Spatio-Temporal Data Fusion in Cerebral Angiography

by

Andrew David Copeland
B.S., Electrical Engineering and Computer Science
Massachusetts Institute of Technology, 2001
M.Eng., Electrical Engineering and Computer Science
Massachusetts Institute of Technology, 2003

Submitted to the Department of Electrical Engineering and Computer Science

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2007

Author .....
Department of Electrical Engineering and Computer Science
May 23, 2007

Certified by .....
Sanjoy K. Mitter
Professor
Thesis Supervisor

Certified by .....
Rami S. Mangoubi
Senior Member of the Technical Staff, C. S. Draper Laboratory
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Thesis Committee Members

Sanjoy K. Mitter Ph.D.
Professor of Electrical Engineering and Computer Science
Massachusetts Institute of Technology

Rami S. Mangoubi Ph.D.
Senior Member of the Technical Staff
Draper Laboratory

Mukund N. Desai Ph.D.
Distinguished Member of the Technical Staff
Draper Laboratory

Alan S. Willsky Ph.D.
Professor of Electrical Engineering and Computer Science
Massachusetts Institute of Technology

Adel M. Malek M.D., Ph.D.
Chief, Division of Neurovascular Surgery
Tufts-New England Medical Center

Spatio-Temporal Data Fusion in Cerebral Angiography

by

Andrew David Copeland

Submitted to the Department of Electrical Engineering and Computer Science
on May 23, 2007, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

This thesis provides a framework for generating the previously unobtained high resolution time sequences of 3D images that show the dynamics of cerebral blood flow. These sequences allow image feedback during medical procedures that can facilitate the detection and observation of stenosis, aneurysms, and clots. The 3D time series is constructed by fusing together a single static 3D image with one or more time sequence of 2D projections. The fusion process utilizes a variational approach that constrains the volumes to have both smoothly varying regions separated by edges and sparse regions of non-zero support. Results are presented on both clinical and simulated phantom data sets. The 3D time series results are visualized using the following tools: time series of intensity slices, synthetic X-rays from an arbitrary view, time series of isosurfaces, and 3D surfaces that show arrival times of contrast using color. This thesis also details the different steps needed to prepare the two classes of data. In addition to the spatio-temporal data fusion algorithm, three new algorithms are presented: a single pass groupwise registration algorithm for registering the time series, a 2D-3D registration algorithm for registering the time series with respect to the 3D volume, and a modified adaptive version of the Cusum algorithm used for determining arrival times of contrast within the 2D time sequences.

Thesis Supervisor: Sanjoy K. Mitter
Title: Professor

Thesis Supervisor: Rami S. Mangoubi
Title: Senior Member of the Technical Staff, C. S. Draper Laboratory

Acknowledgments

First of all, I would like to thank my Draper Thesis Advisor Dr. Rami Mangoubi. He brought me into the nascent stages of what turned out to be a very interesting problem. His caring mentorship and guidance have been instrumental to the success of this thesis.

In addition, I would like to thank my MIT Thesis Advisor Professor Sanjoy Mitter. He has given me a great deal of advice and has pointed me in the right direction on some of the critical components of this work. He has helped me develop a more precise mathematical outlook in this thesis.

Thank you to Dr. Mukund Desai for serving on my doctoral committee. I have spent many a brain storming session discussing various issues that we faced. He has provided me with a great deal of depth and insight based on his experience in medical imaging.

Thank you to Professor Alan Willsky for serving on my doctoral committee. His insightful comments have helped to make this work more complete.

Thank you to Dr. Adel Malek for serving on my doctoral committee. It was his initial contact with Dr. Mangoubi and Dr. Desai that set the project in motion. He has been instrumental in providing us with datasets of images and medical insight into the relevance of the problem.

Thank you to my office mate Larry Chang for accompanying me through the last big push of my thesis work. He has kept me company on the many late nights and weekends at Draper while he completed his thesis work on a UUV mission planning system [22].

Thank you to my wife Liz Copeland. I met her the day after I completed my Masters thesis, which was just in time to be by my side through my final journey at MIT. She has provided me with her unwavering support and love and has kept me going when I did not think that I could. Additionally, she has helped to organize and proof read this document.

In my marriage to Liz, I have gained a whole new family. Thank you to my newly

expanded family of Tom, Cari, Ben, Emily, Alex, and I-Ju.

Thank you to my parents Drs. David and Helen Copland and to my sister Samantha and her husband Erik. Mom you have always kept me on task and helped me keep up with the details, Dad I got here because of a dream you helped me find long ago.

Lastly, I would like to thank my lord and savior Jesus Christ. "Then you will know the truth, and the truth will set you free." John 8:32

This thesis was prepared at The Charles Stark Draper Laboratory, Inc., under Internal Company Sponsored Research and Development, GC DLF Support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

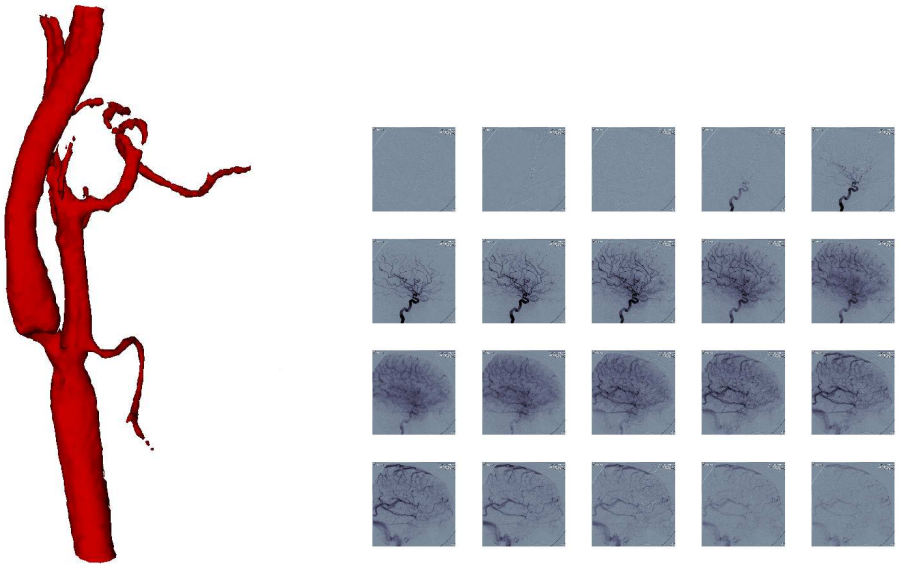
High resolution time sequences of 3D images that show the dynamics of blood flow would allow diagnostic capabilities previously unobtained in cerebral angiography. These sequences allow image feedback during medical procedures that can facilitate the detection and observation of stenosis, aneurysms, and clots. The sequences also allow for the use of offline simple computational fluid dynamic models of the flow. Using current algorithms, angiography systems cannot provide the necessary sequences of images due to insufficient volumetric sampling rates. The methods presented in this thesis take images from current angiography systems and achieve the goal of producing an accurate time sequence of 3D images.

Angiography systems currently produce high resolution 3D volumes (a visualization of a volume is shown in Figure 1-1(a)) by capturing a set of over a hundred X-rays each from a different angle. Using one of several reconstruction algorithms, these projections are combined to generate a single 3D image. The approximately 4 seconds it takes to capture a complete set of projections is clearly insufficient to observe the advancing front of the contrast agent within the blood at up to 100 cm/s [100]. Faster image sampling could capture time histories of the rapid blood flow but would also expose patients to many more X-rays. In addition to these 3D data sets, it is also possible to fix the angle of the angiography imaging system and capture two high temporal resolution X-rays from two fixed view points (an example time series is shown in Figure 1-1(b)). Currently, fusion of the time series information and 3D

data set must be performed mentally by skilled surgeons.

1.1 System Overview

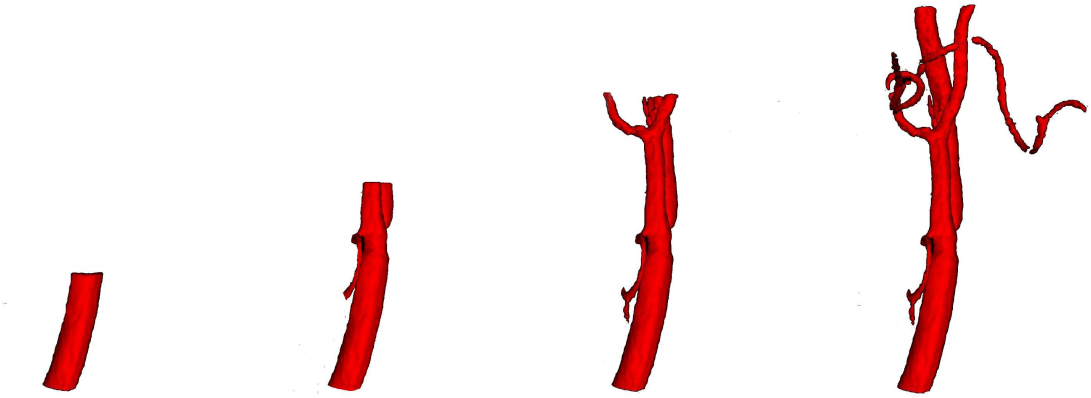
The method of spatio-temporal data fusion in this thesis describes a method for fusing the data sets to produce a time series of 3D volumes as depicted in Figure 1-2. Arrival times in 3D can also be extracted from this series and visualized using color as shown in Figure 1-3. The few projection time series are sparse and do not sufficiently constrain a unique solution for the fusion process. To overcome this, the reconstruction is regularized by taking into account constraints based on prior information such as a high resolution "map" generated from the single high resolution 3D image and other plausible image models such as constraining the image to contain smoothly varying regions separated by edges or simply constraining the reconstruction to have a sparse non-zero support. This method might be extended further to exploit physical laws such as viewing the blood flow as a wave-front propagation.



(a) A visualization of a 3D volume following segmentation.

(b) A 2D background subtracted time series taken from a single perspective.

Figure 1-1: **Utilized data sets** – The two data sets used by the algorithms of this proposal.



(a) Time 1 (b) Time 2 (c) Time 3 (d) Time 4

Figure 1-2: **Visualization of 3D time series** – Depiction of the reconstruction of a simulated 3D time series at four different times.

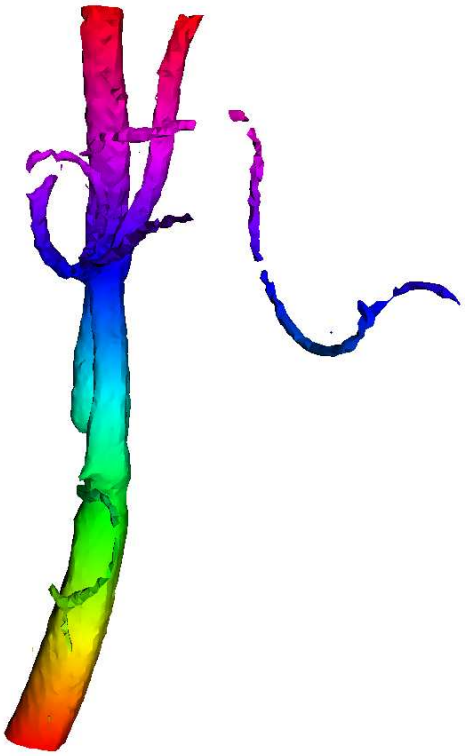


Figure 1-3: **Visualization of 3D arrival times** – Depiction of a simulated set of 3D arrival times. The colors indicate the time at which the contrast arrives at each point.

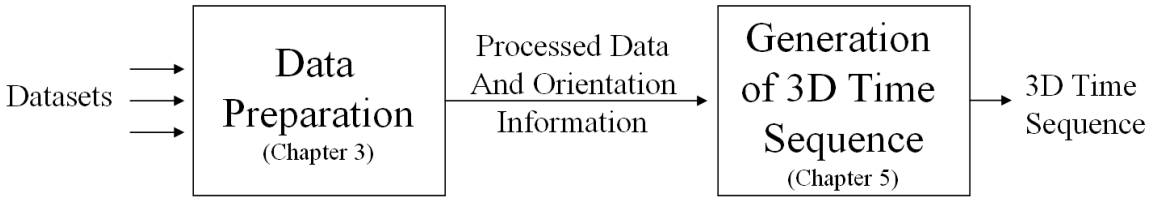


Figure 1-4: **Approach overview** A chart providing an overview of the approach presented. The sets of data are first processed and then used to generate the 3D time sequence.

The overall approach used to generate 3D time series as presented in this thesis can be broken into two main components shown in Figure 1-4. In the first component, the data is prepared by first registering each 2D time series, then segmenting the 3D volume to produce a map of the vasculature where the blood flow takes place, and lastly by finding the relative orientations of the 2D time series with respect to the 3D volume. Once the data is prepared, it can be fused using spatio-temporal data fusion that consists of a variational-based reconstruction of the 3D time series using constraints based on the data (two 2D time series and a segmented volume) and a 3D image model. A high quality 3D time series can be used to extract information about the 3D arrival times.

In addition to the work on 3D time series, a more accurate algorithm for determining contrast arrival times within the 2D time series is presented. This information provides an additional tool for surgeons that complements the information gained in the 3D time series. The 2D arrival time provides information about contrast arrival times in areas that are either outside the acquired volume of the 3D image model or too small to appear in the 3D image model itself.

1.2 Other Works

Several works [28, 31, 45, 66, 73, 97, 98, 103] generate a valid 3D skeleton that is consistent with the skeletons within the 2D time series. The skeleton is simply the centerline of the segmented vasculature. Two of these works [31, 45] produce a recon-

struction with a shell around the 3D skeleton based on estimates of the diameter of the vessels along the 2D skeletons. In [24], a method is presented for finding smoothly varying set of skeletons that show the dynamics of coronary arterial trees. In [87], a smoothly varying snake is found that is consistent with the projections. Several works generate a consistent 3D reconstruction using: binary matrix reconstruction [116], regularized version of several standard reconstruction algorithms [64] on six instead of two projections, and Simulated Annealing [85] on a reconstruction consisting of solid ellipses in each plane. Finally, [99, 100] provide the only algorithm that uses 3D data along with 2D time sets. In an alternative approach to the one presented in this thesis, arrival times are determined using correlation with a template time series from the location that corresponds to the arrival of the first contrast. The orientation is then found between the 3D volume and the 2D time series. The 3D volume information is distilled into a tree-like structure that is combined with the projections while constraining the order of contrast arrival times in 3D.

1.3 Contributions

The methods in this thesis provide both a more accurate reconstruction of the 3D time series and a more accurate determination of 2D and 3D contrast arrival times. Both of these extract more available data than do current methods. This is accomplished by using one of three new contributions:

1. A new approach for 2D arrival time detection that uses a new Cusum-based method [118] to provide a more accurate set of arrival times than the current methods in [51, 96, 99, 100, 120]. Unlike the current methods, this new method does not depend on the time series having the same but shifted (and possibly scaled) time waveform. This Cusum-based method is new because it introduces a second threshold that more accurately determines the arrival time when the primary threshold is surpassed and because it is adaptive to intensity dependent CCD noise.

2. Spatio-temporal data fusion provides a reconstruction that uses both the time series and the 3D map instead of only the time series as in [64, 85, 116]. In addition to the 3D map, this approach constrains the reconstruction to be smoothly varying off of a set of edges in time and space and to contain sparse vasculature structures.

3. The fusion process is a variational approach that can be extended by incorporating other constraints in a natural way.

1.4 Organization

This thesis is organized as follows.

Chapter 2 provides a mathematical foundation for the spatio-temporal data fusion problem. It presents the core of the notation used throughout this thesis. It also provides several numerical techniques used both in tomographic reconstruction and in simultaneous smoothing and segmentation. The notation and these algorithms provides the basis for the overall fusion problem of this thesis and an early look at the numerical methods used to solve it.

Chapter 3 describes the data preparation algorithms used to process the raw data that consists of one or more time series and a 3D volume. The presented algorithms are used to find the groupwise alignment of the raw time series, to segment the 3D volume, and to find the best 2D-3D registration consistent with the segmented volume and the registered time series.

Chapter 4 provides a new Cusum-based method for finding the arrival time of contrast at various points in the vasculature of the brain from a sequence of 2D X-ray angiograms. The 2D arrival time information can aid surgeons in assessing irregularities in the blood flow due to clots or other constriction.

Chapter 5 presents the Spatio-temporal data fusion algorithm that is used on the processed data from Chapter 3. This algorithm extends the variational framework to use constraints from simultaneous smoothing and segmentation, matrix projections, sparseness, time constraints, and 3D map constraint.

Chapter 6 presents the 3D time series results of spatio-temporal data fusion for both real angiographic data and simulated phantom data. The application results motivate the application, while the phantom data provide proof of its accuracy.

Chapter 7 concludes the thesis with a list of contributions and suggestions for future work.

Chapter 2

Background

This chapter provides a mathematical foundation for the spatio-temporal data fusion problem. Section 2.1 presents the core of the notation used throughout this thesis. It also presents a brief description of a numerical model for simulating X-rays.

In Section 2.2, several numerical techniques for tomographic reconstruction are presented. These include an exact procedure for generating both the forward and backward projection operators. In addition, the Simultaneous Algebraic Reconstruction Technique (SART) is presented, which has a framework similar to part of the spatio-temporal data fusion algorithm presented in Chapter 5.

The concept of simultaneous smoothing and segmentation is presented in Section 2.3, along with a more detailed description for solving it using a variational framework. This framework provides the basis for the overall fusion problem of this thesis and a preview of the numerical methods used to solve it.

2.1 Notation and Imaging Model

A notation for both functions of continuous domain and their discretizations are needed to describe the different aspects of the spatio-temporal data fusion problem. The fusion results in the four dimensional (three space and one time) process that is written as $F(x, y, z, t)$ for continuous domain and $F[i, j, k, n]$ for discrete domain. In general, the parentheses are used to denote a function defined on a continuous domain

while the brackets are used to denote a function defined on a discrete domain. When neither parenthesis nor brackets are present, the choice is implied by the context.

The overall 3D volume taken during steady state is notated as $V(x, y, z)$ and $V[i, j, k]$, and the 2D time series are notated similarly as $G(u, v, t)$ and $G[l, m, n]$. Using a vector notation for the spatial arguments, $F(x, y, z, t)$ is written as $F(\mathbf{x}, t)$ where $\mathbf{x}$ is a vector with components x, y, and z. Similarly the vector notation is used for $F[\mathbf{x}_p, n]$, $V(\mathbf{x})$, and $V[\mathbf{x}_p]$. The notation $[\mathbf{x}_p]$ is an abuse of notation that is interpreted as the bilinear interpolated value [69] of the image at the spatial coordinate $\mathbf{x}_p$. The binary valued constraint map $\mathscr{M}$ is produced by segmenting the 3D volume V. The segmentation labels the locations where the vasculature is thought to be with a one and everywhere else with zero. As with the volume, the constraint map $\mathscr{M}$ can be notated as a function of both continuous and discrete domains of either three arguments or a single vector argument. The two 2D time series is also written with the vector notation as $G[\mathbf{x}_q, n]$, where $\mathbf{x}_q$ is instead a 2D coordinate. They are defined so that X-rays from multiple views are represented as a single function. For example, in the continuous case, if the X-rays captured from two separate angles at a given time each have the same compact support of $[0, N] \times [0, M]$, they can be written as a single function with compact support of $[0, 2N] \times [0, M]$. This defines a single function on a larger domain in place of the two functions. An analogous approach is used in the discrete case.

The observation of the continuous 3D volume is modeled as

$$V(x, y, z) = \int_{t_{ss}}^{t_{ss}+\Delta t} F(x, y, z, t)dt + \nu(x, y, z) \tag{2.1}$$

and in the discrete domain as

$$V[i, j, k] = F[i, j, k, n_{ss}] + \nu[i, j, k], \tag{2.2}$$

where t_{ss} is the sample time for the continuous case; n_{ss} the sample time for the discrete; Δt is the exposure time; and $\nu(x, y, z)$ and $\nu[i, j, k]$ are band limited noise.

The continuous 2D time series acquisition can be modeled as

$$G(u,v,t) = [\boldsymbol{\mathcal{A}}F](u,v,t) + \eta(u,v,t) \tag{2.3}$$

and for the discrete domain

$$G[l,m,n] = [\mathbf{A}F][l,m,n] + \eta[l,m,n], \tag{2.4}$$

where $\eta(u,v,t)$ and $\eta[l,m,n]$ are the corresponding band limited noise, and $\boldsymbol{\mathcal{A}}$ and $\mathbf{A}$ are the corresponding X-ray projection operators.

The generic operator $\boldsymbol{\mathcal{A}}$ [57] applied to F can now be defined as

$$[\boldsymbol{\mathcal{A}}F](u,v,t) = \int_{\Omega}\int_{t}^{t+\Delta t} F(\mathbf{x},t')\delta(u-\mathbf{a}_1(u,v)\cdot\mathbf{x}-b_1(u,v))\delta(v-\mathbf{a}_2(u,v)\cdot\mathbf{x}-b_2(u,v))d\mathbf{x}dt', \tag{2.5}$$

where Ω is a closed and connected subset of $\mathbb{R}^3$; δ is the Dirac delta function; and $\mathbf{a}_1(u,v)$, $\mathbf{a}_2(u,v)$, $b_1(u,v)$, and $b_2(u,v)$ define the set of rays as a function of u and v. The Dirac delta function can be defined as the function $\delta(x)$ satisfying the relationship

$$h(y) = \int_{-\infty}^{\infty} h(x)\delta(x-y)dx, \tag{2.6}$$

where $h(x)$ is an arbitrary continuous function. The selection of these parameters allow for any geometry of rays, each parameterized by the pair u, v. An example parameterization is

$$\mathbf{a}_1(u,v) = \cos\theta\hat{x} + \sin\theta\hat{y}, \tag{2.7}$$

$$\mathbf{a}_2(u,v) = \sin\phi\sin\theta\hat{x} - \sin\phi\sin\theta\hat{y} + \cos\phi\hat{z}, \tag{2.8}$$

$$b_1(u,v) = 0, \text{ and} \tag{2.9}$$

$$b_2(u,v) = 0, \tag{2.10}$$

where θ is a rotation around the unit vector $\hat{z}$, and ϕ is a subsequent rotation around the unit vector $\hat{x}$.

If the operator is confined to a single plane, i.e. $z = 0$, then letting $\phi = 0$ and $v = 0$ simplifies the expression to the Radon transform [88] defined as

$$\mathscr{R}(u,\theta)[F(\mathbf{x},t)] = \int_{\Omega \cap z=0} \int_{t}^{t+\Delta t} F(\mathbf{x},t')\delta(u - \cos\theta x - \sin\theta y) d\mathbf{x} dt'. \tag{2.11}$$

Following [84], the discretization of Equation 2.5 can be written as

$$(\mathbf{A}F)[l,m,n] = \sum_{p} \mathbf{A}_{p,l,m} F[\mathbf{x}_p, n], \tag{2.12}$$

where $\mathbf{A}_{p,l,m}$ is a discrete approximation to the two delta functions in Equation 2.5. This expression can be simplified by using the vector notation $\mathbf{x}_q = (l,m)$. This results in the following matrix vector computation

$$(\mathbf{A}F)[\mathbf{x}_q, n] = \sum_{p} \mathbf{A}_{p,q} F[\mathbf{x}_p, n]. \tag{2.13}$$

The exact approach for generating the approximation $\mathbf{A}_{p,q}$ will be outlined later in Section 2.2.1.

The operators $\boldsymbol{\mathcal{B}}$ and $\mathbf{B}$ are defined to be inverse operators of the continuous projection operator $\boldsymbol{\mathcal{A}}$ and discrete projection operator $\mathbf{A}$, respectively. The construction of $\mathbf{B}$ is discussed further in Section 2.2.2. In this study, a back projection [57] type operator is used. As in this application, when there are an insufficient number of angles, an exact inverse algorithms does not exist and there are actually multiple possible reconstructions that are consistent with the set of projections. Further constraints will be developed to regularize this process.

2.2 Tomographic Reconstruction

The objective of tomographic reconstruction algorithms is to find a reconstruction F that satisfies the relationship

$$G = \mathbf{A}F, \tag{2.14}$$

where G is the measurement and $\mathbf{A}$ is the projection matrix. Algorithms used in tomographic reconstruction can be broken into two broad categories: Algebraic and Fourier based. The Algebraic techniques solve for F iteratively without the need to invert A. Some examples from the crowded field of Algebraic algorithms are the Algebraic Reconstruction Technique (ART) [43], the Maximum Entropy Method [77], Simultaneous Algebraic Reconstruction Technique (SART) [8], conjugate gradient methods [58], and convex projections [84]. The Fourier methods take advantage of the Projection Slice Theorem [57]. This relationship allows direct reconstruction by applying an FFT algorithm to an interpolated and appropriately filtered frequency space representation of the projection information [10, 89, 104, 57]. Much of the work in these algorithms concerns the choice of filter and the method of interpolation.

In addition, there are several Fourier based algorithms for the more complicated geometries in cerebral angiography known as cone beam tomography [37, 44, 49, 52]. Cone beam tomography describes a single X-ray point source and an imaging plane that are rotated about a patient. These methods work by exploiting the mathematical structure of the 3D frequency information in a set of projections taken from many angles.

Generally speaking reconstruction of a 2D image containing N^2 pixels requires on the order of N^2 evenly spaced frequency samples from the various projections [61]. If, however, the object to be reconstructed can be generated by a basis of frequencies identified by the corresponding projections, then perfect reconstruction is possible. A set of uniformly spaced views of an object contain more information and are less sensitive to noise than a set of the same number of views confined to a narrow band of angles [61], e.g., 10 angles between 0 and 60 degrees. In [14], it is shown that in some cases a sparse set of frequency samples or projections can lead

to accurate reconstructions by finding the reconstruction with the minimum total variation. By adding the minimization with respect to the total variation term, the reconstruction process in [14] was regularized and gave accurate reconstructions to images that consisted of piecewise constant regions. In Chapter 5, several terms are used to regularize the reconstruction, including a term similar to the total variation.

The spatio-temporal data fusion application considered in this thesis contains projections from only two views. In a two view reconstruction setting, the Algebraic algorithms are a simple and natural choice. In fact, the variational framework that is used in Chapter 5 results in a set of iterative equations that contains a term that is identical to a version of the SART algorithm. To run these algorithm, the discrete projection matrix $\mathbf{A}$ must be generated. A straightforward approach for this is shown in the next section. To motivate the iterative equations, a brief description of both ART and SART are presented in Section 2.2.2 along with a method to produce the final projection matrix $\mathbf{A}$ and the back projection matrix $\mathbf{B}$.

2.2.1 Generating the projection matrix

The generation of the projection matrix $\mathbf{A}_{p,q}$, as defined in Equation 2.13, consists of first tracing each ray q through pixel or voxel space, as shown in Figure 2-1. Each ray is then discretized into a set of points spaced ds apart, depicted as the blue circles in Figure 2-1. Each of these points has a corresponding closest pixel, the red squares in Figure 2-1, or voxel whose indices are collected in the k-tuple $\mathscr{P}(q) = p_1, p_2, \ldots, p_k$. This k-tuple is a list of indices for the qth ray and can contain duplicates. The matrix is then defined to be

$$\mathbf{A}_{p,q} = \text{number of times } p \text{ appears in } \mathscr{P}(q). \tag{2.15}$$

The constraint map $\mathscr{M}$ is used to decrease the size of the k-tuple $\mathscr{P}(q)$. Specifically, the red squares are only chosen in regions where the map is defined. Figure 2-2 shows a possible map in yellow along with the reduced set of red squares. The smaller

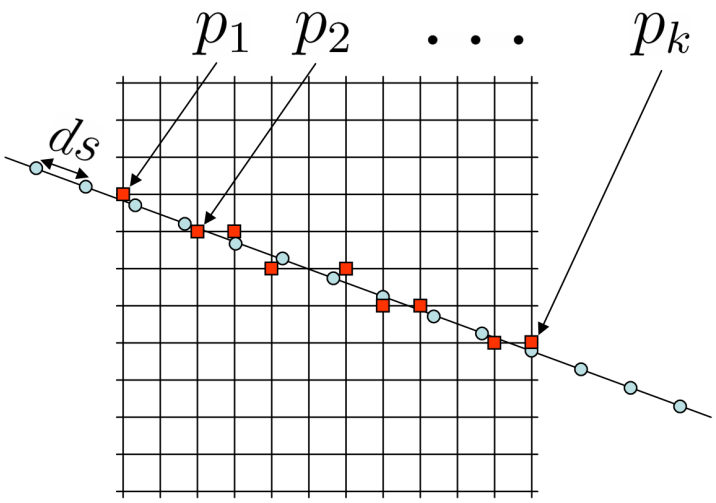


Figure 2-1: **Generation of A** – Figure shows how the coefficients of the matrix **A** are determined for a single ray. The vertices of the grid show the location where the 2D image F is defined. The ray contain circular dots corresponding to the points in F that are sampled to produce the projection. The red squares show the corresponding nearest neighbor to each circle that are used to determine the weights within the matrix.

number of pixel indices not only restrict the reconstruction to the map region, but also reduce the computational overhead involved in using the full set of pixel indices shown in Figure 2-1.

Generation and storage of the **A** matrix can speed up the 2D iterative algorithms as in [111]. However, this does not apply in the 3D angiography case because the matrix is extremely large, approximately 15 million voxels by 1 million pixels. Because most entries in the matrix are zero, the elements can be determine on the fly using precomputed values of the location of the first and second point of each ray, which is a more effective use of computational resources.

2.2.2 ART and SART

This thesis uses projections in the spatio-temporal data fusion algorithm to reconstruct a 3D time series. This use of the projection data is closely related to the work of two iterative reconstruction algorithms ART and SART, which are now described.

