ISTANBUL TECHNICAL UNIVERSITY ★ INFORMATICS INSTITUTE

VESSEL SEGMENTATION AND SURFACE RECONSTRUCTION FROM MRA IMAGES

M.Sc. THESIS

Devran UĞURLU

Department of Informatics

Computational Science and Engineering Programme

JANUARY 2012

ISTANBUL TECHNICAL UNIVERSITY ★ INFORMATICS INSTITUTE

VESSEL SEGMENTATION AND SURFACE RECONSTRUCTION FROM MRA IMAGES

M.Sc. THESIS

Devran UĞURLU
702091014

Department of Informatics

Computational Science and Engineering Programme

Thesis Advisor: Prof. Dr. M. Serdar ÇELEBİ

JANUARY 2012

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ BİLİŞİM ENSTİTÜSÜ

MRA GÖRÜNTÜLERİNDEN DAMAR AYRIŞTIRILMASI VE DAMAR YÜZEYİNİN OLUŞTURULMASI

YÜKSEK LİSANS TEZİ

Devran UĞURLU
702091014

Bilişim Anabilim Dalı

Hesaplamalı Bilim ve Mühendislik Programı

Tez Danışmanı: Prof. Dr. M. Serdar ÇELEBİ

OCAK 2012

Devran Uğurlu, a **M.Sc.** student of ITU **Informatics Institute** student ID **702091014**, successfully defended the **thesis** entitled "**VESSEL SEGMENTATION AND SURFACE RECONSTRUCTION FROM MRA IMAGES**", which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor : **Prof. Dr. M. Serdar ÇELEBİ** ....
İstanbul Technical University

Jury Members : **Prof. Dr. M. Serdar ÇELEBİ** ....
İstanbul Technical University

Doç. Dr. Gözde ÜNAL ....
Sabancı University

Yrd. Doç. Devrim ÜNAY ....
Bahçeşehir University

Date of Submission : 19 December 2011
Date of Defense : 23 January 2012

FOREWORD

I would like to thank my family and advisors for their invaluable support. This work is supported by Istanbul Technical University(ITU) Informatics Institute and Technische Universität München(TUM) Informatics Institute.

December 2011

Devran UĞURLU
Comp. Sci. and Engineering

TABLE OF CONTENTS

Page

FOREWORD .... **vii**
TABLE OF CONTENTS .... **ix**
ABBREVIATIONS .... **xi**
LIST OF TABLES .... **xiii**
LIST OF FIGURES .... **xv**
SUMMARY .... **xvii**
ÖZET .... **xix**
1. INTRODUCTION **1**
2. ANGIOGRAPHIC IMAGING MODALITIES **7**
  2.1 Digital Subtraction Angiography(DSA) 7
  2.2 Computed Tomography Angiography(CTA) 8
  2.3 Magnetic Resonance Angiography(MRA) 9
3. VESSEL SEGMENTATION **13**
  3.1 Objective 13
  3.2 Surface Evolution With the Level Set Method 13
    3.2.1 Representing the surface as level set of a volume 13
    3.2.2 Evolving the embedding volume 16
  3.3 Choosing the Function That Will Guide the Evolution 18
4. SURFACE RECONSTRUCTION **27**
  4.1 Objectives 27
  4.2 Implicit Surface Polygonization 27
  4.3 Mesh Smoothing 31
5. VESSEL SEGMENTATION AND RECONSTRUCTION ALGORITHM AND RESULTS **33**
  5.1 Implementation Details 33
  5.2 Results and Discussion 35
6. CONCLUSION AND RECOMMENDATIONS **45**
  6.1 Future Work 46
REFERENCES **47**
APPENDICES **53**
CURRICULUM VITAE **61**

ABBREVIATIONS

2D	**:** 2-dimensional
3D	**:** 3-dimensional
CFD	**:** Computational Fluid Dynamics
CTA	**:** Computed Tomography Angiography
DSA	**:** Digital Subtraction Angiography
ITK	**:** The Insight Toolkit
MHA	**:** Meta Image
MIP	**:** Maximum Intensity Projection
MRA	**:** Magnetic Resonance Angiography
NRRD	**:** Nearly Raw Raster Data
PDE	**:** Partial Differential Equation
PLY	**:** Polygon File Format
VTK	**:** The Visualization Toolkit

LIST OF TABLES

Page

Table A.1 : Quantitative evaluation of segmentation results for synthetic images with $\sigma_{\text{noise}} = 20$. 57

Table A.2 : Quantitative evaluation of segmentation results for synthetic images with $\sigma_{\text{noise}} = 60$. 58

LIST OF FIGURES

Page

Figure 2.1 : A DSA image of blood vessels in brain. *Figure taken from [43].* 8
Figure 2.2 : a) MIP with bones removed. b) Volume rendering with bones not removed. *Figure taken from [45].* 9
Figure 2.3 : An MIP of an MRA image showing a cerebral aneurysm. 10
Figure 3.1 : Illustration of implicit representation given by the function $\phi(\boldsymbol{x}) = x^2 + y^2 - 9$ and eq. (3.1). Left: Surfaces for different k in the continuous case. Right: Zero level set($k = 0$) in the discrete case....... 14
Figure 3.2 : The continuous Gaussian second derivative function for scales $\sigma = 1$ and $\sigma = 3$. 19
Figure 3.3 : The improvement resulting from our proposed modification to Frangi’s vesselness measure illustrated on a coronal 2D slice of a 3D brain MRA image. Top image: The segmentation result when original Frangi vesselness measure is used. Middle Image: The segmentation result when our modified vesselness measure is used. Bottom Image: The original slice. 25
Figure 4.1 : Left: Surface constructed using only local curvature. Right: Surface constructed using a guidance field. *Figure taken from [34]* 28
Figure 4.2 : Illustration of why the edge length ideally subtends the same angle of the osculating circle. *Figure taken from [58].* 28
Figure 5.1 : Summary of the algorithm. 33
Figure 5.2 : Top row: Segmentation results of the geodesic active contour algorithm. Second row: Segmentation results of the confidence connected region growing algorithm. Third row: Segmentation results of proposed method. Bottom row: Maximum intensity projections of raw data. (a) A brain MRA with $448 \times 448 \times 128$ voxels and $0.51\text{mm} \times 0.51\text{mm} \times 0.8\text{mm}$ voxel size. (b) A brain MRA with $512 \times 512 \times 155$ voxels and $0.47\text{mm} \times 0.47\text{mm} \times 0.6\text{mm}$ voxel size. (c) An abdominal MRA with $512 \times 512 \times 70$ voxels and $0.95\text{mm} \times 0.95\text{mm} \times 1.00\text{mm}$ voxel size. 36
Figure 5.3 : Top row: Segmentation results of the simple geodesic active contour algorithm. Second row: Segmentation results of the confidence connected region growing algorithm. Third row: Segmentation results of proposed method. Bottom row: Maximum intensity projections of raw data. (a) An abdominal MRA with $512 \times 512 \times 50$ voxels and $0.78\text{mm} \times 0.78\text{mm} \times 1.00\text{mm}$ voxel size. (b) A thigh MRA with $512 \times 512 \times 70$ voxels and $0.92\text{mm} \times 0.92\text{mm} \times 1.00\text{mm}$ voxel size. (c) A synthetic image with added Gaussian noise $\sigma_{\text{noise}} = 60$ that has $256 \times 256 \times 256$ voxels and $1.00\text{mm} \times 1.00\text{mm} \times 1.00\text{mm}$ voxel size.... 37

Figure 5.4 : Comparison of using Catmull-Rom splines(top) vs B-splines(bottom) on one of the brain images. $\rho = 0.4,\ \eta^{-1} = 0.8$. 41
Figure 5.5 : Effect of Taubin smoothing. Top: Before Taubin smoothing. Surface created using $\rho = 0.4,\ \eta^{-1} = 0.8$ and B-splines. Bottom: After Taubin smoothing with $\lambda = 0.50$ and $\mu = -0.54$ for 50 iterations. 42
Figure 5.6 : Comparison of our recommended scheme(bottom) against Marching Cubes(top). 43
Figure A.1 : From top to bottom: Segmentation results for a brain MRA with $448 \times 448 \times 128$ voxels and $0.51\text{mm} \times 0.51\text{mm} \times 0.8\text{mm}$ voxel size, an abdominal MRA with $512 \times 512 \times 70$ voxels and $0.95\text{mm} \times 0.95\text{mm} \times 1.00\text{mm}$ voxel size, and the synthetic image with $\sigma_{\text{noise}} = 60$ which has $256 \times 256 \times 256$ voxels and $1.00\text{mm} \times 1.00\text{mm} \times 1.00\text{mm}$ voxel size. From left to right: Varying γ. 54
Figure A.2 : From top to bottom: Segmentation results for a brain MRA with $448 \times 448 \times 128$ voxels and $0.51\text{mm} \times 0.51\text{mm} \times 0.8\text{mm}$ voxel size, an abdominal MRA with $512 \times 512 \times 70$ voxels and $0.95\text{mm} \times 0.95\text{mm} \times 1.00\text{mm}$ voxel size, and the synthetic image with $\sigma_{\text{noise}} = 60$ which has $256 \times 256 \times 256$ voxels and $1.00\text{mm} \times 1.00\text{mm} \times 1.00\text{mm}$ voxel size. From left to right:Varying d. 55
Figure A.3 : From top to bottom: Segmentation results for a brain MRA with $448 \times 448 \times 128$ voxels and $0.51\text{mm} \times 0.51\text{mm} \times 0.8\text{mm}$ voxel size, an abdominal MRA with $512 \times 512 \times 70$ voxels and $0.95\text{mm} \times 0.95\text{mm} \times 1.00\text{mm}$ voxel size, and the synthetic image with $\sigma_{\text{noise}} = 60$ which has $256 \times 256 \times 256$ voxels and $1.00\text{mm} \times 1.00\text{mm} \times 1.00\text{mm}$ voxel size. From left to right:Varying ξ. 56
Figure B.1 : Surface reconstruction results for abdominal image(left) and cerebral image(right) with varying ρ. Top row: $\rho = 0.6$ middle row: $\rho = 0.4$ bottom row: $\rho = 0.2$. B-splines and $\eta^{-1} = 0.8$ are used for all images. 59
Figure B.2 : Surface reconstruction results for abdominal image(left) and cerebral image(right) with varying η. Top row: $\eta^{-1} = 0.8$, middle row: $\eta^{-1} = 0.65$, bottom row: $\eta^{-1} = 0.5$. B-splines and $\rho = 0.4$ are used for all images. 60

VESSEL SEGMENTATION AND SURFACE RECONSTRUCTION FROM MRA IMAGES

SUMMARY

3D surface reconstruction of vascular structures plays a very important role in the medical field as vascular problems can be lethal and are in fact among the leading causes of death. The reconstructed patient specific vessel structure can be visualized and used for diagnostic and surgical planing purposes. Furthermore, analysing the vascular structure and detecting risk areas using CFD simulations can help in making therapeutic decisions and might, in the future, make it possible to prevent some diseases before they show any symptoms. Two very important applications of patient specific CFD simulations are the evaluation of the risk of atherosclerosis and cerebral aneurysm rupture. Atherosclerotic plagues are known to occur more frequently in regions where the flow is not laminar and wall sheer stress is low. Hemodynamic factors are also thought to play a role in the risk of aneurysm rupture. As measuring blood flow in vivo is not reliable or convenient, the only way to assess the risk is by numerical simulation on patient specific 3D vessel structure.

The vessel surface reconstruction problem consists of two main steps. First, all the voxels in the volume data are labeled as vessel or non-vessel. This step is called vessel segmentation. Then, using the segmentation, the vessel surface is reconstructed as a polygonal mesh. If the mesh is to be used for numerical simulations instead of only visualization, then the mesh should be high quality which means that it should contain polygons of similar sizes and angles. Regrettably, manual segmentation of 3D data is a lengthy and cumbersome process and reconstruction of a high quality surface from the segmented data is not trivial. Therefore, both the problem of automatic or semi-automatic vessel segmentation from medical images and the problem of reconstructing a high quality surface from segmented volume data have been extensively studied in literature and there are many different types of approaches. Compared to the amount of research that study vessel segmentation and surface reconstruction seperately, the number of studies that aim to combine these methods for the purpose of patient specific numerical simulations is relatively low and they generally consider only a specific anatomical region.

In this work, we focus on vessel segmentation and reconstruction from MRA data and we aim to design a unified model that works on all anatomical regions, can detect both large and small vessels and uses minimal user interaction.

For the segmentation step, a level-set evolution scheme based on local geometric information is used. More specifically, the vessel surface is represented as the zero level-set of a 3D hypersurface and the hypersurface is evolved under guidance of an evolution function derived from multi-scale Hessian analysis and mean curvature. The evolution is modeled as an initial value problem resulting from a PDE. The surface can be initialized automatically using the Hessian analysis to detect bright

tubular structures. The results are analysed visually on real MRA data and quantitatively on a synthetically created dataset deformed with various noise intensities. It is seen from the results that the proposed method is promising.

The second step of our work is to construct a polygonal representation of the vessel surface using the result of the first part which is an implicit representation of the vessel surface as a discrete sampling of a 3D volume. The standart classical method for polygonizing implicit surfaces is Marching Cubes which constructs a surface with triangles. Although Marching Cubes is fast and guaranteed to be accurate, it generates a very rough surface and can contain low quality triangles(highly acute-angled triangles). Thus, it is not suitable if we want to run numerical simulations using the resulting surface. Therefore, we use an advancing front method which is known to generate smoother surfaces with better triangle quality. Advancing front methods have the downside of being slow but this is not such a big issue when the surface reconstruction is done offline. Finally, Taubin smoothing is used to further smooth the surface. Compared to simpler smoothing methods like Laplacian and Gaussian smoothing which produce shrinkage, Taubin smoothing does not change the topology of the vessel structure if used correctly. The results obtained from using an advancing front method and Taubin smoothing are compared to the classical Marching Cubes results visually to show that much higher triangle quality is achieved while maintaining acceptable accuracy.

MRA GÖRÜNTÜLERİNDEN DAMAR AYRIŞTIRILMASI VE DAMAR YÜZEYİNİN OLUŞTURULMASI

ÖZET

Kalp-damar hastalıklarının özellikle batılı ülkelerde ölüm sebeplerinin başında gelmesi, damar yapısının 3 boyutlu olarak oluşturulmasını çok önemli kılmaktadır. Hastaya özel oluşturulan damar yüzeyi, görselleştirilerek teşhis veya cerrahi planlama amaçlı kullanılabilir. Bunun yanında, oluşturulan yüzey temel alınarak yapılacak olan hesaplamalı sıvı dinamiği(HSD) simülasyonları ile hastalık oluşma riski yüksek olan bölgeler tespit edilebilir ve böylece gelecekte bazı hastalıkların herhangi bir belirti göstermeden önce engellenmesi mümkün olabilir. Hastaya özel HSD simülasyonlarının çok önemli iki uygulaması ateroskleroz ve serebral anevrizma yırtılması riskinin belirlenmesidir. Aterosklerotik plakların, kan akışının düzgün olmadığı ve damar duvarı üzerindeki kayma gerilmesinin düşük olduğu bölgelerde oluşma riskinin daha fazla olduğu bilinmektedir. Anevrizma yırtılması riski için de benzer hemodinamik etkiler rol oynamaktadır. Kan akışının hastanın içinde ölçülmesi güvenilir veya rahat bir işlem olmadığından, akışın hastaya özel damar yapısı üzerinde HSD simülasyonu yapılarak ölçülmesi gerekmektedir.

Damar yapısının 3 boyutlu olarak oluşturulması problemi iki ana adımdan oluşur. İlk önce, hacim verisindeki bütün vokseller damara ait veya değil olarak etiketlenir. Bu adıma damar ayrıştırılması adı verilir. Daha sonra, bu ayrıştırılmış veri kullanılarak damar yapısı poligonal meş şeklinde 3 boyutlu olarak oluşturulur. Eğer meş sadece görselleştirme amaçlı değil, ayrıca simülasyonlar için de kullanılacaksa meşin yüksek kalitede olması gerekir. Yani, meşi oluşturan çokgenlerin açı ve büyüklükleri nümerik simülasyona uygun olmalıdır. Ne yazık ki, 3 boyutlu verilerden damar ayrıştırılmasının manuel olarak yapılması uzun ve zahmetli bir işlemdir. Ayrıca, ayrıştırılmış veriden yüksek kaliteli meş oluşturmak da kolay değildir. Bu nedenle, hem otomatik ve yarı-otomatik damar ayrıştırılması, hem de ayrıştırılmış veriden poligonal meş şeklinde yüzey oluşturulması sorunları bilimsel yazında çokça incelenmiştir ve birçok farklı yaklaşım bulunmaktadır. Bu iki adım için ayrı ayrı yapılan çalışmaların sayısına kıyasla, iki adımı birleştirmek ile ilgili fazla çalışma bulunmamaktadır ve mevcut çalışmalar genelde tek bir anatomik bölgeye odaklıdır.

Biz bu çalışmada MRA verisinden damar ayrıştırılması ve damar yüzeyinin oluşturulmasına odaklanarak, bütün anatomik bölgelerde çalışan, kalın, dar, sağlıklı ve hastalıklı her türlü damarı ayrıştırabilen ve mümkün olduğunca az kullanıcı müdahalesine gerek duyan birleşik bir model tasarlamayı hedefliyoruz.

MRA verisini seçmemizin nedeni MRA’nın, CTA ve DSA gibi anjiyografi tekniklerine kıyasla hasta için daha az risk taşıması ve görüntüdeki en parlak yapıların damar olduğu varsayımının genelde geçerli olmasıdır. Bu varsayım, damar ayrıştırılması adımını kolaylaştıracaktır.

Damar ayrıştırılması adımında yerel geometri bilgisini kullanarak evrimleşen bir level-set yaklaşımı kullanılmıştır. Daha açık olarak söylersek, damar yüzeyi, 3 boyutlu bir hiperyüzeyin sıfır level-seti olarak ifade edilmiş ve yerel multi-scale Hessian ve ortalama eğrilik bilgisinden türetilen bir evrim fonksiyonun rehberliği ile evrimleştirilmiştir. Hessian bilgisi ve damarların görüntüdeki en parlak yapı olduğu varsayımı kullanılarak, parlak ve boruya benzeyen bir yapı içerisinde bulunan noktalar tespit edilebilir. Ortalama eğrilik de eğriliği fazla olan damarlarda ayrıştırmanın devam etmesi için itici bir rol oynar çünkü bu gibi kıvrımlı damarlarda boruya benzerlik düşük olduğundan Hessian bilgisi yetersiz kalabilmektedir. Evrim fonksiyonu, Hessian bilgisinden türetilen bir ölçü ile ortalama eğriliği, kullanıcı tarafından belirleyen katsayılar yardımıyla dengeler.

Evrim süreci, bir kısmi türevli diferansiyel denklem için başlangıç değer probleminin çözümü olarak modellenmiştir. Başlangıç yüzeyi, damar içerisinde olduğu bilinen noktaların etrafında küreler oluşturulmak suretiyle seçilebilir. Noktaların otomatik seçimi için de evrim fonksiyonunda kullandığımız Hessian ölçüsü kullanılabilir. Bu ölçünün en yüksek olduğu noktaların bir damar içerisinde olma olasılığı çok yüksektir. Tabii ki, bazı görüntülerdeki sorunlar, bu otomatik seçilimi etkileyebilmektedir ve böyle durumlarda kullanıcı müdahalesi gerekmektedir. Başlangıç yüzeyi belirlendikten sonra yüzey, evrim fonksiyonu rehberliği altında, yakınsama sağlanıncaya kadar evrimleşir.

Damar ayrıştırılma adımının sonuçları, gerçek MRA verileri üzerinde görsel olarak ve sentetik olarak oluşturulmuş ve gürültü eklenmiş veriler üzerinde sayısal olarak değerlendirilmiştir. Sonuçlara bakıldığında Hessian bilgisinin, ortalama eğriliğe baskınlığı artırıldığında ayrıştırılan bölgenin damar olma olasılığının daha yüksek olduğu ancak özellikle yüksek eğrilikli damarların ayrıştırılmadığı görülmüştür. Tersine olarak, ortalama eğriliğin baskınlığı artırıldığında daha fazla damar ayrıştırılmakta ancak aşırı ayrıştırma durumunun ortaya çıkma şansı da artmaktadır. Ortalama eğrilik, noktanın damar içinde olup olmadığı ile ilgili bir bilgi taşımadığından bu beklenen bir durumdur. Diğer bir önemli gözlem de kullanıcı tarafından belirlenen katsayıların seçiminin ayrıştırılmaya etkisinin, gerçek görüntülerde sentetik görüntülere kıyasla çok daha yüksek olduğudur. Bunun iki nedeni vardır: Birincisi, sentetik görüntülerdeki tek bozulma Gaussian gürültü iken gerçek görüntülerde birçok farklı sorun olabilmesidir. İkinci neden, sentetik görüntülerde eğriliği yüksek olan damar olmamasıdır. Daha önce de bahsettiğimiz gibi, damarın eğriliği yüksek olduğunda Hessian ölçüsü yüksek değerler vermemekte ve ayrıştırmanın devam etmesi için eğrilikten destek gelmesi gerekmektedir. Böylece, önerdiğimiz metodun en önemli sorunu, kullanıcı tarafından belirlenen katsayıların doğru seçilimidir. Üzerinde çalıştığımız veriler için iyi sonuçlar veren katsayılar önermiş olsak da bu katsayıların genelleştirilebilir olduğunu söyleyemeyiz. Tıbbi görüntülerde birçok farklı artifact olması ve damarların çok farklı geometrik şekillerde bulunabilmesi, her görüntü için kullanıcı müdahalesi olmadan iyi sonuç verecek bir metod bulmayı çok zor hale getirmektedir.

