born in Adana on July 2, 1975. He received his B.S. degrees in Computer Education in Education Faculty of METU in 1999. He received a MSc. Degree in Information Systems Department of Informatics Institute, METU in 2003. Since 1999 he has been worked in various software development companies and institutions as software developer, team leader, database developmental administrator, researcher, instructor, senior system engineer, and technical leader. His main areas of interest are image processing, pattern recognition, automatic target recognition, parallel scientific computation, computer graphics, and scientific mobile application development.