One of the earliest iterative algorithms for reconstruction from projections is the

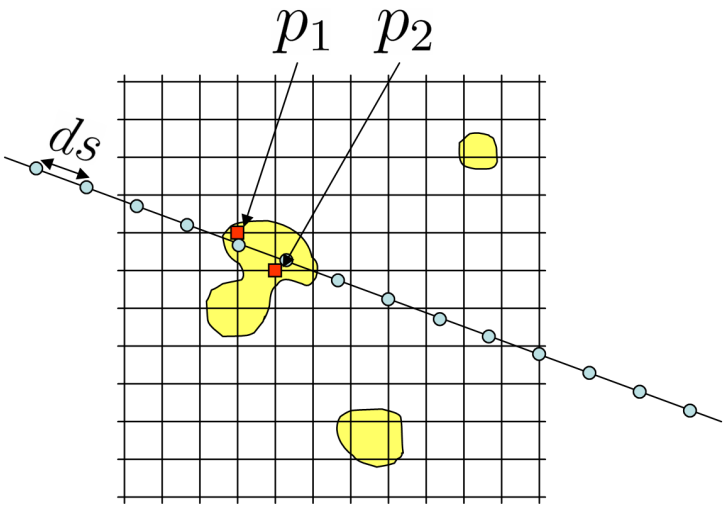


Figure 2-2: **Generation of A using map** – Figure shows the coefficients of the matrix **A** are determined for a single ray in the presence of a constraint map. The figure is similar to Figure 2-1, except the red squares corresponding to the nearest neighbor are restricted to lie within the constraint map $\mathscr{M}$ shown in yellow.

ART algorithms [43]. It can be written as

$$F[\mathbf{x}_p, n, k] = F[\mathbf{x}_p, n, k-1] + \mathbf{A}_{p,q(k)} \frac{(G[\mathbf{x}_{q(k)}, n] - \sum_{\hat{p}} \mathbf{A}_{\hat{p},q(k)} F[\mathbf{x}_{\hat{p}}, n, k-1])}{\sum_{\hat{p}} \mathbf{A}^2_{\hat{p},q(k)}}, \quad (2.16)$$

where $F[\mathbf{x}_p, n, k]$ is the value of F at iteration k and $q(k)$ is the particular ray to be back projected in iteration k. The function $q(k)$ dictates a schedule in which every ray is visited exactly once before any ray is visited a second time. This process then repeats until each ray is visited a second time and so forth until either the algorithm has repeated a set number of iterations or a convergence criteria is satisfied. The algorithm does not specify the order each ray is visited although some orderings can result in faster convergence.

The SART algorithm can be obtained by averaging the back projection of every ray at each pixel or voxel instead of back projecting the rays separately as is done in the ART algorithm. This change results in an algorithm with a faster convergence than the ART algorithm. A proof of this convergence appears in two parallel papers

[18, 56]. The resultant SART algorithm can be written as

$$F[\mathbf{x}_p, n, k] = F[\mathbf{x}_p, n, k-1] + \sum_q \mathbf{A}_{p,q} \frac{(G[\mathbf{x}_q, n] - \sum_{\hat{p}} \mathbf{A}_{\hat{p},q} F[\mathbf{x}_{\hat{p}}, n, k-1])}{\sum_{\hat{q}} \mathbf{A}_{p,\hat{q}} \sum_{\hat{p}} \mathbf{A}_{\hat{p},q}}. \quad (2.17)$$

The back projection operator $\mathbf{B}$ can be found by grouping terms in Equation 2.17,

$$\mathbf{B}_{q,p} = \frac{\mathbf{A}_{p,q}}{\sum_{\hat{q}} \mathbf{A}_{p,\hat{q}} \sum_{\hat{p}} \mathbf{A}_{\hat{p},q}}. \quad (2.18)$$

After making this substitution and reordering F and G, Equation 2.17 becomes

$$F[\mathbf{x}_p, n, k] = F[\mathbf{x}_p, n, k-1] - \sum_q \mathbf{B}_{q,p} (\sum_{\hat{p}} \mathbf{A}_{\hat{p},q} F[\mathbf{x}_{\hat{p}}, n, k-1] - G[\mathbf{x}_q, n]). \quad (2.19)$$

Despite its advantages, the SART algorithm is not appropriate for the spatio-temporal data fusion problem because it requires many more projections than the two that are available. However, the SART algorithm is identical to the term in Equation 5.37 in Chapter 5 used to enforce the data fidelity of the projections of the reconstruction. In addition to the data fidelity term, Equation 5.37 contains several other terms that are used to regularize the reconstruction. In Chapter 5, these terms are discussed in detail.

2.3 Simultaneous Smoothing and Segmentation

Another numerical technique that is drawn from heavily in this thesis is simultaneous smoothing and segmentation. This technique breaks an image into "nearly homogeneous regions, separated by smooth boundaries" [80]. The Mumford and Shah functional presented in [80, 81] poses the simultaneous smoothing and segmentation problem in a functional form that can be solved using Calculus of Variations (see Appendix A.1 or [41]). The process consists of minimizing the functional

$$E(f, \Gamma) = \alpha \int_\Omega (f - g)^2 d\mathbf{x} + \beta \int_{\Omega \setminus \Gamma} ||\nabla_{\mathbf{x}} f||^2 d\mathbf{x} + |\Gamma| \quad (2.20)$$

with respect to f and Γ, where f is the smoothed version of the measurement g, Ω is the image space, α and β are weighting coefficients, and $|\Gamma|$ represents the total length of the edge set Γ. The first term is a data fidelity term that ensures the smoothed version is similar to the measurement. The second term constrains the function to be smooth off the set of edges. The third term penalizes the length of the edge set. This formulation provides a continuous analogy to the algorithms in [42] and [71]. Writing $\exp(-E(f,\Gamma))$ and considering it as a density with respect to an approximate measure on (f,Γ) and discretizing one obtains the model of [42] of a Markov Random Field on the original lattice and dual lattice. These algorithms are based on the Ising model from Markov Random Fields [55] and are solved using the combinatorial optimization process known as simulated annealing [60].

In [5, 6], the edge term Γ was replaced with a spatial edge indicator function $w \in [0,1]$. The indicator equals one where there is an edge, zero where there is not an edge, and somewhere in between where there is uncertainty about the presence of an edge. The value of the indicator function can be interpreted as the probability that it is is an edge. The expression becomes

$$E(f,w) = \int_\Omega [\alpha(f-g)^2 + \beta(1-w)^2||\nabla f||^2 + \frac{\rho}{2}||\nabla w||^2 + \frac{w^2}{2\rho}]d\mathbf{x}, \tag{2.21}$$

where the third term encourages the edge function to be smoothly varying and the fourth term encourages the edge term to have sparse support. This change resulted in a less difficult optimization than that in Equation 2.20 because each of the terms in the new functional are defined on the same domain. This functional can be minimized by using the gradient descent equations found by calculating the Euler Lagrange equations. The two Euler Lagrange equations are

$$\partial_f E = \alpha(f-g) - \nabla \cdot [\beta(1-w)^2 \nabla f] \tag{2.22}$$

$$= \alpha(f-g) - \beta[(1-w)^2\nabla^2 f - 2(1-w)\nabla w \cdot \nabla f], \tag{2.23}$$

found by holding w fixed, and

$$\partial_w E = 2(w-1)\beta||\nabla f||^2 + \frac{w}{\rho} - \rho\nabla^2 w, \tag{2.24}$$

found by holding f fixed. A local minimum is determined by finding the combination of f and w at which both $\partial_f J = 0$ and $\partial_w J = 0$. The solution to these terms is similar to the anisotropic diffusion in [86], where smoothing is preferred in homogeneous regions and discouraged in regions with high gradient according to functions of the gradient $h(||\nabla f||)$.

It is shown in [5, 6] that as ρ goes to zero the functional in Equation 2.21 converges to the functional in Equation 2.20. In [90], this property was shown for a broader class of approximation functionals. This work also introduced the concept of the approximation matching Γ on a smaller and smaller scale as ρ goes toward zero. Lastly, in [62] it is shown that certain discretizations of the continuous functional in Equation 2.21 converges to the solution of the continuous functional as the lattice size tends to zero.

Another approach to segmentation is to use the techniques of level set methods. These methods use an implicit curve represented by the zero level set of the signed distance from the curve. The curve moves in a direction normal to itself according to a speed term [2, 79, 82, 83, 101]. The speed term changes the values of the level set, which in turn moves the embedded curve. Techniques that move such a curve in the direction normal to itself are known as curve evolution. In these iterative methods, no smoothing is performed. Other methods, such as those in [20, 21], use global metrics to control changes in the level sets and hence in the curves. In [112, 113], this type of method for segmentation was extended to also perform smoothing in regions tracked using a level set method with a globally defined metric. A curve evolution technique that uses a variational-based method instead of level sets is presented in [102]. This technique modifies the approach in [5, 6] to emulate the dynamics of curve evolution. This technique is more computationally intensive than the level set methods, however, it is not as sensitive to initial conditions on the locations of the edges. These curve

evolution techniques can be generalized in many ways, including multi-dimensional problems involving 3D time series [33] and diffusion tensor data [32].

The following two sections provide a summary of a version of the algorithm presented in [91]. In Section 2.3.1, Equation 2.21 is discretized and, In Section 2.3.2, it is extended to provide an implementable numerical algorithm.

2.3.1 Discretization

In [91], a straightforward and accurate discretization and the Euler Lagrange equations for the 2D smoothing and segmentation problem is presented. The 2D discretization is extended to 4D and adapted to the spatio-temporal data fusion application in Chapter 5. The first step of the discretization defines the lattices for both f and w and defines two auxiliary functions for determining neighboring lattice points of the respective functions. The smoothed and segmented image $f(\mathbf{x}_i)$ is defined on a rectangular lattice. The edge function $w(\mathbf{x}_i)$ is defined on a similar rectangular lattice produced by placing a point between each pair of neighboring points of the lattice that supports $f(\mathbf{x}_i)$. A depiction of these two lattices for a three by three example is shown in Figure 2-3

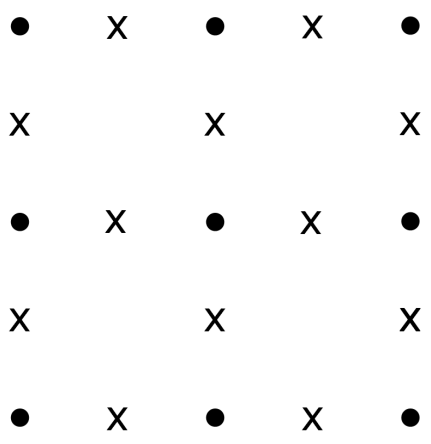


Figure 2-3: **Example lattices** – This figure shows an example three by three lattice of f using dots and the corresponding lattice of w using the symbol $\times$. The lattice points of w are placed between each of the lattice points of f.

The lattice for $f(\mathbf{x}_i)$ is defined on the subset of $\mathbb{Z}^2$ as

$$\mathscr{L}_f = \{(i,j) : i \in \{1,\ldots,M\}, j \in \{1,\ldots,P\}\}. \tag{2.25}$$

The nearest neighbors of the image at point $\mathbf{x}_i \in \mathscr{L}_f$ at time n are the points in the set

$$\mathscr{N}_f(\mathbf{x}_i) = \{\mathbf{x}_j \in \mathbb{Z}^2 : |\mathbf{x}_j - \mathbf{x}_i| = \delta\}, \tag{2.26}$$

where δ is the lattice spacing. The lattice for the edge function $w(\mathbf{x}_i)$ is defined as the lattice between each image point and its neighbor

$$\mathscr{L}_w = \{\frac{\mathbf{x}_l + \mathbf{x}_m}{2} : \mathbf{x}_l \in \mathscr{L}_f, \mathbf{x}_m \in \mathscr{N}_f(\mathbf{x}_l) \cap \mathscr{L}_f\}. \tag{2.27}$$

The resultant nearest neighbors of $\mathbf{y}_i \in \mathscr{L}_w$ are

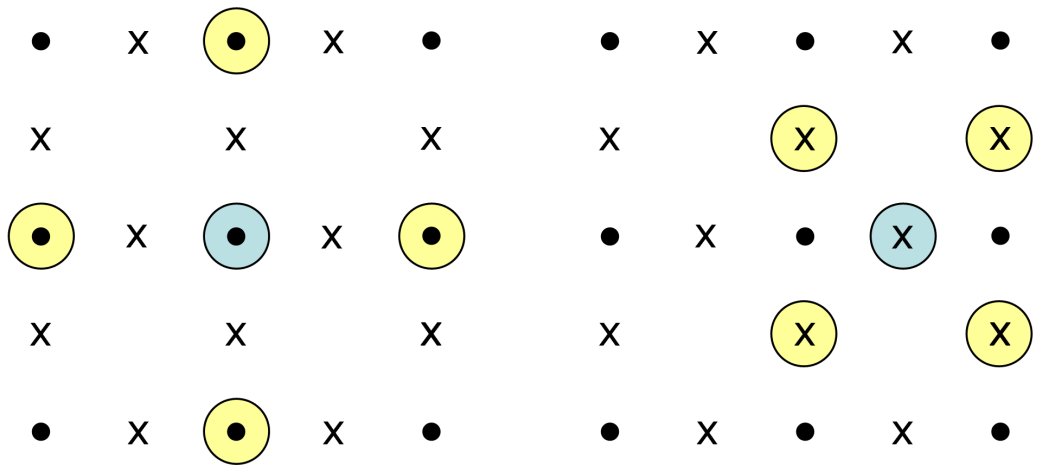
$$\mathscr{N}_w(\mathbf{y}_i) \ = \{\mathbf{y}_j = \frac{\mathbf{x}_l + \mathbf{x}_m}{2} : \mathbf{x}_l \in \mathscr{L}_f, \mathbf{x}_m \in \mathscr{N}_f(\mathbf{x}_l), |\mathbf{y}_j - \mathbf{y}_i| = \delta/\sqrt{2}\}. \tag{2.28}$$

Figure 2-4 shows the resultant nearest neighbors to both the lattice points of f and the nearest neighbors of each lattice point of w. Note that the distance between each lattice point of w and its nearest neighbors is $1/\sqrt{2}$ times that of f.

To simplify the implementation, a transformation is applied by replacing every appearance of w with $1 - w$. This reverses the interpretation of the values of 1 and 0. The resultant discrete version of the functional in Equation 2.21 is

$$E(f,w) = \sum_{\mathbf{x}_i \in \mathscr{L}_f} \left[\alpha(f[\mathbf{x}_i] - g[\mathbf{x}_i])^2 + \sum_{\mathbf{x}_j \in \mathscr{N}_f(\mathbf{x}_i)} \frac{\beta}{2} \frac{(f[\mathbf{x}_i] - f[\mathbf{x}_j])^2}{\delta^2} w[\frac{\mathbf{x}_i + \mathbf{x}_j}{2}]^2 \right] + \sum_{(\mathbf{y}_i) \in \mathscr{L}_w} \left[\frac{(1 - w[\mathbf{y}_i])^2}{2\rho} + \sum_{(\mathbf{y}_j) \in \mathscr{N}_w(\mathbf{y}_i)} \frac{\rho}{4} \frac{(w[\mathbf{y}_i] - w[\mathbf{y}_j])^2}{\delta^2/2} \right]. \tag{2.29}$$

The term $\delta^2/2$ is used because the mesh for w is twice as dense as that for f. From this point forward the expression is simplified by setting $\delta = 1$.



(a) Grid point neighbors (b) Edge point neighbors

Figure 2-4: **Nearest neighbors** – This figure shows the nearest neighbors to a lattice point of f (left) and to a lattice point of w (right). The lattice point of interest is placed on top of a small blue circle and the neighboring points are placed on top of yellow circles. The neighboring points are based on the nearest neighbors functions $\mathcal{N}_f(\mathbf{x}_i)$ (left) or $\mathcal{N}_w(\mathbf{y}_i)$ (right).

2.3.2 Numerical implementation

The Euler Lagrange equations for the discrete functionals are

$$\partial_{f_i} E = \alpha(f[\mathbf{x}_i] - g[\mathbf{x}_i]) + \sum_{\mathbf{x}_j \in \mathcal{N}_f(\mathbf{x}_i)} \beta(f[\mathbf{x}_i] - f[\mathbf{x}_j])(w[\frac{\mathbf{x}_i + \mathbf{x}_j}{2}])^2 \tag{2.30}$$

and

$$\partial_{w_i} E = \beta(f[\mathbf{x}_l] - f[\mathbf{x}_m])^2(w[\mathbf{y}_i]) + \frac{w[\mathbf{y}_i] - 1}{2\rho} + \sum_{(\mathbf{y}_j) \in \mathcal{N}_w(\mathbf{y}_i)} \rho(w[\mathbf{y}_i] - w[\mathbf{y}_j]). \tag{2.31}$$

These terms are equivalent to a discretization of Equations 2.22 and 2.24 after substituting $w - 1$ for w. These equations can be used to implement an iterative gradient descent towards a local solution to the minimization of the functional in Equation 2.29.

The variables can be modified to denote the value at a particular iteration k by adding the additional argument to the respective function, such as in $f[\mathbf{x}_i, k]$ and

$w[\mathbf{y}_j, k]$. The iterative steps are written as

$$f[\mathbf{x}_i, k] = f[\mathbf{x}_i, k-1] - c_f \cdot \partial_{f_i} E \tag{2.32}$$

and

$$w[\mathbf{y}_i, k] = w[\mathbf{y}_i, k-1] - c_w \cdot \partial_{w_i} E, \tag{2.33}$$

where c_f and c_w are terms that control the speed and stability of the descent process. The results in [91] provide the choices for the terms of

$$c_f = \frac{1}{2}(\alpha + 4\beta)^{-1} \tag{2.34}$$

and

$$c_w = \frac{1}{2}(\beta(f[\mathbf{x}_l, k-1] - f[\mathbf{x}_m, k-1])^2 + \frac{1}{2\rho} + 4\rho)^{-1}. \tag{2.35}$$

To see how c_w is determined, isolate each of the terms that multiply $w[\mathbf{y}_i, k-1]$ in Equation 2.33

$$w[\mathbf{y}_i, k] = \Big(1 - c_w\big(\beta(f[\mathbf{x}_l, k-1] - f[\mathbf{x}_m, k-1])^2 + \frac{1}{2\rho} + 4\rho\big)\Big) w[\mathbf{y}_i, k-1] + c_w(\frac{1}{2\rho} + \sum_{\mathbf{y}_j \in \mathcal{N}_w(\mathbf{y}_i)} \rho w[\mathbf{y}_j, k-1]), \tag{2.36}$$

then set the value of c_w so that the term multiplying $w[\mathbf{y}_i, k-1]$ is equal to $1/2$. This forces both the first line and second lines in equation 2.36 to be less than or equal to $1/2$, which is therefore a stable numerical scheme. An approximation to a similar term was used for determining c_f.

With this choice for c_w, the iterative equation for $w[\mathbf{y}_i, k]$ simplifies to

$$w[\mathbf{y}_i, k] = \frac{1}{2} w[\mathbf{y}_i, k-1] + c_w(\frac{1}{2\rho} + \sum_{\mathbf{y}_j \in \mathcal{N}_w(\mathbf{y}_i)} \rho w[\mathbf{y}_j, k-1]). \tag{2.37}$$

This choice for c_f results in the iterative equation

$$f[\mathbf{x}_i, k] = f[\mathbf{x}_i, k-1] - c_f \left[\alpha(f[\mathbf{x}_i, k-1] - g[\mathbf{x}_i]) + \sum_{\mathbf{x}_j \in \mathcal{N}_f(\mathbf{x}_i)} \beta(f[\mathbf{x}_i, k-1] - f[\mathbf{x}_j, k-1])(w[\frac{\mathbf{x}_i + \mathbf{x}_j}{2}, k-1])^2 \right]. \quad (2.38)$$

This final equation concludes the description of a simplified version of the numerical algorithm for smoothing and segmentation presented in [90, 91]. This numerical framework will be expanded in Chapter 5 to support the spatio-temporal data fusion algorithm.

Chapter 3

Data Preparation

Data preparation is required to convert the raw data from the medical procedure into a form that the spatio-temporal data fusion algorithm can use. This chapter describes the data preparation algorithms shown in Figure 3-1. This figure provides a flow chart for how the raw data, the volume and one or more time series, is processed for use later in the thesis. Before the raw data can be processed it must first be acquired, as described in Section 3.1. As shown in Figure 3-1, the data preparation begins with the groupwise alignment of the raw time series described in Section 3.2. In a parallel step, the 3D volume is segmented using the algorithm presented in Section 3.3. Finally, the relative orientation of the segmented volume with respect to the registered time series is found using the 2D-3D registration algorithm in Section 3.4.

3.1 Data Acquisition

During a procedure, doctors are able to observe the flow of blood in the brain by releasing an X-ray visible contrast agent into an artery in the neck. The effect of the contrast is then observed using an angiography system such as a Siemens AXIOM Artis. The observation process results in data in each of two forms: 1. a 3D volume $V[\mathbf{x}_p]$ taken in steady state by reconstructing X-rays taken from more than a hundred different perspectives (see Figure 1-1(a) for a view of a visualization of segmented volume), 2. one or more time series of X-rays $I_n[\mathbf{x}_p]$ taken from a fixed angle during

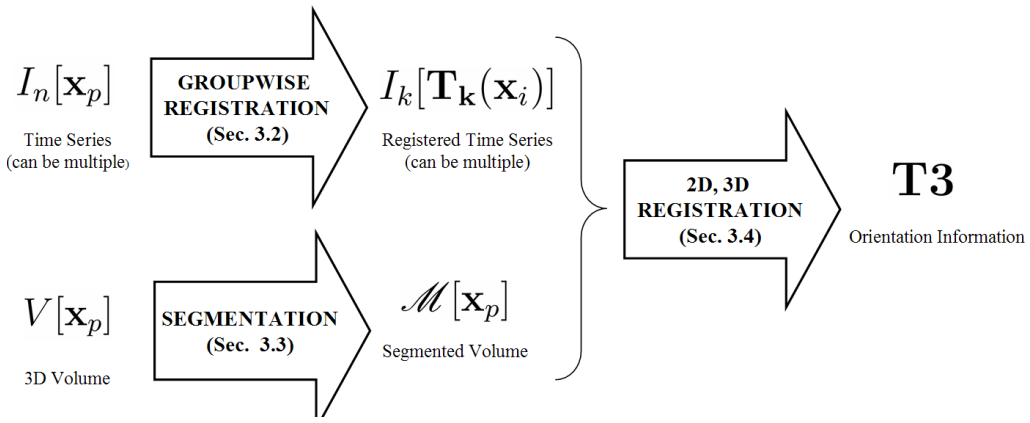


Figure 3-1: **Data preparation** – Flowchart for the preparation of the data set. The data is prepared and the appropriate orientations are found for use in the spatio-temporal data fusion in Chapter 5

the injection of contrast (see Figure 1-1(b) for a view of a time series). The 2D image sequences used in this work are captured from two separate angles at rates of up to 8 frames per second.

3.2 Groupwise Registration

The first step of data preparation removes the effect of patient motions from the time sequences using the new method for groupwise image registration known as the Once Through Sweep. The Once Through Sweep is a fast and accurate algorithm that relies on the least squares similarity metric to align the unimodal angiography time series. In addition to the least squares algorithm, a fast Joint Entropy-based groupwise algorithm that is more appropriate in multi-modal imaging is proposed for use in future work.

A comprehensive account of earlier work in registration in the field of angiography can be found in the survey [74]. The work in [74], however, deals mainly with pairwise image alignment of each image to a background image, even though angiography data sequences consist of several images. Groupwise registration takes these ideas one step further by finding the best alignment between a set of images. This can be thought

of as allowing features not present in all images to be weighed together. A groupwise algorithm was chosen instead of the common approach of pairwise registration with the first image both because it allows vasculature, which is the focus of this study, to have an effect on the registration and because it can better take it advantage of the greater similarity between images at adjacent times.

Past work on groupwise registration can be found in [65, 75, 114, 113, 123]. Multiple or groupwise registration of images is formulated in [65, 75] using the entropy similarity metric. The focus is on two applications: aligning large sets of different versions of individual written characters, and infant brain images from multiple subjects. The objective is to extract a central tendency from a collection. The computational performance of the algorithm in [65, 75] is enhanced by using a stochastic gradient descent method in [123]. Another groupwise algorithm that also relies on the least squares criterion is presented in [113], and further elaborated on in [114]. The algorithm in [114] uses a one versus many least squares comparison as opposed to the pairwise least squares comparison used in the algorithm developed in this section.

The recursive Once Through Sweep algorithm, depicted in Figure 3-2, begins on a set of images by first registering a pair of images with respect to one another and then finding a representative image of that pair. Subsequently, the next of the unregistered images in the set is registered with the representative image. Once registered, the image is added appropriately to the set of registered images and a new representative image is found. The process then repeats until all of the images have been registered. Each step of the groupwise registration scheme reduces to simple pairwise comparison with a "mean" image (see Equations 3.5 and 3.6, and Figure 3-2) as in [30, 115]. In both [30] and [115] the mean is calculated over all images instead of the set of aligned images as is done here. Other choices for the representative image include the median image and the minimum entropy image.

The algorithms in [75, 114] consists of iteratively moving each image, one at a time, into alignment with all the other images until the set is aligned. In contrast, the Once Through Sweep provides a simple recursive variation on these algorithms that utilizes any of the extensive numbers of pairwise registration algorithms [9, 13,

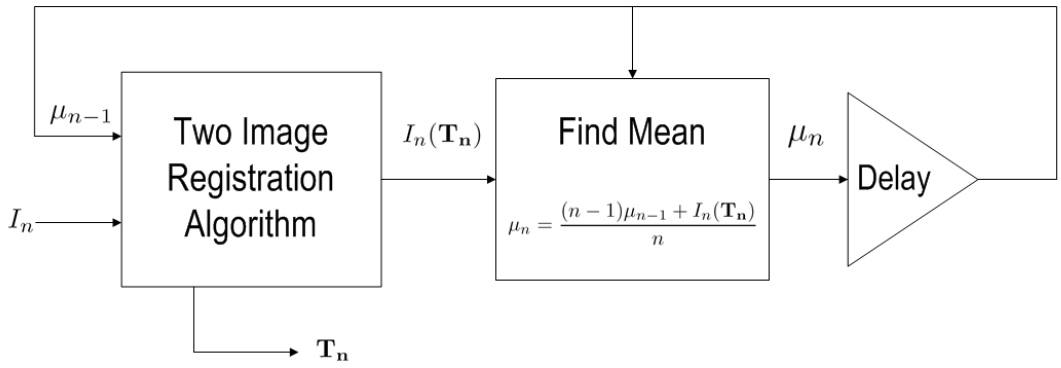


Figure 3-2: **A groupwise representative image registration algorithm.**. The algorithm begins with $n = 2$ and $\mu_1 = I_1$. This process upgrades any two image registration algorithm into a groupwise algorithm. In each iteration the algorithm begins by registering μ_{n-1} with I_n, then the registered version of I_n is added appropriately to the mean μ_{n-1} to produce μ_n, which is saved for the next iteration, when it is registered with I_{n+1}. In Section 3.2, this algorithm is generalized by instead using a representative image algorithm based on the entropy criterion.

26, 27, 54, 93, 106, 109, 110, 117, 121] to produce a groupwise algorithm. These algorithms may be sped up by sampling to decrease the number of points in the image [3] or by estimating the distribution of pixel intensities with sparse numbers of pixels [117]. For this application, the two-image registration algorithm developed in [121] is used. It provides both a highly accurate registration algorithm and a rich set of possible transformations.

This section next presents a mathematical derivation of the Once Through Sweep. Following the derivation of the least squares case, it is generalized to the Joint Entropy case. In Section 3.2.1 results are presented on the least squares algorithm. The section concludes in Section 3.2.2 with a discussion.

Registration

For a time series from a single viewpoint, the notation $G(u, v, t)$ for continuous domain and $G[l, m, n]$ for discrete domain are replaced by $I_t(\mathbf{x})$ and $I_n[\mathbf{x}_p]$, respectively, where p denotes the index of the coordinates of interest $\mathbf{x}_p \in \mathbb{R}^2$. The terms $\mathbf{T_n}$ and $\bar{\mathbf{T}}_\mathbf{n}$

are defined as a coordinate transformation, and the best coordinate transformation with respect to the chosen error metric, respectively, on the coordinates of image n. In general, a wide variety of transformations are possible, both rigid and elastic. Here, the algorithm based on [121] is used along with a globally rigid homogeneous transformation.

The objective of registration is to find the set of transformations $\mathbf{T_j}$ for each image I_j, where $j \in \{1, 2, \ldots, N\}$, that best aligns the group with respect to some metric. A cost function $\Phi(\mathbf{T_1}, \mathbf{T_2}, \ldots, \mathbf{T_N}; \mathbf{X}, I_1, I_2, ..., I_N)$ is constructed that measures the overall similarity amongst a collection of images, where $\mathbf{X}$ is the set of all pixel locations. A general expression is

$$\Phi^* = \min_{\mathbf{T_1}, \mathbf{T_2}, ..., \mathbf{T_N}} \Phi(\mathbf{T_1}, \mathbf{T_2}, \ldots, \mathbf{T_N}; \mathbf{X}, I_1, I_2, ..., I_N). \tag{3.1}$$

This formulation is restricted to transformations that are bijective (one to one and onto) mappings of compact regions contained within the respective images. This set is further restricted in the presented implementation to homogeneous transformations.

Following [76], Equation 3.1 is adapted to a pixel based entropy formulation with the objective function

$$E = \min_{\mathbf{T_1}, \mathbf{T_2}, ..., \mathbf{T_N}} \sum_{i=1}^{L} H(I_1(\mathbf{T_1}(\mathbf{x}_i)), I_2(\mathbf{T_2}(\mathbf{x}_i)), \ldots, I_N(\mathbf{I_N}(\mathbf{x}_i))), \tag{3.2}$$

where L is the total number of pixels under consideration and H is the entropy function for a stack of N pixel values at location x_i.