Çalışmamızın ikinci kısmı, birinci kısmın sonucunu, yani damar yüzeyini kapalı olarak ifade eden 3 boyutlu bir hacmin ayrık örneklemesini, kullanarak yüzeyi çokgenler yardımıyla ifade etmektir. Kapalı ifade edilmiş yüzeyleri çokgenleştirmek için kullanılan standart yöntem üçgenler kullanarak bu işlemi yapan Marching Cubes'dur. Bu yöntem çok hızlı ve isabetli olmakla beraber düşük kaliteli üçgenlerden (çok dar açılı veya çok küçük) oluşan oldukça çıkıntılı yüzeyler oluşturmaktadır. Bu nedenle, sonuçta elde ettiğimiz yüzeyi nümerik simülasyonlarda kullanmak istiyorsak, Marching Cubes uygun bir yöntem değildir. Dolayısıyla, yüksek kaliteli üçgenlerden oluşan daha pürüzsüz yüzeyler oluşturduğu bilinen bir advancing front metodunu tercih ettik. Advancing front metodlarının temel dezavantajı yavaş olmalarıdır ancak nümerik simülasyonlar için kullanılacak yüzeylerin oluşturulma aşamasının gerçek zamanlı olması gerekmediğinden bu soruna tahammül edilebilir. Yöntemde, yüzeyin pürüzsüzlüğü ile verilen veriye bağlılığını dengeleyen, kullanıcı tarafından tanımlanan katsayılar bulunmaktadır. Bu katsayılar, oluşturulan yüzeyin nasıl bir uygulamada kullanılacağına ve girdi olarak alınan verinin ne kadar gürültülü olduğu gözönünde bulundurularak seçilmelidir. Ayrıca, yüzey girdi verisine ne kadar bağlı olursa yöntemin çalışma süresinin de o kadar uzayacağı hesaba katılmalıdır.

Son olarak, yüzeyin daha da pürüzsüzleştirilmesi için Taubin pürüzsüzleştirmesi kullanılmıştır. Bu yöntem, Laplacian veya Gaussian gibi klasik pürüzsüzleştirme yöntemlerinin aksine, doğru kullanıldığında, yüzeyde daralma oluşturmamaktadır.

Sonuçlar, Marching Cubes yöntemiyle elde edilen sonuçlarla görsel olarak karşılaştırılmış ve önerilen yöntemin Marching Cubes'a kıyasla çok daha yüksek kalitede yüzeyler oluşturduğu ve kıyaslanabilir isabette olduğu görülmüştür.

Çalışmamızın bütünü gözönüne alındığında üzerinde durulması gereken en önemli nokta, her adımda istenilen bazı özelliklerin elde edilmesi için istenen başka özelliklerden vazgeçilmesi gerektiğidir. Damar ayrıştırılması kısmında isabetlilik ile kullanıcı müdahalesine gereksinim, hesaplama pahası (computation cost) ve genellik, damar yüzeyi oluşturulması kısmında ise isabetlilik ile pürüzsüzlük ve hesaplama pahası dengelenmelidir. Bu dengeleme işlemini yaparken, elde edilecek olan yüzeyin kullanılacağı özel tıbbi uygulamanın gereksinimleri dikkate alınmalıdır. Birkaç örnek vermek gerekirse, tehşis için yeterli olan bir isabet oranı, beyin ameliyatı planlaması için yeterli olmayabilir. Damar yüzeyinin pürüzsüz olması nümerik simülasyonlar için gerekli olmakla beraber tehşis veya ameliyat planlaması için istenmeyen bir durum olabilir. Sonuç yüzeyindeki çokgen sayısı ameliyat planlaması veya tehşis için bir sorun oluşturmayacak ancak nümerik simülasyonun hızını ciddi şekilde etkileyecektir. İdeal olarak gelecekte ulaşılmak istenilen nokta, verilen herhangi bir tıbbi süreç için herhangi bir tıbbi görüntüyü hiçbir kullanıcı müdahalesi veya düzeltmesine gerek olmaksızın anlamlı şekilde işleyecek bir programa sahip olunmasıdır. Ancak şu an bu noktadan uzak olunduğundan, bir algoritmanın her durumda işe yaramasını beklememek gerekir. Dolayısıyla, tıbbi uygulamalar üzerinde çalışan bilgisayar bilimcilerin, hekimlerle yakın bir işbirliği içerisinde çalışmaları ve böylece geliştirdikleri yöntemin kullanılacağı tıbbi uygulamanın gereksinimlerinden haberdar olmaları büyük önem arz etmektedir.

1. INTRODUCTION

With advances in medical imaging technology, medical imaging data available have rapidly increased and this trend continues. Although this is good news, it brings along the problem of how to analyse and use this data correctly and efficiently. An important subset of this area is vessel segmentation and surface reconstruction. Vessel segmentation is the extraction of vessel structures from medical images by labeling pixels/voxels as vessel or non-vessel. Then, surface of the vasculature can be reconstructed as a polygonal mesh from this data.

Importance of vessel segmentation and surface reconstruction is clear from the fact that vascular diseases are among the leading causes of death [1]. Reconstructed patient specific vessel structure can be visualized and used for diagnostic and surgical planing purposes. Furthermore, analysing the vascular structure and detecting risk areas using CFD simulations can help in making therapeutic decisions and might, in the future, make it possible to prevent some diseases before they show any symptoms. Two very important applications of patient specific CFD simulations are the evaluation of the risk of atherosclerosis and cerebral aneurysm rupture. Atherosclerotic plagues are known to occur more frequently in regions where the flow is not laminar and wall sheer stress is low [2]. Hemodynamic factors are also thought to play a role in the risk of aneurysm rupture [3]. As measuring blood flow in vivo is not reliable or convenient, the ideal way to assess the risk is by numerical simulation on patient specific 3D vessel structure. More examples can be found in a good review of patient specific CFD simulation applications by Löhner et. al. [4].

Unfortunately, segmenting 3D images manually is a lengthy and cumbersome task. Moreover, surface reconstruction from segmented data is not trivial and poorly constructed surfaces are not suitable for numerical simulations and can give wrong results even if the segmentation is accurate. Furthermore, most of the studies in literature focus seperately on vessel segmentation and surface reconstruction. As there exist a huge amount of research in these areas, there are lots of different input and output types. Therefore, how to combine these two steps is not immediately clear. Compared to the amount of research done seperately on vessel segmentation and surface reconstruction, studies that aim to combine vessel segmentation and surface reconstruction for patient specific applications are relatively few and target specific anatomical regions.

Not surprisingly, there is a tremendous amount of research on the problem of vessel segmentation. Two great reviews of the area are given in [5,6]. As seen in these reviews, there are many different approaches and target application areas. For this study, we will focus on local geometry based level-set evolution methods as it is impossible to investigate all types of approaches due to timing constraints. Admittedly, this choice is mostly subjective as the huge number of methods and application areas coupled with the lack of a public database with available ground truth segmentations and standard quantitative evaluation parameters make comparison of methods very difficult even if the methods are targeted for similar applications. Therefore, it is unfortunately not possible to quantitatively select a best approach for a certain application and our judgement has to be based on visual results and what seems to be interesting and promising. As local geometry based level-set evolution methods are flexible and can be applied to any type of vessel or shape, we decided to focus on this type of method for this study.

The idea of level set evolution methods is to represent the surface implicitly as the zero level-set of a hypersurface and evolve the hypersurface. The hypersurface is initialized using seed points that are known to be inside vessels. This initialization can be manual, semi-automatic or automatic depending on the image. Numerical theory of surface evolution using level sets was first introduced by Osher and Sethian in [7] and is described in more detail in [8]. Level-set evolution methods proved to be superior to classical snake methods as they are able to represent topological changes in the surface without a need for reparametrization and can evolve surfaces of arbitrary shape.

Caselles et al. [9] and Malladi et al. [10] independently applied the level set framework to segment shapes from images. They used image gradients and mean curvature of the surface in the level set evolution equation. The expansion speed of the surface is inversely proportional to the gradient magnitude. This makes sense as the gradient magnitude is expected to be low inside the object of interest and high on the boundary. Mean curvature is used as a volume minimizing force to promote stability and smoothness. This method can give sufficient results for segmenting large vessels in low noise images but it will not work on small vessels as the gradient magnitude will be large inside the vessel too. A now well-known method called CURVES was later developed by Lorigo et al. [11] for vessel segmentation. The main idea of CURVES was to evolve the underlying centerline curve of the vessel instead of the surface. In practice however, they evolved a tube around the curve and thus the method was similar to the method of Caselles et al. [9] and Malladi et al. [10] but it used the smaller of the principal curvatures instead of mean curvature for regularization. This means that it avoided regularizing against the curvature that corresponds to the radii of the vessel and instead regularized against the curvature of the underlying curve. This allowed segmentation of more vessels but vessels whose underlying curve has high curvature can still be left out of the segmentation. Yan and Kassim [12] proposed adding a capillary force term to the evolution function proposed in [9] to facilitate segmentation along thin vessels and reported that their algorithm outperforms CURVES. Law and Chung [13] derived a level set evolution equation by modelling the problem as an elastic solid deforming under pressure from a liquid. Liquid is the vasculature and background is the elastic solid. The terms in the obtained equation involve multi-scale Hessians and mean curvature. The method is reported to achieve better results than the CURVES algorithm. In a different approach, Frangi et al. developed a vessel enhancement filter based on the Hessian of the image intensity [14]. The Hessian is calculated at different scales for multi-scale detection of vessels with different radii. The vesselness measure function defined in this filter was later incorporated into the level set segmentation framework by Yu et al. [15]. They combined the gradient and mean curvature terms with a new term derived from Frangi's vesselness measure and the terms are weighted by user-defined parameters. The algorithm was only tested on 2D images.

The surface reconstruction problem is to polygonize a surface that is defined by other means, e.g. point cloud, parametric function, implicit function. We will of course constrain ourselves according to the output of the vessel segmentation method, that is, a regular uniform 3D grid sampling of an implicit function. The area of implicit surface polygonization has been extensively studied in the past three decades and an overview of different types of methods is given in [16]. The methods can be classified as volume based and surface tracking based. Although most methods assume a continuous implicit function, they have to sample the function anyway and when the input is already a discrete sampling, depending on the method, it can either be used directly as the sampling or interpolation is used to find the value at required points.

The classic and most popular volume based method is without a doubt the Marching Cubes algorithm [17,18]. It works by sampling the implicit function at corners of cubes in a regular uniform grid. Then the values at each corner of a cube is checked to determine if and how the surface intersects the cube. If the surface intersects the cube, a polygon is created by combining points on the edges of the cube where the surface intersects. This method is simple, fast and known to accurately represent the surface up to the chosen grid resolution. It is also so widely used that it can safely be used as a validation tool when a new algorithm is tried. Another advantage is that as it samples the volume cubicly, it lends itself well to discrete input on 3D volumes. On the other hand, the fact that the sampling is a fixed grid means the number of triangles near a certain point on the surface is also fixed. This is not desirable because the ideal number of triangles is different in different parts of the surface. For example, areas with high curvature require more triangles and using a too large triangle size would mean some details are not represented in the final surface. If the triangle size is reduced, this will unnecessarily increase computation time because other parts of the surface may not require such a small triangle size. Another problem of the Marching Cubes algorithm is that the surface is very rough with staircase artifacts and most triangles have low quality(which means they are too acute-angled). Thus, this method is not suitable for running numerical simulations. Other volume based methods are mostly based on the Marching Cubes. Some of the famous variations are Bloomenthal's Marching Tethedra [19] which uses tetrahedra instead of cubes for sampling and Carr et. al.'s method [20] which uses octahedra.

Due to the popularity of Marching Cubes [17,18] , many remeshing methods have been proposed to improve the surface quality of its output and when a volume based method is used, it is usually necessary to apply a remeshing method or a combination of remeshing methods to get a surface with acceptable quality. A review of remeshing methods can be found in [21]. After remeshing, an extra smoothing step(e.g. [22-26]) may also be applied.

Another volume-based approach is grid adaptation which warpes the grid for better performance. This approach was first introduced by Moore and Warren [27]. They warped the grid to reduce the number of triangles created by the Marching Cubes algortihm. Azernikov and Fischer [28] improved this idea by warping the grid according to the local differential properties of the implicit function, thus making the warping adaptive to surface shape.

Surface tracking based methods aim to directly generate a high-quality surface from the implicit function thereby eliminating the need to use any post-processing methods. The polygonization is initialized with some seed points and progressively grown by adding new close-to-equilateral traingles until the whole surface is polygonized. The resulting surface therefore, usually has very high quality triangles. Early surface tracking based methods [29,30] are not adaptive and have the same fixed triangle size problem like volume based methods but adaptive versions were later developed(e.g. [31,32,33]). These methods determined triangle sizes based on curvature and other local properties but this brings a problem too. If the curvature on the surface changes too rapidly, the size of triangles will also change rapidly and this can cause low quality triangles. Also, some details in the high-curvature area may be missed because of the large triangle size determined in a close but low-curvature area. Schreiner et. al. construct a guidence field taking into account all the curvatures at points sampled from the surface to deal with this problem [34]. This method constructs very high quality surfaces but is slow.

Compared to the amount of research done seperately on vessel segmentation and surface reconstruction, there are only a few previous studies that use vessel segmentation and surface reconstruction together in a pipeline from the original angiography image to a surface mesh. These studies mostly focus on a specific anatomic region. Cebral, Hernandez and Frangi propose a full pipeline from CTA and 3D rotational angiography data to simulating blood flow for cerebral aneurysms [35]. For vessel segmentation, they use a fast-marching method [36] to get a rough initial segmentation and then refine it using a combination of [9] and [14]. For generating a surface mesh, they use the marching tetrahedra method developed by Cebral and Löhner [37]. Antiga, Ene-Iordache and Remuzzi generate a volume mesh for arteries from CTA and MRA [38]. They use a similar level set based vessel segmentation method proposed in [39] and surface reconstruction is achieved by

Marching Cubes and several mesh improvement methods. Zhang et. al. constructs hexahedral volume meshes from CT and MRI data [40]. They first use a preprocessing step of contrast enhancement, filtering and classification. They then use a fast-marching method [41] for segmentation and the method proposed by Ju et. al. for surface reconstruction [42].

In this study, we propose some improvements on Yu et. al.’s method[15] and use it for the vessel segmentation step. The term derived from Frangi’s vesselness measure is adapted for 3D images, the gradient term is removed as it does not help in small vessels and can cause problems in noisy images and finally, the mean curvature is used as an expansion force rather than a regularization force. Using the curvature in this manner greatly helps the segmentation of high curvature vessels but can cause irregular surfaces and oversegmentation if the weighting parameters are selected incorrectly. The result of the proposed vessel segmentation method is fed to Schreiner et. al.’s advancing front method [34] which is known to generate high-quality surfaces. Although this method is slow, surface quality is essential for our purpose of using the resulting surface in CFD simulations. Finally, Taubin smoothing [26] is used to smooth the surface. We expect our method to prove useful in the pipeline from MRA data to CFD simulations and aid in diagnostic and therapeutic decision making process related to vascular diseases.

2. ANGIOGRAPHIC IMAGING MODALITIES

Before describing our vessel segmentation and surface reconstruction method, we would first like to give brief information on widely used angiography techniques and discuss their advantages and disadvantages.

2.1 Digital Subtraction Angiography(DSA)

Planar X-ray is an imaging technique based on the fact that different tissues have different X-ray absorption rates. X-rays are passed through the patient and a detector detects these X-rays. The areas behind tissues with high X-ray absorption appear brighter in the final image. DSA technique works based on the same principle. First, a normal planar X-ray image is taken. Then, a contrast agent with high X-ray absorption is injected into the bloodstream using a catheter. A second image is taken after this injection and subtracted from the first image. Blood vessels are thus seen as black in this final subtracted image [43]. A DSA of the brain is shown in Figure 2.1.

DSA images have the advantage of very high accuracy, detail and clarity. Also, using the catheter allows for targeting specific vessels. The main disadvantage of DSA is its invasiveness. The catheter has to be navigated inside the patient which is uncomfortable and risky. The required time and cost are also high. X-ray radiation poses some risk but this risk is very small compared to more X-ray intensive modalities like CTA.

Classical DSA is 2D but rotational 3D DSA has also become available with the C-arm, which is a device that rotates around the patient to acquire 2D images from different angles and combines them to get a 3D image. The radiation dose is of course increased for 3D DSA but it includes more information than 2D DSA [44].

DSA is considered the gold standart most reliable angiography method but its invasiveness make it undesirable and non-invasive methods like CTA and MRA are preferred when very high accuracy and detail are not a necessity.

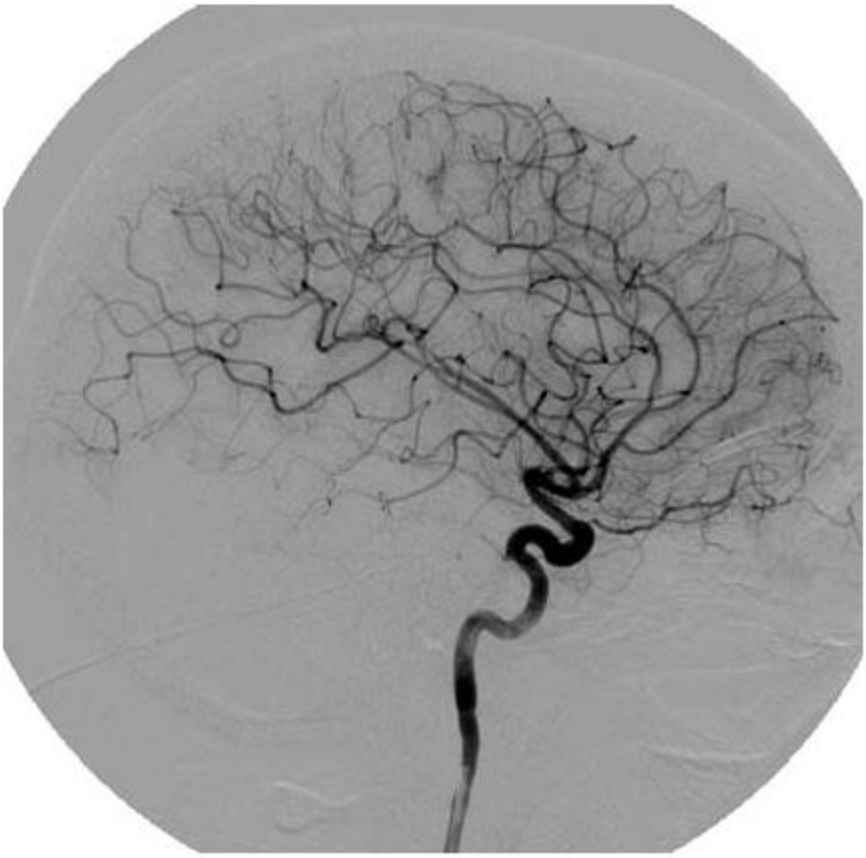


Figure 2.1 : A DSA image of blood vessels in brain. *Figure taken from [43]*.

2.2 Computed Tomography Angiography(CTA)

CTA uses the same physical principles used in DSA. The difference is that CTA produces a full 3D volume by combining many images taken from different angles and positions. Modern CTA devices have multiple X-ray sources and detectors and can acquire many slices at once making CTA a very fast technique. Another difference from DSA is that the contrast material is injected into a small vein in hand or arm with an intravenous(IV) line thus making it a much less invasive technique than DSA. Finally, no subtraction is performed in CTA and blood vessels appear bright because contrast material has high X-ray absorption [43]. As bone also appears bright in CTA, this can make vessel segmentation very difficult. Recently, bone removal by using dual energy CTA was introduced but it is not widely used yet [45]. CTA data is usually visualized by volume rendering [46] or maximum intensity projection(MIP) which is a method that involves projecting parallel rays through the volume and choosing the highest encountered intensity along the path to get a 2D projection. An MIP and volume rendering of supraaortic vessels are shown in Figure 2.2.

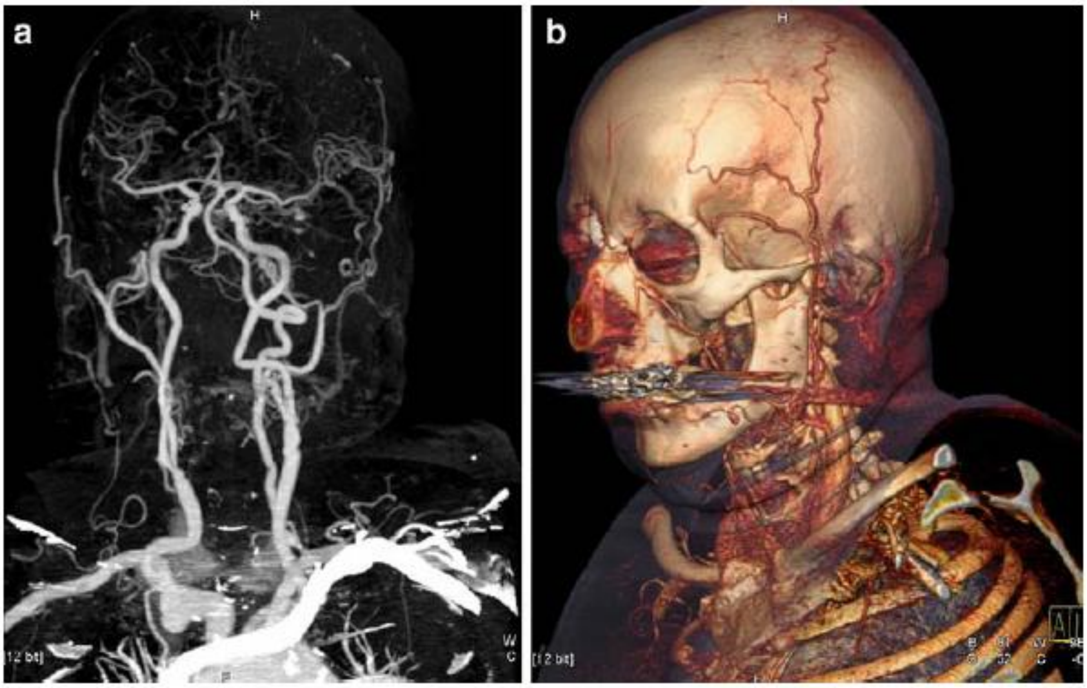


Figure 2.2 : a) MIP with bones removed. b) Volume rendering with bones not removed. *Figure taken from [45]*.

The main advantages of CTA are its quasi non-invasiveness, speed, comparatively low cost and wide availability. Its disadvantages are low accuracy compared to DSA and high radiation dose.

2.3 Magnetic Resonance Angiography(MRA)

The physical principles of MRA are entirely different from DSA and CTA. Unlike the other two modalities that depend on X-rays, MRA uses electromagnetism and radio waves. MR equipment creates an extremely strong external magnetic field and protons in the patient's body start precessing around this magnetic field. The precession is then disturbed by radio waves and it takes a while for the protons to return to equilibrium. This return to equilibrium is called relaxation. Different tissues have different relaxation times and MR technology uses this fact to differentiate tissue in the final image which is full 3D. Because of the nature of MR physics, very different images can be obtained by changing acquisition schemes. For example, blood can appear dark in one MR image while appearing white on another. The MR technician is responsiple for acquiring an image that will be useful for the case being investigated. Details of physical principles and how the acquisition works can be found in [47]. An MIP of an MRA image showing a cerebral aneurysm is shown in Figure 2.3.

MRA is non-invasive and pretty safe. As the technology is not based on X-rays, there is no radiation. Also, angiographic images can be obtained even without using contrast material. This lowers quality of course but may be necessary for patients that are allergic to contrast materials. On the other hand, accuracy and details are low compared to DSA, it cannot be used on patients that have metallic objects in their bodies, it is more costly and less available than CTA, it is slower than CTA which means there are more motion artifacts and it has difficulty in correctly depicting very small vessels [43].

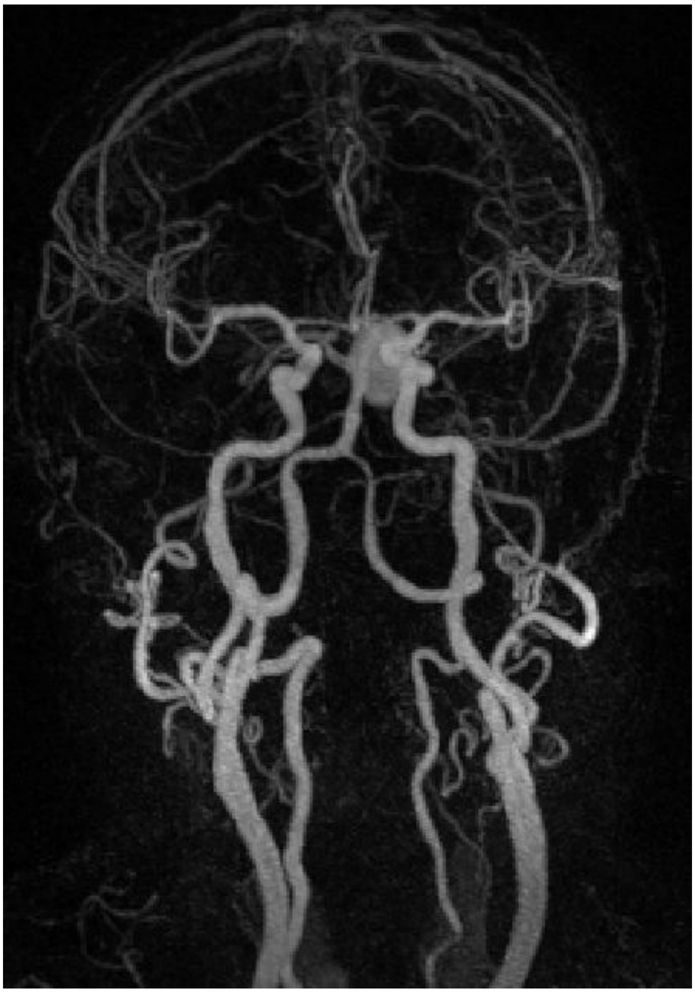


Figure 2.3 : An MIP of an MRA image showing a cerebral aneurysm.

In this study, we chose to work on MRA data because we believe that it will be more dominant in the future as the technology advances. The main reasoning behind our belief is that the physical principles of MRA are the safest. Also, vessels are the brightest structures in MRA images and this makes segmentation easier compared to CTA where bones are also bright. Note that recently introduced dual energy CTA images with bone removal look like MRA images . Therefore, although we have only tested our algorithm with MRA, it can be expected to work with dual energy CTA too.

3. VESSEL SEGMENTATION

3.1 Objective

The objective of the vessel segmentation step is to take a 3D MRA image as input and give the vessel surface represented as the zero level set of a 3D volume as output. Voxels with negative values will be inside of vessels and voxels with positive values will be outside of vessels.

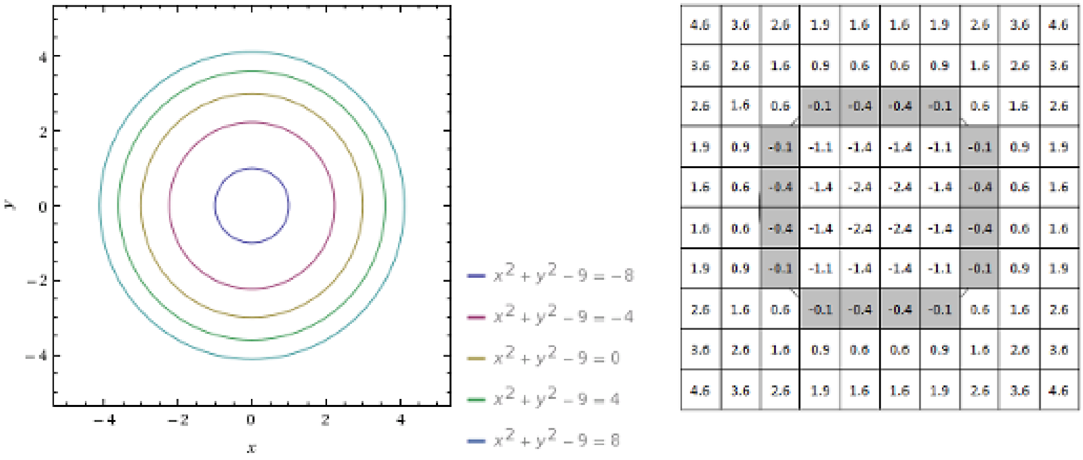
3.2 Surface Evolution With the Level Set Method

3.2.1 Representing the surface as level set of a volume

Let $U \subset \Re^3$ be the domain of a volume and let ϕ be a scalar function from U to $\Re$. Then, a surface can be represented as

$$S = \{\boldsymbol{x} \mid \phi(\boldsymbol{x}) = k\}. \tag{3.1}$$

Here, k can be chosen arbitrarily and choosing different k gives different surfaces. When $k = 0$, the surface S is called the zero level-set of ϕ. Sometimes ϕ is called the embedding and S is called an isosurface. It is important to mention that the surface will be closed as long as it does not intersect with the boundary of the domain U [48]. There is no consensus in literature whether negative values of ϕ are considered inside or outside the surface. In this text, we will define negative values to be inside the surface to be consistent with the convention used in the Insight Toolkit(ITK) [49] which is the software framework we have used for implementation. So namely, when $\phi(\boldsymbol{x}) < 0,$ the point $\boldsymbol{x}$ is inside the surface, when $\phi(\boldsymbol{x}) = 0,$ $\boldsymbol{x}$ is on the surface and when $\phi(\boldsymbol{x}) > 0,$ $\boldsymbol{x}$ is outside the surface. For implementation, the volume will be a grid sampling of $\phi.$ An illustration of how this representation defines a family of surfaces and how it would seem in the discrete case is given Fig 3.1. The volume is chosen as 2D for easier visualization.



4.6	3.6	2.6	1.9	1.6	1.6	1.9	2.6	3.6	4.6
3.6	2.6	1.6	0.9	0.6	0.6	0.9	1.6	2.6	3.6
2.6	1.6	0.6	-0.1	-0.4	-0.4	-0.1	0.6	1.6	2.6
1.9	0.9	-0.1	-1.1	-1.4	-1.4	-1.1	-0.1	0.9	1.9
1.6	0.6	-0.4	-1.4	-2.4	-2.4	-1.4	-0.4	0.6	1.6
1.6	0.6	-0.4	-1.4	-2.4	-2.4	-1.4	-0.4	0.6	1.6
1.9	0.9	-0.1	-1.1	-1.4	-1.4	-1.1	-0.1	0.9	1.9
2.6	1.6	0.6	-0.1	-0.4	-0.4	-0.1	0.6	1.6	2.6
3.6	2.6	1.6	0.9	0.6	0.6	0.9	1.6	2.6	3.6
4.6	3.6	2.6	1.9	1.6	1.6	1.9	2.6	3.6	4.6

Figure 3.1 : Illustration of implicit representation given by the function $\phi(\boldsymbol{x}) = x^2 + y^2 - 9$ and eq. (3.1). Left: Surfaces for different k in the continuous case. Right: Zero level set($k = 0$) in the discrete case.

Two variables we will need for the evolution are the surface normal and mean curvature so let us discuss how they are defined in this representation. The outward surface normal is simply

$$\boldsymbol{n}(\boldsymbol{x}) = \frac{\nabla\phi(\boldsymbol{x})}{|\nabla\phi(\boldsymbol{x})|} \tag{3.2}$$

where ∇ denotes the gradient operator and the mean curvature is the divergence of the surface normal:

$$\kappa = \nabla . \frac{\nabla\phi(\boldsymbol{x})}{|\nabla\phi(\boldsymbol{x})|}. \tag{3.3}$$

In the discrete case, these variables have to be numerically approximated. The mean curvature can be expressed in terms of partial derivatives of ϕ as

$$\kappa = \frac{\left\{ \begin{array}{r} \left(\phi_{yy} + \phi_{zz}\right)\phi_x^2 + \left(\phi_{xx} + \phi_{zz}\right)\phi_y^2 + \left(\phi_{xx} + \phi_{yy}\right)\phi_z^2 \\ - 2\phi_x\phi_y\phi_{xy} - 2\phi_x\phi_z\phi_{xz} - 2\phi_y\phi_z\phi_{yz} \end{array} \right\}}{\left(\phi_x^2 + \phi_y^2 + \phi_z^2\right)^{3/2}} \tag{3.4}$$

and the partial derivatives can be approximated with central differences [8]. The surface normal however cannot be approximated with a simple difference scheme because the direction of the surface normal abruptly jumps at corners of ϕ and if the chosen evolution function is sensitive to the direction of the normal, this can significantly affect the evolution. The surface normal should thus be approximated by calculating the one-sided difference approximations in all possible directions and averaging them [50]. More openly, let ϕ_{ijk} denote $\phi(x_{ijk})$ and let us define:

$$D_x^-\phi_i = \frac{\phi_i - \phi_{i-1}}{\Delta x},\ D_x^+\phi_i = \frac{\phi_{i+1} - \phi_i}{\Delta x}. \tag{3.5}$$

Then, the approximation to the surface normal is written as:

$$n_{ijk} = -\frac{1}{8}\left\{\begin{array}{l}
\frac{\left(D_x^+\phi_{ijk}, D_y^+\phi_{ijk}, D_z^+\phi_{ijk}\right)}{\sqrt{\left(D_x^+\phi_{ijk}\right)^2 + \left(D_y^+\phi_{ijk}\right)^2 + \left(D_z^+\phi_{ijk}\right)^2}} \\
+\frac{\left(D_x^+\phi_{ijk}, D_y^+\phi_{ijk}, D_z^-\phi_{ijk}\right)}{\sqrt{\left(D_x^+\phi_{ijk}\right)^2 + \left(D_y^+\phi_{ijk}\right)^2 + \left(D_z^-\phi_{ijk}\right)^2}} \\
+\frac{\left(D_x^+\phi_{ijk}, D_y^-\phi_{ijk}, D_z^+\phi_{ijk}\right)}{\sqrt{\left(D_x^+\phi_{ijk}\right)^2 + \left(D_y^-\phi_{ijk}\right)^2 + \left(D_z^+\phi_{ijk}\right)^2}} \\
+\frac{\left(D_x^+\phi_{ijk}, D_y^-\phi_{ijk}, D_z^-\phi_{ijk}\right)}{\sqrt{\left(D_x^+\phi_{ijk}\right)^2 + \left(D_y^-\phi_{ijk}\right)^2 + \left(D_z^-\phi_{ijk}\right)^2}} \\
+\frac{\left(D_x^-\phi_{ijk}, D_y^+\phi_{ijk}, D_z^+\phi_{ijk}\right)}{\sqrt{\left(D_x^-\phi_{ijk}\right)^2 + \left(D_y^+\phi_{ijk}\right)^2 + \left(D_z^+\phi_{ijk}\right)^2}} \\
+\frac{\left(D_x^-\phi_{ijk}, D_y^+\phi_{ijk}, D_z^-\phi_{ijk}\right)}{\sqrt{\left(D_x^-\phi_{ijk}\right)^2 + \left(D_y^+\phi_{ijk}\right)^2 + \left(D_z^-\phi_{ijk}\right)^2}} \\
+\frac{\left(D_x^-\phi_{ijk}, D_y^-\phi_{ijk}, D_z^+\phi_{ijk}\right)}{\sqrt{\left(D_x^-\phi_{ijk}\right)^2 + \left(D_y^-\phi_{ijk}\right)^2 + \left(D_z^+\phi_{ijk}\right)^2}} \\
+\frac{\left(D_x^-\phi_{ijk}, D_y^-\phi_{ijk}, D_z^-\phi_{ijk}\right)}{\sqrt{\left(D_x^-\phi_{ijk}\right)^2 + \left(D_y^-\phi_{ijk}\right)^2 + \left(D_z^-\phi_{ijk}\right)^2}}
\end{array}\right\}. \tag{3.6}$$

If any of the denominators are zero, that term is ignored and the rest are weighted accordingly.

3.2.2 Evolving the embedding volume

As the surface will evolve, let us denote the embedding hypersurface $\phi(\boldsymbol{x},t)$. The surface is defined as the zero level set of $\phi(\boldsymbol{x},t)$ at all times so for any point on the surface that moves with the surface as it evolves, we have

$$\phi(\boldsymbol{x}(t),t)=0. \tag{3.7}$$

Differentiating with respect to time yields

$$\phi_t+\nabla\phi(\boldsymbol{x}(t),t)\cdot\boldsymbol{x}'(t)=0. \tag{3.8}$$

Now let us assume that the surface is moving along the direction of its outward surface normals at every point with speed *F*. Then;

$$\boldsymbol{x}'(t)\cdot\boldsymbol{n}=F. \tag{3.9}$$

Remembering eq. (3.2) and comparing eqs. (3.8) and (3.9), it is seen that we have arrived at the following PDE:

$$\phi_t+F\left|\nabla\phi\right|=0. \tag{3.10}$$

Given $\phi(\boldsymbol{x},t=0)$, the equation can be solved as an initial value problem. This is the formulation given in [8].

The remaining problem is to implement all this in a computer. We first need a stable numerical scheme and then take into account the computational cost because with the level set formulation, a 2D problem has been increased to a 3D problem.

Let the approximation to $\phi_t(x_{ijk},t=n)$ be denoted by $\Delta\phi_{ijk}^n$. Then we will have the following update scheme:

$$\phi_{ijk}^{n+1}=\phi_{ijk}^n+\Delta t\Delta\phi_{ijk}^n. \tag{3.11}$$

If *F* has the form

$$F=g(\boldsymbol{x})\left|\nabla\phi\right|+\kappa\left|\nabla\phi\right|, \tag{3.12}$$

$F = \kappa|\nabla\phi|$ can be approximated with central differences but approximating $g(\boldsymbol{x})|\nabla\phi|$ with simple finite differences is unstable as it tends to overshoot and the up-wind scheme proposed in [7] is used [8,16,50,51]. The up-wind scheme is as follows:

$$g(\boldsymbol{x})|\nabla\phi| = \left\{ \begin{array}{l} \left[\max\left(g(\boldsymbol{x}),0\right)D_x^{+}\phi + \min\left(g(\boldsymbol{x}),0\right)D_x^{-}\phi\right]^2 \\ +\left[\max\left(g(\boldsymbol{x}),0\right)D_y^{+}\phi + \min\left(g(\boldsymbol{x}),0\right)D_y^{-}\phi\right]^2 \\ +\left[\max\left(g(\boldsymbol{x}),0\right)D_z^{+}\phi + \min\left(g(\boldsymbol{x}),0\right)D_z^{-}\phi\right]^2 \end{array} \right\}^{\frac{1}{2}} \qquad \textbf{(3.13)}$$

and the time step is chosen so that the largest change in one time step cannot be larger than 1. This means that the fastest moving front on the surface cannot move more than 1 grid point. Formally;

$$\Delta t \leq \frac{1}{\sup_{i,j,k\in U}\left\{\left|\nabla g(i,j,k,t)\right|\right\}}. \qquad \textbf{(3.14)}$$

The only remaining problem now is computational cost. It obviously does not make sense to update the whole volume when we are interested in only one surface, i.e. the zero level set of ϕ. Two algorithms have been proposed in literature to compute the updates efficiently when we are interested in only one surface: the narrow band method [52] and the sparse field method [51].

The narrow band method constrains the updates to a band of width *m* around the surface and the embedding is calculated as the signed distance transform. The downside of the method is that as the surface evolves, it will come near the boundary of the narrow band and the band has to be reinitialized. Thus, the choice of *m* is a trade-off between the number of updates each iteration and the number of reinitializations. If *m* is large, a low number of reinitializations will be sufficient but updates for ϕ will have to be calculated for more points.

Our method of choice, the sparse field method, is the extreme case of the narrow band method where the update of ϕ is only calculated on points that are adjacent to the surface, i.e. the points that have a value between -0.5 and 0.5. The author calls this set the *active set*. Of course, to calculate the update which requires finite difference approximations, the nearest neighbors of all the points in the *active set* are necessary. Thus, 2 additional inside and 2 additional outside layers are used. The values in the additional layers are computed from the values of the *active set*. A linked list data structure is used to keep track of and update the layers as the surface evolves.

3.3 Choosing the Function That Will Guide the Evolution

Until this point, nothing that is specific to vessel segmentation has been developed. What will make the method perform vessel segmentation is the choice of the function *F*. *F* must be chosen such that the surface evolves and converges to the vessel boundaries in the image. Remembering that ϕ takes negative values inside the surface and from eq. (3.10), it is evident that the surface expands at points where *F* is positive. Therefore, *F* should ideally be positive inside vessels, negative outside and zero on the boundary. For defining our *F*, we will make use of the Frangi vesselness measure proposed in [14]. The measure is based on the idea of making use of the fact that vessels are mostly tubular and brighter(or darker in some modalities) than surrounding tissue in angiographic images. To measure tubularity, analysis of the Hessian is used. As it is well known, computing derivatives on an image always requires smoothing because otherwise, noise would be amplified by the operation. Computing the Hessian thus has an intiuitive justification as the second order derivative at scale σ is given by convolving the image with a Gaussian second derivative kernel computed at scale σ. This will measure the intensity difference between the region $\left(\boldsymbol{x}-\sigma, \boldsymbol{x}+\sigma\right)$ and the outside along the direction of the derivative. The continuous Gaussian second derivative function for two different scales is illustrated in Figure 3.2. As can be noticed, the Gaussian second derivative function has significantly smaller values in the larger scale. Thus, a scale normalization is used when computing the Hessian. All second derivatives are multiplied by the variance of the current scale, namely σ^2.