Similarly, Equation 3.1 is adapted to the root mean square error (RMSE) metric by minimizing the expression

$$\varepsilon = \min_{\mathbf{T_1}, \mathbf{T_2}, ..., \mathbf{T_N}} \left(\sum_{i=1}^{L} \sum_{n=1}^{N} \sum_{\substack{j=1 \\ j \neq n}}^{N} \frac{[I_j(\mathbf{T_j}(\mathbf{x}_i) - I_n(\mathbf{T_n}(\mathbf{x}_i))]^2}{L \cdot N \cdot (N-1)} \right)^{\frac{1}{2}}. \tag{3.3}$$

Equation 3.3 represents taking the square root of the average of the squared difference

between each image, I_n, with all the other images, I_j, over each pixel, $\mathbf{x}_i$. The set of translations that minimize this equation results in each image being most similar to all others in the least squares sense. This work will focus on the least squares algorithm in Equation 3.3 because it is more appropriate for the single modality angiography application.

The Representative Image

The optimization problem in Equation 3.3 is difficult to solve because it simultaneously searches for N transformations. A more tractable alternative is to iteratively solve each $\mathbf{T_n}$ separately while fixing each $\bar{\mathbf{T}}_\mathbf{j}$ from the most recent iteration for $j \neq n$ as in the expression

$$\bar{\mathbf{T}}_\mathbf{n} = \underset{\mathbf{T_n}}{\operatorname{argmin}} \sum_{\substack{j=1 \\ j \neq n}}^{N} \sum_{i=1}^{L} \frac{[I_j(\bar{\mathbf{T}}_\mathbf{j}(\mathbf{x}_i)) - I_n(\mathbf{T_n}(\mathbf{x}_i))]^2}{N-1}. \tag{3.4}$$

The bar denotes that the value of $\mathbf{T_j}$ is fixed to the optimal value obtained in an earlier iteration. To start the algorithm, each of the $\mathbf{T_j}$'s is initialized to the Identity operator. The algorithm continues by sequentially solving Equation 3.4 in a loop across each of the images until convergence to a local minimum. The result is an algorithm that registers one image versus all other images. This process is a least squares analogy to the iterative gradient descent process used to align images under the entropy metric known as "congealing" that is described in [65, 75, 76]. The algorithm can be applied repeatedly on each image of a group versus the other images until convergence is reached.

The pairwise representation of 3.4 can be arrived at by solving for the decoupled set of $\bar{\mathbf{T}}_\mathbf{j}$'s, $\{j = 1, \ldots, N\}$ that solve the equivalent optimization problem

$$\bar{\mathbf{T}}_\mathbf{n} = \underset{\mathbf{T_n}}{\operatorname{argmin}} (I_n(\mathbf{T_n}(\mathbf{x}_i)) - \mu_n(\mathbf{x}_i))^2, \tag{3.5}$$

where

$$\mu_n(\mathbf{x}_i) = \frac{1}{N-1} \sum_{\substack{j=1 \\ j \neq n}}^{N} I_n(\bar{\mathbf{T}}_\mathbf{n}(\mathbf{x}_i)) \tag{3.6}$$

The result in equation 3.5 is the pairwise registration between I_n and $\mu_n(\mathbf{x}_i)$.

Once Through Sweep

Finally, a further simplification is obtained in a recursive procedure known as the Once Through Sweep. This procedures begins by aligning two images. Next, the mean of the two aligned images is aligned with the next image in the set. This process is repeated until each image in the original set has been registered. In general, at each iteration the definition of the mean $\mu_n(\mathbf{x}_i)$ in Equation 3.6 is replaced by

$$\bar{\mu}_n(\mathbf{x}_i) = \sum_{j=1}^{n-1} \frac{I_j(\bar{\mathbf{T}}_\mathbf{j}(\mathbf{x}_i))}{n-1}. \tag{3.7}$$

This process can be run in the forward direction (starting with image 1 and ending with image N), in the reverse direction (starting with image N and concluding with image 1), or in any arbitrary order. The alignment of the n^{th} image in the sequence is found by solving for the $\mathbf{T_n}$ that minimizes

$$\bar{\mathbf{T}}_\mathbf{n} = \arg\min_{\mathbf{T_n}} \sum_{i=1}^{L} \left[I_n(\mathbf{T_n}(\mathbf{x}_i)) - \bar{\mu}_{n-1}(\mathbf{x}_i) \right]^2, \tag{3.8}$$

where $\bar{\mu}_n(\mathbf{x}_i)$ is calculated recursively using the relation

$$\bar{\mu}_n(\mathbf{x}_i) = \frac{(n-1)\bar{\mu}_{n-1}(\mathbf{x}_i) + I_n(\mathbf{T_n}(\mathbf{x}_i))}{n}. \tag{3.9}$$

This change simplifies the recursive algorithm in Equation 3.5 to an algorithm consisting of a single iteration. The benefits of the Once Through Sweep are discussed next.

Algorithm Discussion

By using the Once Through Sweep, the groupwise registration algorithm is M times faster than an algorithm based on Equation 3.4, where M is the number of iterations required for convergence of the algorithm. This greatly reduces the number of individual comparisons from N to one, where N is the total number of images to be aligned. This reduction translates directly to a factor of N reduction in the registration algorithm.

The pairwise registration algorithm used here is guaranteed to monotonically decrease the square difference between I_n and the mean image. This decrease in the error between the image pair results in a monotonic decrease in the overall error term as each image is added to the set of aligned images. Because the error term is both monotonically decreasing and bounded below by zero, the convergence to a local minima is guaranteed.

As noted in [75] and [113], there are multiple sets of $\mathbf{T_n}$'s that produce a minimum, so the set must be constrained by some metric. An extreme case for a minimum is shrinking each of the images to a single point. This outcome was avoided in [75] by constraining the mean of each parameter across all of $\mathbf{T_n}$'s to zero following an iteration in which the error is minimized with respect to each of the $\mathbf{T_n}$'s. Another example of this is shown in [113], where a normalization term was created to constrain the solution. Another choice for a metric, that is more appropriate for the Once Through Sweep, is to set $\mathbf{T_1}$ to the identity matrix. In general, if the set of transformations $\{\mathbf{T_1}, \mathbf{T_2}...\mathbf{T_N}\}$ results in an accurate registration so does the set $\{\mathbf{A} \cdot \mathbf{T_1}, \mathbf{A} \cdot \mathbf{T_2}...\mathbf{A} \cdot \mathbf{T_N}\}$, where $\mathbf{A}$ is any arbitrary non zero transformation.

To replicate the gradient descent portion of the congealing algorithm found in [75], the magnitude of $\mathbf{T_n}$ from the previous estimate is normalized to be less than a chosen value of $\delta\mathbf{T_n}$. This limits the transformation applied to image I_n to a perturbation in the direction of the alignment. This difference does not affect the convergence of the registration algorithm, although it will affect the speed and point of convergence. For more on the convergence of congealing in the context of the minimum entropy

criteria, see the work in [75]. Small step descent processes, such as congealing, are of interest in non quadratic similarity measures such as entropy. With the least squares similarity measure, the small steps are not necessary and each image can be brought into maximum alignment with the other images at each iteration.

A Representative Image for Joint Entropy Groupwise Registration

Following a direct application of the Chain Rule of Entropy [29], a similar formulation to the least squares metric can be written for the joint entropy of a set of N images as

$$\min_{\mathbf{T_1},...,\mathbf{T_N}} H(I_N(\mathbf{T_N}(\mathbf{x})), I_{N-1}(\mathbf{T_{N-1}}(\mathbf{x})), ..., I_1(\mathbf{T_1}(\mathbf{x}))) = \min_{\mathbf{T_1},...,\mathbf{T_N}} \sum_{j=1}^{N} H(I_j(\mathbf{T_j}(\mathbf{x})|I_{j-1}(\mathbf{T_{j-1}}(\mathbf{x}), .., I_1(\mathbf{T_1}(\mathbf{x}))). \quad (3.10)$$

Here, the conditional entropy H of a random variable X given random variable Y is defined as

$$H(X|Y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p_{X,Y}(x,y) \log p_{X|Y}(x|y) dxdy \quad (3.11)$$

$$= E\left[\log p_{X|Y}(x|y)\right], \quad (3.12)$$

where $p_{X|Y}(x|y)$ is the conditional probability distribution of random variables X conditioned on Y.

Instead of solving for all of the $\mathbf{T_i}$'s simultaneously, as done previously, they are now solved for sequentially. A general algorithm, that provides a direct analogy of the Once Through Sweep for joint entropy based registration, works by replacing the one versus all other images registration with two image registration using a representative image. The representative image $\mathscr{O}^*$ is found by minimizing the expression

$$\mathscr{O}_n^* = \arg\min_{\mathscr{O}} H(\mathscr{O}|\mathcal{I}_n), \quad (3.13)$$

at each step, where $\mathcal{I}_n = \{I_n(\bar{\mathbf{T}}_\mathbf{n}(\mathbf{x})), ..., I_1(\bar{\mathbf{T}}_\mathbf{1}(\mathbf{x}))\}$. Note that all registered images are used to calculate $\mathscr{O}_n^*$. This differs from the recursive process used in determining the mean as the representative image, shown in Equation 3.9.

One approach for determining an approximation of the representative image $\mathscr{O}_n^*$ is to first convert each of the images $\mathcal{I}_n$ into a common modality, and then calculate the mean image as done in Equation 3.9. The conversion is performed by estimating the functional relationship between the intensities of images under one modality with intensities of the reference modality and inverting. The particular modality is not important, although there is a computational benefit to using the modality common to the largest number of images. This strategy leads to a recursive calculation of $\mathscr{O}_n^*$ analogous to Equation 3.9. In other words, two images are first aligned toward one another using a joint entropy method, such as in [106, 117], and then fused. Next, a third image is registered to the fused representative image. The registered third image is then fused with the representative image and the process is repeated on additional images.

Other strategies might use different weights across location and modality. For example, a modality should be weighted more heavily in regions where it contains more detail than the other modalities. For example, in [46] the wavelet decomposition of two images are fused in a process that also minimizes the Rényi entropy of the resultant fused image.

Implementation Details

To find a local minimum solution for pairwise registration under the least squares error metric, this work uses an iterative algorithm that is similar to Newton's method known as Levenberg-Marquardt Algorithm (LMA) [1, 107, 109, 121].

A Gaussian pyramid approach is used to avoid getting trapped in local minima and to speed up convergence. This approach both smoothes and down-samples the images being aligned [13], which allows shifts that were relatively large to be reduced to sub-pixel motions that can be found directly.

A pixel mask restricts the use of pixels to within the area of the image outlined by

a stationary screen used in angiography. This mask removes the impact of the static screen on the registration of the moving patient.

In addition to registration based on the mask, the registration algorithm was validated on a region that does not contain contrast. This region contained fiducial markers, such as bones that contain sharp features, that might aid in registration. The error metric evaluated on pixels within the more selective region was used to measure the relative performance of the registration algorithm based on original mask to the registration algorithm based on the more selective mask.

3.2.1 Results

Image Comparison. An example image from the 2D time series is shown in Figure 3-3. The standard deviation, a metric that is equivalent to the mean square error, of each pixel stack before and after registration is shown in Figure 3-4. Clear differences can be seen between the two panels in the areas corresponding to the skull in the upper left and upper right and the eye sockets in the lower right (see arrows). The effect of the mask is also apparent in the registered image stack. This affect was ignored in the registration process. To give a flavor for the quality of registration, results are included that show background subtraction for Frame 5 minus Frame 1, both before and after the Once Through Sweep algorithm is applied, in Figure 3-5 in gray scale and Figure 3-6 in color. The lower right portion of both images in Figure 3-6 are shown in Figure 3-7. The removal of the thick artifacts due to the misalignment in the two images are shown in the figure. The remaining differences can be attributed to either errors in the registration or observed out of plain rotations

The Root Mean Square Error. A useful metric for measuring the performance of the Once Through Sweep, shown in Figure 3-2, is the square root of the right hand side of Equation 3.3, known as the error term ε. The first line of Table 3.1 shows the value of ε for the following methods: no registration, registration of each image to image 1, the sweep in the forward direction, and the sweep in the backward direction. The registration versus Frame 1 provides a modest increase in ε of approximatley

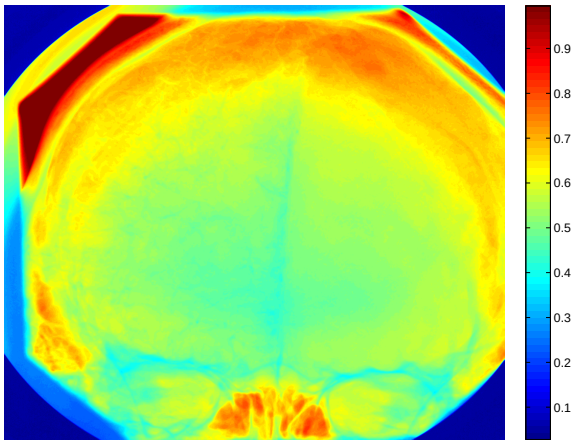


Figure 3-3: **A typical frame.** This figure shows a typical frame. In this case it is Frame 8 without having subtracted the background image. Notice that there is a mask region in the background.

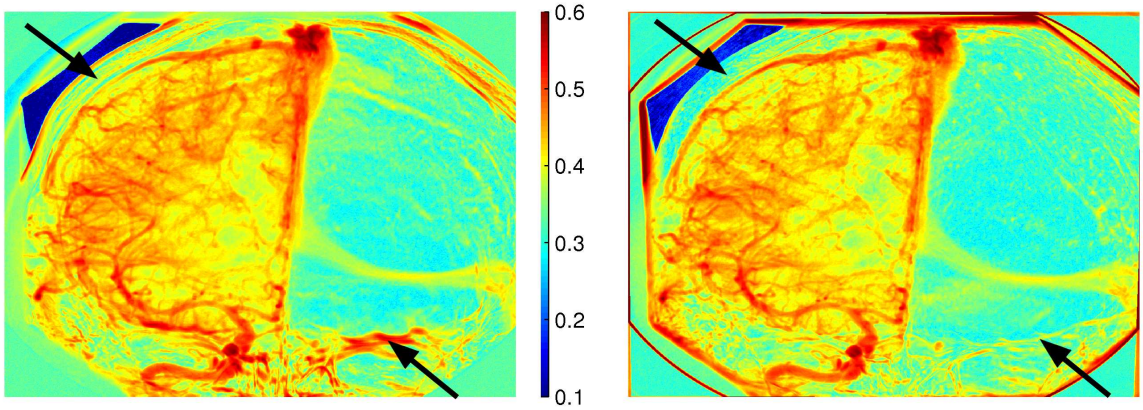


Figure 3-4: **RMSE visualization.** The figure shows a contrast enhanced image of the standard deviation of each pixel across all of the images for the unregistered (left) and registered stack (right). Arrows have been added to highlight two areas where differences due to the registration can be observed.

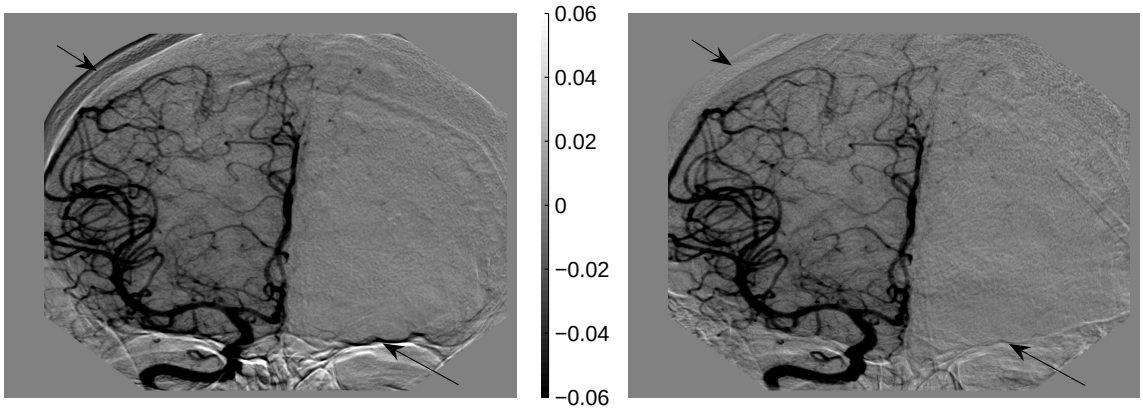


Figure 3-5: **Unaligned and aligned images - grayscale.** Background subtraction of digital angiography images frame 5 - frame 1, shown in grayscale (a format familiar to clinicians). The left panel shows the difference between the unaligned frame and the right panel shows the difference with the registered frame (using Once Through Sweep algorithm). Arrows have been added to highlight two areas where differences due to the registration can be observed.

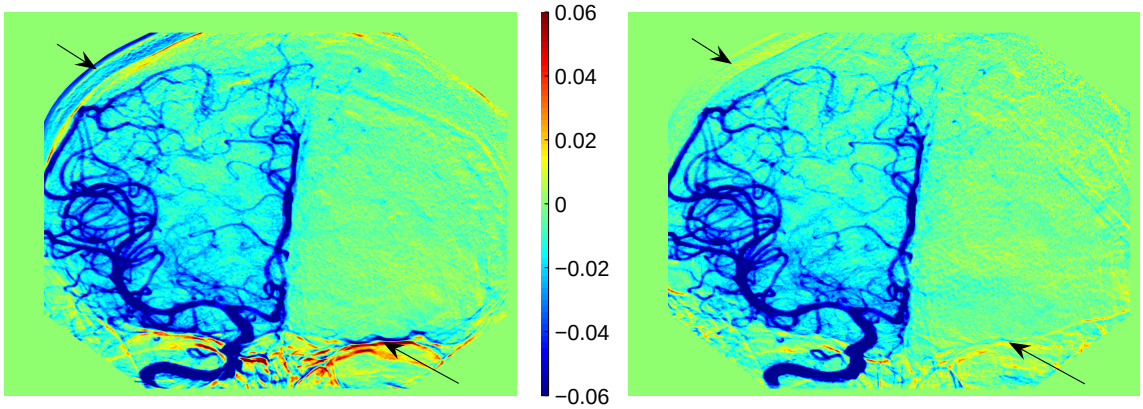


Figure 3-6: **Unaligned and aligned images - color.** Background subtraction of digital angiography images frame 5 - frame 1. The left panel shows the difference between the unaligned frame and the right panel shows the difference with the registered frame (using the Once Through Sweep algorithm). Arrows have been added to highlight two areas where differences due to the registration can be observed.

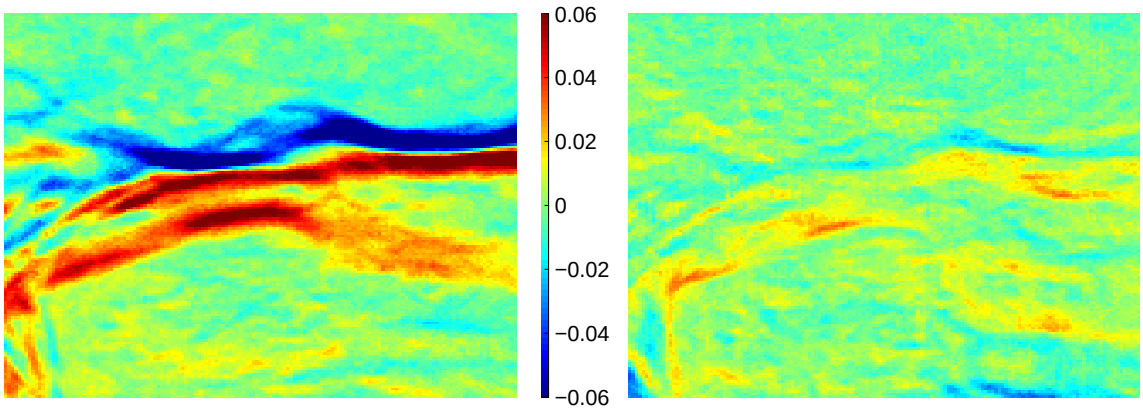


Figure 3-7: **A zoomed in view of Figure 3-6.** A zoomed in view of the eye socket in the bottom right corner of Figure 3-6 showing the background subtraction of Digital Angiography Images Frame 5 - Frame 1. The left panel shows the difference between the unaligned frame and the right panel shows the difference with the registered frame (using a sweep in the forward direction).

Total RMSE				
Method	No Reg.	Reg. vs Frame 1	Forward	Backward
Total Area	0.0213	0.0215 (-1%)	0.0193 (9%)	0.0193 (9%)
Unchanged Area	0.0181	0.0186 (-3%)	0.0114 (37%)	0.0112 (38%)

Table 3.1: The RMSE of the total error ε between each pair of the 16 images as specified in Equation 3.3. The first row shows these errors for multiple methods where the error is calculated over the entire mask region, while the second row shows the errors where only the pixels within the more selective region are used. The methods considered are: no registration (baseline), registration of each to the first image, the forward groupwise algorithm, and the backward groupwise algorithm.

1%, while both the forward and backward methods reduce ε by approximately 10%. The second line shows the results when the error is calculated using pixels from the more restrictive region as later described. Within this restrictive region, the pairwise technique still performs worse than no registration at all while the two groupwise methods provide a reduction in ε of approximately 38%.

To observe what is happening with the value of ε, the RMSE is calculated between each of the registered frames and the first. Table 3.2 shows this for unaligned and for the forward and the backward algorithms. Note that in both the unregistered and registered images, the RMSE is greatest around the middle frames and decreases toward the two ends. There is also a significant contrast present in the middle frames

while there is little or no contrast at the ends.

To eliminate any impact of the contrast on the RMSE between the two images, a restrictive region that contains little or no contrast and visible fiducial markers is used for the comparison. Table 3.3 shows the RMSE for these different measurements. This table shows the same trend as in Table 3.2, the largest difference between the first image and each of the other images occurred in the middle, although there is a larger reduction in the RMSE than shown in Table 3.2. This suggests that the contrast had a meager impact on the quality of the registration.

Additional iterations of aligning each of the images versus all other images following the forward Once Through Sweep algorithm did not reduce the RMSE. A variation of the congealing method was performed by registering each image separately versus the mean of the other images while holding the first image fixed. Each iteration thereafter uses the new shifted version of the registered images. This algorithm resulted in an identical RMSE as was found with the Forward and Backward applications of the Once Through Sweep.

The least squares algorithm presented here was tested on ten other time series and on data from other patients. The detailed results given above are qualitatively similar across all the other data sets. In general, the RMSE was lowest for the Once Through Sweep algorithm, somewhere in between for the alignment to a single background image, and largest for no alignment. The presence of large well-defined vasculature structures improved the relative performance of the Once Through Sweep with respect to the performance of the alignment to a background image.

3.2.2 Discussion

Image Comparison. Although the registration algorithm visually improves the registration in the zoomed images in Figure 3-7, there still remain noticeable defects. These defects can be attributed to the either observed out of plane rotations or deficiencies of the registration algorithm. To deal with this, rotations models could be developed to allow compensation of the out of plane rotations. The algorithm might be improved by leveraging an alternative algorithm such as one that uses deformable

RMSE Between Each Image and Image 1			
Frame	Original	Forward	Backward
2	0.0035	0.0032	0.0033
3	0.0118	0.0102	0.0102
4	0.0195	0.0172	0.0171
5	0.0243	0.0225	0.0225
6	0.0277	0.0256	0.0256
7	0.0308	0.0267	0.0266
8	0.0341	0.0284	0.0284
9	0.0321	0.0276	0.0276
10	0.0317	0.0280	0.0280
11	0.0296	0.0282	0.0281
12	0.0256	0.0239	0.0239
13	0.0204	0.0174	0.0175
14	0.0130	0.0116	0.0117
15	0.0125	0.0085	0.0086
16	0.0108	0.0069	0.0070

Table 3.2: The RMSE between the particular frame and the first frame in the unaligned, and Forward and Backward algorithms. Only pixels that are within the masked region are considered.

RMSE Between The Two Images			
Frame	Original	Forward	Backward
2	0.0047	0.0043	0.0044
3	0.0165	00119	0.0116
4	0.0244	0.0174	0.0172
5	0.0219	0.0163	0.0165
6	0.0232	0.0169	0.0173
7	0.0285	0.0162	0.0162
8	0.0354	0.0197	0.0196
9	0.0298	0.0161	0.0160
10	0.0273	0.0146	0.0144
11	0.0190	0.0131	0.0126
12	0.0203	0.0131	0.0129
13	00217	0.0117	0.0118
14	0.0126	0.0093	0.0097
15	0.0160	0.0080	0.0082
16	0.0147	0.0074	0.0075

Table 3.3: The RMSE between the particular frame and the first frame in the unaligned, and Forward and Backward algorithms. Only pixels that are within the more selective region where no contrast is present are considered.

models, or the correspondences of identified fiducial markers.

Root Mean Square Error. The superior results for the groupwise algorithms shown in Table 3.1 suggest that the groupwise algorithms are preferable for our application. In section 3.2.1 it was shown that the addition of iterations of the one versus all of the other images algorithm did not reduce the RMSE. This suggests that the Once Through Sweep algorithm is a good approximation to the more computationally expensive methods.

In Tables 3.2 and 3.3, a peak was observed in the central frames that was not due to the presence of the contrast. This peak suggests that there are larger errors in the registration of the middle frames. This error may be due to the presence of more contrast in the center frames that may have a negative effect on the quality of the registration. The presence of the contrast causes the set of images to be more different then it would otherwise be. This difference can degrade the quality of the overall registration. Additionally, larger motions were observed in these frames, which might include out of plane rotations that would change the underlying structure of the X-ray and have a negative impact on the registration. These observations suggest that a higher quality registration would be obtained with an algorithm that leaves out the negative impact of the contrast and accounts for out of plane motions.

Additional Testing. Additional testing performed on ten image sequences verified the improved performance of the algorithm, but highlighted one potential weakness of the algorithm. Because of the recursive dependence of the representative image, any single image misalignment can lead to poor alignments of later images. During these tests, an image to be registered was inadvertently left unregistered and caused the algorithm to break down. To eliminate outlier images with poor registration, an alternative representative image such as the median image might be used. However, he median image is not used here both because it does not have a recursive formulation and because it does not reduce the noise across a set of images as well as the mean. This sensitivity to the individual registrations suggests that particular care should be taken in registering each image with the representative image.

3.3 3D Segmentation

To fuse the two data sets, the 3D volume and the 2D time series, the location of the blood vessels needs to be extracted from the 3D volume. This process results in a segmentation of the volume or simply a 3D binary map that contains a 1 everywhere contrast is present and a 0 everywhere else. A great deal of literature exists on such algorithms that use smoothing followed by thresholding [53, 94] and variational methods [59, 102, 113]. Because the volumes used in the study may be as large as 512^3 voxels, any algorithm needs to be computational simple to be practical. The algorithm used in this thesis is that of a simple threshold. This simple approach yielded good results. Once this map is generated it can be used in 2D-3D registration, in addition to the fusion algorithm that is later discussed. The segmentation of $V(\mathbf{x})$ and $V[\mathbf{x}_p]$ results in the maps $\mathscr{M}(\mathbf{x})$ and $\mathscr{M}[\mathbf{x}_p]$ respectively.

3.4 2D-3D Registration

The final hurdle in preparing the data is to find the relative orientation of the 2D time series in relation to the 3D volumes. Because the time between the acquisition of the 3D volume and the time series is typically much larger (on the order of minutes) than between successive frames in the 2D time series (on the order of tenths of seconds), there is a greater likelihood of patient motion. The relative orientation can be found by finding the projection of the volume, known as a digital reconstructed radiograph or DRR, that is most similar to a representative image of the registered time series. The section presents a new algorithm that projects a triangular mesh representation of the blood vessel structure contained within the 3D volume.

3.4.1 Digital reconstructed radiograph (DRR) generation

There are several algorithms for generating DRR's in the literature [19, 38, 48, 95, 119, 122]. Determining the DRRs can be a computationally intensive task, so an algorithm that simplifies this task is preferred. There are various speed ups that

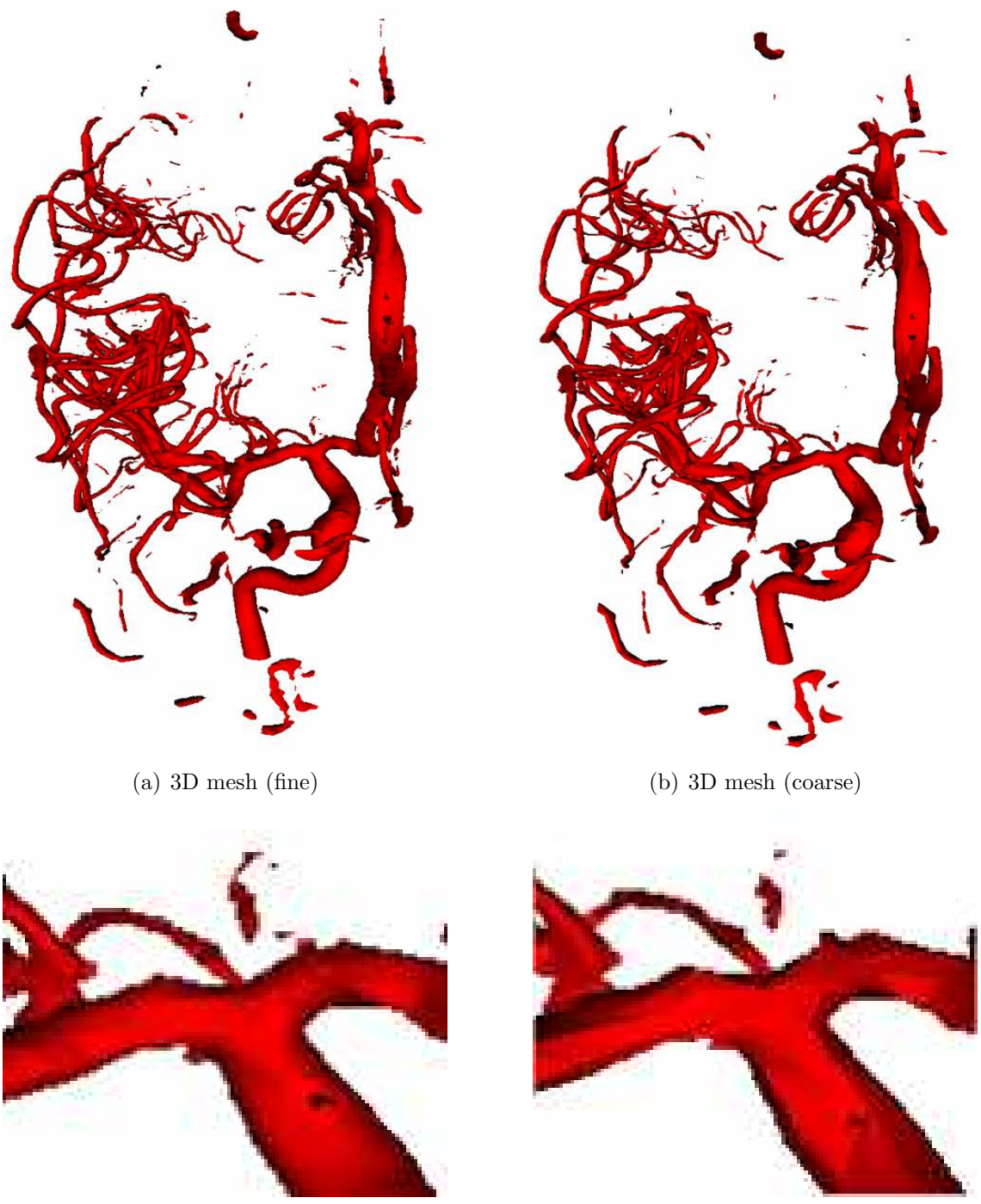
allow computation to be reduced, such as only calculating a small subset of the rays in the DRR [122], transforming the 3D volume into another space [95, 119], projecting a binary image or vessel probabilities of the 3D volume [48], and projecting spheres on the skelatonized segmented volume [19, 38]. The resultant DRR images can be either grayscale or binary images as in [19]. Each of these, with the exception of [122] requires a significant amount of computation to transform the data into a more convenient form for the generation of the DRR. In this thesis the new form is a triangular mesh placed on an isosurface in the 3D volume.