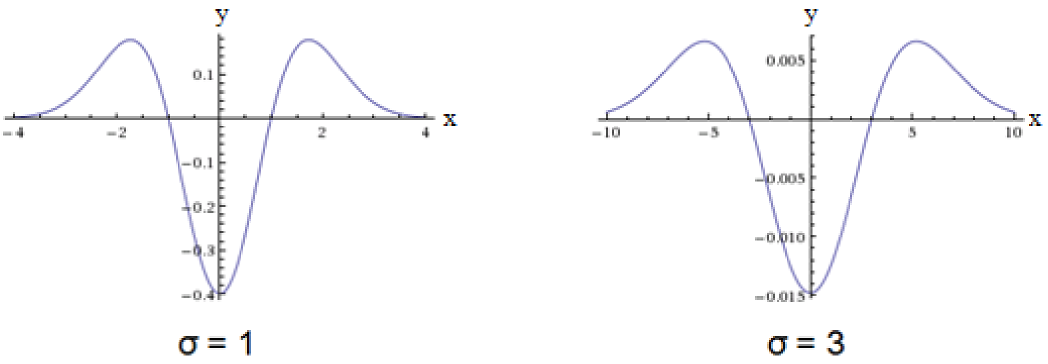
Let $\lambda_1, \lambda_2, \lambda_3$ be the eigenvalues of the Hessian and $\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3$ be the corresponding eigenvectors. Also let $|\lambda_1| \le |\lambda_2| \le |\lambda_3|$. Then, if $\boldsymbol{x}$ is inside a vessel and σ matches the radius of the vessel, $\boldsymbol{u}_1$ will be the direction along the vessel, $\boldsymbol{u}_2$ and $\boldsymbol{u}_3$ will span the cross-sectional plane of the vessel and the corresponding eigenvalues will measure the convolution of the second derivative Gaussian with the image along the directions of the eigenvectors [14]. Therefore, in modalities where the vessels are brighter than surrounding tissue, λ_1 should have small magnitude while λ_2 and λ_3 should both be negative and have large magnitude. Using this analysis, the vesselness measure in 3D for modalities where the vessels are brighter than surrounding tissue is defined as follows [14]:

$$v(\boldsymbol{x},\sigma)=\begin{cases}0, & \text{if } \lambda_2>0 \text{ and } \lambda_3>0\\ \left\{\left(1-\mathrm{e}^{-\frac{R_A^2}{2\alpha^2}}\right)\mathrm{e}^{-\frac{R_B^2}{2\beta^2}}\left(1-\mathrm{e}^{-\frac{S^2}{2c^2}}\right)\right\}, & \text{otherwise}\end{cases} \tag{3.15}$$

where

$$R_A=\frac{|\lambda_2|}{|\lambda_3|},\ R_B=\frac{|\lambda_1|}{\sqrt{|\lambda_2\lambda_3|}},\ S=\|\boldsymbol{H}_F\|=\sqrt{\sum_{j\le D}\lambda_j^2}.$$

Here, α,β,c are user-defined variables that control the weight of each term and D is the dimension of the image. Taking $\alpha=0.5,\beta=0.5$ and c equal to the half of the maximum Hessian norm in the image is reported to work well in most cases [14]. Of course, as we have mentioned, the vesselness measure only makes sense if the σ matches or is close to the radius of the vessel. Therefore, to detect vessels with different radii, the vesselness measure v should be computed for multiple scales and the maximum value should be selected as the effective value.



$\sigma = 1$ $\sigma = 3$

Figure 3.2 : The continuous Gaussian second derivative function for scales $\sigma=1$ and $\sigma=3$.

Using the Frangi vesselness measure in a level set vessel segmentation framework was already proposed in [15] but some of the terms included were redundant and the method had several shortcomings [53]. The evolution function used in [15] was:

$$F = \gamma p(I) + (1-\gamma) g(\nabla I)(c_1 + c_2\kappa) - \nabla g(\nabla I).\boldsymbol{n} \tag{3.16}$$

where

$$p(I(\boldsymbol{x})) = \begin{cases} 1, & \text{if } v(\boldsymbol{x}) \geq a \\ v(\boldsymbol{x}), & \text{if } v(\boldsymbol{x}) < a \text{ and } v(\boldsymbol{x}) \geq b \\ -(1-v(\boldsymbol{x})), & \text{otherwise.} \end{cases} \tag{3.17}$$

I is image intensity, κ is mean curvature, $\boldsymbol{n}$ is the outward surface normal and g is a function of ∇I that is usually chosen as

$$g(\nabla I) = \frac{1}{1+|\nabla I|^k} \tag{3.18}$$

with $k=1$ or $k=2$ [15]. Parameters γ, c_1, c_2, a, b are all user-defined and good results are reported for $\gamma = 0.5, c_1 = 1.0, c_2 = -0.1, a = 0.4, b = 0.2$ on 2D images [15].

Let us analyze the terms in eq. (3.16). First, $p(I(\boldsymbol{x}))$ expands or shrinks the surface based on Frangi's vesselness measure. A value of 1 gives the strongest expansion effect and a value of -1 gives the strongest shrinkage effect. $g(\nabla I)$ is close to 1 when the gradient magnitude is small(e.g. inside large vessels) and values close to 0 when the gradient magnitude is large(e.g. on vessel boundary). Thus, its purpose is to expand the surface inside vessels. The term $-\nabla g(\nabla I).\boldsymbol{n}$ usually gives positive values inside vessels and negative outside so its purpose is to help snap the surface boundary to the vessel boundary. Finally, mean curvature is used as a volume minimizing force for smoothing and regularization.

A first observation is that the $g(\nabla I)$ term is good for segmenting large vessels but it does not help the segmentation of tiny vessels because the gradient magnitude will be large inside tiny vessels. Furthermore, a smoothing scale has to be chosen for computing gradients which is not desirable in a multi-scale method and it also means more user-defined parameters to optimize. Another problem with the $g(\nabla I)$ term is that if the surface leaks beyond the vessel into the background, it will help the leakage to spread everywhere because gradient magnitude will be small in the background.

A second observation is that as the Frangi vesselness measure ignores any vesselness values except the highest, this loss of information could cause oversegmentation because two tiny vessels that are close and roughly aligned to each other appear as one bigger vessel at larger scales and is segmented as such. The negative values between the vessels that occur at smaller scales are ignored. For the same reason, if the brightness of vessels vary significantly in the image, brighter vessels appear larger than they actually are as they make a larger area around themselves bright at larger scales.

In [53], we have removed the terms that include $g(\nabla I)$, as we consider them to be redundant, to make the method simpler and faster and proposed a modification to the Frangi vesselness measure to deal with the identified shortcomings. The main drawback was that the selection of user defined variables was difficult and some images required manual initialization. We now propose the following very similar scheme that includes slight modifications to overcome those drawbacks.

$$F = (\gamma p(I) + (1-\gamma)\left|d\kappa\right|), \tag{3.19}$$

which makes the evolution function

$$\phi_t = -(\gamma p(I) + (1-\gamma)\left|d\kappa\right|)\left|\nabla\phi\right|. \tag{3.20}$$

Here,

$$p(I(\boldsymbol{x})) = \begin{cases} 1, & \text{if } v(\boldsymbol{x}) \geq a \\ \dfrac{v(\boldsymbol{x}) - b}{a-b}, & \text{if } v(\boldsymbol{x}) < a \text{ and } v(\boldsymbol{x}) \geq b \\ \dfrac{v(\boldsymbol{x})}{b} - 1, & \text{if } v(\boldsymbol{x}) < b \text{ and } v(\boldsymbol{x}) \geq 0 \\ -1, & \text{otherwise,} \end{cases} \tag{3.21}$$

$$v(\boldsymbol{x},\sigma)=\begin{cases}-\left(1-\mathrm{e}^{-\frac{\lambda_3^2}{0.5c^2}}\right), & \text{if } \lambda_3>0 \text{ and } \sigma\le k\\ 0, & \text{if } \lambda_3>0 \text{ and } \sigma>k\\ 0, & \text{if } \lambda_3\le 0 \text{ and } \lambda_2>0\\ \left\{\left(1-\mathrm{e}^{-\frac{R_A^2}{2\alpha^2}}\right)\mathrm{e}^{-\frac{R_B^2}{2\beta^2}}\left(1-\mathrm{e}^{-\frac{S^2}{2c^2}}\right)\right\}, & \text{otherwise,}\end{cases} \qquad \textbf{(3.22)}$$

κ is the mean curvature and a,b,d,γ,k are user-defined variables. As can be seen from (3.21), we are trying to scale $p(I)$ to be between -1 and 1. This choice is arbitrary and does not matter as the fastest moving front is constrained to move at most 1 voxel because of stability issues. The vesselness response is simply scaled so that anything above *a* gives 1, values between *a* and *b* are scaled to range [1,0], [*b*,0] is scaled to [0,-1] and anything with negative vesselness is given -1. An important property is the modification to the vesselness function. A term is added that gives negative values if λ_3 is positive and of course the effective scale will now be selected according to the maximum absolute value. Doing this prevents the problems we have identified because λ_3 will be highly positive just outside the vessel border and give highly negative vesselness values preventing oversegmentation. This negative term is constrained to be active only in scales smaller than k because this fixes the identified problems and it could prevent segmentation of thin vessels that are near large vessels if it were also active in larger scales. We choose k to be twice the minimum standart deviation of the selected scale range. The improvement offered by this modification is illustrated in Figure 3.3. It is seen that using the original Frangi vesselness measure can result in oversegmentation as it can ignore information on small scales.

It can be noticed that *F* is different from the one proposed in [53] as the absolute value of the mean curvature is now taken. This seems like a strange choice at first because the mean curvature is usually used as a smoothing, volume minimizing force in literature [7,8,9,10,13,14,15]. We have however shown in [53] that using the curvature oppositely can facilitate segmentation of tiny vessels and give better results. This idea works because at the tip of the surface that is evolving along the vessel, both principal curvatures are expected to be positive with large magnitude.

Using the curvature in this manner is made possible by our modification to the Frangi vesselness measure. In our scheme, the vesselness measure robustly gives negative values just outside of vessels and this prevents the expanding curvature force from making the surface expand everywhere. Our way of using the curvature as an expanding force by reversing the sign of *d* was however not ideal. With a reversed sign, the mean curvature can still act as a shrinking force at points where the vessel curves toward the surface normal. This serves no purpose and can be disadvantageous. Therefore, it is better to take the absolute value and make sure that the curvature tries to expand the surface at every point. It is important to stress that although using the curvature in this manner facilitates segmentation of thin and high curvature vessels, it also increases the risk of oversegmentation and irregularization of the surface, and may cause the solution to be unstable. Thus, the user-defined variable *d* that weigh the curvature should be selected carefully. Taking $d = 0.2$ gives good results in all our datasets which is consistent with the results from [53].

Another difference from [53] is that *c* is now chosen as one fifth of the highest intensity in the image instead of the half of the maximum Hessian norm at the current scale. The reasoning behind this change is to prevent oversegmentation caused by selecting the maximum scale too high. Even if the maximum scale is slightly higher than the radius of the thickest vessel in the image, the maximum Hessian norm will be lower compared to other scales. This will make the maximum scale have unusually high vesselness values and can cause oversegmentation. Of course, this new selection method will result in optimal *a* and *b* values to be very different for different images. Thus, we now propose the following selection formula:

$$\begin{aligned} b &= \frac{\xi v_{\text{avg}}}{\left(\sigma_{\max} / \sigma_{\min}\right)}, \\ a &= 2b, \end{aligned} \tag{3.20}$$

where ξ is a user-defined parameter, v_{avg} is the average of positive vesselness values in the image and $\sigma_{\min}$ and $\sigma_{\max}$ are the minimum and maximum standart deviations that define the selected scale range. We have found in our experiments that this selection formula makes the choice of user-defined variables much easier. Obviously, increasing the value of ξ will segment less vessels but decrease the chance of oversegmentation. $\xi = 20$ gave satisfactory results for most of our images.

We do not claim that this is the optimal way to select these parameters. As images from different anatomical regions have different characteristics, finding an absolute optimal way is probably not possible.

The variable γ controls the weighting of vesselness and curvature terms. As the vesselness term should be dominant, γ should be selected larger than 0.5 and obviously smaller than 1. Choosing this variable between 0.6 and 0.8 seems to be logical and $\gamma = 0.7$ consistently gave good results in all our datasets.

Finally, initialization can be performed automatically by choosing the top 0.05% of the voxels with a positive vesselness according to their vesselness measure computed at scale $\sigma_{\text{avg}} = (\sigma_{\min} + \sigma_{\max}) / 2$ as seeds and the initial surface is spheres of radius 1 around these seed points. The reason for using σ_{avg} instead of the normal vesselness measure computed at multiple scales is that this is more robust. The multi-scale measure has more false positives especially at low scales caused by noise.

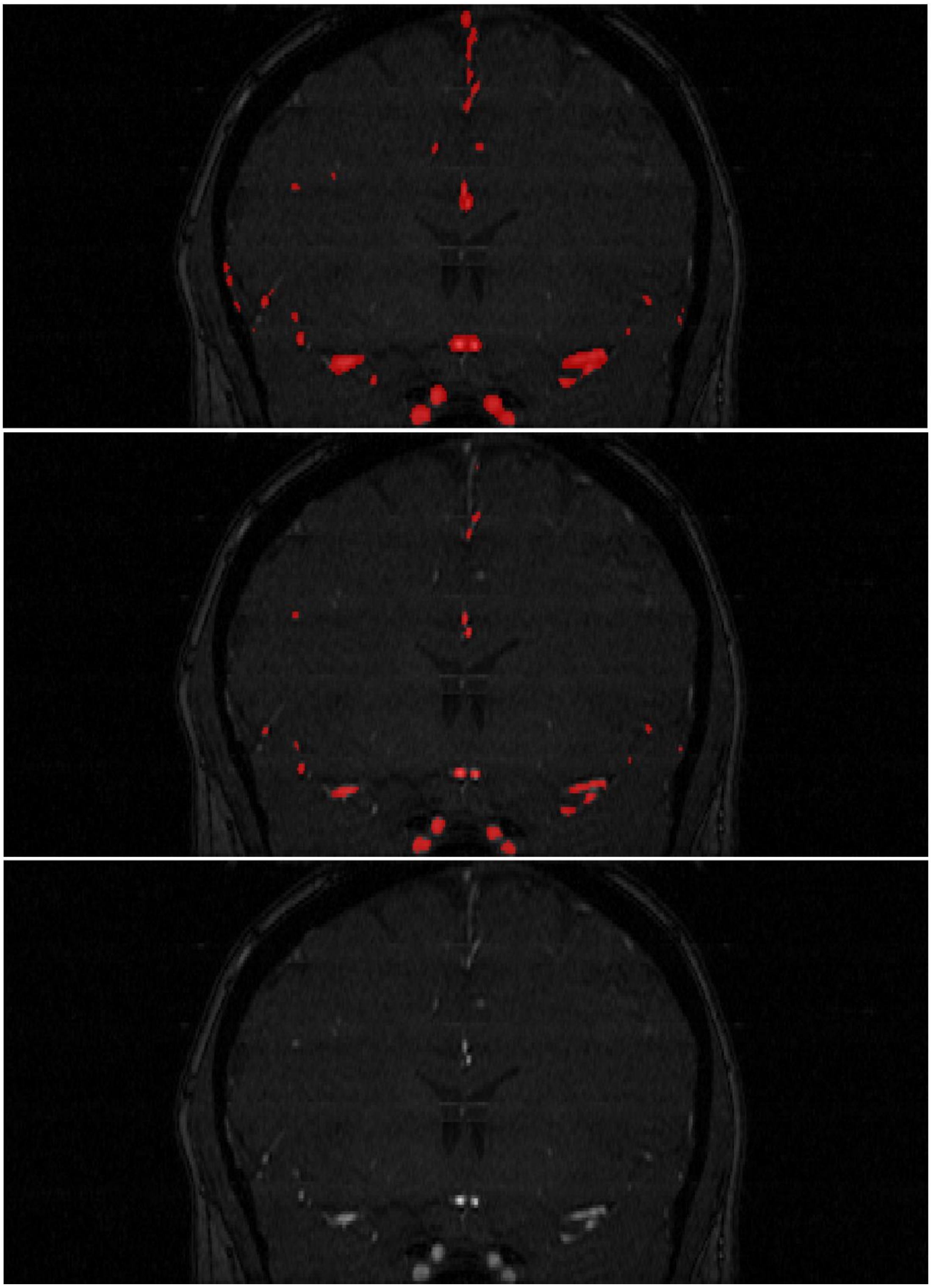


Figure 3.3 : The improvement resulting from our proposed modification to Frangi’s vesselness measure illustrated on a coronal 2D slice of a 3D brain MRA image. Top image: The segmentation result when original Frangi vesselness measure is used. Middle Image: The segmentation result when our modified vesselness measure is used. Bottom Image: The original slice.

4. SURFACE RECONSTRUCTION

4.1 Objectives

The aim of this chapter is to represent the vessel surface with polygons using the output of the vessel segmentation algorithm which is a 3D regular grid sampling of an implicit function. The resulting mesh should contain high quality triangles to be suitable for numerical simulation.

4.2 Implicit Surface Polygonization

As our aim is to run numerical simulations on the surface, triangle quality is essential and the speed of the surface reconstruction is not so important. Therefore, Schreiner et. al.'s advancing front method [34] seems like a good candidate for this task. Availability of source code is also advantageous as we do not need to worry about implementation. For the sake of completeness, let us summarize this method.

As all advancing front methods, the fundamental idea of the algorithm is to start from seed points and grow the mesh from those seeds until the whole surface is covered. Triangle size adaptation to surface shape is based on curvature. Similar schemes already exist (e.g. [35,36]) but as curvature is computed locally, on areas where the curvature changes abruptly, badly shaped triangles can be created or surface details can be missed. To overcome this problem, a guidance field that takes into account all curvatures is constructed before starting the polygonization. The difference between using only local curvature and using a guidance field is illustrated in Figure 4.1. Notice that the triangles gradually get smaller before they reach the high curvature area. This idea was already proposed by Scheidegger et. al for polygonizing point set surfaces [58] and used by Schreiner et. al. for remeshing [59]. In [34], Schreiner et. al. improve this guidance field framework by giving a way to calculate the necessary number of sample points at which to calculate the guidance field such that all the details of the surface are captured.

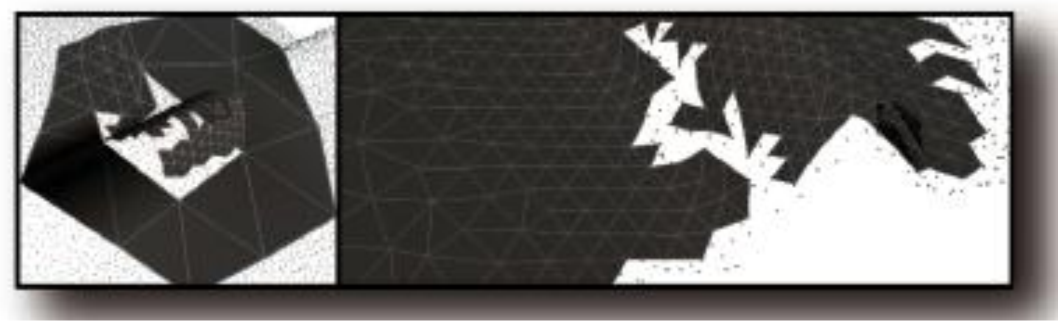


Figure 4.1 : Left: Surface constructed using only local curvature. Right: Surface constructed using a guidance field. *Figure taken from [34].*

Let us first describe the guidance field construction. Let *s* be a point on the surface. Then, the ideal edge length at that point is defined as:

$$l(\boldsymbol{s}) = \frac{2\sin(\rho/2)}{\kappa_{\max}(\boldsymbol{s})}, \tag{4.1}$$

where $\kappa_{\max}$ is the maximum curvature and $0 \le \rho \le 2\pi$. In other words, the ideal edge length always subtends the angle ρ of the osculating circle. This is illustrated in Figure 4.2. The user can control the accuracy of the method by specifying ρ.