Generation of DRR's from a mesh

The new 2D-3D registration algorithm presented here builds upon many meshing algorithms to produce a fast $O(N^2)$ method of calculating DRR's from any arbitrary pose. The mesh-based algorithm is most similar to [19, 38], where a skeleton like 3D structure is projected. In this work the triangles within the mesh are instead projected. These meshes are already calculated in many tomography settings and will not require specialized codes such as in [67, 108] to generate a skeleton.

The algorithm can be further sped up by using a mesh reduction algorithm that will reduce the number of triangles in the mesh. Two possible mesh reduction algorithms are [40] and the Matlab function reducepatch [72]. This work used the Matlab function reducepatch [72] to decrease the number of triangles in the mesh. Large reductions in the overall number of triangles can be achieved without significantly affecting the qualitative shape of the mesh. In contrast, decreasing the number of spheres in the other algorithm will degrade the approximate shape more quickly. In other words, the triangular mesh provides a more efficient representation of the surface than do the spheres. Specifically, reducing the number of triangles by 90% resulted in a factor of 6 speedup. Figure 3-8 shows a sample mesh and the effect of the reduction in the number of triangles on the overall shape.

The DRR is generated by first producing a mesh V_{MESH} using the Marching Cubes algorithm [70] on the segmented data. This creates a collection of points that combine to form a surface that is inherently sparse in size compared to the 3D volume. Each



(a) 3D mesh (fine)

(b) 3D mesh (coarse)

(c) zoomed in view (fine)

(d) zoomed in view (coarse)

Figure 3-8: **Mesh decimation** – The left panels show the original mesh containing 205,000 triangles derived from the 3D volume. The right panels show the impact of reducing the number of triangles to 21,000.

point in the mesh can then be projected to form $I_{DRR}(\mathbf{x}_p; S_{MESH}; \mathbf{T3})$, which is the DRR at point $\mathbf{x}_p \in \mathbb{R}^2$ of the mesh S_{MESH} after the transformation $\mathbf{T3} : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ has been applied. After the mesh is projected into the imaging plane, the pixels locations within each triangle is set to one. Figure 3-9 shows a DRR generated by this process. Figure 3-10 shows the residual between the projection before and after decimation. A factor of ten reduction in the mesh size results in only a 1% change in the pixels within the DRR.

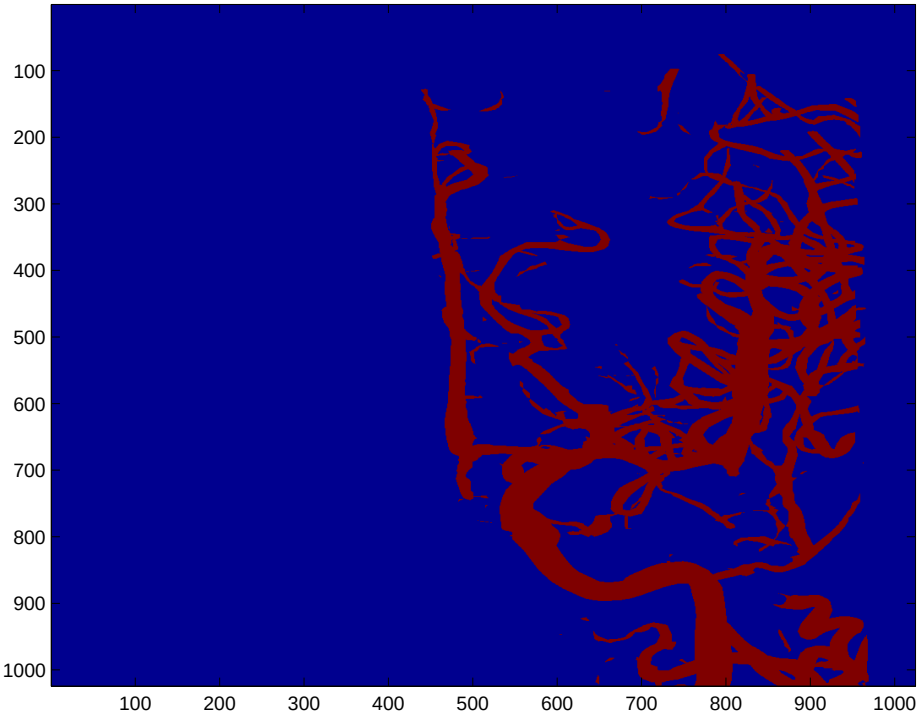


Figure 3-9: **DRR of 3D mesh** – The digital reconstructed radiograph of the 3D mesh. Each triangle in the mesh is transformed into the 2D imaging plane where the interior parts of the triangle are filled in with ones.

3.4.2 Generation of vasculature image from 2D time series

The vasculature image is determined by subtracting the minimum image across the arterial cycle from the maximum image. This process can be written as

$$I_{vasc}(\mathbf{x}_p) = \max_n I(\mathbf{x}_p, n) - \min_n I(\mathbf{x}_p, n), \tag{3.14}$$

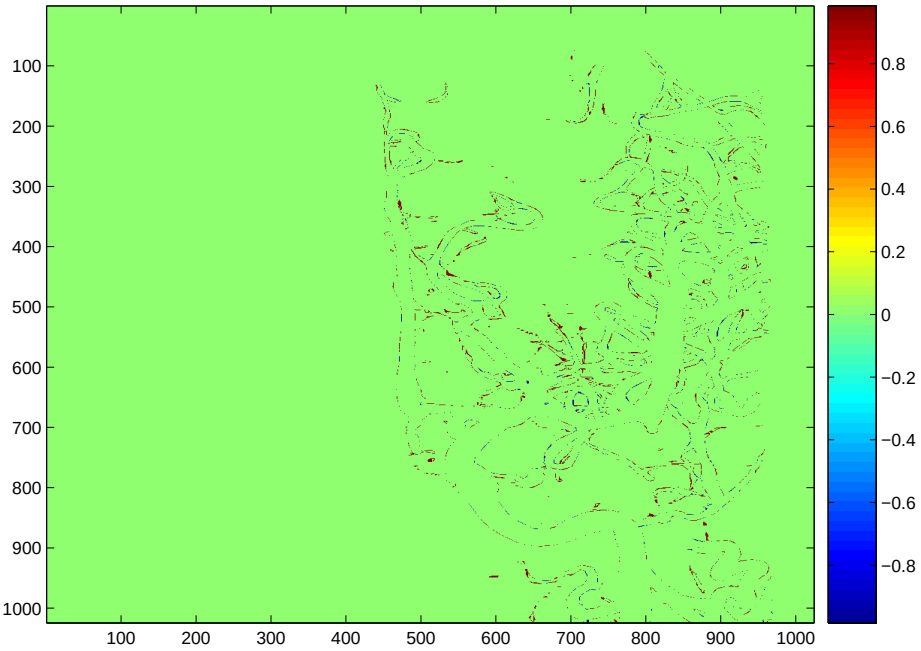


Figure 3-10: **Difference between fine and coarse mesh** – The residual between the projections of the fine and coarse meshes. Fewer than one percent of the pixels have been changed.

where n ranges over the times corresponding to when contrast was in the arterial cycle.

This captures the largest effect of the contrast at each point thereby accentuating the location of the bloodvessels.

3.4.3 Search for best transformation

The meshed volume S_{MESH} is a collection of vertices $\mathbf{v_i} = (x_i, y_i, z_i)^T$ representing points in space and triangles $Triag_j = \{j_1, j_2, j_3\}$ containing triplets of vertex indices. The transformation **T3** is applied to the list of vertices only, since it does not affect the connectedness of the individual triangles. The transformation **T3** is a function of three rotations about the x, y, and z axes in the volume reference frame; three shifts in the $\hat{x}$, $\hat{y}$, and $\hat{z}$ directions in the volume reference frame; and two shifts in the 2D image reference frame. A general expression for the transformation **T3** is

$$\mathbf{x}_p = \mathcal{P}(\boldsymbol{R}\mathbf{v_i} + d\mathbf{v}) + d\mathbf{x}, \tag{3.15}$$

where $\mathcal{P}$ represents the projection of the 3D coordinates into the 2D imaging plane, $\boldsymbol{R}$ is a rotation matrix, $d\mathbf{v}$ is the shift in the volume reference frame, and $d\mathbf{x}$ is the shift in the imaging reference frame. To apply the function $\mathcal{P}$ on the shifted and rotated vertices it is necessary to know both the geometries of the conebeam [37] setup and the spatial resolutions of the 3D map. This information completes the set of information used to construct a DRR.

The registration process can be described as the search for

$$\overline{\mathbf{T3}} = \arg\max_{\mathbf{T3}} \rho(I_{DRR}[\mathbf{x}_p; S_{MESH}; \mathbf{T3}], I_{vasc}[\mathbf{x}_p]), \tag{3.16}$$

where ρ is the correlation

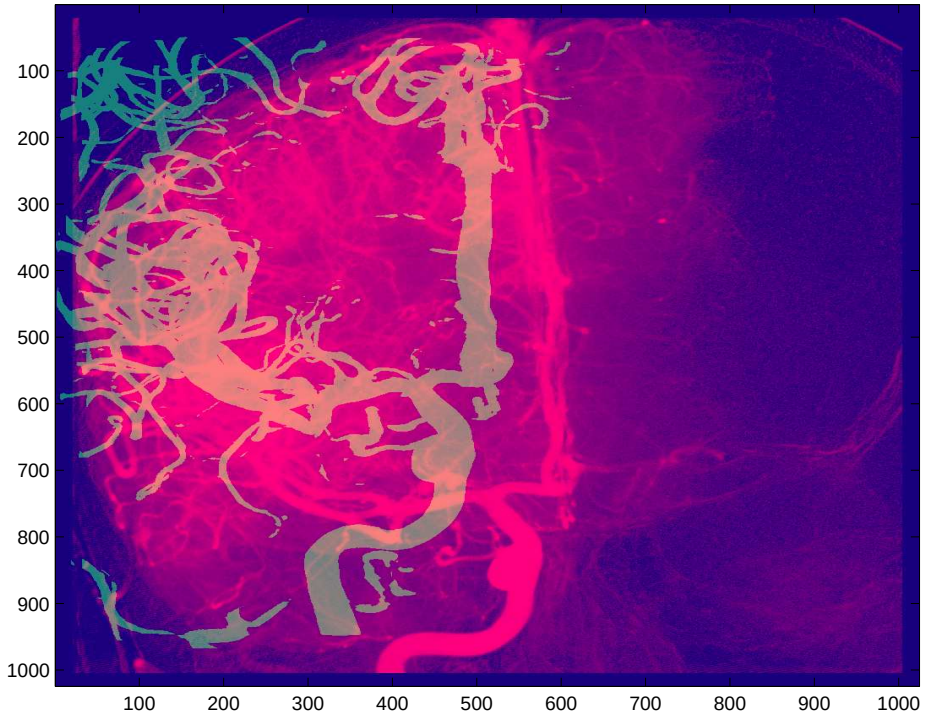
$$\rho(I_1, I_2) = \frac{\sum_i (I_1(\mathbf{x}_i) - \mu_{I_1})(I_2(\mathbf{x}_i) - \mu_{I_2})}{\sigma_{I_1}\sigma_{I_2}} \tag{3.17}$$

between the two images. The variables μ and σ are the mean and variance respectively of the subscripted image, in this case it is either I_1 or I_2. The binary DRR can be thought of as a labeling or segmentation of the vasculature image. By varying **T3**, one varies the labeling. The correlation metric provides a good measure of the quality of the segmentation.

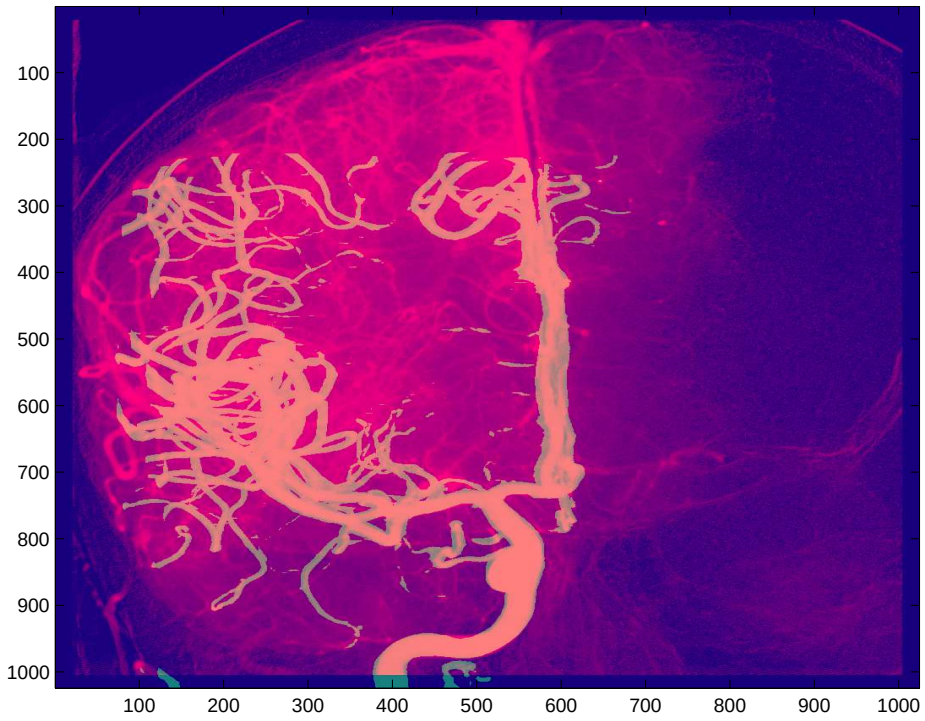
The search is initialized by performing a search in the range of variables centered at those set in the DICOM header information. The eight variables are solved for separately using Brent's method [11] as implemented in the Matlab function fminbnd [72]. The process repeats by looping through each parameter, finding the best choice, updating, then continuing. The results of the algorithm are shown in Figure 3-11 and 3-12. The results show that the algorithm greatly improves the match between the DRR and the time series.

3.4.4 Discussion

The 2D-3D registration assumes that the similarity between projections of the 3D blood vessel surface and the representative 2D image. This similarity is strong in the

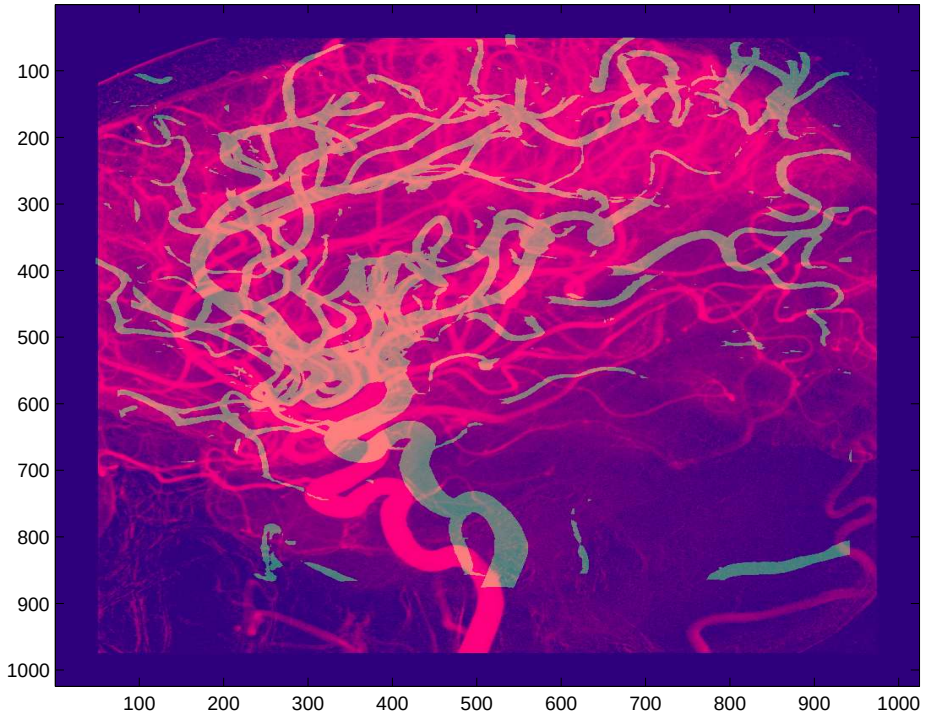


(a) Unregistered

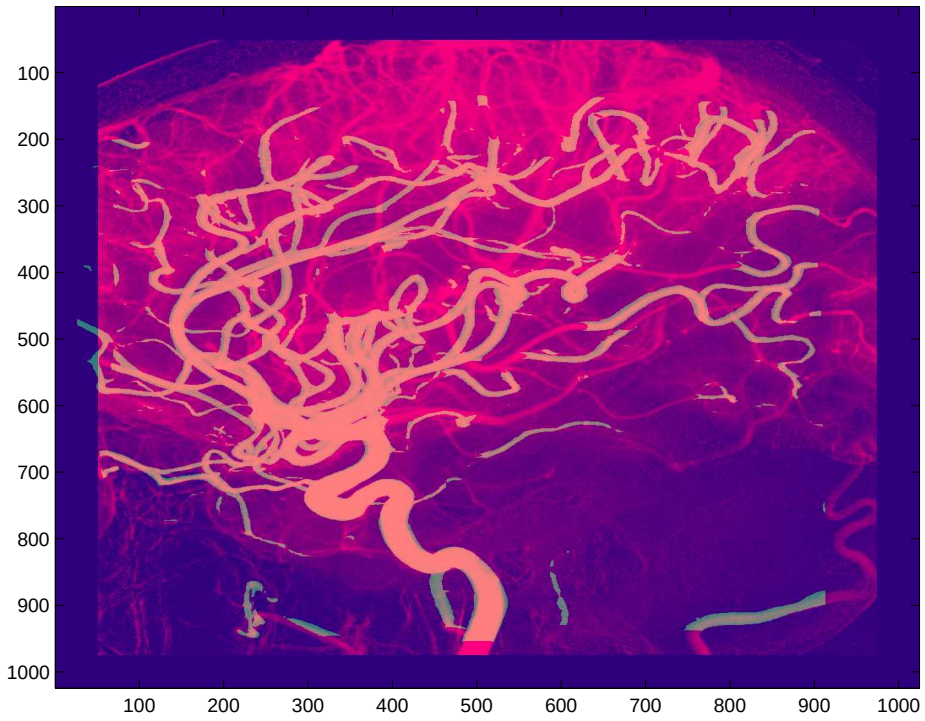


(b) Registered

Figure 3-11: **Before and after 2D-3D registration (front view)** – Overlay of unregistered (top) and registered (bottom) DRR in green over the representative image of the time series in pink. When the two are aligned the result is peach in color.



(a) Unregistered



(b) Registered

Figure 3-12: **Before and after 2D-3D registration-(side view)** – Overlay of unregistered (top) and registered (bottom) DRR in green over the representative image of the time series in pink. When the two are aligned the result is peach in color.

angiography case because contrast within the blood vessel is relatively uniform and the representative image isolates contrast within the 2D sequence.

Another assumption for the 2D-3D registration is that the shape of the vasculature needs to stay the same. An example that violates this assumption is aligning a map from one patient with a 2D sequence from another patient. In most cases, this produces incorrect registration. If, however, the morphologies of the two patients are similar, a reasonable registration without high accuracy can be achieved. To perform registration, the algorithm relies on the orientation and signal information in the vessels. An example of orientation information is a single branch on an otherwise straight vessel. This branch allows the algorithm to find the proper rotation of the vessel. If a blockage restricts the blood flow to that branch, the orientation information is lost and the algorithm is unable to resolve the proper rotation of the vessel. If instead the vessel has many branches, the blockage of one would not have much impact on the registration because the other branches provide sufficient orientation information.

This section provides a simple yet fast algorithm for calculating digital reconstructed radiographs of a 3D map from an arbitrary pose. The fast speed of the algorithm comes from the information reduction obtained in the mesh extraction. The mesh process provides accurate results because it is able to capture features of the sparse 3D blood vessels corresponding to those in the 2D time series.

Chapter 4

Finding 2D Arrival Times

This chapter presents a method for finding the arrival time of contrast within the vasculature of the brain at various points within a sequence of 2D X-ray angiograms. Determining the arrival times of the contrasts collapses the time series information down to a single image containing the arrival time at each pixel. The 2D arrival time information provides a visual aid that helps surgeons assess irregularities in the blood flows caused by clots or other constrictions. This aid allows surgeons to more easily observe delays in contrasts, enabling them to quickly diagnose the presence of an anomaly. This information can be presented through "color coded images which indicate the arrival time of the contrast medium" [51]. The work in this Chapter provides a complementary tool for surgeons to the spatio-temporal data fusion algorithm presented in the next chapter.

This thesis uses a modified version of the Cusum algorithm [118] as developed in the context of failure detection. The Cusum-based algorithm provides a more accurate set of arrival times that does not rely on either a priori knowledge of the shape of the time series at each location, as was done in [50, 51, 96, 120], or on the time series at each location having the same shape under translation, as was done in [99]. The Cusum algorithm is used because it is robust to the wide variations that exist in the shape of the time series across different locations. In general, the time series consists of a flat region followed by a rapid drop in intensity, corresponding to when the contrast first arrives, that subsequently flattens for a brief time before

slowly returning to its initial intensity. The variations in shape are due to both the existence of a superposition of a complicated network of overlapping blood vessels and the large range of blood vessel diameters.

To make the Cusum algorithm capable of detecting contrast arrival times it is modified in two new ways. First, it is modified by adding a second threshold that allows the algorithm to find a more precise arrival time without introducing a high false alarm rate. The second modification is the inclusion of a noise estimate that depends on intensity. This approach more accurately captures the characteristics of the CCD noise present in the angiographic time sequence. This intensity based noise level is similar to adaptive approaches in [25, 34].

This chapter begins with an overview of the Cusum-based algorithm in Section 4.1. Results of the proposed algorithm are compared to those of the correlation-based algorithm in [99] using both an angiography time series and synthetic data in Section 4.2. Additional results are also included that show the arrival times for a patient both before and after the development of a clot. Finally, the chapter concludes with a discussion of the Cusum-based algorithm in Section 4.3.

4.1 Determining Arrival Times

4.1.1 Overview

A general method that looks for prolonged changes in the intensity at each pixel location due to contrast is desired for determining the arrival time of contrast at each point in the 2D time series. By only looking for a prolonged change, e.g., a drop in intensity below the mean that lasts for several frames, a method that is robust to the various possible time series is provided. An appropriate choice for this method is an adaptive variation of the Cusum algorithm [118]. The Cusum algorithm can be interpreted as a special application of the repeated Sequential Probability Ratio Test or SPRT as shown in [105]. The Cusum algorithm works by thresholding an indicator

function $h_n[\mathbf{x}_i]$

$$h_n[\mathbf{x}_i] = \left[h_{n-1}[\mathbf{x}_i] - \frac{I_n[\bar{\mathbf{T}}_\mathbf{n}(\mathbf{x}_i)] - \mu_{n-1}[\mathbf{x}_i]}{q(\mu_{n-1}[\mathbf{x}_i])} - \tau \right]^+, \tag{4.1}$$

where $I_n[\bar{\mathbf{T}}_\mathbf{n}(\mathbf{x}_i)]$ is the registered time series described in Section 3.2,

$$\bar{\mu}_n(\mathbf{x}_i) = \frac{(n-1)\bar{\mu}_{n-1}(\mathbf{x}_i) + I_n(\mathbf{T}_\mathbf{n}(\mathbf{x}_i))}{n}. \tag{4.2}$$

as in Equation 3.9, τ is a threshold, $h_0[\mathbf{x}_i]$ is zero, $[\cdot]^+ = \max(\cdot, 0)$, and q is a function representing the noise level at a point. The addition of the intensity based noise level to the Cusum algorithm makes the resultant algorithm adaptive. This adaptive noise level is similar to both the addition of an adaptive worst case error term to the repeated SPRT test in [34] and to another approach presented in [25]. The expression inside the one-sided test is the previous indicator function plus the number of standard deviations that the current image at location $\mathbf{x}_i$ is below the cumulative mean $\mu_{n-1}[\mathbf{x}_i]$ minus a threshold τ. To remove outliers during the generation of the indicator function, a median filter [69, 92] is applied to the image defined by the fraction in Equation 4.1 at each time.

4.1.2 Estimating the noise level

As is typical with CCD images, the X-ray noise level depends linearly on the intensity of the detected value [47]. The affine model

$$q(v) = v \cdot a + b, \tag{4.3}$$

where q is the estimate of the standard deviation of a pixel based on the scalar intensity v at that pixel, provides a reasonable fit to the observed noise levels and is used here.

The first step in fitting the noise is calculating the coefficients a and b. To do this, the noise levels in the image are determined by generating the joint histogram

between the first two registered images. These images are chosen so that the image to image variations are due to noise in the image acquisition and not due to the presence of contrast or patient motions. The joint histogram is used to determine the conditional PDF $P_{I_2|I_1}(I_2[\bar{\mathbf{T}}_\mathbf{2}(\mathbf{x}_i)] = v_2|I_1[\bar{\mathbf{T}}_\mathbf{1}(\mathbf{x}_i)] = v_1)$. The conditional standard deviation $\sigma(I_2|I_1 = v_1)$ for each v_1, which will be denoted by the shorthand $\sigma(v_1)$, is determined from these conditional probabilities. Then, a least square affine fit is performed to find the best estimator of $\sigma(v_1)$, $q(v_1) = a \cdot v_1 + b$. The fit is performed by using a weighting on the variance of each term based on the total number of image points with a particular intensity. This fit is equivalent to the minimization of

$$\{a^*, b^*\} = \underset{a,b}{\operatorname{argmin}} \sum_{v_1 \in Range(I_1)} (\sigma(v_1) - av_1 - b)^2 P_{I_1}(v_1). \tag{4.4}$$

4.1.3 Determining arrival times

The indicator function in Equation 4.1 is thresholded to determine the contrast arrival time. Using a high threshold avoids many false alarms, but also makes it difficult to detect when the contrast first arrives. After the contrast is detected, when the higher threshold is triggered, the algorithm steps back in time until it reaches the time immediately after it crossed a second lower threshold. The addition of the lower threshold is unique to the algorithm presented in this thesis. The time the intensity goes above the lower threshold is the arrival time assigned to that point in the 2D image.

The Cusum algorithm detects that contrast is present at time n_f when the indicator function h_{n_f} first exceeds the threshold τ_f. The Cusum algorithm is then modified in this thesis with the second threshold to find the arrival time n_a at each location using the equation

$$n_a = \underset{n \in [1, n_f - 1]}{\operatorname{argmax}} \; h_{n-1} < \tau_a, \tag{4.5}$$

for some threshold τ_a. This final step provides the desired 2D set of arrival times $n_a[\mathbf{x}_i]$.

4.2 Results

4.2.1 Angiographic Sequence

Figure 4-1 shows the arrival time estimates found using Equations 4.1-4.5 on a sequence of 105 images, where contrast was injected into the vertebral artery. Similar results are shown in Figure 4-2 for one of the carotid arteries. These results are for the values of the thresholds τ, τ_f, and τ_a shown in the figure captions. The values of the thresholds can be tuned to increase the sensitivity of the algorithm or to avoid assigning arrival times to locations where no contrast appears.