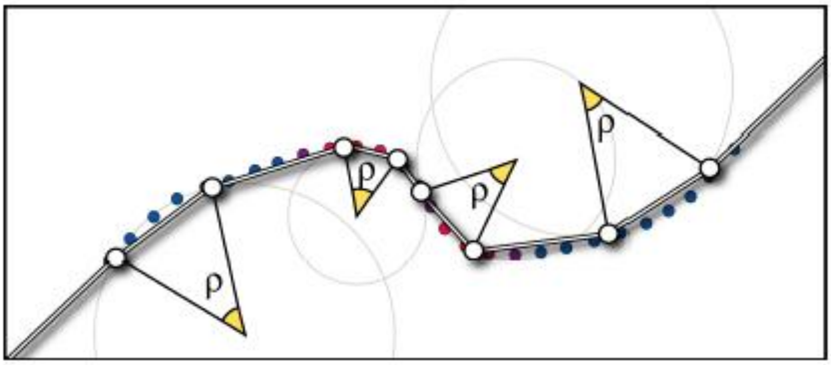


Figure 4.2 : Illustration of why the edge length ideally subtends the same angle of the osculating circle. *Figure taken from [58].*

As previously mentioned, the guidance field should be constructed such that triangle sizes start getting gradually smaller before reaching a high curvature area so that triangle size does not change too rapidly. To this end, a function $\tilde{g}_s(\boldsymbol{x})$ from the embedding ϕ to $\Re$ is defined for all ***s*** as follows:

$$\tilde{g}_s(\boldsymbol{x}) = \left(1-\eta^{-1}\right)\left|\boldsymbol{x}-\boldsymbol{s}\right| + \eta^{-1} l(\boldsymbol{s}). \tag{4.2}$$

Notice that the function gives smaller values as $\boldsymbol{x}$ gets closer to $\boldsymbol{s}$. The variable η is user-defined and controls how fast triangle size increases as $\boldsymbol{x}$ gets away from $\boldsymbol{s}$. The guidance field function is obtained by taking the minimum $\tilde{g}_s(\boldsymbol{x})$ for all $\boldsymbol{s}$. Formally,

$$g(\boldsymbol{x}) = \min_{s \in \phi} \tilde{g}_s(\boldsymbol{x}). \tag{4.3}$$

For the function to make sense, η should be chosen $\eta \geq 1$. When η is close to 1, triangle sizes will change very slowly, resulting in extremely high quality triangles but the number of triangles will also be very high everywhere resulting in slower computation times both for the construction of the surface and for any algorithms that will be run on the resulting surface. Notice that $\eta = 1$ removes adaptability to curvature and makes all edge lengths equal to the smallest ideal edge length. This is of course not practical and η should be chosen such that there is a good balance between adaptability and triangle quality. It is important to mention that the ideal edge length $l(\boldsymbol{s})$ is scaled in the guidance field by η^{-1} which is a value between 0 and 1. Although this introduces no theoretical problem, it may be confusing for the user when selecting ρ and η. Ideally, the accuracy should depend solely on ρ while triangle quality and adaptivity to curvature depend solely on η for the selection to be more intuitive. This however is not the case in this formulation. Here we propose a quick rough fix for this problem that can be used without changing the implementation. As ρ will typically be small, we can roughly assume that

$$\frac{\sin(\rho/2)}{(\rho/2)} \approx 1. \tag{4.4}$$

Then we can define $\rho_2 = \eta\rho$ and use ρ_2 instead of ρ in eq. (4.1) in which case the accuracy will roughly only depend on ρ while triangle quality and adaptivity to curvature depends on η.

The algorithm starts by first calculating the number of points on which to sample the surface for creating the guidance field. A sufficient sampling ensures that $\tilde{g}_s(\boldsymbol{s}) \leq l(\boldsymbol{s})$ for all $\boldsymbol{s}$. After the number of points required for sufficient sampling is determined, the points are created as a set of jittered random points in cells that neighbor the surface and projected on the surface with gradient descent. Some of these points are actually irrelevant and do not contribute anything to the guidance field so these are removed to reduce memory consumption and increase running speed. See [34] for details on how the necessary number of points is computed and how irrelevant points are determined.

After the guidance field is constructed, we are ready to start the triangulation process. First, seed points are found near the surface and projected on the surface with gradient descent. One seed point is enough for each connected component. The seed points can be calculated efficiently as described in [60].

The surface is then progressively grown from seed points until all the surface has been constructed. The method used is the same as in [59]. New triangles are formed by connecting an existing edge to a vertex. A new vertex is created by using a combination of the value of the guidance field at the vertices of the edge and ordering and normals of edge vertices to create a new point and projecting that point on the surface. If the vertex is a new point on the surface, the triangle formed is called a free triangle. If this free triangle winds up too close to an existing front, a vertex that is already part of the triangulation is used instead and the created triangle is called a connection triangle. If the edge and the new vertex are part of the same front, the front is split in two. If the new vertex is part of another existing front, the fronts are merged. Free triangles are placed before connection triangles and within each class of triangle, triangles which have larger ratios of incircle radius to circumcircle radius are placed first. Triangle quality is further improved by keeping a band of triangles behind the advancing front and allowing edge flip operations on this band. An edge flip operation is performed if the two new triangles have larger minimum angles than the two original triangles adjacent to the edge and if the normals of the new triangles are not in opposition to the original triangles. A triangle is moved out of the band and finalized when all three vertices of the triangle are no longer on a front.

The framework described so far is a general advancing front scheme and is not specific to how the implicit function is defined. Now, let us describe how to handle a 3D regular grid sampling of an implicit function. Schreiner et. al. define two ways for regular grids [34]: Catmull-Rom splines [61] and B-splines [62]. The splines are used to generate piecewise cubic trivariate polynomials. Then, curvature and gradients can be easily computed analytically. Schreiner et. al. use Catmull-Rom splines to interpolate and B-splines to approximate the input data [34]. Thus, Catmull-Rom splines represent the data much more accurately but this may not be desirable when the input is not perfect and contains noise. Noise causes high curvature in many areas and significantly more triangles are created when Catmull-Rom is used. Moreover, the smoothing nature of B-splines may be desirable depending on application. The downside of B-splines is that some details of the surface may be missed because of the approximation. For our vessel segmentation input, this means some very tiny vessels may be lost.

4.3 Mesh Smoothing

Although the advancing front method constructs a high quality surface, a further smoothing step may be desirable. Here, we describe Taubin smoothing [26] which is a simple but effective method that modifies the Gaussian smoothing idea [63] to not produce shrinkage.

Gaussian smoothing method works by changing the position of a vertex by convolving it and the vertices that share an edge with it with a discrete Gaussian kernel. As this operation is very local, many iterations are necessary to achieve a significant amount of smoothing. This however, also causes significant shrinkage which is unacceptable for a vessel segmentation method.

To explain the modification of Taubin [26], let us first describe the Gaussian smoothing method more formally. First let us denote our set of vertices $V = \{v_i\}$. Then, let M_i be the set of vertices that share an edge with v_i. An average difference between vertices is first calculated as follows:

$$\Delta v_i = \sum_{j \in M_i} w_{ij} \left(v_j - v_i \right). \tag{4.5}$$

The weigths w_{ij} can be taken from a discrete Gaussian kernel or any other weighting scheme can also be chosen as long as all w_{ij} are positive and their sum is 1. The vertex positions are then updated using

$$v_i' = v_i + \lambda \Delta v_i \tag{4.6}$$

with $0 < \lambda < 1$ at each iteration. To prevent the shrinkage caused by this algorithm, Taubin [26] propose defining a second update using a scale factor that has opposite sign to λ:

$$v_i' = v_i + \mu \Delta v_i \tag{4.7}$$

where $0 > \mu > -\lambda$. These two updates are rotated at each iteration preventing shrinkage. The magnitude of μ should be slightly larger than λ because Δv_i will get smaller at each iteration and eq (4.6) is the first update used.

5. VESSEL SEGMENTATION AND RECONSTRUCTION ALGORITHM AND RESULTS

5.1 Implementation Details

The original MRA data is in DICOM series or MetaImage(MHA) format. If the data is in DICOM series format, the series are read and converted to greyscale volume data as MHA files. Higher greyscale values ideally correspond to vessels. As a pre-processing step, a sigmoid filter[49] is used on the MHA file to emphasize the expected intensity range of vessels. A curvature flow smoothing filter[49] is also used in noisy images. The resulting MHA file is used by the vessel segmentation step to produce another scalar volume data which is the implicit representation of the surface.

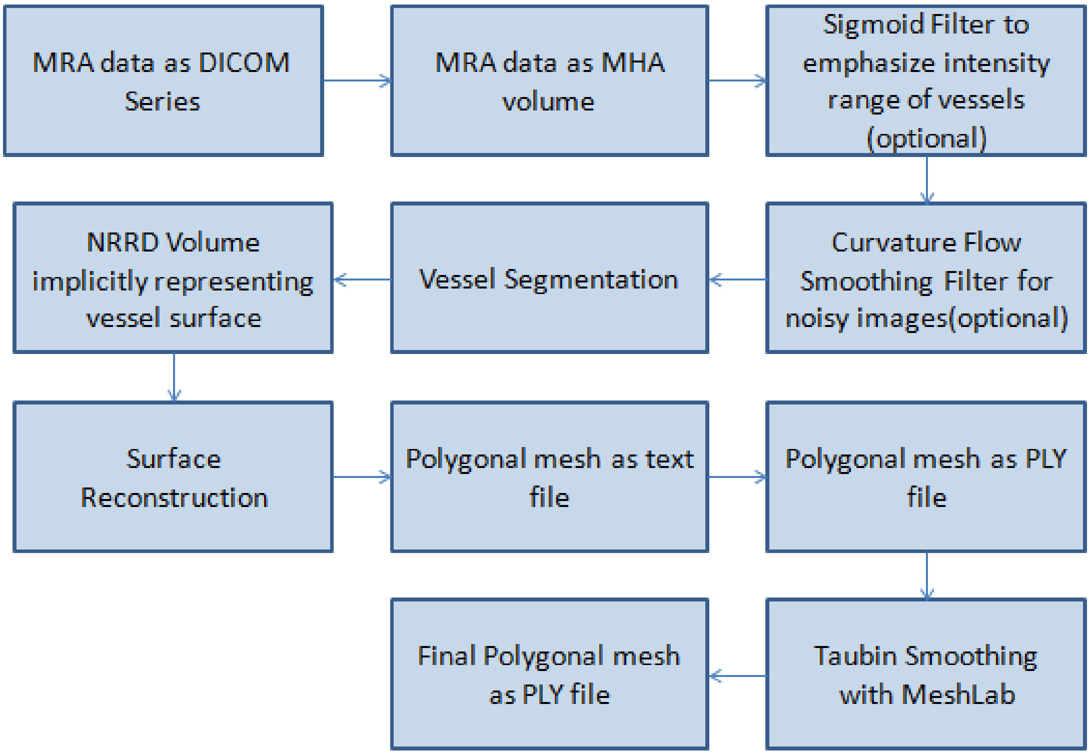

MRA data as DICOM Series
MRA data as MHA volume
Sigmoid Filter to emphasize intensity range of vessels (optional)
Curvature Flow Smoothing Filter for noisy images(optional)
Vessel Segmentation
NRRD Volume implicitly representing vessel surface
Surface Reconstruction
Polygonal mesh as text file
Polygonal mesh as PLY file
Taubin Smoothing with MeshLab
Final Polygonal mesh as PLY file

Figure 5.1 : Summary of the algorithm.

The sparse field algorithm is used for evolution as described by Whitaker [51] and as implemented in ITK [49]. For scale selection, the range of vessel radii that the user is interested in, is manually chosen by the user. The number of scales is chosen based on the scale range as:

$$N = \mathrm{ceil}\left(\log_{1.5}\left(\sigma_{\max} - \sigma_{\min}\right)\right) \tag{5.1}$$

and the scales in between are chosen logarithmically as suggested by Sato et. al. [54]:

$$\sigma_i = \exp\left(\ln\left(\sigma_{\min}\right) + i\frac{\ln\left(\sigma_{\max}\right) - \ln\left(\sigma_{\min}\right)}{N-1}\right),\ i = 0,1,\ldots,N-1. \tag{5.2}$$

For calculating the Hessians, the image is first Gaussian smoothed with different scales. The Gaussian kernel is used with maximum error equal to 0.01 and maximum kernel size equal to 30. Then, central differences are calculated using a 3x3x3 neighborhood. The step size in each direction is taken to be equal to the standart deviation of the current scale. When the reached point is out of the boundary, the value of the closest boundary is used. This is preferred instead of mirroring because mirroring can cause unusually high vesselness response and adversely affect the algorithm, especially the automatic initialization.

Convergence is checked every 100 iterations by computing the change in the level set function per segmented voxel over 100 iterations. Evolution is stopped when the current change is below 0.05 and the difference between previous change and current change is below 0.005. In all computations, linear interpolation is used when required.

The pseudocode for the vessel segmentation step is as follows:

```
VESSEL_SEGMENTATION(input_MRA_data, sigma_min, sigma_max, $\gamma$, $d$, $\xi$)
    Find maximum intensity in the image and set $c$ as one sixth of it.
    Compute the vesselness measure for the whole image.
    Calculate $a$ and $b$ as explained in eq 3.20.
    Set seeds as explained in Section 3.3.
    Use fast marching and seeds to initialize.
    Use the sparse field method with the evolution function proposed in Section 3.3 until convergence.
```

For the surface reconstruction step, the advancing front algorithm of Schreiner et. al. [34] kindly provided by the authors, is used without changes. The program takes Nearly Raster Raw Data(NRRD) files as input so the output of the segmentation should be NRRD. Also, spacing is not considered so spacing ratios are equalized in a pre-processing step. The output mesh is given as a text file and we convert it to PLY for further processing. PLY files are visualized with Paraview [64] and MeshLab [65] is used for Taubin smoothing. The Marching Cubes result that we use for comparison is computed with VTK [66].

5.2 Results and Discussion

The proposed method was tested on 34 real MRA images and a synthetic image with intensity range [0,255] created by the software VascuSynth [55]. 20 of the real images are healthy brain images which were collected and made available by the CASILab at The University of North Carolina at Chapel Hill and were distributed by the MIDAS Data Server at Kitware, Inc. [56]. The remaining 14 real images were acquired from Sonomed Medical Imaging Center and Laboratory in Istanbul and contain both healthy and unhealthy images from different anatomical regions. For comparison with existing methods, we have used the confidence connected region growing method and the geodesic active contour method implemented in ITK [49]. Some of the results are shown in Figure 5.2 and Figure 5.3 along with the corresponding maximum intensity projections. For the proposed algorithm, $\gamma = 0.7$, $d = 0.2$ and $\xi = 20$ were used for all images. For the other methods, we have tried several sets of user-defined parameters for each image and the best result is shown. The same seeds were given to all methods for initialization. The results are visualized with ITK-SNAP [57] which uses Marching Cubes to triangulate a binary segmentation result. As our output is a grid sampling of an implicit function, we use a zero-crossing filter to convert it to binary. It is true that some information is lost in this process but Marching Cubes is limited by the grid resolution anyway. Marching Cubes is sufficient for visually evaluating segmentation results because at the current state of the art, no segmentation algorithm is expected to achieve subpixel accuracy.

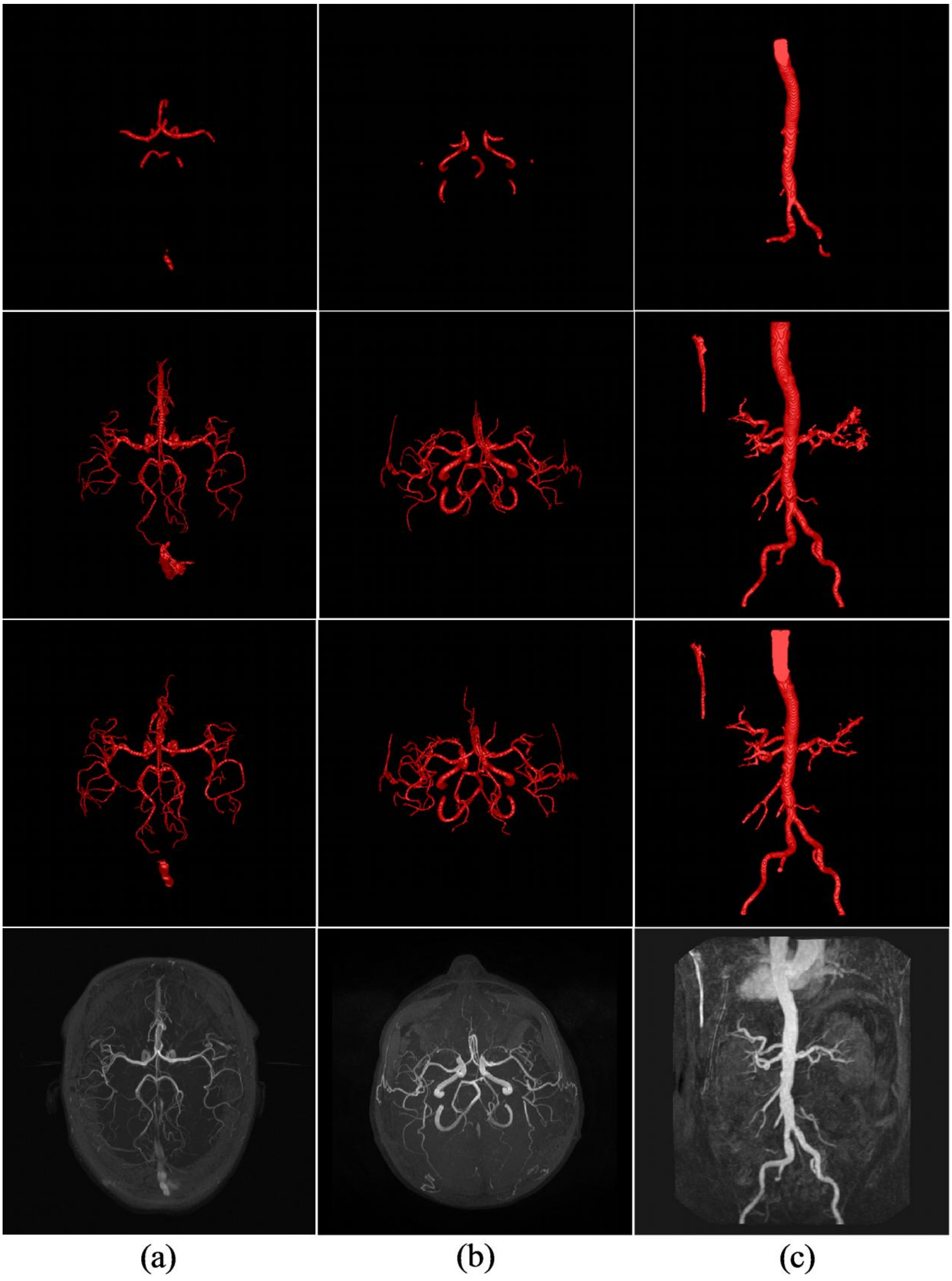


Figure 5.2 : Top row: Segmentation results of the geodesic active contour algorithm. Second row: Segmentation results of the confidence connected region growing algorithm. Third row: Segmentation results of proposed method. Bottom row: Maximum intensity projections of raw data. (a) A brain MRA with $448 \times 448 \times 128$ voxels and $0.51\text{mm} \times 0.51\text{mm} \times 0.8\text{mm}$ voxel size. (b) A brain MRA with $512 \times 512 \times 155$ voxels and $0.47\text{mm} \times 0.47\text{mm} \times 0.6\text{mm}$ voxel size. (c) An abdominal MRA with $512 \times 512 \times 70$ voxels and $0.95\text{mm} \times 0.95\text{mm} \times 1.00\text{mm}$ voxel size.