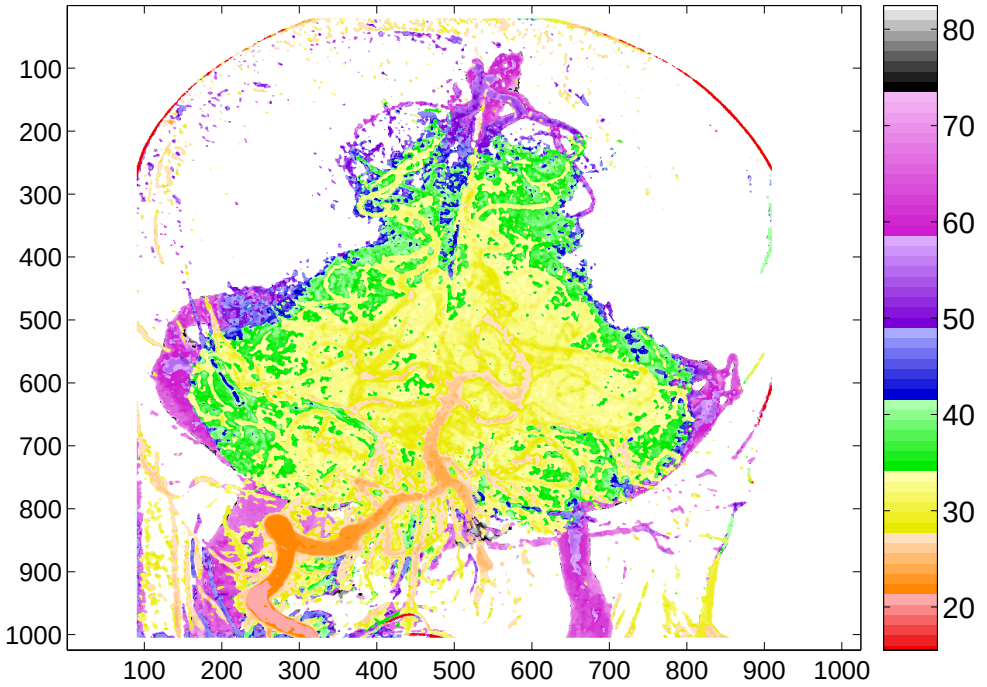


Figure 4-1: **2D arrival times** – Arrival times measured by the Cusum-based algorithm. Contrast was injected into the vertebral artery. For this figure $\tau_f = 2$, $\tau_a = 0.1$, and $\tau = 1$. The color bar is used to show the arrival times (frame number) of the contrast.

Figure 4-3(a) plots the time series at four separate locations (the blue and green solid lines show time series with pronounced arrival times, while the cyan and red solid lines show time series where the arrival of the contrast was more subtle). This figure also shows the recursively calculated mean $\mu_{n-1}[\mathbf{x}_i]$ used in Equation 4.1 using dotted lines. The blue and green time series are clearly not simply shifted and scaled versions

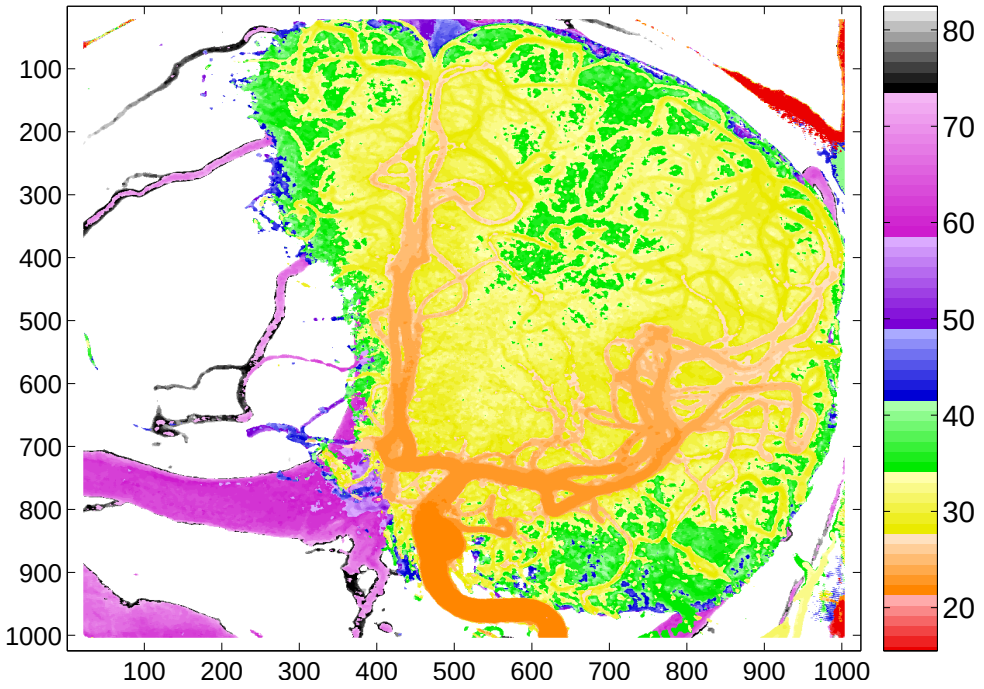
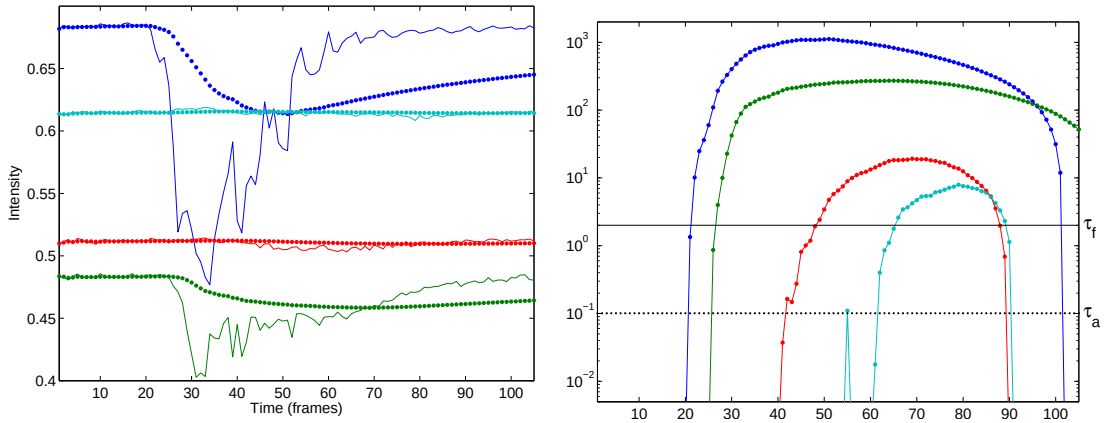


Figure 4-2: **2D arrival times** – Arrival times measured by the Cusum-based algorithm. Contrast was injected into the carotid artery. For this figure $\tau_f = 2$, $\tau_a = 0.1$, and $\tau = 1$. The color bar is used to show the arrival times (frame number) of the contrast.

of one another as earlier algorithms assumed [50, 51, 96, 99, 120]. This suggests that the proposed Cusum-based method is more appropriate because it does not make this assumption. If the image values drop below the recursively calculated means, the indicator function in Equation 4.1 will be non-zero as shown in Figure 4-3(b). The indicator function for the blue and green time series crosses the two thresholds more quickly than the function for the cyan and red plots because of the smaller change in the image intensities. It is expected that the algorithm is less accurate on the weaker signals because the change is less pronounced.

The Cusum-based arrival time algorithm finds the correct contrast arrival times for the blue and the green time series to be 21 and 26 respectively. This was verified by visual inspection. Correct results could not be obtained using the algorithm in [99], even when modified so that the first and lowest values of the blue time series were forced to be the same as that of the green time series. This modified algorithm resulted in a relative shift of 25 frames that was far from the correct relative shift of



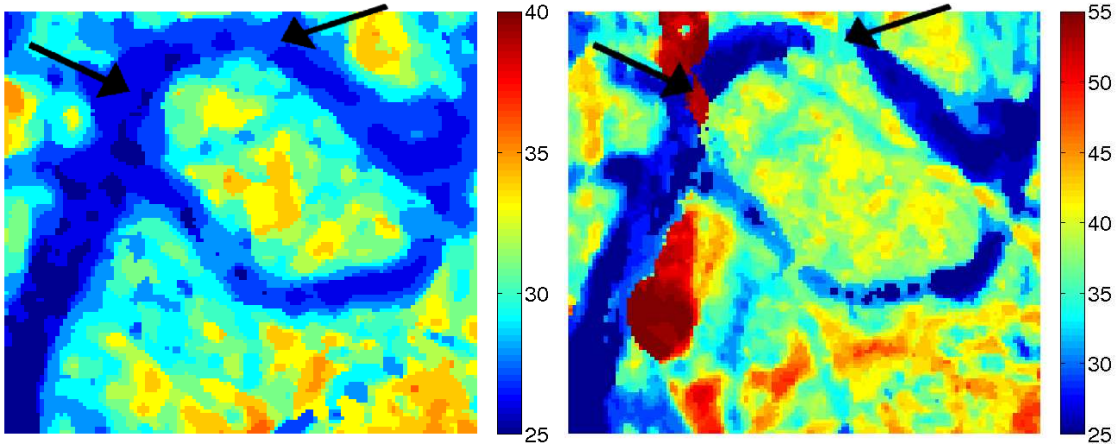
(a) Image time series (b) Indicator function of time Series

Figure 4-3: **Time series and indicator function.** – The blue, green, cyan, and red lines are from data located at the points (253,934), (578,620), (300,460), and (454,155) in Figure 4-1 respectively. Intensity plots are shown in the left panel for the 4 locations. A dotted line for each plot shows the recursively calculated mean $\mu_{n-1}[\mathbf{x}_i]$ used in Equation 3.9. The indicator function in Equation 4.1 for the four locations are shown in the right panel. The solid and dotted black lines show the location of thresholds τ_f and τ_a, respectively.

5 frames as measured by the Cusum-based algorithm. This difference is due to the green time series taking longer to return to normal than the blue time series. The higher accuracy of the Cusum-based method, which makes no assumption about the waveform, suggests that it is the best approach for this application.

Figure 4-4 shows a zoomed in version of the arrival times for both of the Cusum-based and for the correlation-based methods. The zoomed in version is shown here because the differences were not as visible while using the larger spatial and temporal scales of the un-zoomed version. The regions with similar arrival times tend to be well ordered along the vessels, although there are clearly some areas where the estimates of the contrast arrival times are out of order. The results for the Cusum-based algorithm show fewer breaks along the blood vessels. In addition, the correlation method picked up arrival times consistent with the exit of contrast, as opposed to the arrival, from the brain. This takes place because of the superposition of small arteries, where blood enters the brain, with large veins, where blood exits the brain. The signal for the blood exiting the brain is stronger at these locations causing the correlation algorithm to break down. This effect of superposition on both the Cusum-based and

the correlation-based algorithms are studied in great detail in the next section.



(a) Cusum-based method (b) Correlation method

Figure 4-4: **Zoomed in view of 2D arrival times** – Zoomed in view of arrival times for set of images used in Figure 4-1 for both the Cusum-based and the correlation methods. *Note that the range of the two colorbars are different.* The left arrow in both panels shows a break in the arrival times that appears only in the correlation plot, where times were assigned to a stronger signal for contrast exiting the brain. The arrival times should follow the blood vessel shape and should be consistent with the flow of the blood. The right arrow in both panels shows a less drastic difference, where the arrival times in the gap were off by 8 frames from the vessel as opposed to 3 frames.

4.2.2 Validation

A set of test data was generated using a smooth fit to the blue time series in Figure 4-3(a). The smoothed fit provides a template time series as shown in Figure 4-5. Once formed this template can be both replicated and shifted and can have noise added to it. Resultant time series for the superposition of a stationary version and a shifted version of the template with added noise are shown in Figure 4-6 for four different shifts and in Figure 4-7 for four separate noise levels. These two factors, noise and shift, provide a battery of tests for comparing the Cusum-based algorithm of this thesis with the correlation-based algorithm.

The zero shift case provides a test for the performance of the algorithm under the assumption that the template is the same at different spatial locations and no superposition of signals is present. The smaller three noise levels are chosen to replicate the

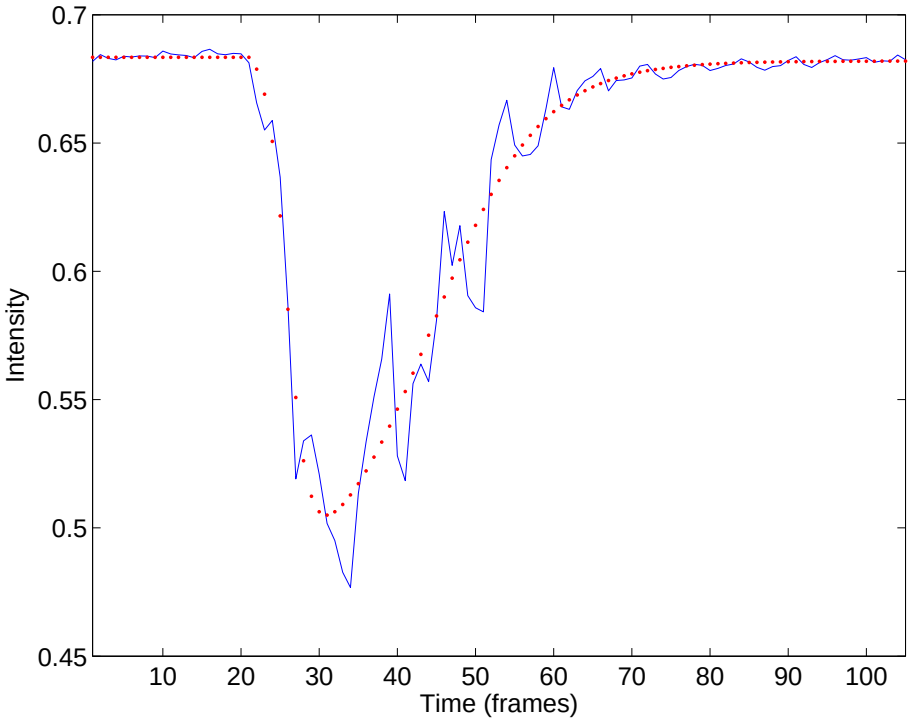
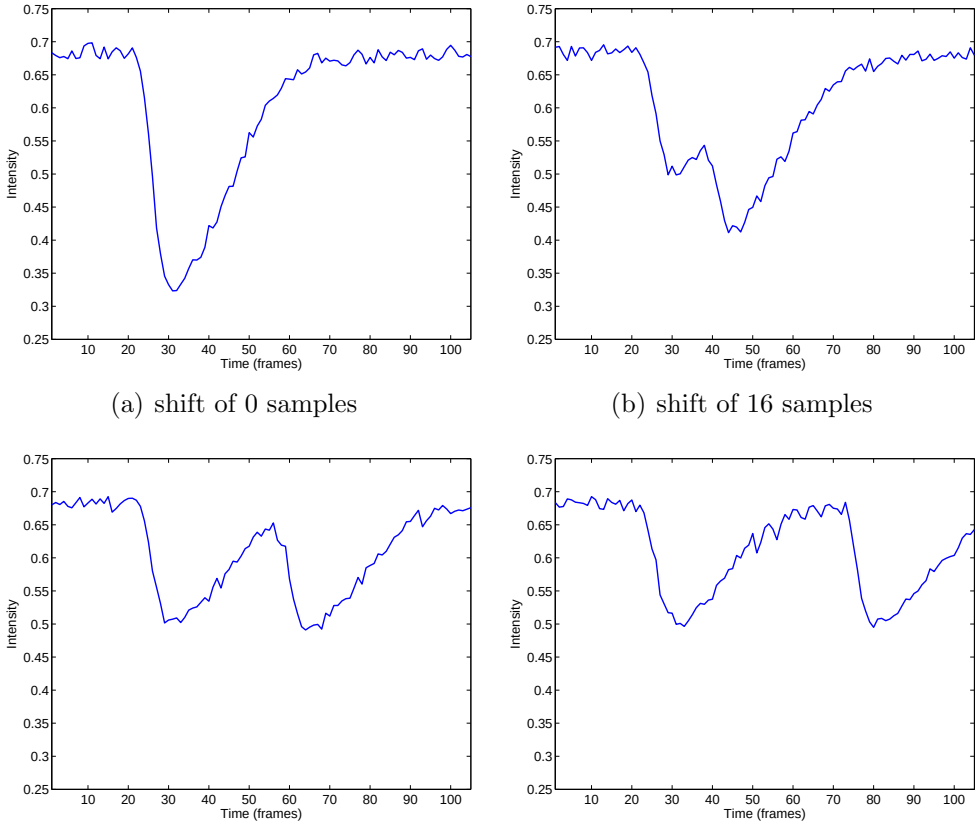


Figure 4-5: **Template time series** – The blue time series is a repeat of the blue plot shown in Figure 4-3(a) and the red dotted line is a smoothed version of the blue time series. The smoothed version is used as a template to generate test data to validate the algorithm.



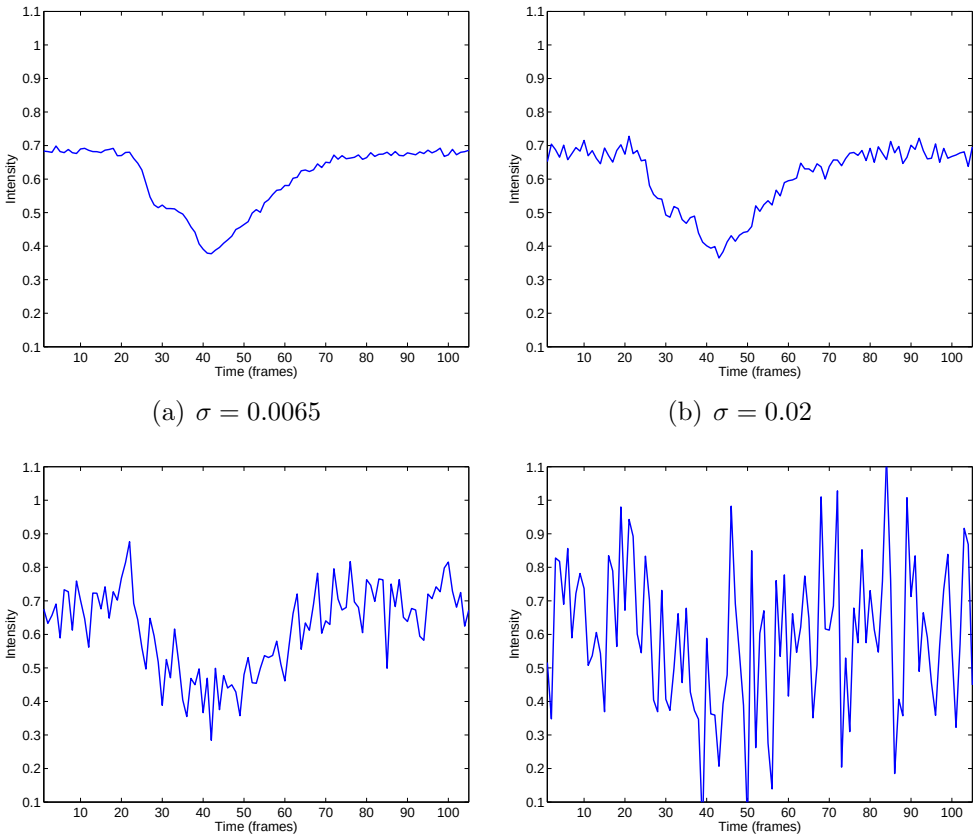
(a) shift of 0 samples

(b) shift of 16 samples

(c) shift of 34 samples

(d) shift of 50 samples

Figure 4-6: **Simulated time series with various shifts** – The simulated time series based on four different shifts. These time series are determined through the superposition of two versions of the template one with no delay and a second that was delayed by 0, 16, 34, and 50 samples. In this example, zero mean Gaussian noise with standard deviation 0.0065 was added to each of the time series.



(a) $\sigma = 0.0065$ (b) $\sigma = 0.02$

(c) $\sigma = 0.065$ (d) $\sigma = 0.2$

Figure 4-7: **Simulated time series with added noise** – Simulated time series with four levels of added noise. The added noise is zero mean Gaussian with standard deviation of 0.0065, 0.02, 0.065, and 0.2. These time series are based on the superposition of two versions of the template shown in Figure 4-5: one static and the other after a delay of 25 samples.

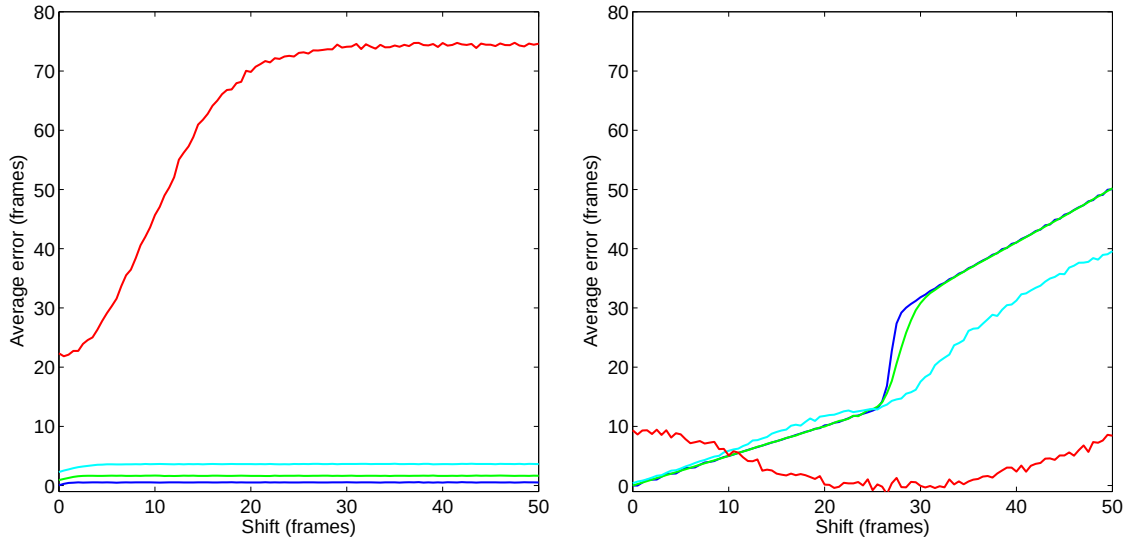
range found in the angiographic time sequence above, while the last level is chosen so that the Cusum algorithm breaks down. Figure 4-8 shows summary plots for the performance of the two algorithms under these four noise levels and on 101 evenly spaced shifts between 0 and 50. These plots show the mean and standard deviation of the error in frames for both algorithms after 10,000 trials at each shift and noise level.

In Figure 4-8, the mean and standard deviation error for the first three noise levels provided by the Cusum-based algorithm are predominately flat. Under the correlation-based algorithm the mean error is strictly increasing as the shift increases. In the middle range of shifts there is an increase in the slope of the mean error. Under the lowest two noise levels the standard deviation of the correlation-based method is flat except for a spike in the middle shifts. The standard deviation of the error in the correlation-based algorithm for the 0.065 noise level is increasing up to the middle shifts where it then plateaus. For the lowest noise level the cusum-based algorithm is always better, while for the next two higher noise levels the correlation algorithm is preferred up until a shift of three or four.

For the highest noise level, neither algorithm provides accurate results. While the average error of the cusum-based algorithm is always much worse than the average error for the correlation-based algorithm, the standard deviations are very high and cannot be trusted. Despite the high standard deviation, the mean error in the intermediate shift range does drop close to zero for the correlation-based method.

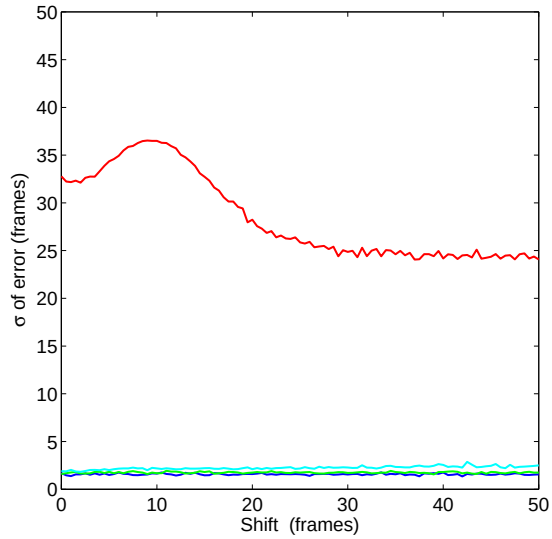
4.2.3 Detection of Clot

Figure 4-9 shows the resultant arrival times (front and side view) for a patient before a clot has developed. Figure 4-10 shows the arrival time pair after a clot has developed. The arrows in the two figures show where the clot formed. The arrival times downstream of the blockage in Figure 4-10 are clearly delayed from those in Figure 4-9.

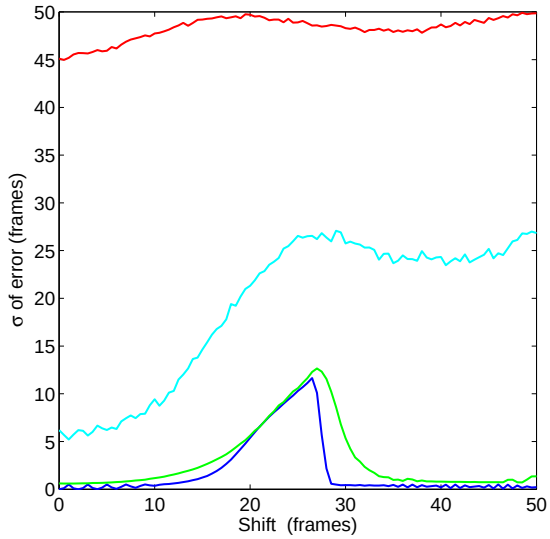


(a) Mean error for Cusum

(b) Mean error for correlation

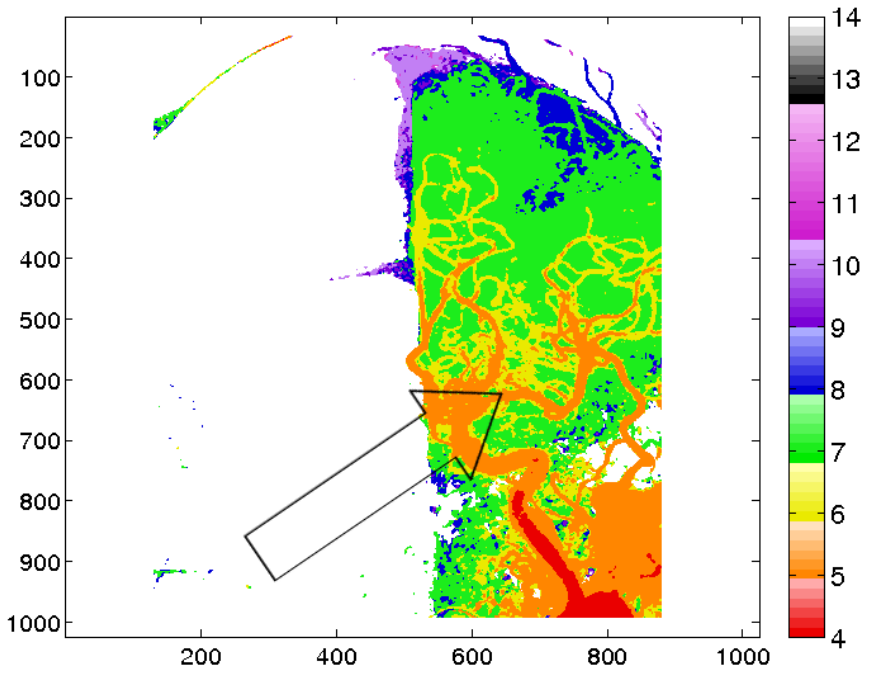


(c) σ of error for Cusum

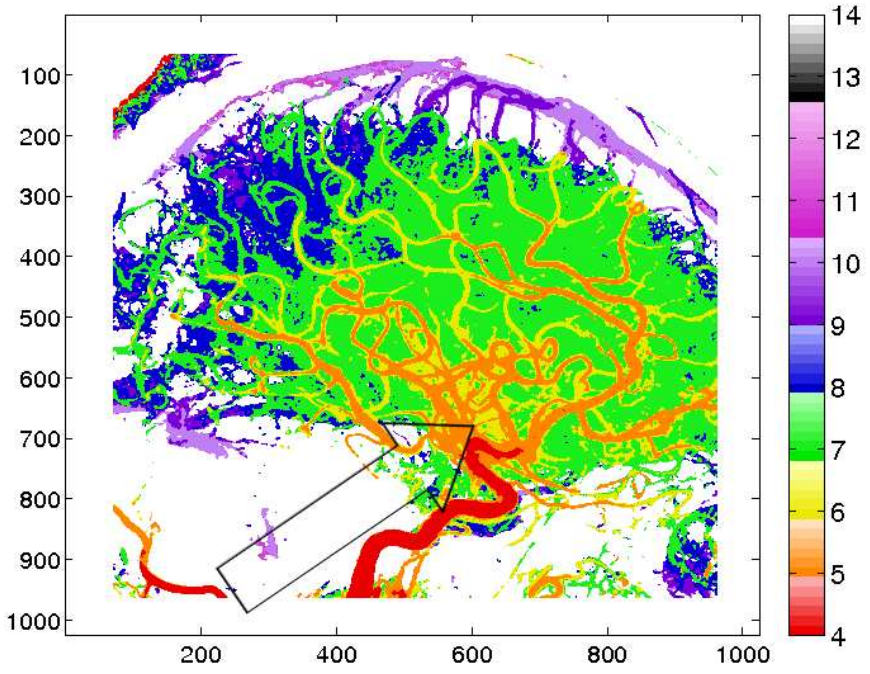


(d) σ of error for correlation

Figure 4-8: **Performance of the two arrival time algorithms** – Plots are shown for both the mean (top) and standard deviation σ (bottom) of the error after 10,000 iterations of both the Cusum-based algorithm (left) and the minimum square difference of correlation based algorithm (right) as a function of the shift applied to the second template. Shifts from zero to fifty are applied on the second template, some examples are shown in Figures 4-6 and 4-7. The results are shown for four levels of zero mean Gaussian noise with standard deviation of 0.0065 (blue), 0.02 (green), 0.065 (cyan), and 0.2 (red).

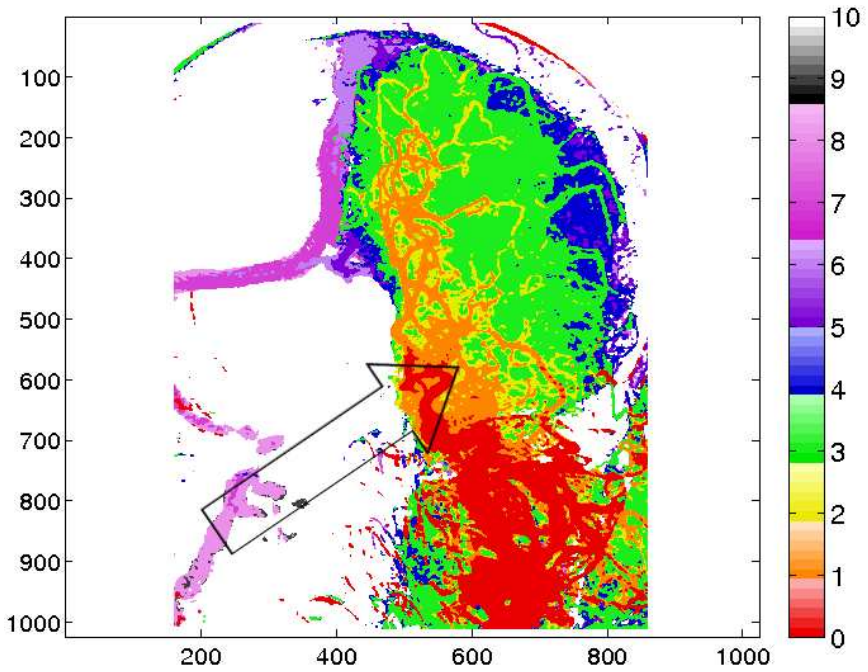


(a) Front view

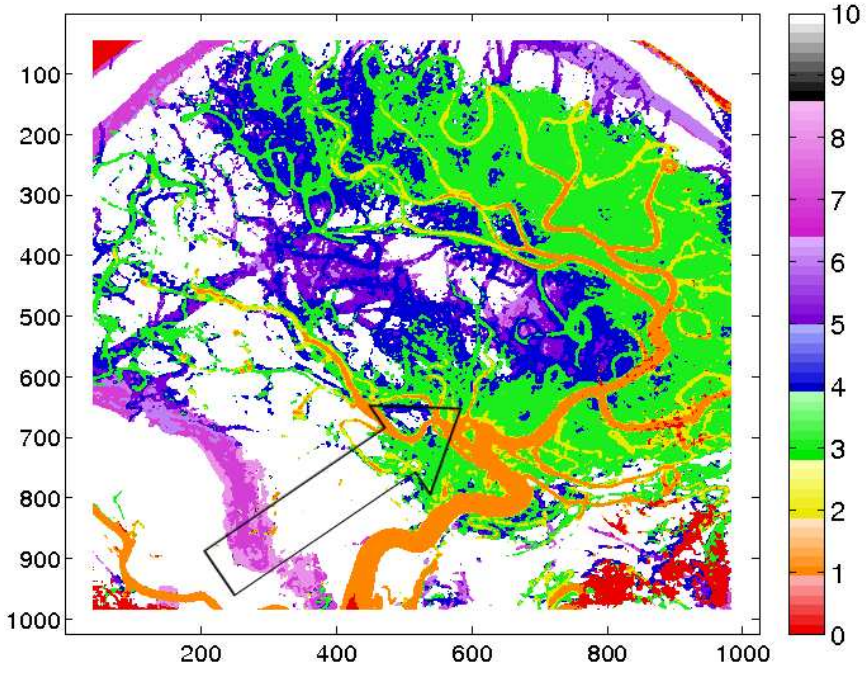


(b) Side view

Figure 4-9: **2D arrival times** – Arrival times measured by the Cusum-based algorithm on images from the same patient as in Figure 4-10. Contrast was injected into the carotid artery. An arrow was added to both views to emphasize the location of the clot in Figure 4-10. For this figure $\tau_f = 4$, $\tau_a = 1$, and $\tau = 1$. The color bar is used to show the arrival times (frame number) of the contrast relative to the first frame.



(a) Front view



(b) Side view

Figure 4-10: **2D arrival times with blockage** – Arrival times measured by the Cusum-based algorithm on images from the same patient as in Figure 4-9. Contrast was injected into the carotid artery and encountered a blockage in the Middle Cerebral Artery. An arrow was added to both views to emphasize the location of the clot. For this figure $\tau_f = 4$, $\tau_a = 1$, and $\tau = 1$. The color bar is used to show the arrival times (frame number) of the contrast relative to the first frame.

4.3 Conclusion

The new noise adaptive Cusum-based algorithm provides a more accurate measurement of the 2D times of arrival than existing methods. The method is more accurate because it does not depend on the shapes of the time series, an assumption that is often violated in practice. The results on the angiographic data show the superiority of the proposed methods over an existing techniques. These advantages were validated in the test data.

The test data show that the Cusum-based algorithm is better at dealing with the effects of superposition than the correlation-based algorithm. The standard deviation of the error remains relatively unchanged for the first three noise levels. For these noise levels the error goes up slightly as the noise level increases. This is because the threshold is increased proportionally with the standard deviation of the added noise. The algorithm finally fails when the threshold is so large that it is not always surpassed.

The mean error of the correlation algorithm increases as the shift increases because the effect of the superposition blurs the location of the first arrival. The sudden increase in the middle shifts (between 20 and 30) along with the spike in the standard deviation is due to the competition between the two sequences. In some instances, the template best matches with the first and in others it best matches with the second delayed sequence. After that middle shifts, it almost always best matches with the second delayed sequence and the standard deviation accordingly drops.

This study observed the impact of both noise and superposition, but it did not directly observe the impact of the amount of time it takes for the intensity to return to its pre-contrast value. It did however, provide an indirect test during the middle shifts when the effect of the superposition elongated the dynamics of the generated time series. Because there is not a perfect match for the correlation, there is a larger standard deviation in the assigned arrival time.

During these tests, it was observed that for lower values of the thresholds the mean error did decrease, however, the standard deviation went up. Also, as the

threshold increases so does the mean error. If the threshold is too high the arrival of contrast can be missed entirely. The small positive bias can also be subtracted from the measurement. Based on these observations, the values of the thresholds should be chosen carefully following tests such as those described in this chapter.

Figures 4-9 and 4-10 show a clear case where the arrival time algorithm can aid a surgeon in assessing changes in the timing of blood flow. The arrival time information provides an automatic and quantitative approach in place of a surgeon looking over every branch from one time to the next to search for the presence of an ailment. Not only does the surgeon know that a clot exists, he can also know how long it has delayed the flow and at which points contrast does not reach.

Another useful application of the Cusum-based algorithm also exists. Assuming the observation time is long enough to flush contrast from the head, the arrival time algorithm can be run in reverse to determine the exit time of the contrast. This practice highlights structures where the contrast leaves the brain. These structures are often occluded by other structures at the same location within the X-ray where contrast arrived at an earlier time. This algorithm provides surgeons a more accurate visualization for use during procedures.

Chapter 5

Spatio-Temporal Data Fusion Formulation

Spatio-temporal data fusion is the process of combining spatial and temporal constraints to produce a time series of 3D spatial data. This thesis work uses spatio-temporal data fusion to reconstruct a 3D time series that shows the flow of blood during a medical procedure. The process of spatio-temporal data fusion is shown in Figure 5-1, where the processed data from Chapter 3 is fused together to form the desired 3D time series. As shown in the figure, the data consists of the two registered time series $I_k[\mathbf{T}_k(x_i)]$, the constraint map $\mathscr{M}$, and the 3D orientation information $\mathbf{T_3}$.

Spatio-temporal data fusion constrains the reconstruction by taking advantage of the structure and attributes of blood vessels in the following five ways: 1. images are consistent with the time projection model, 2. images are smoothly varying off of edges in space [81], 3. blood flow over time follows physical laws, 4. the blood in the images generally appears in sparse convex structures, and 5. images are non-zero only within the 3D spatial map. Although not all of these constraints are necessary to produce a reconstruction, using all five helps to regularize the reconstruction. The constraints are imposed using energy functionals with term that penalize the violation of constraints. The energy functionals are minimized using gradient descent in function space while the gradients are obtained through Euler Lagrange equations

of the variational problem [41] (See Appendix A.1). The solution of this algorithm is the 3D time series. The 3D time series provides additional information that is not observable in the 2D arrival times that were determined in the previous chapter.

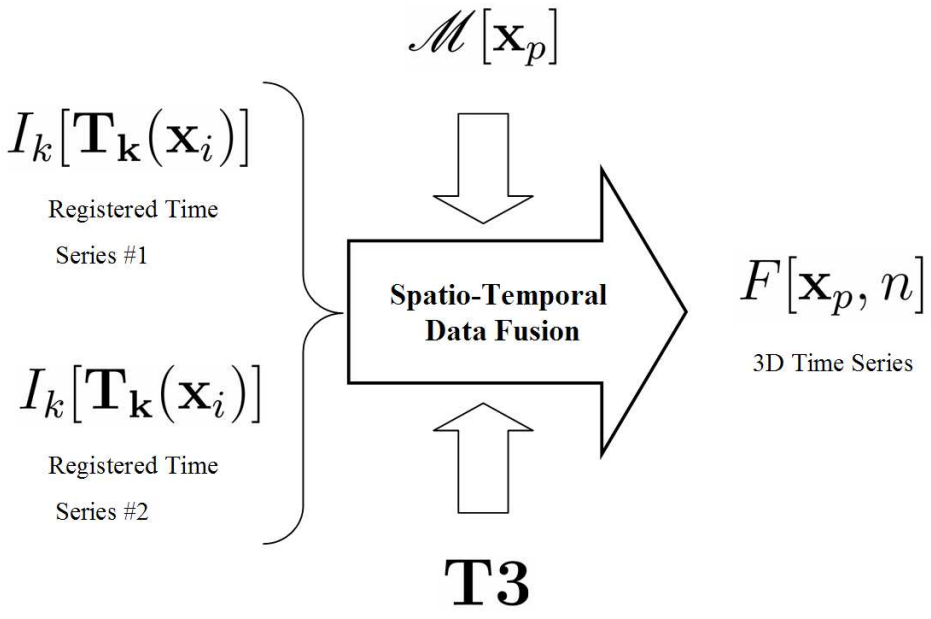


Figure 5-1: **Spatio-Temporal Data Fusion** – Flowchart showing the Spatio-Temporal Data Fusion process. The prepared data in Chapter 3, the two registered time series $I_k[\mathbf{T}_k(x_i)]$, the constraint map $\mathscr{M}$, and the 3D orientation information $\mathbf{T_3}$, is combined to produce a time series of 3D volumes.

This chapter presents the mathematical formulation for the spatio-temporal data fusion shown in Figure 5-1. Section 5.1 gives a brief description of a set of 2D test images that are used later in the chapter to visualize the algorithm. The basic functional that constrains the reconstruction to have consistent time projections and to be smoothly varying off a set of edges is developed in Section 5.2. The functional is then extended to encourage tight convex clusters of contrast using the minimum support constraint in Section 5.3. The clusters mimic the distribution of contrast within vasculature structures and discourage back projection artifacts. Next, it is restricted to the blood vessel structure using a spatial constraint map in Section 5.4. In Section 5.5, the functional is modified to use a separate temporal and spatial edge term. The functional is discretized in Section 5.6. The final algorithm is produced

by adapting the discrete equations to use a decoupled Gauss-Seidel-type algorithm to exploit the space-time structure of the angiography problem in Section 5.7. The final algorithm is discussed in Section 5.8. The 3D time series results from both real angiographic data and phantom data are saved for Chapter 6.

5.1 2D Visualizations

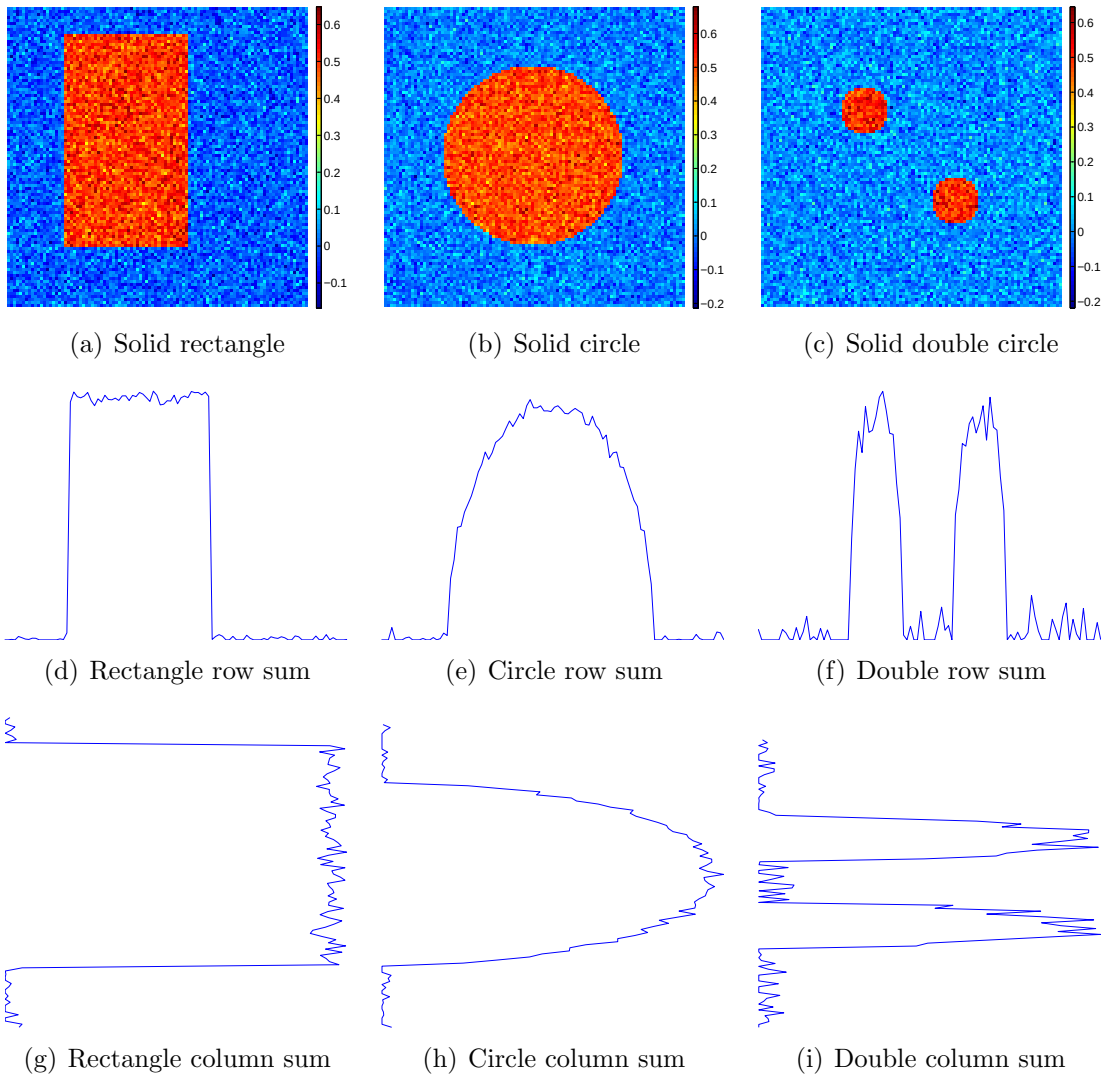
The 4D spatio-temporal process, whose modeling is the objective of this thesis, does not lend itself to easy visualization. Fortunately, many of the salient features of the 4D reconstruction can be observed in a 2D spatial reconstruction problem. The 2D example consists of a pair of a projections of a 2D image, one in the row and the other in the column directions, and a 2D constraint map. Throughout this chapter several 2D results are included to illustrate the impact of some of the terms in the functionals on the reconstruction. In Chapter 6, 4D results from both real angiographic data and artificial phantom data are presented.

Figure 5-2 displays the set of three different test images along with a pair of projections for each. The rectangle and circle are basic shapes whose reconstructions test the performance of the algorithm. The third column displays a two circle image that has an ambiguous reconstruction. This image is used in Section 5.4 to show the need for a constraint map to be incorporated in the algorithms functional. The column and row sums were set to zero when the sums were below zero.

5.2 Consistent Projections

A variational formulation similar to [81] that instead uses projections of the reconstruction F in the data fidelity term to constrain $F(\mathbf{x}, t)$ is

$$E(F, w) = \int_{S\times[0,t_{end}]} [\alpha([\boldsymbol{\mathcal{A}}F](\mathbf{s}, t) - G(\mathbf{s}, t))^2] d\mathbf{s}dt + \int_{\Omega\times[0,t_{end}]} [\beta(1-w)^2||\nabla F||^2 + \frac{\rho}{2}||\nabla w||^2 + \frac{w^2}{2\rho}] d\mathbf{x}dt, \quad (5.1)$$



(a) Solid rectangle (b) Solid circle (c) Solid double circle

(d) Rectangle row sum (e) Circle row sum (f) Double row sum

(g) Rectangle column sum (h) Circle column sum (i) Double column sum

Figure 5-2: **2D raw data** – 2D Raw data used to test the reconstruction algorithm.

where F the reconstructed 3D time series; w is the edge function; $\boldsymbol{\mathcal{A}}$ is the projection operator described in Chapter 2 ; α, β, and ρ are weighting constants; S is the support of G the measured data; and t_{end} is the duration of the time sequences. The first term ensures that the projections of the reconstructed volume are consistent with the time series of the projections. The last three terms are from the simultaneous smoothing segmentation problem discussed in greater detail in Section 2.3. The second term ensures that the reconstruction is smoothly varying off of the edges. The third term ensures that the edge function is smoothly varying, while the fourth term ensures that the edge function does not contain too many edges.

To solve this optimization using calculus of variations, each term must be defined on the same domain and integrated with respect to $d\mathbf{x}dt$. This is accomplished by using the back projection operator $\boldsymbol{\mathcal{B}}$ on the data fidelity term in Equation 5.1. This process treats each pixel by a weight proportional to the errors in the data fidelity. The new functional is now

$$E(F,w) = \int_{\Omega\times[0,t_{end}]} [\alpha\boldsymbol{\mathcal{B}}([\boldsymbol{\mathcal{A}}F]-G)^2 + \beta(1-w)^2||\nabla F||^2 + \frac{\rho}{2}||\nabla w||^2 + \frac{w^2}{2\rho}]d\mathbf{x}dt. \quad (5.2)$$

Note that this approach is different from using the inverse of the projection data with the reconstructed image in the data fidelity term, i.e., $F - \boldsymbol{\mathcal{B}}G$. The goal of the data fidelity term in Equation 5.2 is to match the projection of the reconstructed data, not to match the final image to the output of any particular inverse tomographic algorithm. If $\beta = 0$ only the data fidelity term remains, whose solution is equivalent to a version of the SART [8] algorithm discussed in Section 2.2.2. This is different from [63] where Lagrange multipliers were used to weight any differences in the projections. A different approach that instead uses level sets to solve a variational formulation of the tomography problem is presented in [4]. Another level set technique described in [39] instead solves for a set of 2D texture coefficients that is consistent with the measured projections. These level set approaches are not appropriate for this appli-

cation because they do not allow a straightforward application of the sparsity and map constraints presented later in this chapter.

Figure 5-3 shows the results of the reconstruction process on the first two test images, the third image will be addressed later in Section 5.4. The results show the actual reconstruction in the top row and the edge function in the bottom row. Both reconstructions show typical artifacts of back projection. There are negative values in both images that counteract overshoots in the central portions of the images. The edges for the reconstructed rectangle appear where the original rectangles edges did, but they incorrectly appear at edges of the artifacts. The edges of the circle only appear in the four corners due to the odd shape of the reconstructed image. In Section 5.3, a strategy is presented that eliminates these artifacts and produces more accurate reconstructions.

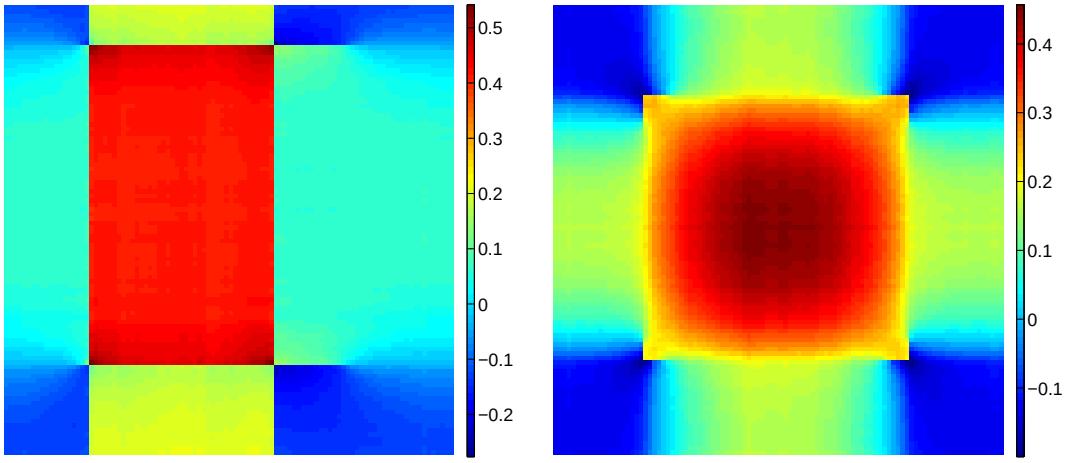
5.3 Minimum Support Constraint

The formulation in Equation 5.2 produces reconstructions with projections that are equal to the measured projections but not physically plausible. This is because the underdetermined set of measurements result in commonly known back projection artifacts [57], such as those in Figure 5-3. Because contrast typically appears in tight convex small regions, a support constraint $C(F)$ is introduced to the functional to reduce these artifacts. The resulting functional is

$$E(F,w) = \int_{\Omega\times[0,t_{end}]} [\alpha\boldsymbol{\mathcal{B}}([\boldsymbol{\mathcal{A}}F]-G)^2 + \beta(1-w)^2||\nabla F||^2 + \gamma C(F) + \frac{\rho}{2}||\nabla w||^2 + \frac{w^2}{2\rho}]d\mathbf{x}dt, \quad (5.3)$$

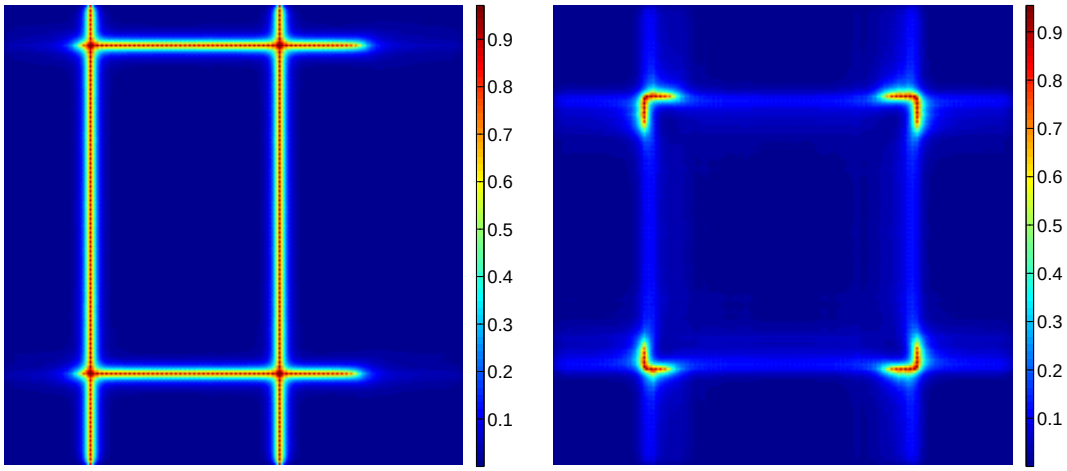
where γ is an additional weighting term and C penalizes the measure of the non-vanishing support of F.

Recently, there has been a great deal of work on using sparsity terms to regularize reconstruction based on an under determined set of constraints [14, 17, 23, 68]. In [14, 68], $C(F) = ||F||_0$ represents the desired non-vanishing support of F. Under



(a) Solid rectangle

(b) Solid circle



(c) Rectangle edge function

(d) Circle edge function

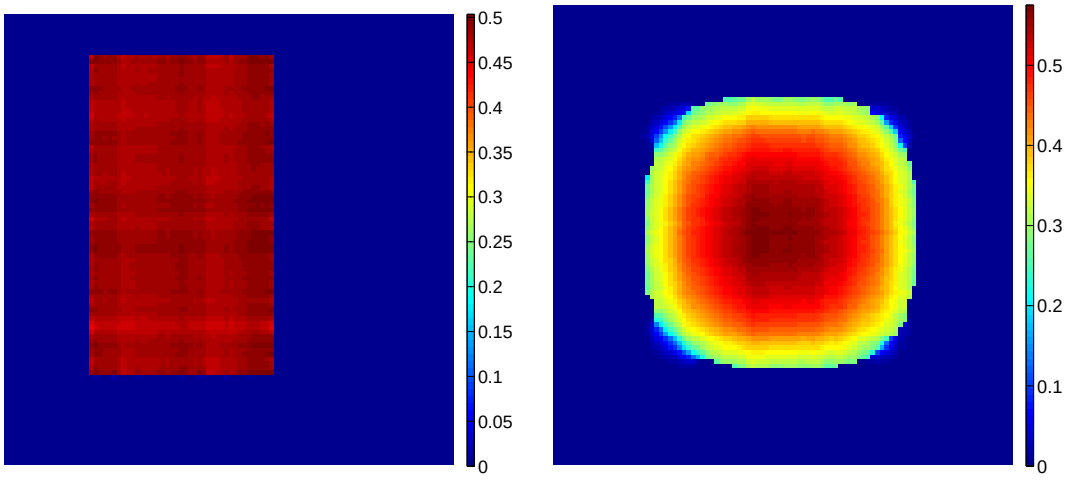
Figure 5-3: **2D reconstruction using edge term** – Reconstructed data and edge functions using a numerical solution to equation 5.2.

this sparsity term, the minimization is not convex and is thus difficult to solve. To create an optimization process that is convex with respect to F while w fixed and with respect to w with F fixed,

$$C(F) = ||F||_1 \tag{5.4}$$

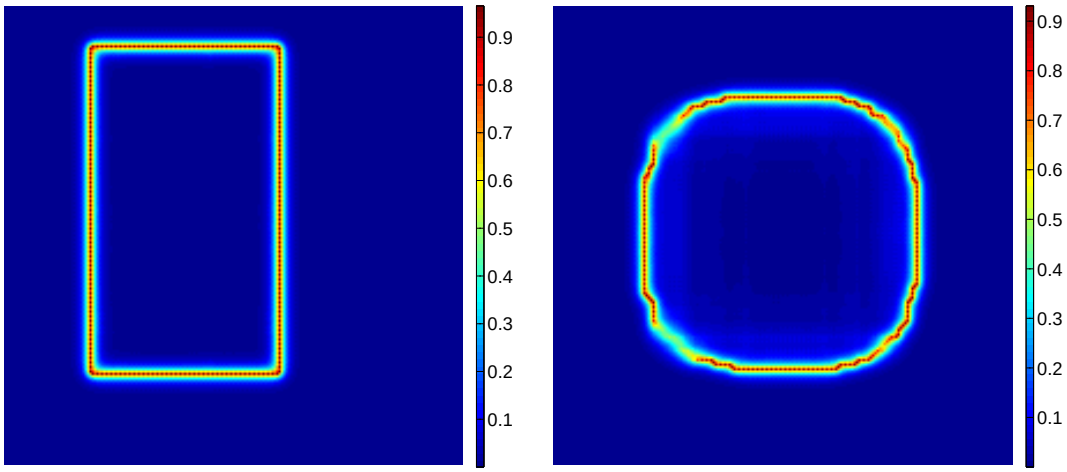
is instead used [16, 36]. Under certain conditions on the overall sparseness, number of non zero terms measured by the 0-norm, of the underlying data, the 0-norm and 1-norm optimization problems achieve identical results [14, 16]. Unlike the work in [15], where linear programming is used to minimize the sparseness under a separate constraint on the projections, the formulation in Equation 5.3 minimizes the two in a single term. For appropriate choices of γ the results should be similar. The results are similar because both optimization techniques minimize the same two terms. Alternatives to the 1-norm are the p-norm for $p \geq 1$. For work with $p = 1.1$ see [68]. The 1.1 norm is favored in this work because it is differentiable around zero. In the thesis work the derivative of the 1 norm is set to zero at zero. The 1-norm is chosen in the thesis because it provides convex solution with respect to F and w separately to the variational problem, while providing a good approximation to the non-convex 0-norm approach.

Figure 5-4 shows the impact of the additional term in Equation 5.3 on the reconstructed test images in Figure 5-2(a) and 5-2(b). As desired, the back projection artifacts seen in Figure 5-3 are now eliminated. The rectangle is very close to the original, while in the circle more intensities appear at the center than at the periphery. Overall, it provides a good reconstruction for one based on only two projections. The edge function produced for the rectangle is exact. For the circle, the edge function is not tight all of the way around and is somewhat more square than it should be. This affect can be largely mitigated by increasing the smoothness weighting term β. Other than this, it provides a reasonable approximation to the original.



(a) Solid rectangle

(b) Solid circle



(c) Rectangle edge function

(d) Circle edge function

Figure 5-4: **2D reconstruction using convexity term** – Reconstructed data and edge functions found using gradient descent to find a minimum of Equation 5.3. The artifacts found in Figure 5-3 are removed.