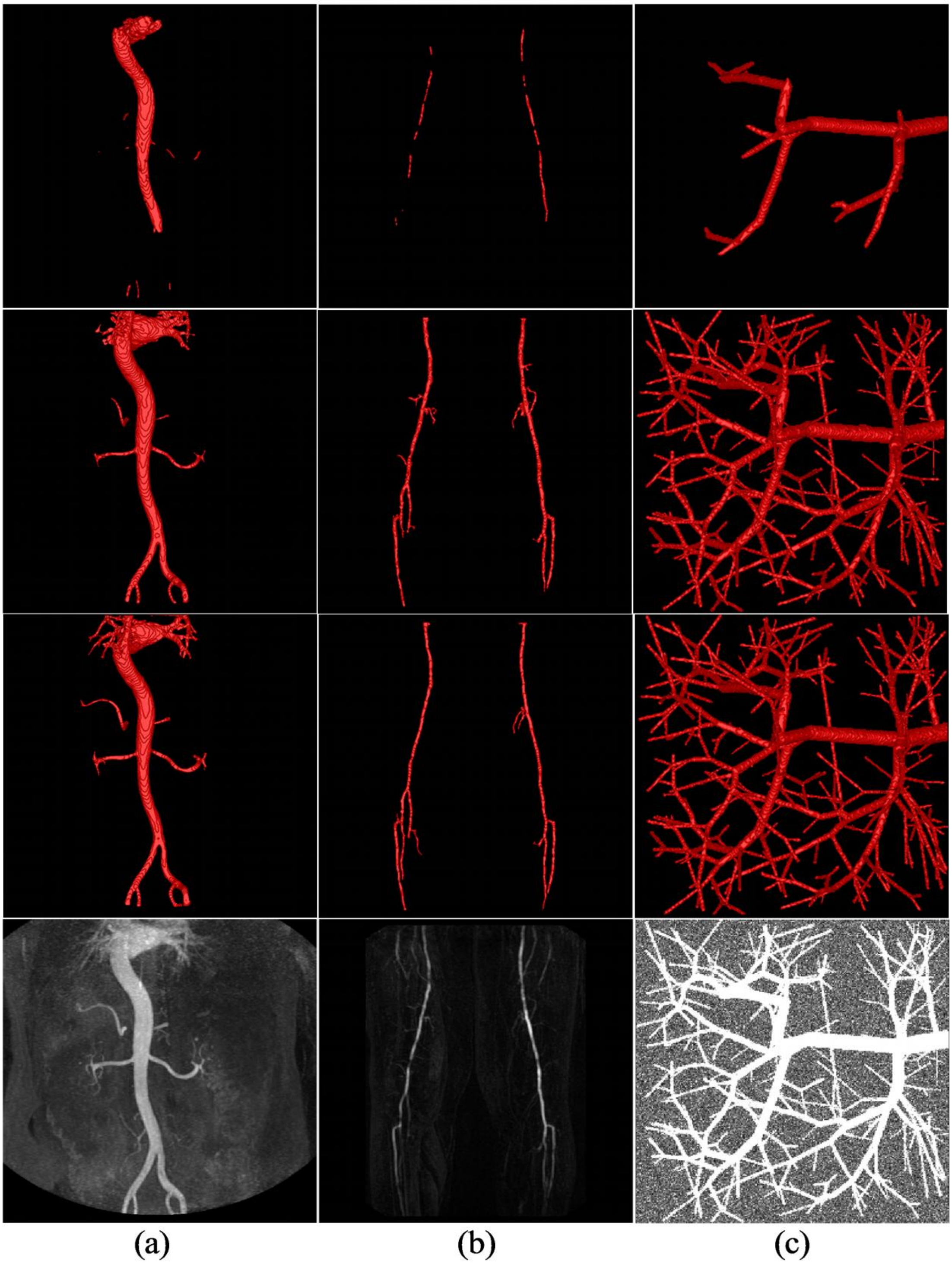


Figure 5.3 : Top row: Segmentation results of the simple geodesic active contour algorithm. Second row: Segmentation results of the confidence connected region growing algorithm. Third row: Segmentation results of proposed method. Bottom row: Maximum intensity projections of raw data. (a) An abdominal MRA with $512\times512\times50$ voxels and $0.78\text{mm}\times0.78\text{mm}\times1.00\text{mm}$ voxel size. (b) A thigh MRA with $512\times512\times70$ voxels and $0.92\text{mm}\times0.92\text{mm}\times1.00\text{mm}$ voxel size. (c) A synthetic image with added Gaussian noise $\sigma_{\text{noise}}=60$ that has $256\times256\times256$ voxels and $1.00\text{mm}\times1.00\text{mm}\times1.00\text{mm}$ voxel size.

It is seen that the proposed method clearly outperforms the two previous methods we have chosen for comparison. The geodesic active contour method which is based on [9] fails on tiny vessels because surface propagation is only high on areas of low gradient magnitude and gradient magnitude is usually high inside tiny vessels. The confidence connected region growing method is only based on image intensity so it can successfully segment most vasculature provided it is initialized with good seeds and correct user-defined parameters are chosen but it has a high risk to cause irregularities. Proposed method seems to segment most of the vasculature without causing severe problems.

To give a better feeling of how the user-defined variables γ, d and ξ affect segmentation, the effect of varying these parameters are illustrated in Figure A.1, Figure A.2, Figure A.3, Table A.1 and Table A.2. As expected, increasing the weight of the curvature term by increasing d or lowering γ segments more vasculature but increases the chance of oversegmentation. Increasing ξ obviously reduces oversegmentation but increases the chance of missing some vessels. Table A.1 and Table A.2 also serve as quantitative evaluation of the algorithm as results are compared against ground truth. Ground truth image is acquired by thresholding the noise-free synthetic image with the value 127. VascuSynth creates the image by dividing each voxel into 8 subvoxels and whether a vessel is contained in each subvoxel is calculated. For every subvoxel that contains a vessel, 32 is added to the image intensity for that voxel. Thus, thresholding with 127 means we accept all voxels which have 4 or more subvoxels that contain a vessel. For quantitative evaluation, 5 parameters are used. True Positive Ratio(TPR), False Positive Ratio(FPR), True Negative Ratio(TNR), False Negative Ratio(FNR) and Overlap Ratio. These are defined in eq. (5.3).

A very important observation is that changing the user-defined variables has a bigger impact on real images compared to synthetic images and the algorithm performs much better in synthetic images. This has two causes: First, real images contain a variety of image artifacts while the synthetic image is only distorted with Gaussian noise. Second, vessels in the synthetic image do not have high curvature. Vessels with high curvature do not give a high vesselness response to our Hessian based measure and segmentation of these vessels are mostly facilitated by the curvature term. Of course, relying on the curvature term is not desirable as it can cause irregularities and oversegmentation. Thus, the method is much more sensitive to user-defined variable selection and more prone to errors when the image contains vessels with high curvature and different types of artifacts.

$$
\begin{aligned}
&\mathrm{TPR} = \frac{\text{Number of True Positive Voxels}}{\text{Total Number of Voxels}}, \\
&\mathrm{FPR} = \frac{\text{Number of False Positive Voxels}}{\text{Total Number of Voxels}}, \\
&\mathrm{TNR} = \frac{\text{Number of True Negative Voxels}}{\text{Total Number of Voxels}}, \\
&\mathrm{FNR} = \frac{\text{Number of False Negative Voxels}}{\text{Total Number of Voxels}}, \\
&\text{Overlap Ratio} = \frac{2\mathrm{TPR}}{2\mathrm{TPR} + \mathrm{FPR} + \mathrm{FNR}}.
\end{aligned} \qquad \textbf{(5.3)}
$$

For the surface reconstruction part, let us start by first comparing the different spline types. The result of using Catmull-Rom splines and B-splines is shown in Figure 5.4. As Catmull-Rom splines are used to interpolate and B-splines are used approximate, Catmull-Rom gives a surface that is very loyal to the input data while B-splines smooth the data. As our segmentation output is not perfect and contains a lot of noise, Catmull-Rom splines create a rough surface with too many triangles. B-Splines seem to be a better choice but it should not be forgotten that very tiny structures may be lost.

Secondly, the effect of varying ρ and η is illustrated in Figure B.1 and Figure B.2 respectively. As we have already discussed, increasing ρ reduces the number of triangles but it is less accurate and can lose some structures. Increasing η reduces the number of triangles by making the triangle size more adaptive to curvature but this reduces triangle quality.

Next, the possible improvement provided by using Taubin smoothing is illustrated in Figure 5.5. 50 smoothing iterations are used with $\lambda = 0.50$ and $\mu = -0.54$ on the surfaces acquired by using $\rho = 0.4$, $\eta^{-1} = 0.8$ and B-splines.

Finally, the smoothed surfaces from Figure 5.5. are compared against the classical Marching Cubes result in Figure 5.6. It is seen that the recommended reconstruction scheme can achieve a smooth surface with high triangle quality while remaining fairly accurate.

Radiologist Dr. med. Zafer Kaya was asked to rate the final meshes resulting from the segmentation and surface reconstruction of our 34 real images with a scale of 1 to 5(1 worst, 5 best) and comment on what problems there are with the reconstruction. The average rating was 2.94 and the problems were as follows: Despite our improvement to the Frangi vesselness measure, two separate vessels can still sometimes appear as one vessel and vessels can appear thicker than they actually are. Also, vessel contours can be irregular especially in tiny vessels and some vessels are not segmented especially in low contrast leg images. Although the radiologist evalution turned out worse than desired, it should be noted that all the images were automatically seeded and the same user-defined variables were used despite the fact that the images were acquired from different sources and anatomical regions. It may be possible to improve the results by optimizing the user-defined parameters on a specific anatomical region and images taken by the same MR machine.

It is important to note that at the current state of the art, no algorithm can be expected to be readily usable for all medical procedures and all images. There are many different types of artifacts and vessels in medical images and each medical procedure has different requirements. It is extremely difficult for a method to cope with all possible scenarios. It should be noticed that both the segmentation and reconstruction steps involve a tradeoff. The segmentation part has to balance accuracy with the amount of user interaction, computation cost and generalness, surface reconstruction step have to balance accuracy with smoothness and computation cost. These tradeoffs always have to be considered depending on the specific medical application. Requirements of an application should be determined by close collaboration with medical partners and vessel segmentation and surface reconstruction methods should be selected accordingly. Assessing the usefulness of our proposed method for specific medical applications remains as future work.

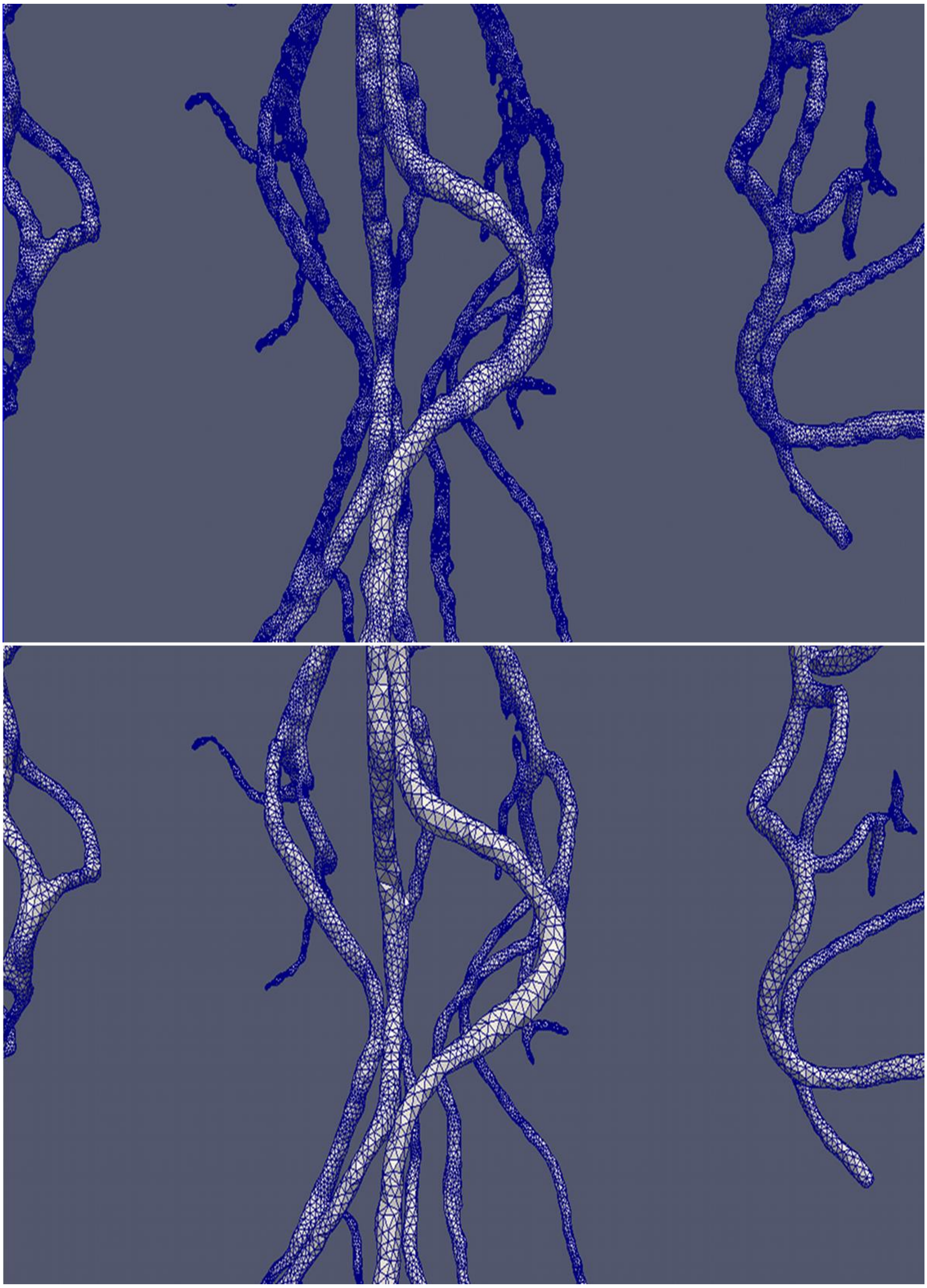


Figure 5.4 : Comparison of using Catmull-Rom splines(top) vs B-splines(bottom) on one of the brain images. $\rho = 0.4,\ \eta^{-1} = 0.8$.

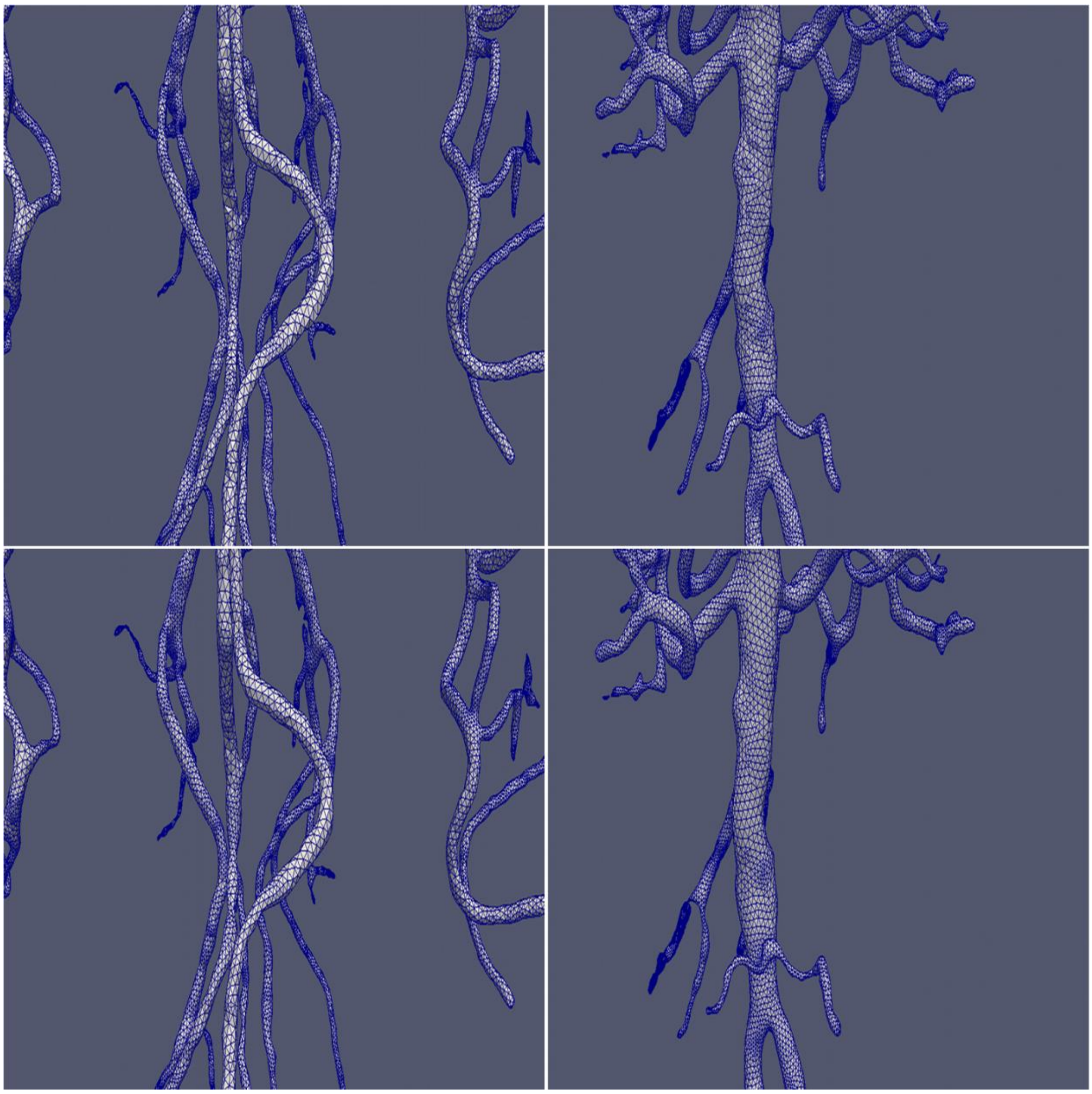


Figure 5.5 : Effect of Taubin smoothing. Top: Before Taubin smoothing. Surface created using $\rho = 0.4$, $\eta^{-1} = 0.8$ and B-splines. Bottom: After Taubin smoothing with $\lambda = 0.50$ and $\mu = -0.54$ for 50 iterations.

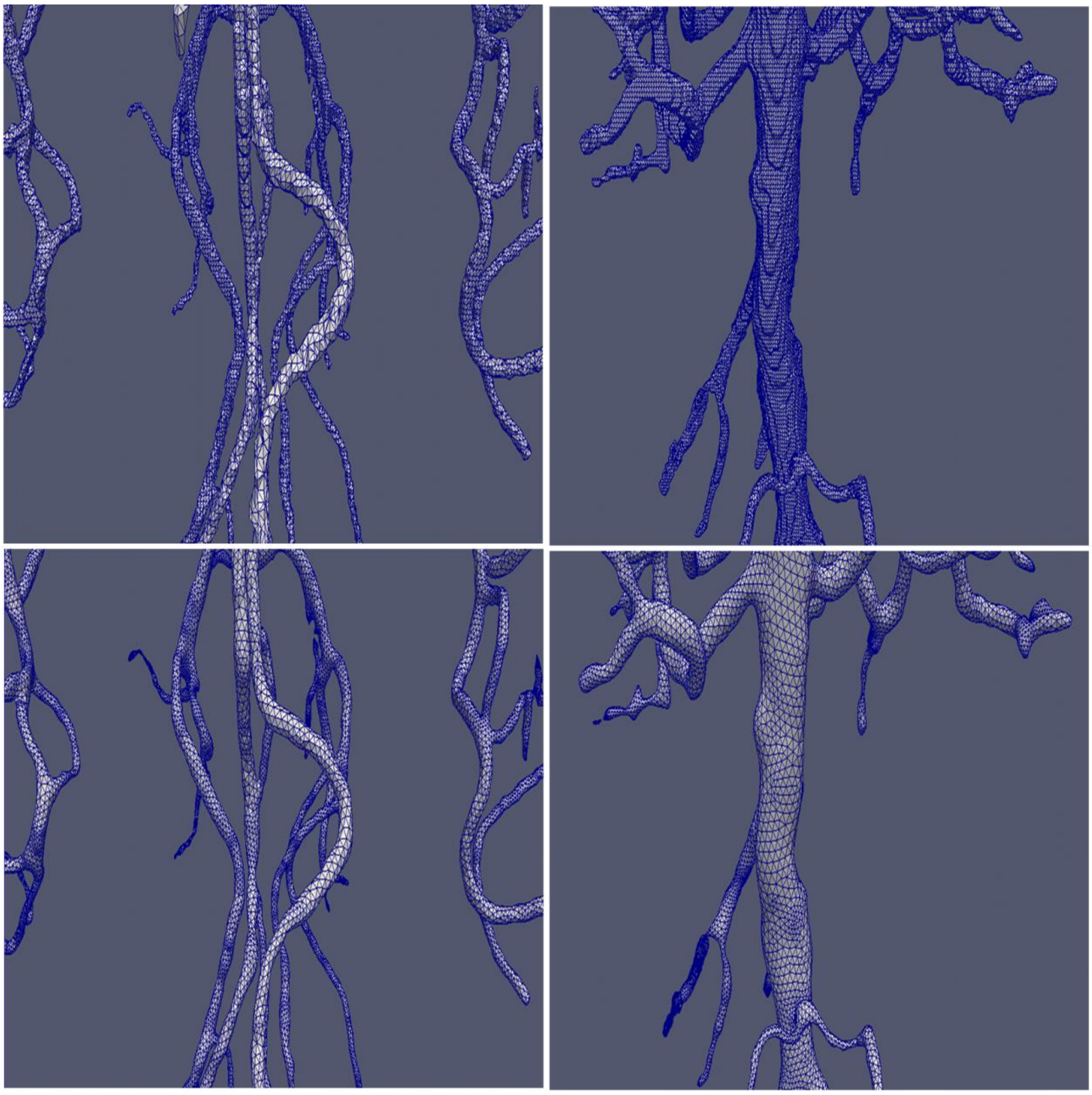


Figure 5.6 : Comparison of our recommended scheme(bottom) against Marching Cubes(top).

6. CONCLUSION AND RECOMMENDATIONS

We have presented a combination of methods to extract a high-quality polygonized surface that represents the vasculature from MRA images. Two main steps involved are vessel segmentation and surface reconstruction.

The vessel segmentation method takes as input MRA image data and produces as output a 3D discrete grid sampling of the implicit function that represents the vessel surface. The method was designed to work for all anatomical regions and segment both tiny and large vessels while requiring minimal user interaction. To this end, local multi-scale Hessian and curvature information were used to guide the evolution of the surface in a level set framework. Visually good-looking results were obtained for our dataset but careful inspection by our radiologist revealed some inaccuracies. The variability of medical image data make it extremely difficult to come up with a fully automatic method that works for all anatomical regions and all types of vessels. If very high accuracy is desired, a general method like the one proposed may not be desirable. In these cases, a general method could be used for initializing a more specialized algorithm.

The result of the segmentation step is fed to an existing advancing front implicit surface polygonizer that can construct a high-quality surface suitable for numerical simulation. Finally, Taubin smoothing is used to further smooth the surface without changing its topology.

It is important to stress that each step involves a tradeoff. The segmentation part has to balance accuracy with the amount of user interaction, computation cost and generalness, surface reconstruction and post-processing steps have to balance accuracy with smoothness and computation cost. These tradeoffs always have to be considered depending on the specific medical application. Although the ultimate goal is to have a computer that can process any imaging data in a meaningful way for a given medical procedure without requiring any user-interaction or correction, we are still far from achieving this and an algorithm cannot be expected to work for all scenarios. Thus, computer scientists that work on developing medical algorithms should closely collaborate with clinicians and know the requirements of different medical procedures. To give some examples, an accuracy that is enough for diagnostics may not be acceptable for planning brain surgery. While smoothness is required for numerical simulations, it may not be desirable for diagnostics. The computational cost may not be an issue for planning and diagnostics but the number of created triangles may be prohibitive for a numerical simulation.

6.1 Future Work

As future work, the method should be optimized and validated for specific applications. Using the surface constructed by the method, a volume mesh should be generated and used in CFD simulations. The results of these CFD simulations should also be analysed and validated by clinical partners. The speed of the whole process should be improved by optimizing the code and making full use of computational resources by using all the CPU and GPU cores efficiently. A visually pleasing and intuitive user interface that can easily be used by clinicians should be developed. The user interface should ideally include all the pipeline from imaging data to CFD simulation and tools that allow for corrections in case the result of automatic algorithms contain inaccuracies.

REFERENCES

[1] **World Health Organization(WHO).** (2008). The top ten causes of death - Fact sheet N310.

[2] **Lusis, A. J.** (2000). Atherosclerosis. *Nature*, ***407***, 233-241.

[3] **Kayembe, K. N., Sasahara, M. and Hazama F.** (1984). Cerebral aneurysms and variations in the circle of Willis**.** *Stroke*, ***15***, 846-850.

[4] **Löhner, R., Cebral, J., Soto, O., Yim, P. and Burgess, J. E.** (2003). Applications of patient-specific CFD in medicine and life sciences**.** *International Journal for Numerical Methods in Fluids,* ***43***, 637-650.

[5] **Kirbas, C. and Quek, F.** (2004). A Review of Vessel Extraction Techniques and Algorithms. *ACM Computing Surveys,* Vol. ***36***, no. 2, pp. 81-121.

[6] **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,* Vol. ***13***, no. 6, pp. 819-845.

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

[8] **Sethian, J. A.** (1999). *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision and materials science.* Cambridge, U.K. New York: Cambridge University Press.

[9] **Caselles, V., Kimmel, R. and Sapiro, G.** (1997). Geodesic Active Contours. *International Journal of Computer Vision,* Vol. ***22***, no. 1, pp. 61-79.

[10] **Malladi, R., Sethian, J. A. and Vemuri, B. C.** (1995). Shape modeling with front propagation: A level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. ***17***, no. 2, pp. 158-175.

[11] **Lorigo, L. M., Faugeras, O. D., Grimson, W. E. L., Keriven, R., Kikinis, R., Nabavi, A. and Westin C. F.** (2001). CURVES: Curve evolution for vessel segmentation. *Medical Image Analysis,* ***5***, 195-206.

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

[13] **Law, M. W. K. and Chung, A. C. S.** (2009). A deformable surface model for vascular segmentation. *Lecture Notes in Computer Science,* ***5762***, 59-67.

[14] **Frangi, R. F., Niessen, W. J., Vincken, K. L. and Viergever, M. A.** (1998). Multiscale vessel enhancement filtering. *Lecture Notes in Computer Science*, ***1496***, 130-137.

[15] **Yu, G., Miao, Y., Li, P. and Bian, Z.** (2006). Region-based vessel segmentation using level set framework. *International Journal of Control, Automation and Systems*, Vol. ***4***, no. 5, pp. 660-667.

[16] **Bloomenthal, J. and Bajaj, C.** (1997). *An Introduction to Implicit Surfaces.* Morgan Kaufmann Publishers.

[17] **Lorensen, W. and Cline, H.** (1987). Marching Cubes: A high-resolution 3D surface construction algorithm. *Computer Graphics,* Vol. ***21***, no. 4, pp. 163-168.

[18] **Nielson, G. M. and Hamann, B.** (1991). The asymptotic decider: Removing the ambiguity in marching cubes. *IEEE Visualization*, pp. 83-91.

[19] **Bloomenthal, J.** (1994). An implicit surface polygonizer, in *Graphics Gems IV*, pp. 324-349, Academic Press, Boston.

[20] **Carr, H., Theußl, T. and Möller T.** (2003). Isosurfaces on optimal regular samples. *Proceedings of the Symposium on Data Visualization*, pp. 39-48.

[21] **Alliez, P., Ucelli, G., Gotsman, C. and Attene M.** (2005). Recent advances in remeshing of surfaces. In *State-of-the-art report of the AIM@SHAPE EU network.*

[22] **Chen, L.** (2004). Mesh smoothing schemes based on optimal Delaunay triangulations. *Proceedings of 13th International Meshing Roundtable*, pp. 109-120.

[23] **Desbrun, M., Meyer, M., Schröder, P. and Barr, A. H.** (1999). Implicit fairing of irregular meshes using diffusion and curvature flow. *Proceedings of SIGGRAPH 99*, pp. 317-324.

[24] **Jones, T., Durand, F. and Desbrun M.** (2003). Non-iterative, feature-preserving mesh smoothing, *ACM Transcations on Graphics*, Vol. ***22***, no. 3, pp. 943-949.

[25] **Ohtake, Y. and Belyaev, A. G.** (2002). Dual/primal mesh optimization for polygonized implicit surfaces, *7th ACM Symposium on Solid Modeling and Applications*, pp. 171-178.

[26] **Taubin, G.** (1995). Curve and surface smoothing without shrinkage, *ICCV 95*, pp. 852-857.

[27] **Moore, D. and Warren, J.** (1991). An improved contouring method for trivariate data, *Technical Report COMP TR91-166*, Department of Computer Science, Rice University, P.Ø. Box 1892, Houston, TX.

[28] **Azernikov, S. and Fischer, A.** (2006). A new volume warping method for surface reconstruction, *Journal of Computing and Information Science in Engineering*, ***6***, 355-363.

[29] **Hartmann, E.** (1998). A marching method for the triangulation of surfaces, *The Visual Computer*, Vol. ***14***, no. 2, pp. 95-108.

[30] **Hilton, A., Stoddart, A. J., Illingworth, J. and Windeatt, T.** (1996). Marching triangles: range image fusion for complex object modelling, *Proceedings of International Conference on Image Processing, **2***, 381-384.

[31] **Akkouche, S. and Galin, E.** (2001). Adaptive implicit surface polygonization using marching triangles, *Computer Graphics Forum*, Vol. ***20***, no. 2, pp. 67-80.

[32] **de Araujo, B. R. and Jorge, J. A. P.** (2004). Curvature dependent polygonization of implicit surfaces, *Proceedings of 17th Brasilian Symposium on Computer Graphics and Image Processing*, pp. 266-273.

[33] **Karkanis, T. and Stewart, A. J.** (2001). Curvature dependent triangulation of implicit surfaces, *IEEE Computer Graphics and Applications*, Vol. ***21***, no. 2, pp. 60-69.

[34] **Schreiner, J., Scheidegger, C. E. and Silva, C. T.** (2006). High-quality extraction of isosurfaces from regular and irregular grids, *IEEE Transactions on Visualization and Computer Graphics,* Vol. ***12***, no. 5, pp. 1205-1212.

[35] **Cebral, J. R., Hernandez, M. and Frangi, A. F.** (2003). Computational analysis of blood flow dynamics in cerebral aneurysms from CTA and 3D rotational angiography image data, *Proceedings of the International Congress on Computational Bioengineering*, Vol. ***1***, pp. 191-198.

[36] **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***, 1591-1595.

[37] **Cebral, J. R. and Löhner, R.** (2001). From medical images to anatomically accurate finite element grids. *International Journal for Numerical Methods in Engineering*, ***51***, 985-1008.

[38] **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. *IEEE Transactions on Medical Imaging*, Vol. ***22***, no. 5, pp. 674-684.

[39] **Malladi, R. and Sethian, J. A.** (1998). A real-time algorithm for medical shape recovery. *Proceedings of the International Conference on Computer Vision*, ***196***, 304-310.

[40] **Zhang, Y., Bazilevs, Y., Goswami, S., Bajaj, C. L. and Hughes, T. J. R.** (2007). Patient-specific vascular NURBS modeling for isogeometric analysis of blood flow. *Computer Methods in Applied Mechanics and Engineering*, pp. 2943-2959.

[41] **Yu, Z. and Bajaj, C.** (2002). Image segmentation using gradient vector diffusion and region merging. *16th International Conference on Pattern Recognition*, ***2***, 941-944.

[42] **Ju, T., Losasso, F., Schaefer, S., and Warren, J.** (2002). Dual contouring of hermite data. *SIGGRAPH*, pp. 339-346.

[43] **Url-1** <*http://www.radiologyinfo.org/*>, date retrieved 15.12.2011.

[44] **Sugahara, T., Korogi, Y., Nakashima, K., Hamatake, S., Honda, S. and Takahashi M.** (2002). Comparison of 2D and 3D digital subtraction angiography in evaluation of intracranial aneurysms. *American Journal of Neuroradiology, **23***, 1545-1552.

[45] **Johnson, T. R. C., Krauß, B., Sedlmair, M., Grasruck, M,. Bruder, H., Morhard, D., Fink, C., Weckbach, S., Lenhard, M., Schmidt, B., Flohr, T., Reiser, M. F. and Becker, C. R.** (2007). Material differentiation by dual energy CT: initial experience. *European Radiology, **17***, 1510-1517.

[46] **Drebin, R. A., Carpenter, L. and Hanrahan, P.** (1988). Volume rendering. *Computer Graphics,* Vol. ***22***, no. 4, pp. 65-74.

[47] **Suetens, P.** (2009). *Fundamentals of Medical Imaging 2nd Edition.* Cambridge, U.K. New York: Cambridge University Press.

[48] **Whitaker, R**. (2004). Isosurfaces and Level Sets, in *Insight into Images: Principles and Practice for Segmentation, Registration and Image Analysis,* pp. 193-217, Ed. Yoo T. S., A K Peters Ltd., Wellesey.

[49] **Ibanez, L., Schroeder, W., Ng, L. and Cates, J.** (2005). *The ITK Software Guide Second Edition.* Kitware Inc.

[50] **Sethian, J. A. and Strain, J.** (1992). Crystal Growth and Dendritic Solidification. *Journal of Computational Physics,* Vol. ***92***, no. 2, pp. 231-253.

[51] **Whitaker, R.** (1998). A level-set approach to 3D reconstruction from range data. *International Journal of Computer Vision,* Vol. ***29***, no. 33, pp. 203-231.

[52] **Adelsteinsson, D. and Sethian, J. A.** (1995). A Fast Level Set Method for Propagating Interfaces, *Journal of Computational Physics, **118***, 269-277.

[53] **Ugurlu, D., Demirci, S., Navab, N. and Celebi, M. S.** (2011). A Vessel Segmentation Method for MRA Images Based on Multi-scale Analysis and Level Set Framework, *3rd International MICCAI-Workshop on Computation and Visualization on (Intra)Vascular Imaging.* September 18, 2011 Toronto, Canada.

[54] **Sato, Y., Nakajima, S., Shiraga, N., Atsumi, H., Yoshida, S., Koller, T., Gerig, G. and Kikinis, R.** (1998). Three-dimensional multi-scale line filter for segmentation and visualization of curvilinear structures in medical images, *Medical Image Analysis,* Vol. ***2***, no. 2, pp. 143-168.

[55] **Hamarneh, G. and Jassi, P.** (2010). VascuSynth: Simulating vascular trees for generating volumetric image data with ground truth segmentation and tree analysis, *Computerized Medical Imaging and Graphics,* Vol. ***34***, no. 8, pp. 605-616.

[56] **Url-2** <*http://www.insight-journal.org/midas/community/view/21*>, date retrieved 20.10.2011.

[57] **Yushkevich, P. A., Piven, J., Hazlett, H. C., Smith, R. G., Ho, S., Gee, J. C. and Gerig, G.** (2006). User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability, *Neuroimage,* Vol. ***31***, no. 3, pp. 1116-1128.

[58] **Scheidegger, C. E., Fleishman, S. and Silva, C. T.** (2005). Triangulating point set surfaces with bounded error, *Symposium on Geometry Processing,* pp. 63-72.

[59] **Schreiner, J., Scheidegger, C. E., Fleishman, S. and Silva, C. T.** (2006). Direct (re)meshing for efficient surface processing, *Computer Graphics Forum,* Vol. ***25***, no. 3, pp. 527-536.

[60] **Bajaj, C. L., Pascucci, V. and Schikore, D. R.** (1996). Fast isocontouring for improved interactivity, *Volume Visualization Symposium*, pp. 39-46.

[61] **Catmull, E. and Rom, R.** (1974). *Computer Aided Geometric Design*, chapter A Class of Local Interpolating Splines, Academic Press.

[62] **Piegl, L. and Tiller, W.** (1997). *The NURBS Book*, Springer.

[63] **Lindeberg, T.** (1990). Scale-space for discrete signals, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. ***12***, no. 3, pp. 234-254.

[64] **Handerson, A.** (2007). *ParaView guide, a parallel visualization application,* Kitware Inc.

[65] **Url-3** <*http://meshlab.sourceforge.net/*>, date retrieved 10.10.2011.

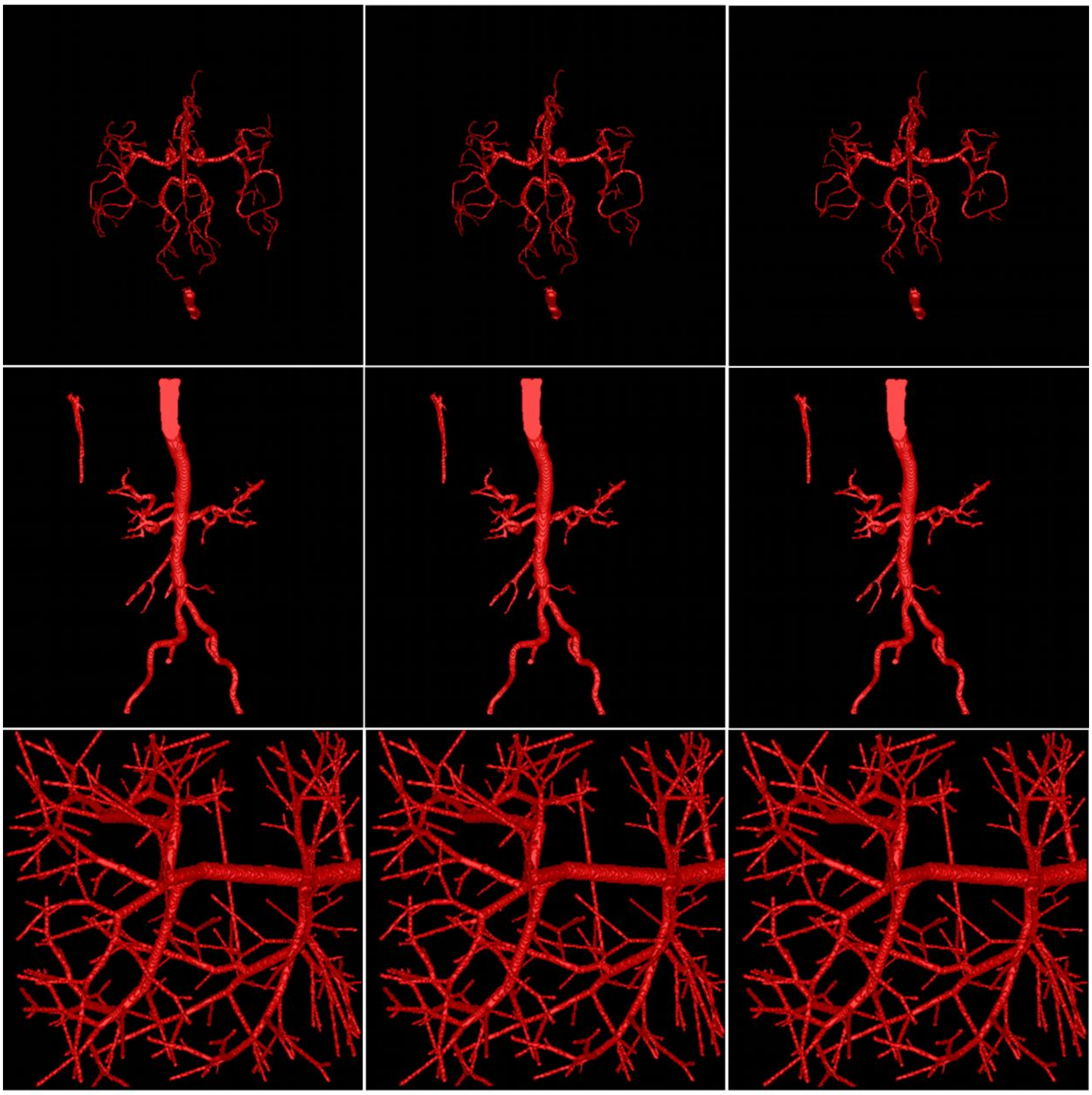
[66] **Schroeder, W., Martin, K. and Lorensen, B.** (2004). *The Visualization Toolkit: An Object Oriented Approach to 3D Graphics 3rd Edition,* Kitware Inc.