5.4 Spatial Constraints

After the background image is subtracted from the aligned X-ray time series, contrast, noise, and structural artifacts from motion between frames remain. Assuming the registration in Section 3.2 limits the artifacts, the dominating term is the contrast itself. Because the contrast only appears within the blood vessels themselves, the knowledge of their locations can further regularize the fusion process. Specifically, the reconstruction can be confined to the map $\mathscr{M}$ of location of the blood vessels. The map takes on a binary value of one at the locations in V where the vasculature is thought to be inside and zero otherwise. The first of two possible techniques restricts the functional to the support of $\mathscr{M}$ as in the functional

$$E(F,w) = \int_{\mathscr{M}\times[0,t_{end}]} [\alpha\boldsymbol{\mathcal{B}}([\boldsymbol{\mathcal{A}}F]-G)^2 + \beta(1-w)^2||\nabla F||^2 + \gamma C(F) + \frac{\rho}{2}||\nabla w||^2 + \frac{w^2}{2\rho}]d\mathbf{x}dt. \quad (5.5)$$

The second approach penalizes non-zero values of F that occur off of the support of $\mathscr{M}$. This weighting can be written as

$$E(F,w) = \int_{\Omega\times[0,t_{end}]} [\alpha\boldsymbol{\mathcal{B}}([\boldsymbol{\mathcal{A}}F]-G)^2 + \beta(1-w)^2||\nabla F||^2 + \gamma C(F) + \eta(\mathbf{x})F^2 + \frac{\rho}{2}||\nabla w||^2 + \frac{w^2}{2\rho}]d\mathbf{x}dt, \quad (5.6)$$

where $\eta(\mathbf{x})$ is a spatial varying penalty term. The penalty is large where contrast is known to be absent, zero where contrast is present, and an intermediate value in areas where the presence of contrast is uncertain. A simple choice for $\eta(\mathbf{x})$ is

$$\eta(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{x} \in \mathscr{M}, \\ \lambda & \text{if } \mathbf{x} \notin \mathscr{M}, \end{cases} \quad (5.7)$$

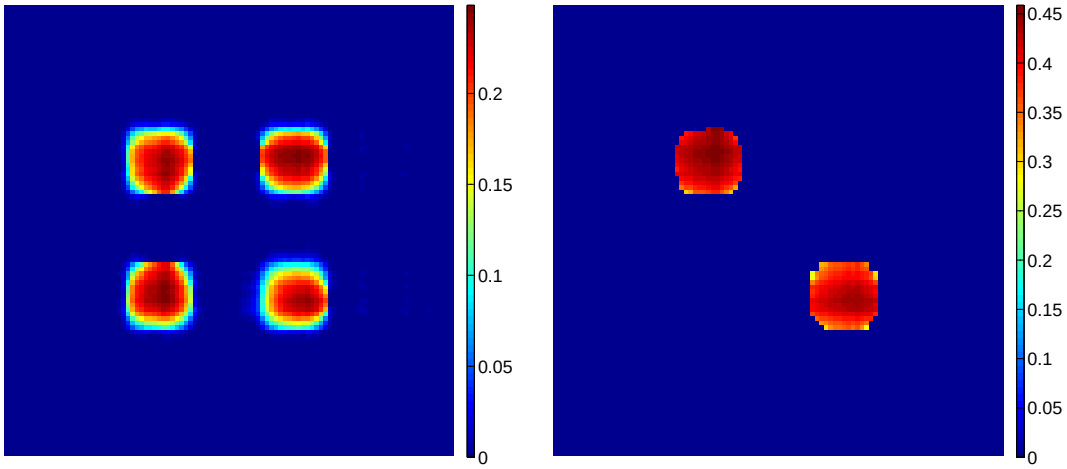
where λ is a constant chosen to appropriately penalize F in regions where no contrast is present, while providing a stable numerical scheme.

Both of these functionals were used to determine consistent reconstructions. The first approach allowed the projection operators to be modified to take advantage of the zero value of F outside of the map. This allowed many more values of the discrete operators in $\mathcal{A}$ and $\mathcal{B}$ defined in Section 2.2.1 to be set to zero. This can be thought of $\mathcal{B}$ smearing the residual evenly over a smaller map region. This resulted in faster convergence because the value of F is modified at fewer locations.

The first column in Figure 5-5 shows the reconstruction of the double circle image in Figure 5-2 using both the functional in Equation 5.3 and the functional in Equation 5.5 along with the map in Figure 5-6. The left panels show the presence of two "ghost" circles where none appeared in the original. This result is consistent with the two projections and can be interpreted as giving an equal weighting to each of the circles when insufficient information is available to decide between them. By introducing the map in Figure 5-6, the ambiguity is resolved as shown in the right panels. Similar results were obtained using a map that restricted only the area containing the two "ghost" circles. Figure 5-5 shows the results using Equation 5.5, which were almost identical to the results using Equation 5.6 for the 1000 iterations used in the simulation. The use of the constraint map for the other test images does not impact the reconstruction because no ambiguity exists.

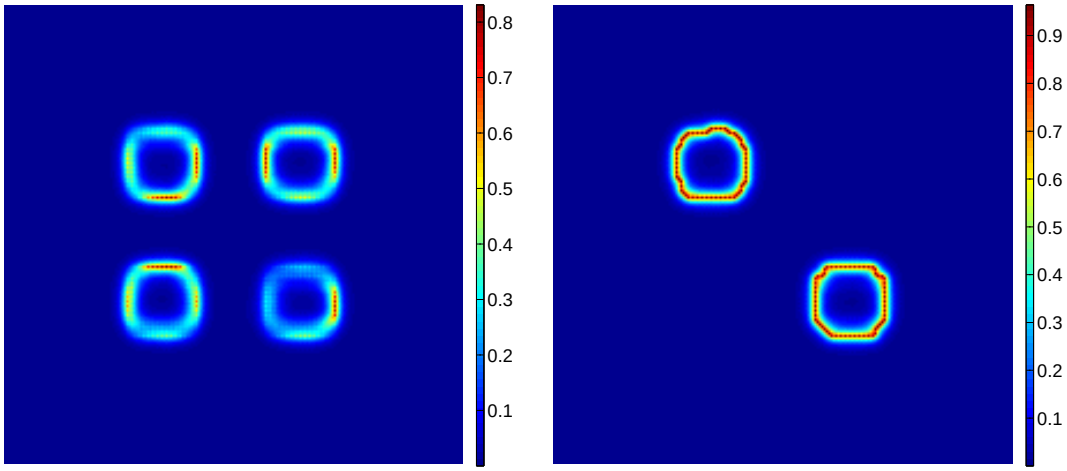
5.5 Time Constraints

The functional in Equation 5.5 uses the same edge indicator for both edges in space and in time. This model does not capture the different nature of edges in space and in time. Edges in time occur only on the advancing boundary of the contrast, whereas the temporal edges at a given time occur at all points in space where contrast appears or disappears in the pair of volumes the temporal edge lies between in time. To capture this distinction the single edge indicator function w is broken up into a spatial edge indicator function w and a temporal edge indicator function v. After



(a) Reconstruction using L1 norm

(b) Reconstruction using L1 norm and map



(c) Edge using L1 norm and map

(d) Edge using L1 norm and map

Figure 5-5: **Resolving ambiguity using map** – The two panels on the right show the reconstruction and edge function of a situation that is ambiguous. The map shown in Figure 5-6 is used in Equation 5.5 to constrain the reconstruction eliminates the ambiguity as shown in the right panels.

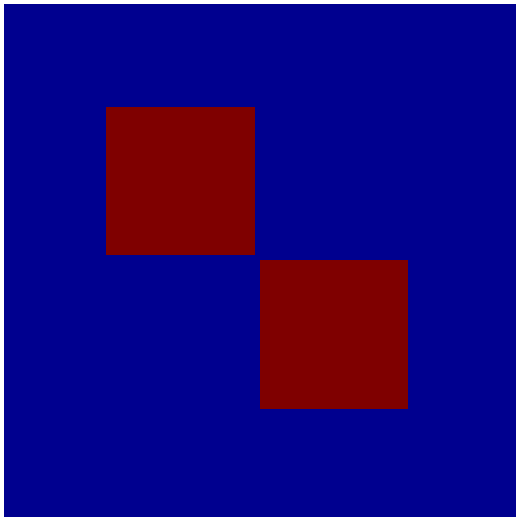


Figure 5-6: **Map** – The figure shows the map used to further constrain the reconstruction process in Figure 5-5. The red portion of the map specifies the non-zero support of the reconstruction.

breaking up the edge function the functional becomes

$$E(F,w,v) = \int_{\mathscr{M}\times[0,t_{end}]} [\alpha\boldsymbol{\mathcal{B}}([\boldsymbol{\mathcal{A}}F]-G)^2 + \beta(1-w)^2||\nabla_x F||^2 + \beta(1-v)^2||\nabla_t F||^2 + \gamma|F|_1 + \frac{\rho}{2}||\nabla_x w||^2 + \frac{w^2}{2\rho} + \frac{\rho}{2}||\nabla v||^2 + \frac{v^2}{2\rho}]d\mathbf{x}dt. \quad (5.8)$$

Following the Calculus of Variations (See Appendix A.1), the Euler Lagrange equations for Equation 5.8 are

$$\partial_F E = \alpha\boldsymbol{\mathcal{B}}([\boldsymbol{\mathcal{A}}F]-G) - \nabla_x\cdot[\beta(1-w)^2\nabla_x F] - \nabla_t\cdot[\beta(1-v)^2\nabla_t F] + \frac{\gamma}{2}\frac{F}{|F|} \quad (5.9)$$

for F,

$$\partial_w E = 2\beta(w-1)||\nabla_x F||^2 + \frac{w}{\rho} - \rho\nabla_x^2 w \quad (5.10)$$

for w, and

$$\partial_v E = 2\beta(v-1)||\nabla_t F||^2 + \frac{v}{\rho} - \rho\nabla^2 v \quad (5.11)$$

for v. These coupled equations are used to find the desired minimum of Equation 5.8 that occurs when the two gradients are equal to zero and the second derivatives are

positive. Unless the function in Equation 5.8 is globally convex with respect to F, w, and v, the minimum is not necessarily global.

The temporal smoothness constraint in Equation 5.8 provides an implicit fluid flow model. This model smoothes the intensities through time off of a set of edges or discontinuities in the flow. The set of discontinuities is penalized to be smoothly varying in space and time. A higher fidelity fluid flow model would capture a more precise numerical model for the flow such as an approximation to the Navier-Stokes equation. The temporal flow of the contrast can be captured by the temporal derivative of the intensities or $\nabla_t F$. If $\nabla_t F$ satisfies the relation

$$\frac{\partial}{\partial t}(\nabla_t F) = \mathscr{L}(\nabla_t F), \tag{5.12}$$

where $\mathscr{L}$ is an operator, possibly nonlinear, that approximates the flow model, then $\nabla_t F$ is a valid flow. An appropriate functional for incorporating models of this form is

$$E(F,w) = \int_{\mathscr{M}\times[0,t_{end}]} [\alpha \boldsymbol{\mathcal{B}}([\boldsymbol{\mathcal{A}}F] - G)^2 + \beta(1-w)^2 ||\nabla_x F||^2 + \zeta||\frac{\partial}{\partial t}(\nabla_t F) - \mathscr{L}(\nabla_t F)||^2 + \gamma|F|_1 + \frac{\rho}{2}||\nabla_x w||^2 + \frac{w^2}{2\rho} d\mathbf{x}dt, \tag{5.13}$$

where ζ is a penalty term that penalizes deviations from the flow model. In order for algorithms of this form to be practical, the fluid model approximations in the term $\mathscr{L}$ must not be too computationally demanding.

5.6 Discretization

In this section, the discrete equations used to solve the variational optimization problem are presented. The approach in this section finds the Euler Lagrange equations for the discrete approximation to the formulation in Equation 5.8. This result can also be obtained through the discretization of the continuous Euler Lagrange Equations shown in Equations 5.9, 5.10, and 5.11. However, if the gradient terms in Equation

5.9 are carried through using the product rule the expression is different and is difficult to discretize. The notation and choice of edge function for the work presented here draw heavily from the 2D work done in [91]. This work was chosen because it provides an accurate discretizations and one of first few implementable algorithms in the literature [113]. The work presented here extends the 2D work in [91] that was described in 2.3.1 into four dimensions, 3 space and 1 time, and includes different terms used in this angiography application.

The reconstructed volumes $F(\mathbf{x}_i, n)$ are defined on a rectangular lattice that only includes points where the map function is equal to one. The spatial edge function $w(\mathbf{x}_i, n)$ is defined on a similar rectangular lattice produced by placing a point between each pair of adjacent points in space of the lattice that supports $F(\mathbf{x}_i, n)$. Similarly, the temporal edge function $v(\mathbf{x}_i, n)$ is defined by placing a point between each pair of adjacent points in time of the lattice that supports $F(\mathbf{x}_i, n)$. This discretization results in an accurate and simple implementation. The lattice for $F(\mathbf{x}_i, n)$ is defined on the subset of $\mathbb{Z}^4$ where the map $\mathscr{M} = 1$ as

$$\mathscr{L}_F = \{(i,j,k,n) : i \in \{1,\ldots,M\}, j \in \{1,\ldots,P\}, \\ k \in \{1,\ldots,Q\}, n \in \{1,\ldots,N\}, \mathscr{M}[i,j,k] = 1\}. \quad (5.14)$$

The nearest neighbors of the image at point $(\mathbf{x}_i, n) \in \mathscr{L}_F$ are the points in the set

$$\mathscr{N}_F(\mathbf{x}_i, n) = \{(\mathbf{x}_j, m) \in \mathbb{Z}^4 : |\mathbf{x}_j - \mathbf{x}_i| + |n - m| = \delta\}, \quad (5.15)$$

where δ is the lattice spacing. The lattices for the spatial edge function $w(\mathbf{x}_i, n)$ and the temporal edge function $v(\mathbf{x}_i, n)$ are respectively,

$$\mathscr{L}_w = \{\frac{\mathbf{x}_l + \mathbf{x}_m}{2}, q : (\mathbf{x}_l, q) \in \mathscr{L}_F, (\mathbf{x}_m, r) \in \mathscr{N}_F(\mathbf{x}_l, q) \cap \mathscr{L}_F, q = r\} \quad (5.16)$$

and

$$\mathscr{L}_v = \{\mathbf{x}_l, \frac{q+r}{2} : (\mathbf{x}_l, q) \in \mathscr{L}_F, (\mathbf{x}_m, r) \in \mathscr{N}_F(\mathbf{x}_l, q), \mathbf{x}_l = \mathbf{x}_m, |q - r| = 1\}. \quad (5.17)$$

The resultant nearest neighbors for w are

$$\mathscr{N}_w(\mathbf{y}_i, n) = \{\mathbf{y}_j = \frac{\mathbf{x}_l + \mathbf{x}_m}{2}, n : (\mathbf{x}_l, q) \in \mathscr{L}_F, \\ (\mathbf{x}_m, r) \in \mathscr{N}_F(\mathbf{x}_l, q), q = r = n, |\mathbf{y}_j - \mathbf{y}_i| = \delta/\sqrt{2}\} \quad (5.18)$$

and for v are

$$\mathscr{N}_v(\mathbf{z}_i, n) = \{\mathbf{z}_j \in \mathbb{Z}^3, (p + 1/2) \in \mathbb{Z}; |\mathbf{z}_j - \mathbf{z}_i| + |n - p| = \delta\}. \quad (5.19)$$

Using these definitions for the lattice and neighbor functions the discrete formulation is

$$E(F, w, v) = \sum_{(\mathbf{x}_i, n) \in \mathscr{L}_F} \left[\alpha \sum_l \boldsymbol{B}_{l,i} (\sum_j \boldsymbol{A}_{j,l} F[\mathbf{x}_j, n] - G[\mathbf{x}_l, n])^2 + \gamma |F[\mathbf{x}_i, n]| \right. \\ \sum_{(\mathbf{x}_j, n) \in \mathscr{N}_F(\mathbf{x}_i, n)} \frac{\beta}{2} \frac{(F[\mathbf{x}_i, n] - F[\mathbf{x}_j, n])^2}{\delta^2} (1 - w[\frac{\mathbf{x}_i + \mathbf{x}_j}{2}, n])^2 + \\ \left. \sum_{(\mathbf{x}_i, m) \in \mathscr{N}_F(\mathbf{x}_i, n)} \frac{\beta}{2} \frac{(F[\mathbf{x}_i, n] - F[\mathbf{x}_i, m])^2}{\delta^2} (1 - v[\mathbf{x}_i, \frac{n + m}{2}])^2 \right] + \\ \sum_{(\mathbf{y}_i, n) \in \mathscr{L}_w} \left[\frac{w[\mathbf{y}_i, n]^2}{2\rho} + \sum_{(\mathbf{y}_j, m) \in \mathscr{N}_w(\mathbf{y}_i, n)} \frac{\rho}{4} \frac{(w[\mathbf{y}_i, n] - w[\mathbf{y}_j, n])^2}{\delta^2/2} \right] + \\ \sum_{(\mathbf{z}_i, n) \in \mathscr{L}_v} \left[\frac{v[\mathbf{z}_i, n]^2}{2\rho} + \sum_{(\mathbf{z}_j, m) \in \mathscr{N}_v(\mathbf{z}_i, n)} \frac{\rho}{4} \frac{(v[\mathbf{z}_i, n] - v[\mathbf{z}_j, m])^2}{\delta^2} \right]. \quad (5.20)$$

The term $\delta^2/2$ is used because the mesh for w is twice as dense as that for F. From this point forward the expression is simplified by setting $\delta = 1$. For a simpler implementation, the term w and v that ranges between zero and one are replaced with $1 - w$ and $1 - v$, respectively. The interpretation of the value of 1 and 0 are

simply reversed. The resultant discrete functional is

$$E(F,w,v) = \sum_{(\mathbf{x}_i,n)\in\mathscr{L}_F} \Bigg[\alpha \sum_l \boldsymbol{B}_{l,i} (\sum_j \boldsymbol{A}_{j,l} F[\mathbf{x}_j,n] - G[\mathbf{x}_l,n])^2 + 2\gamma |F[\mathbf{x}_i,n]| \\ \sum_{(\mathbf{x}_j,n)\in\mathscr{N}_F(\mathbf{x}_i,n)} \frac{\beta}{2}(F[\mathbf{x}_i,n] - F[\mathbf{x}_j,n])^2 (w[\frac{\mathbf{x}_i+\mathbf{x}_j}{2},n])^2 + \\ \sum_{(\mathbf{x}_i,m)\in\mathscr{N}_F(\mathbf{x}_i,n)} \frac{\beta}{2}(F[\mathbf{x}_i,n] - F[\mathbf{x}_i,m])^2 (v[\mathbf{x}_i,\frac{n+m}{2}])^2 \Bigg] + \\ \sum_{(\mathbf{y}_i,n)\in\mathscr{L}_w} \left[\frac{(1-w[\mathbf{y}_i,n])^2}{2\rho} + \sum_{(\mathbf{y}_j,m)\in\mathscr{N}_w(\mathbf{y}_i,n)} \frac{\rho}{2}(w[\mathbf{y}_i,n] - w[\mathbf{y}_j,n])^2 \right] + \\ \sum_{(\mathbf{z}_i,n)\in\mathscr{L}_v} \left[\frac{(1-v[\mathbf{z}_i,n])^2}{2\rho} + \sum_{(\mathbf{z}_j,m)\in\mathscr{N}_v(\mathbf{z}_i,n)} \frac{\rho}{4}(v[\mathbf{z}_i,n] - v[\mathbf{z}_j,m])^2 \right]. \quad (5.21)$$

The Euler Lagrange Equations for the discrete functional in Equation 5.21 are

$$\partial_{F_{i,n}} E = \alpha \sum_l \boldsymbol{B}_{l,i} (\sum_j \boldsymbol{A}_{j,l} F[\mathbf{x}_j,n] - G[\mathbf{x}_l,n]) + \gamma \frac{F[\mathbf{x}_i,n]}{2|F[\mathbf{x}_i,n]|} + \\ \sum_{(\mathbf{x}_j,n)\in\mathscr{N}_F(\mathbf{x}_i,n)} \beta(F[\mathbf{x}_i,n] - F[\mathbf{x}_j,n])(w[\frac{\mathbf{x}_i+\mathbf{x}_j}{2},n])^2 + \\ \sum_{(\mathbf{x}_i,m)\in\mathscr{N}_F(\mathbf{x}_i,n)} \beta(F[\mathbf{x}_i,n] - F[\mathbf{x}_i,m])(v[\mathbf{x}_i,\frac{n+m}{2}])^2, \quad (5.22)$$

$$\partial_{w_{i,n}} E = \beta(F[\mathbf{x}_l,n] - F[\mathbf{x}_m,n])^2 (w[\mathbf{y}_i,n]) \\ + \frac{w[\mathbf{y}_i,n]-1}{2\rho} + \sum_{(\mathbf{y}_j,n)\in\mathscr{N}_w(\mathbf{y}_i,n)} \rho(w[\mathbf{y}_i,n] - w[\mathbf{y}_j,n]), \quad (5.23)$$

and

$$\partial_{v_{i,n}} E = \beta(F[\mathbf{x}_i,q] - F[\mathbf{x}_i,r])^2 (v[\mathbf{x}_i,n]) \\ + \frac{w[\mathbf{x}_i,n]-1}{2\rho} + \sum_{(\mathbf{x}_i,m)\in\mathscr{N}_w(\mathbf{x}_i,n)} \rho(v[\mathbf{x}_i,n] - v[\mathbf{x}_i,m]). \quad (5.24)$$

Following the substitution of w with $w-1$ and v with $v-1$, these terms are equivalent to a discretization of Equations 5.9, 5.10, and 5.11. These equations can be used to implement an iterative gradient descent towards a local solution to the minimization of the discrete functional in Equation 5.21. Without loss of generality, the term $\gamma/2$ is replaced with γ.

The variables can be modified to denote the value at a particular iteration k by adding the additional argument to the respective function such as in $F[\mathbf{x}_i, n, k]$ and $w[\mathbf{y}_j, m, k]$. The iterative steps are

$$F[\mathbf{x}_i, n, k] = F[\mathbf{x}_i, n, k-1] - c_F \cdot \partial_{F_{i,n}} E, \tag{5.25}$$

$$w[\mathbf{y}_i, n, k] = w[\mathbf{y}_i, n, k-1] - c_w \cdot \partial_{w_{i,n}} E, \tag{5.26}$$

and

$$v[\mathbf{z}_i, n, k] = v[\mathbf{z}_i, n, k-1] - c_v \cdot \partial_{v_{i,n}} E, \tag{5.27}$$

where c_F, c_w, and c_v are terms that control the speed and stability of the descent process. These terms should be carefully selected because if they are too large convergence is not achieved and if they are too small convergence is very slow. Following a similar result to that in [91] that was outlined in Section 2.3.2, a good choice that produces a Newton type descent is

$$c_F = \frac{1}{2}(\alpha + 8\beta + \frac{\gamma}{|F[\mathbf{x}_i, n, k-1]|})^{-1}, \tag{5.28}$$

$$c_w = \frac{1}{2}(\beta(F[\mathbf{x}_l, n, k-1] - F[\mathbf{x}_m, n, k-1])^2 + \frac{1}{2\rho} + 8\rho)^{-1}, \tag{5.29}$$

and

$$c_v = \frac{1}{2}(\beta(F[\mathbf{z}_i, q, k-1] - F[\mathbf{z}_i, r, k-1])^2 + \frac{1}{2\rho} + 8\rho)^{-1}. \tag{5.30}$$

These terms vary with location, time, and iteration. The coefficient 8 in Equations 5.28-5.30 corresponds to the number of neighbors points for each lattice point. With this choice for c_w and c_v, the iterative equations for $w[\mathbf{y}_i, n, k]$ and $v[\mathbf{z}_i, n, k]$ simplify

to

$$w[\mathbf{y}_i, n, k] = \frac{1}{2} w[\mathbf{y}_i, n, k-1] + c_w(\frac{1}{2\rho} + \sum_{(\mathbf{y}_j, m) \in \mathcal{N}_w(\mathbf{y}_i, n)} \rho w[\mathbf{y}_j, n, k-1]) \tag{5.31}$$

and

$$v[\mathbf{z}_i, n, k] = \frac{1}{2} v[\mathbf{z}_i, n, k-1] + \frac{c_v}{2}(\frac{1}{\rho} + \sum_{\substack{(\mathbf{z}_j, m) \in \mathcal{N}_v(\mathbf{z}_i, n) \\ n=m}} \rho v[\mathbf{z}_j, n, k-1] + \sum_{\substack{(\mathbf{z}_j, m) \in \mathcal{N}_v(\mathbf{z}_i, n) \\ \mathbf{z}_j = \mathbf{z}_i}} \rho v[\mathbf{z}_i, n, k-1]) \tag{5.32}$$

respectively.

Because the term $1/|F[\mathbf{x}_i, n, k-1]|$ in c_F in Equation 5.28 is not bounded as $|F[\mathbf{x}_i, n, k-1]|$ approaches zero, it is now replaced with the term $1/(|F[\mathbf{x}_i, n, k-1]| + \epsilon)$, where ϵ is chosen to be a small positive real number. This choice for c_F results in the iterative equation,

$$\begin{aligned} F[\mathbf{x}_i, n, k] = F[\mathbf{x}_i, n, k-1] - \\ c_F \Bigg[\alpha \sum_l \boldsymbol{B}_{l,i} (\sum_j \boldsymbol{A}_{j,l} F[\mathbf{x}_j, n, k-1] - G[\mathbf{x}_l, n]) + \gamma \frac{F[\mathbf{x}_i, n, k-1]}{|F[\mathbf{x}_i, n, k-1]|} + \\ \sum_{(\mathbf{x}_j, m) \in \mathcal{N}_F(\mathbf{x}_i, n)} \beta (F[\mathbf{x}_i, n, k-1] - F[\mathbf{x}_j, n, k-1])(w[\frac{\mathbf{x}_i + \mathbf{x}_j}{2}, n, k-1])^2 + \\ \sum_{(\mathbf{x}_i, m) \in \mathcal{N}_F(\mathbf{x}_i, n)} \beta (F[\mathbf{x}_i, n, k-1] - F[\mathbf{x}_i, m, k-1])(v[\mathbf{x}_i, \frac{n+m}{2}, k-1])^2 \Bigg], \end{aligned} \tag{5.33}$$

This choice for c_F causes the gradient step to decrease as $F[\mathbf{x}_i, n, k-1]$ gets closer and closer to zero. This prevents the sparsity term from causing the gradient step to overshoot.

5.7 Exploiting Structure of Time Constraints

The process used to solve the problem is now modified to simplify the calculation and to take advantage of the structure of the constraints. Instead of using a Jacobi-type descent a Gauss-Seidel-type method is used that alternates between solving for reconstructions at a individual times and solving for the time coupling term. Note that Equations 5.31-5.33 rely on the value of both F, w at the current, previous, and next time. This suggests a Jacobi-type algorithm in which all the variables at each spatial location and time at iteration $(k-1)$ are saved for determining the variables at the next iteration (k). This type of algorithm requires keeping two copies of the entire state F,w, and v, one at iteration (k) and another at iteration $(k-1)$. This doubles the memory requirements of this already demanding application. An alternative to the Jacobi-type descent is to use a method similar to the Gauss-Seidel [7] method to solve this coupled PDE. Extending the Gauss-Seidel type algorithm to the gradient descent use of the Euler equations does not change the values of c_f, c_w, or the recursive calculation of w. The equation

$$
\begin{aligned}
F[\mathbf{x}_i,n,k] = F[\mathbf{x}_i,n,k-1] - \\
c_F \Bigg[\alpha \sum_l \boldsymbol{B}_{l,i} (\sum_j \boldsymbol{A}_{j,l} F[\mathbf{x}_j,n,k-1] - G[\mathbf{x}_l,n]) + \gamma \frac{F[\mathbf{x}_i,n,k-1]}{|F[\mathbf{x}_i,n,k-1]|} + \\
\sum_{(\mathbf{x}_j,m)\in\mathscr{N}_F(\mathbf{x}_i,n)} \beta(F[\mathbf{x}_i,n,k-1] - F[\mathbf{x}_j,n,k-1])(w[\frac{\mathbf{x}_i+\mathbf{x}_j}{2},n,k-1])^2 + \\
\beta(F[\mathbf{x}_i,n,k-1] - F[\mathbf{x}_i,n+1,k-1])(v[\mathbf{x}_i,n+\frac{1}{2},k-1])^2 + \\
\beta(F[\mathbf{x}_i,n,k-1] - F[\mathbf{x}_i,n-1,k])(v[\mathbf{x}_i,n-\frac{1}{2},k-1])^2 \Bigg] . \quad (5.34)
\end{aligned}
$$

is the same as Equation 5.33 except in the last line where the value of the volume at the previous time instance is at the current iteration $F[\mathbf{x}_i,n-1,k]$ instead of the previous iteration $F[\mathbf{x}_i,n-1,k-1]$. This reduces the memory needs of the algorithm and provides faster convergence by allowing the value of $F[\mathbf{x}_i,n-1,k-1]$ to be discarded after $F[\mathbf{x}_i,n-1,k]$ is determined for all values of $\mathbf{x}_i$. This process results

in a sequential algorithm that first performs a single iteration of the reconstruction of volume F and edge function w at time one. Next, the new values of F and w for time one are saved over the previous values. The process continues at time two with a single iteration of the reconstruction of F and w that is saved at completion. Subsequently, a single iteration is performed of the reconstruction of v at time 1.5 and saved at completion. After v is determined, the values F and w at time three are moved forward an iteration and saved. The algorithm then returns to determining v at time 2.5. When the end of the time series is reached, the algorithm returns to the beginning of the time series.

Similar to the substitution in Equation 5.34, the equation

$$v[\mathbf{z}_i, n, k] = \frac{1}{2} v[\mathbf{z}_i, n, k-1] + \frac{c_v}{2}\Big(\frac{1}{\rho} + \sum_{\substack{(\mathbf{z}_j, m) \in \mathcal{N}_v(\mathbf{z}_i, n) \\ n=m}} \rho v[\mathbf{z}_j, n, k-1] + \rho v[\mathbf{z}_i, n-1, k] + \rho v[\mathbf{z}_i, n+1, k-1]\Big), \quad (5.35)$$

substitutes the term $v[\mathbf{z}_i, n-1, k-1]$ in Equation 5.32 with $v[\mathbf{z}_i, n-1, k]$. The expression for finding c_v changes to

$$c_v = \frac{1}{2}\Big(\beta\big(F[\mathbf{z}_i, n+\frac{1}{2}, k] - F[\mathbf{z}_i, n-\frac{1}{2}, k]\big)^2 + \frac{1}{2\rho} + 8\rho\Big)^{-1} \quad (5.36)$$

by using the values of F at iteration (k) instead of $(k-1)$. These changes allow the edge term v to be calculated at iteration (k) after the kth iteration of F has been determined at the adjacent times. As with F, only the most recent iterations for the two adjacent times of F and w are required to determine $v[\mathbf{z}_i, n, k]$.