APPENDICES

APPENDIX A: Vessel segmentation results with different sets of user-defined parameters

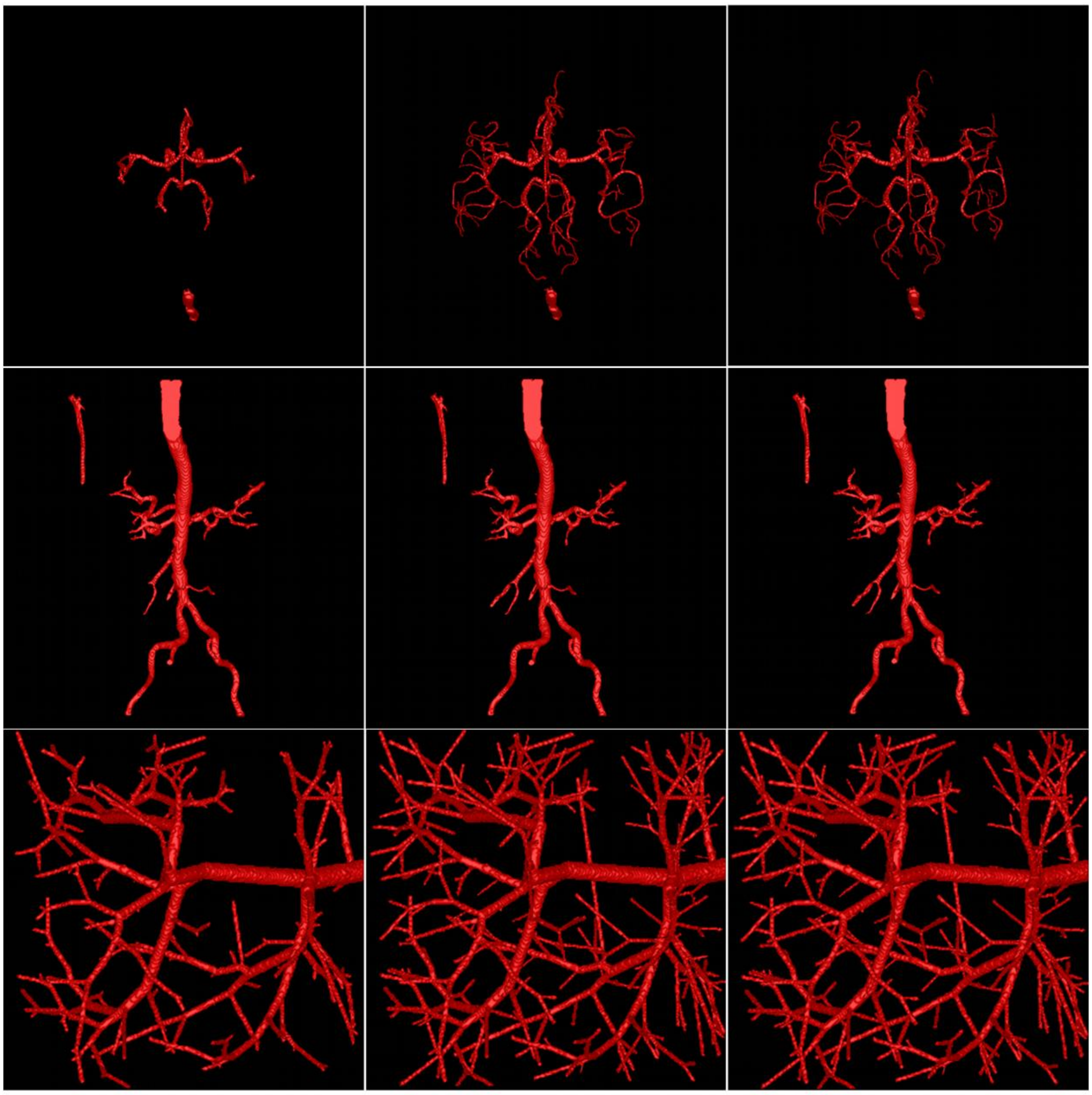
APPENDIX B: Surface reconstruction results with different sets of user-defined parameters

APPENDIX A



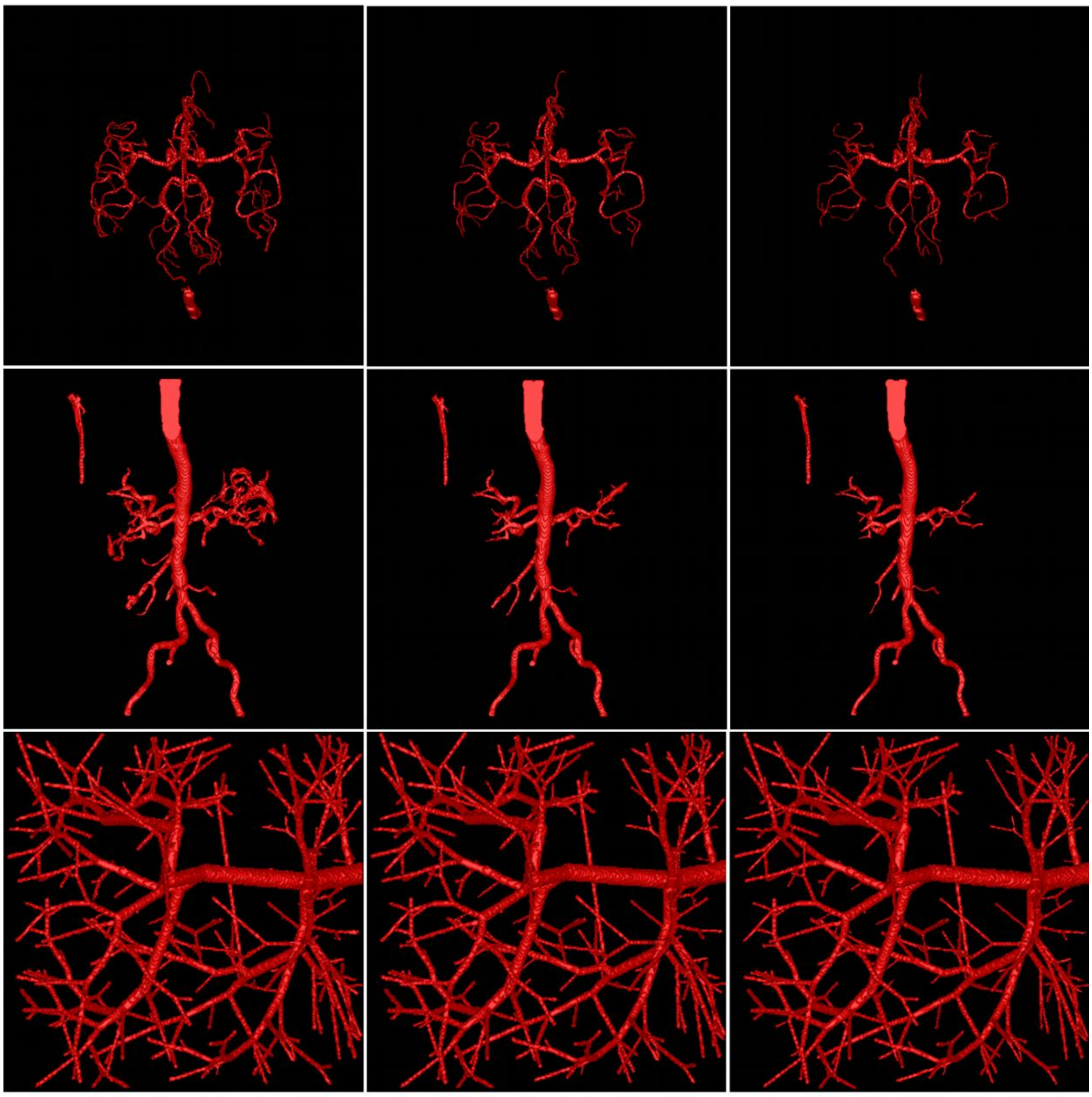
$\gamma=0.6,\ d=0.2,\ \zeta=20$ $\gamma=0.7,\ d=0.2,\ \zeta=20$ $\gamma=0.8,\ d=0.2,\ \zeta=20$

Figure A.1 : From top to bottom: Segmentation results for a brain MRA with $448\times448\times128$ voxels and $0.51\text{mm}\times0.51\text{mm}\times0.8\text{mm}$ voxel size, an abdominal MRA with $512\times512\times70$ voxels and $0.95\text{mm}\times0.95\text{mm}\times1.00\text{mm}$ voxel size, and the synthetic image with $\sigma_{\text{noise}}=60$ which has $256\times256\times256$ voxels and $1.00\text{mm}\times1.00\text{mm}\times1.00\text{mm}$ voxel size. From left to right: Varying γ.



$\gamma = 0.7,\ d = 0.0,\ \zeta = 20$ $\gamma = 0.7,\ d = 0.2,\ \zeta = 20$ $\gamma = 0.7,\ d = 0.4,\ \zeta = 20$

Figure A.2 : From top to bottom: Segmentation results for a brain MRA with $448 \times 448 \times 128$ voxels and $0.51\text{mm} \times 0.51\text{mm} \times 0.8\text{mm}$ voxel size, an abdominal MRA with $512 \times 512 \times 70$ voxels and $0.95\text{mm} \times 0.95\text{mm} \times 1.00\text{mm}$ voxel size, and the synthetic image with $\sigma_{\text{noise}} = 60$ which has $256 \times 256 \times 256$ voxels and $1.00\text{mm} \times 1.00\text{mm} \times 1.00\text{mm}$ voxel size. From left to right:Varying d.



$\gamma = 0.7,\ d = 0.2,\ \zeta = 10$ $\gamma = 0.7,\ d = 0.2,\ \zeta = 20$ $\gamma = 0.7,\ d = 0.2,\ \zeta = 40$

Figure A.3 : From top to bottom: Segmentation results for a brain MRA with $448 \times 448 \times 128$ voxels and $0.51\text{mm} \times 0.51\text{mm} \times 0.8\text{mm}$ voxel size, an abdominal MRA with $512 \times 512 \times 70$ voxels and $0.95\text{mm} \times 0.95\text{mm} \times 1.00\text{mm}$ voxel size, and the synthetic image with $\sigma_{\text{noise}} = 60$ which has $256 \times 256 \times 256$ voxels and $1.00\text{mm} \times 1.00\text{mm} \times 1.00\text{mm}$ voxel size. From left to right: Varying ξ.

Table A.1 : Quantitative evaluation of segmentation results for synthetic images with $\sigma_{\text{noise}} = 20$.

γ	d	ξ	TPR	FPR	TNR	FNR	Overlap Ratio
0.60	0.00	10	0.677	0.020	99.206	0.098	92.013
0.60	0.00	20	0.673	0.019	99.207	0.101	91.837
0.60	0.00	40	0.646	0.013	99.212	0.128	90.137
0.60	0.20	10	0.753	0.035	99.191	0.021	96.433
0.60	0.20	20	0.752	0.034	99.192	0.023	96.380
0.60	0.20	40	0.746	0.027	99.198	0.028	96.403
0.60	0.40	10	0.753	0.035	99.191	0.021	96.451
0.60	0.40	20	0.752	0.034	99.192	0.022	96.394
0.60	0.40	40	0.747	0.030	99.196	0.027	96.318
0.70	0.00	10	0.677	0.020	99.206	0.098	92.013
0.70	0.00	20	0.673	0.019	99.207	0.101	91.826
0.70	0.00	40	0.646	0.013	99.212	0.128	90.137
0.70	0.20	10	0.753	0.035	99.191	0.021	96.434
0.70	0.20	20	0.751	0.034	99.192	0.023	96.333
0.70	0.20	40	0.745	0.026	99.200	0.029	96.396
0.70	0.40	10	0.753	0.035	99.191	0.021	96.448
0.70	0.40	20	0.752	0.034	99.192	0.022	96.396
0.70	0.40	40	0.747	0.028	99.198	0.028	96.390
0.80	0.00	10	0.677	0.020	99.206	0.098	92.013
0.80	0.00	20	0.673	0.019	99.207	0.101	91.841
0.80	0.00	40	0.646	0.013	99.212	0.128	90.138
0.80	0.20	10	0.752	0.034	99.191	0.022	96.393
0.80	0.20	20	0.751	0.034	99.192	0.023	96.341
0.80	0.20	40	0.744	0.025	99.200	0.023	96.426
0.80	0.40	10	0.753	0.035	99.191	0.021	96.444
0.80	0.40	20	0.751	0.034	99.192	0.023	96.353
0.80	0.40	40	0.745	0.027	99.199	0.029	96.400

Table A.2 : Quantitative evaluation of segmentation results for synthetic images with $\sigma_{\text{noise}} = 60$.

γ	d	ξ	TPR	FPR	TNR	FNR	Overlap Ratio
0.60	0.00	10	0.622	0.031	99.195	0.152	87.156
0.60	0.00	20	0.613	0.027	99.199	0.161	86.748
0.60	0.00	40	0.573	0.010	99.215	0.201	84.454
0.60	0.20	10	0.728	0.062	99.164	0.046	93.119
0.60	0.20	20	0.723	0.048	99.177	0.051	93.540
0.60	0.20	40	0.689	0.018	99.208	0.085	93.052
0.60	0.40	10	0.729	0.064	99.162	0.045	93.029
0.60	0.40	20	0.724	0.052	99.174	0.050	93.407
0.60	0.40	40	0.698	0.023	99.202	0.076	93.363
0.70	0.00	10	0.619	0.030	99.195	0.155	86.946
0.70	0.00	20	0.613	0.027	99.199	0.161	86.748
0.70	0.00	40	0.573	0.010	99.215	0.201	84.454
0.70	0.20	10	0.728	0.061	99.165	0.046	93.148
0.70	0.20	20	0.721	0.047	99.179	0.053	93.545
0.70	0.20	40	0.685	0.016	99.209	0.089	92.870
0.70	0.40	10	0.729	0.062	99.163	0.045	93.111
0.70	0.40	20	0.723	0.050	99.176	0.051	93.512
0.70	0.40	40	0.693	0.020	99.206	0.082	93.192
0.80	0.00	10	0.622	0.031	99.195	0.152	87.156
0.80	0.00	20	0.613	0.027	99.199	0.161	86.748
0.80	0.00	40	0.573	0.010	99.215	0.201	84.454
0.80	0.20	10	0.727	0.060	99.165	0.047	93.132
0.80	0.20	20	0.721	0.046	99.180	0.053	93.578
0.80	0.20	40	0.681	0.015	99.211	0.093	92.629
0.80	0.40	10	0.729	0.061	99.164	0.046	93.149
0.80	0.40	20	0.723	0.048	99.178	0.052	93.583
0.80	0.40	40	0.686	0.017	99.209	0.089	92.884

APPENDIX B

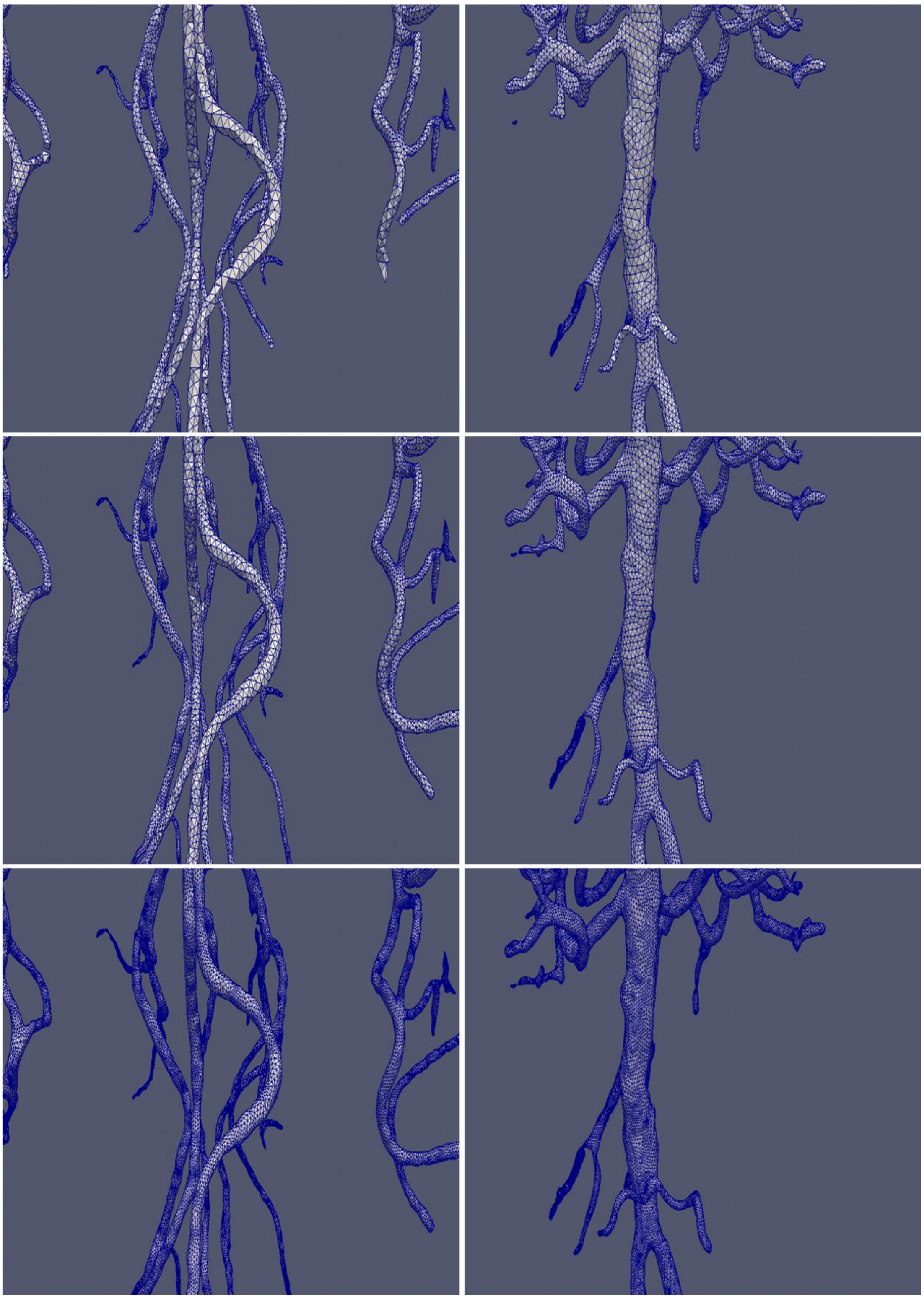


Figure B.1 : Surface reconstruction results for abdominal image(left) and cerebral image(right) with varying ρ. Top row: $\rho = 0.6$ middle row: $\rho = 0.4$ bottom row: $\rho = 0.2$. B-splines and $\eta^{-1} = 0.8$ are used for all images.

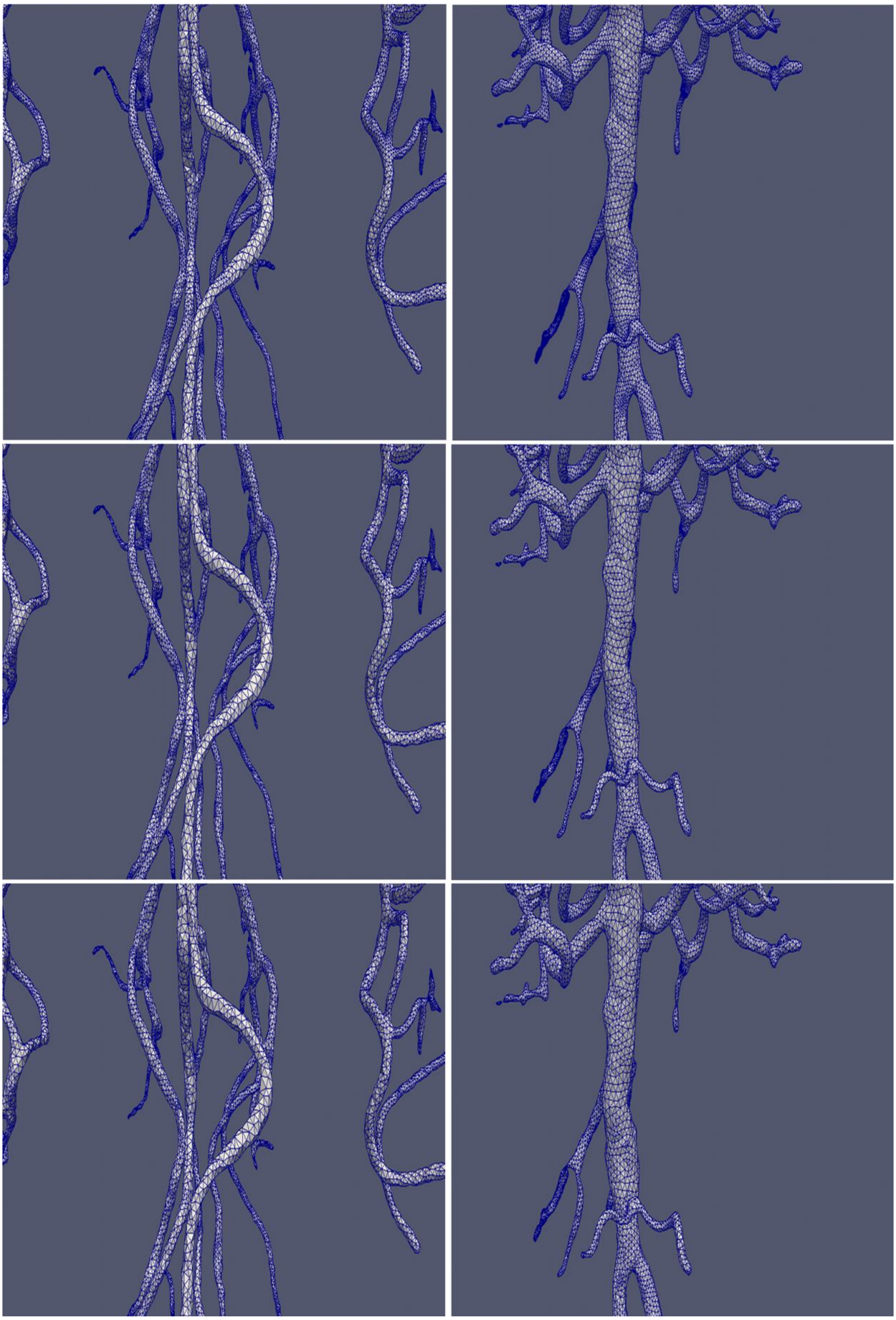


Figure B.2 : Surface reconstruction results for abdominal image(left) and cerebral image(right) with varying η. Top row: $\eta^{-1} = 0.8$, middle row: $\eta^{-1} = 0.65$, bottom row: $\eta^{-1} = 0.5$. B-splines and $\rho = 0.4$ are used for all images.

CURRICULUM VITAE

Name Surname: Devran Uğurlu

Place and Date of Birth: İstanbul, 16.03.1987.

Address: Sındı köyü Zeytincik mah. No:54 Datça, Muğla

E-Mail: devran.ugurlu@gmail.com

B.Sc.: Mathematics Engineering, Istanbul Technical University

List of Publications and Patents:

PUBLICATIONS/PRESENTATIONS ON THE THESIS

- **Ugurlu D.,** Demirci S., Navab N., and Celebi M. S., 2011: A Vessel Segmentation Method for MRA Images Based on Multi-scale Analysis and Level Set Framework, *3rd International MICCAI-Workshop on Computation and Visualization on (Intra)Vascular Imaging,* September 18, 2011 Toronto, Canada.