The Gauss-Seidel type algorithm provides an in place algorithm for finding the reconstruction of F, w, and v at each time. This process can be extended to further decouple the determination of v at each time. Specifically, given v for each spatial location at time $n + 1/2$ and $n - 1/2$, $w[\mathbf{y}_i, n, k+p]$ and $F[\mathbf{x}_i, n, k+p]$ can be determined for several gradient descent iterations instead of just one. The choice of P determines the range $p \in 1, 2, \cdots, P$. The number of iterations P can be

set to avoid the cost of switching from the determination of both w and F to the determination of $v[\mathbf{z}_i, n-1/2, k+p]$. For P iterations, this process decouples the individual 3D reconstructions from the smoothing of each spatial location across time. The reconstruction still begins at time one, but now P iterations are performed at each time instead of just one. The resulting algorithm for simultaneously determining $F[\mathbf{x}_i, n, k+p]$ and $w[\mathbf{y}_i, n, k+p]$ is the evaluation of

$$\begin{aligned}
F[\mathbf{x}_i, n, k+p] &= F[\mathbf{x}_i, n, k+p-1] - \\
c_F \Bigg[&\alpha \sum_l \boldsymbol{B}_{l,i} (\sum_j \boldsymbol{A}_{j,l} F[\mathbf{x}_j, n, k+p-1] - G[\mathbf{x}_l, n]) + \gamma \frac{F[\mathbf{x}_i, n, k+p-1]}{|F[\mathbf{x}_i, n, k+p-1]|} + \\
&\sum_{(\mathbf{x}_j, m) \in \mathcal{N}_F(\mathbf{x}_i, n)} \beta (F[\mathbf{x}_i, n, k+p-1] - F[\mathbf{x}_j, n, k+p-1]) (w[\frac{\mathbf{x}_i + \mathbf{x}_j}{2}, n, k+p-1])^2 + \\
&\beta (F[\mathbf{x}_i, n, k+p-1] - F[\mathbf{x}_i, n+1, k]) (v[\mathbf{x}_i, n+\frac{1}{2}, k])^2 + \\
&\beta (F[\mathbf{x}_i, n, k+p-1] - F[\mathbf{x}_i, n-1, k+P]) (v[\mathbf{x}_i, n-\frac{1}{2}, k])^2 \Bigg] \quad (5.37)
\end{aligned}$$

and

$$w[\mathbf{y}_i, n, k+p] = \frac{1}{2} w[\mathbf{y}_i, n, k+p-1] + c_w (\frac{1}{2\rho} + \sum_{(\mathbf{y}_j, m) \in \mathcal{N}_w(\mathbf{y}_i, n)} \rho w[\mathbf{y}_j, n, k+p-1]) \quad (5.38)$$

using the weighting terms

$$c_f = \frac{1}{2} (\alpha + 8\beta + \frac{\gamma}{|F[\mathbf{x}_i, n, k+p-1]| + \epsilon})^{-1} \quad (5.39)$$

and

$$c_w = \frac{1}{2} (\beta (F[\mathbf{x}_l, n, k+p-1] - F[\mathbf{x}_m, n, k+p-1])^2 + \frac{1}{2\rho} + 8\rho)^{-1}. \quad (5.40)$$

After the variables $F[\mathbf{z}_i, n-\frac{1}{2}, k+P]$ and $F[\mathbf{z}_i, n+\frac{1}{2}, k+P]$ are obtained using P

iterations of Equations 5.37 and 5.38, the term

$$v[\mathbf{z}_i, n, k+p] = \frac{1}{2}v[\mathbf{z}_i, n, k+p-1] + \frac{c_v}{2}(\frac{1}{\rho} + \sum_{\substack{(\mathbf{z}_j,m)\in\mathcal{N}_v(\mathbf{z}_i,n) \\ n=m}} \rho v[\mathbf{z}_j, n, k+p-1] + \rho v[\mathbf{z}_i, n-1, k+P] + \rho v[\mathbf{z}_i, n+1, k]), \quad (5.41)$$

is evaluated for P iterations using the weight

$$c_v = \frac{1}{2}(\beta(F[\mathbf{z}_i, n+\frac{1}{2}, k+P] - F[\mathbf{z}_i, n-\frac{1}{2}, k+P])^2 + \frac{1}{2\rho} + 8\rho)^{-1}. \quad (5.42)$$

In order to start the now completed algorithm F and w are initialized to one and v is initialized to zero.

5.8 Discussion

The spatio-temporal data fusion algorithm breaks up the total 3D time series reconstruction into several successive 3D reconstructions. This process replaces the Jacobi iteration with a Gauss-Seidel-type iteration that alternates between solving for the individual 3D reconstructions and in solving for the temporal edge term. This decoupling allows the algorithm to use less memory by writing variables to disk that are complete after the final iteration on a volume and by loading needed variables from disk before iterations are run on the next volume. By adapting the algorithm to the Gauss-Seidel-based method the memory requirements were reduced from 10 gigabytes to 600 megabytes for a series of 10 volumes.

The initial condition of $F = 1$ was used for the spatio-temporal data fusion. The use of other initial conditions change the algorithms speed and point convergence. For example, if F is set to zero instead of one, the convergence is very slow because the $1/(F+\epsilon)$ term in c_F makes the gradient step very small. A corollary to this observation is that larger values of F yield bigger gradient steps. As desired, this difference in speed causes the functional to favor regions with smaller support.

A few alternates to the forward direction algorithm specified in this chapter are

now discussed. The algorithm can be run in the reverse direction (starting at the last time and moving backward in time) or with alternating forward and reverse sweeps. Because each volume is linked to its immediate neighbor, another interesting choice is to calculate every other volume in a single sweep (e.g., the odd times) and then go back to calculate the volumes at the remaining times (e.g., the even times) in a second sweep. This type of algorithm is similar to a red-black Gauss-Seidel [12] algorithm, where every other item is labeled red and each of its neighbors are labeled black. Such an algorithm consists of "red" iterations followed by "black" iterations. Determining each volume of a given color is independent of all other volumes of that same color. An algorithm of this form is parallelizable and can lead to significant gains in computational performance.

The spatio-temporal data fusion algorithm provides a memory efficient and stable method for reconstructing the blood flow. Restriction of the reconstruction to the map allows for further optimizations that yield improved speed and reduced memory constraints.

Equation 5.8 is an approximation to the expression $E(F, \Gamma_x, \Gamma_t)$, where Γ_x is the set of edges in space and Γ_t is the set of edges in time. Similarly to what was mentioned in Section 2.3, the expression $\exp(-E(F, \Gamma_x, \Gamma_t))$ can be interpreted as a density with respect to an approximate measure (F, Γ_x, Γ_t). The values of (F, Γ_x, Γ_t) that minimize $E(F, \Gamma_x, \Gamma_t)$ maximize the distribution and are thus the Maximum A Posteriori estimates. Because the reconstruction is merely an estimate, it is necessary to understand its distribution and confidence intervals before any assertion can be made about that reconstruction. An example of this is combining features about arrival times of contrast from different reconstructed time series of the same patient. As suggested in [78], the use of Monte-Carlo might provide the necessary information about the distribution.

Chapter 6

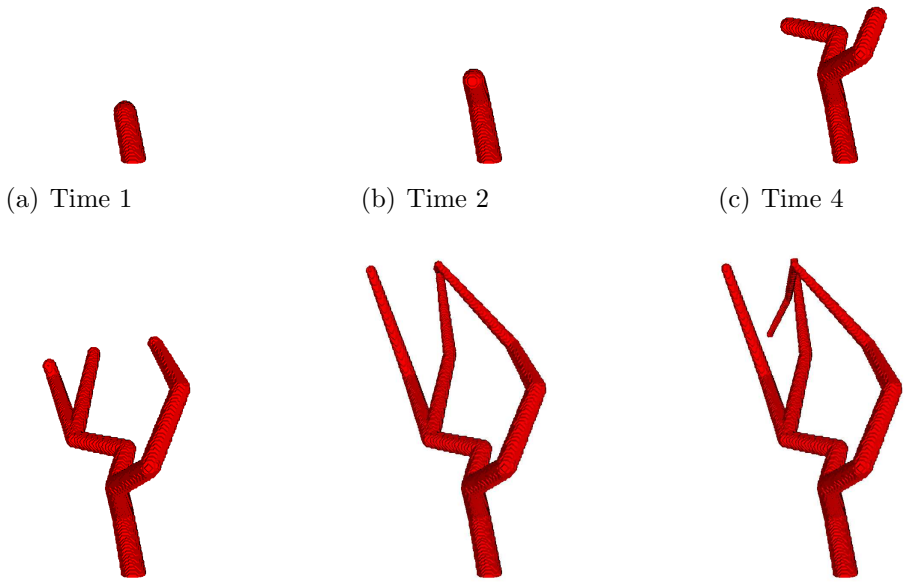
4D Results and Validation

This chapter shows results for the spatio-temporal data fusion algorithm on both real angiographic data and synthetic/phantom data. Section 6.1 presents the synthetic phantom data sets, which include a test case both before and after a simulated clot is introduced, used to validate the algorithms and the results of the overall fusion algorithm. In Section 6.2, results are presented on data from two patients. The data for patient A contains a time series acquired at a high frame rate, while data for patient B contains time sequences acquired at a lower frame rate taken both before and after the patient developed a clot.

6.1 Validation on Phantom Data

Figure 6-1 shows a visualization of the isosurfaces placed on the underlying time series of phantom volumes that are used to validate the spatio-temporal data fusion algorithm. The volumes are used to generate 2D time series. This allows the accuracy of reconstructions using the spatio-temporal data fusion algorithm to be compared against a ground truth. The many plots in Figure 6-1 can be summarized in a single surface mesh that uses color to represent the time information in the time series as shown in Figure 6-2. This figure contains the arrival time of contrast in 3D from four different views.

The volumes visualized in Figure 6-1 and 6-2 can be projected to produce digital



(a) Time 1 (b) Time 2 (c) Time 4

(d) Time 6 (e) Time 8 (f) Time 10

Figure 6-1: **Original phantom time series** – A visualization of six of the ten test volumes that are used to validate the reconstruction algorithm.

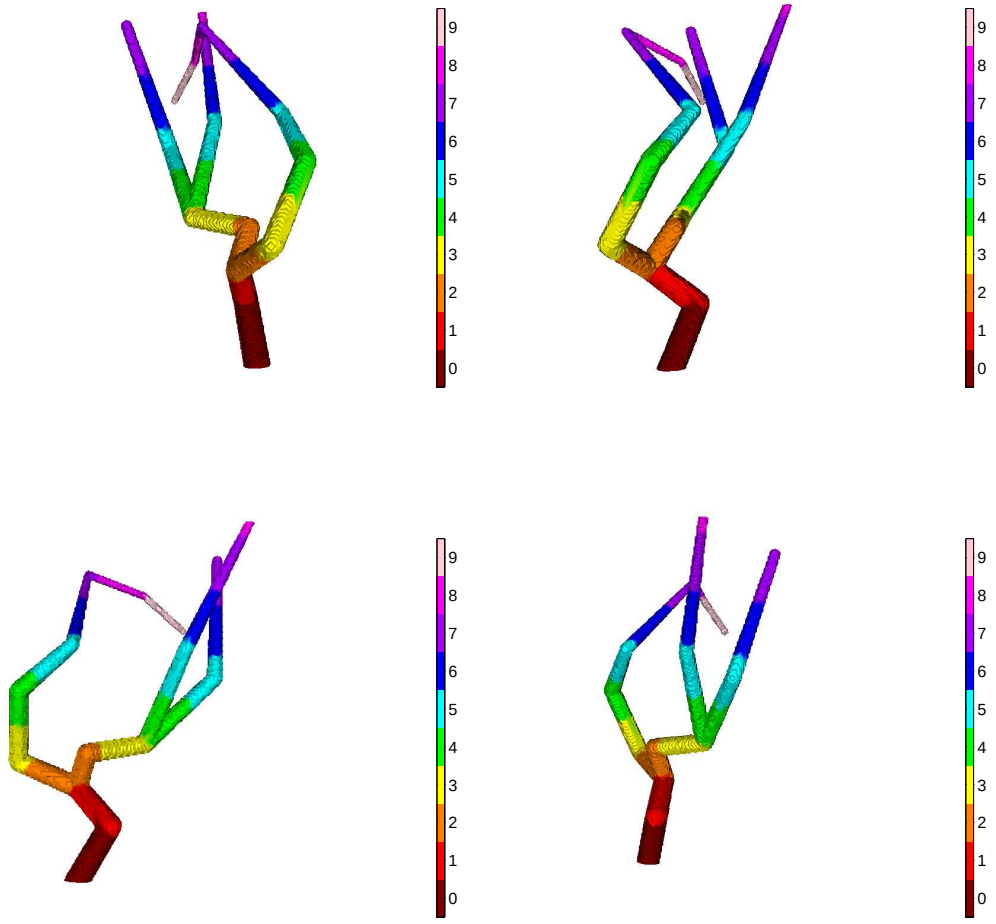


Figure 6-2: **3D arrival times for original phantom time series** – Visualization of the underlying phantom time series using color to denote the arrival time of contrast at the different points on the surface of the mesh. This visualization is shown for four separate views. The units of time are in frames.

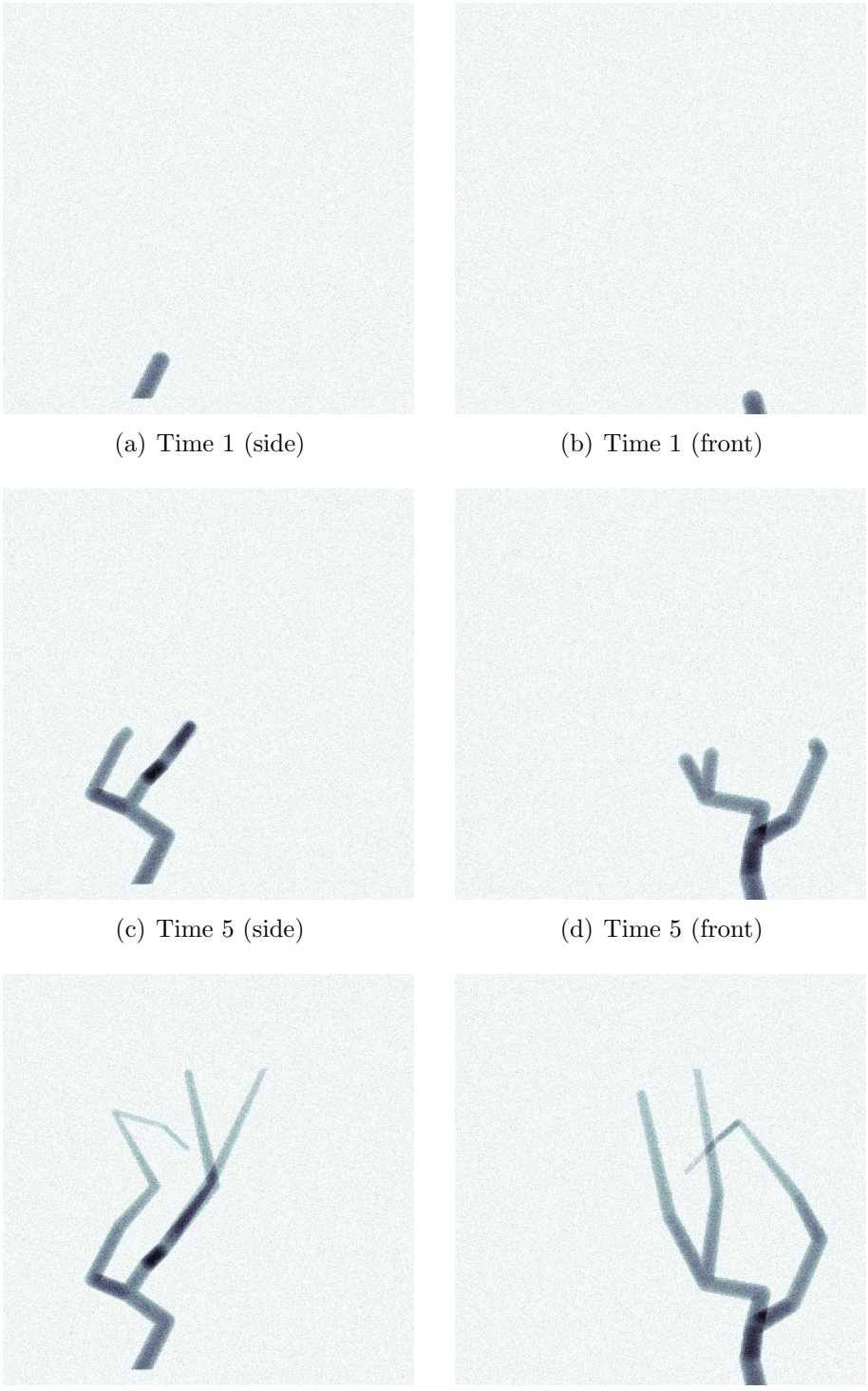
reconstructed radiographs from any view. Figure 6-3 shows a few such images after Gaussian noise has been added that are taken from both side and front views. These time series can then be reconstructed using the spatio-temporal data fusion algorithm.

A visualization of the reconstruction produced by the spatio-temporal data fusion algorithm applied to the phantom time series is shown in Figure 6-4 using a series of isosurfaces and in Figure 6-5 using color to denote the arrival time of the simulated contrast. The results in Figure 6-4 are similar to the original data shown in 6-1. The reconstructed volumes contain a few points that are above the threshold used to generate the isosurfaces. These points appear as red dots in the individual images. Figure 6-5 shows the arrival times based on the reconstructed time series at or near the surface mesh. A few locations contain erroneous arrival times due to noise in the reconstruction observed in Figure 6-4 as red dots.

Figure 6-6 shows the effect of time coupling on the reconstruction of the phantom data set. The left panels show the reconstruction where the spatio-temporal data fusion algorithm was used and the right panels show the reconstruction when a modified version of the algorithm with no time coupling was used. The time coupling reduces the amount of incorrectly labeled times by smoothing away isolated points in time. Even though the coupling improves the performance of the algorithm, both algorithms provide results similar to the original data shown in Figure 6-2.

Figure 6-7 provides a visualization of a second set of phantom data in which a simulated clot is introduced. For comparison, the panels on the left are as they were in Figure 6-2 and the panels on the right contain the simulated clot. The middle of the three branches contain arrival times that are delayed in the right panels relative to the arrival times in the left panels. A similar comparison is provided for the projected phantom data at frame ten in Figure 6-8.

Figure 6-9 provides a similar comparison to that in Figure 6-7, where instead the reconstructions are compared. The delay in the middle branch shown in the right panels is easily observed in the reconstructed data and consistent to that in Figure 6-7. These reconstructions show the accuracy of the algorithm along with an example



(a) Time 1 (side)

(b) Time 1 (front)

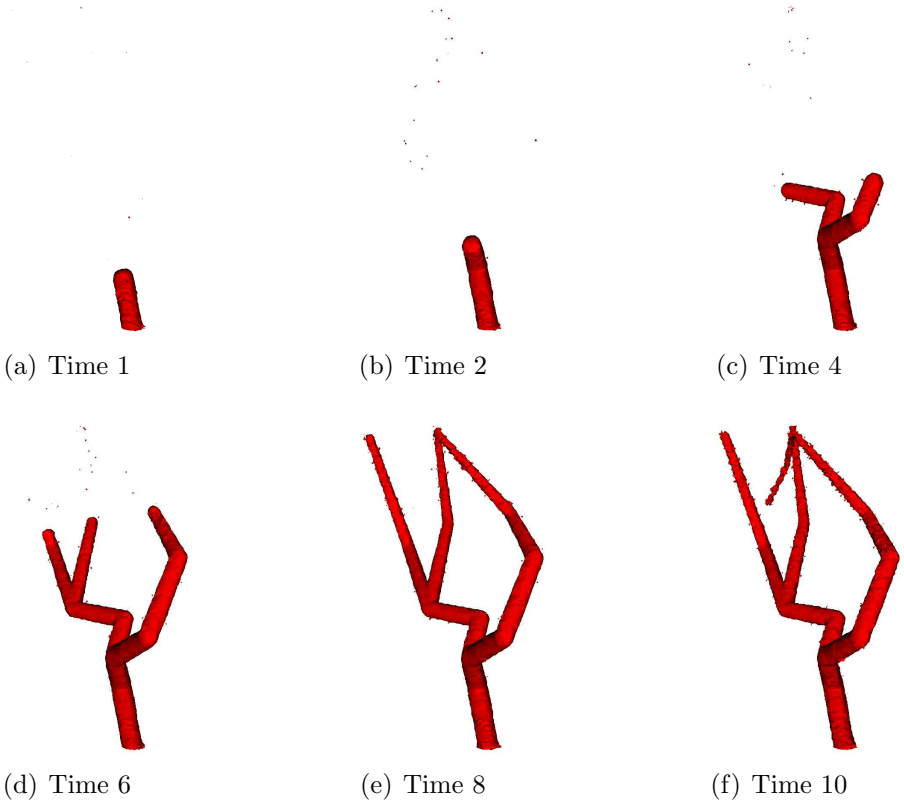
(c) Time 5 (side)

(d) Time 5 (front)

(e) Time 10 (side)

(f) Time 10 (front)

Figure 6-3: **Phantom time series projections** – Digital reconstructed radiographs of the underlying 3D phantom time series from two separate views. The left panels are from the side view and the right panels are from the front view. Gaussian noise has been added to the projections.



(a) Time 1 (b) Time 2 (c) Time 4

(d) Time 6 (e) Time 8 (f) Time 10

Figure 6-4: **Reconstructed phantom time series** – A visualization of reconstructions of six of the ten phantom volumes that are used to validate the Spatio-Temporal Data Fusion algorithm.

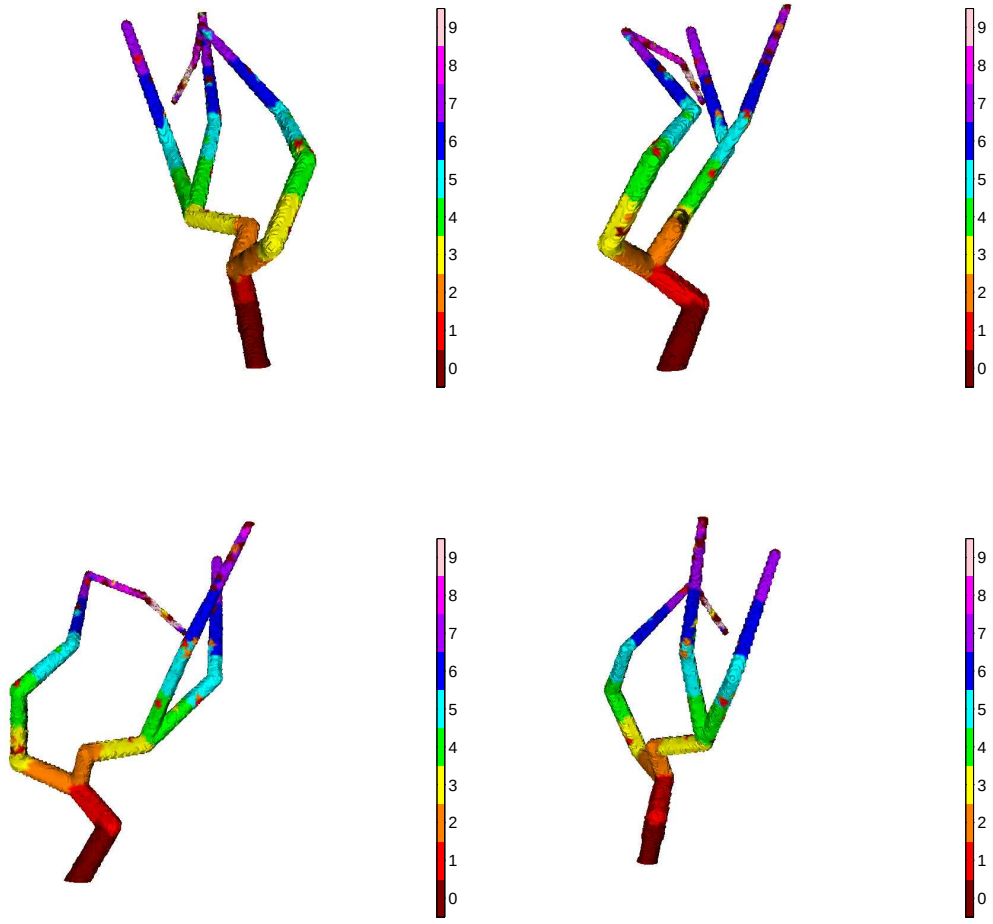
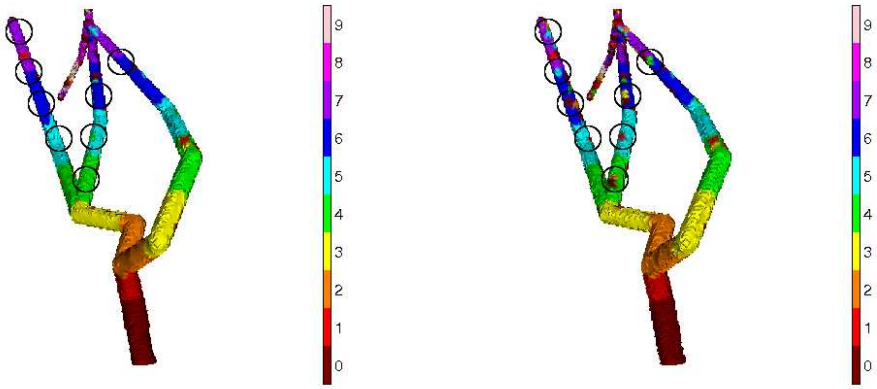
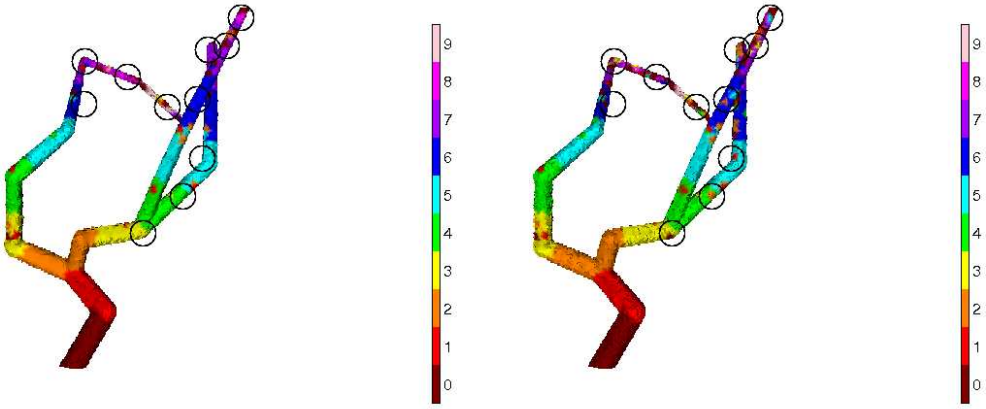


Figure 6-5: **3D arrival times for reconstructed phantom time series** – The visualization of the reconstructed phantom time series using color to denote the arrival time of contrast at the different points on the surface of the mesh. This visualization is shown from four separate views. The units of time are in frames.



(a) With Coupling

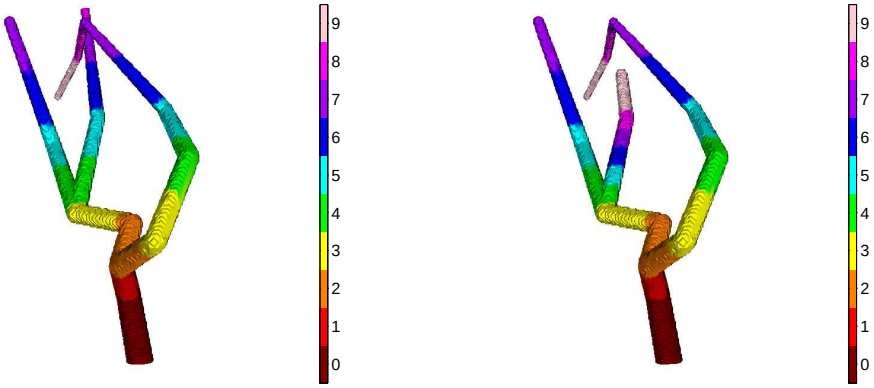
(b) Without Coupling



(c) With Coupling

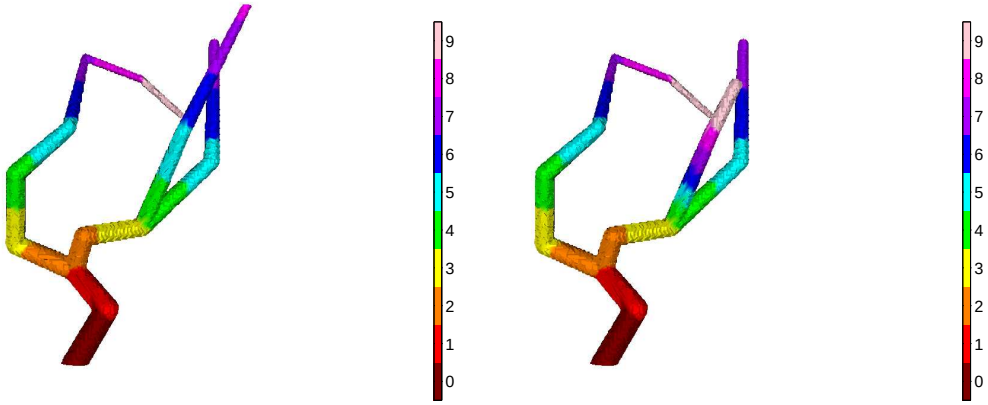
(d) Without Coupling

Figure 6-6: **Effect of time coupling on reconstructed phantom series** – 3D time series visualization for the reconstructed phantom time series using time coupling (left) and not using time coupling (right). Circles have been added to the images to highlight locations that both did not contain errors in the reconstruction with time coupling and contained errors in the reconstruction without time coupling. The visualizations are shown for two separate views. The units of time are in frames.



(a) No Clot

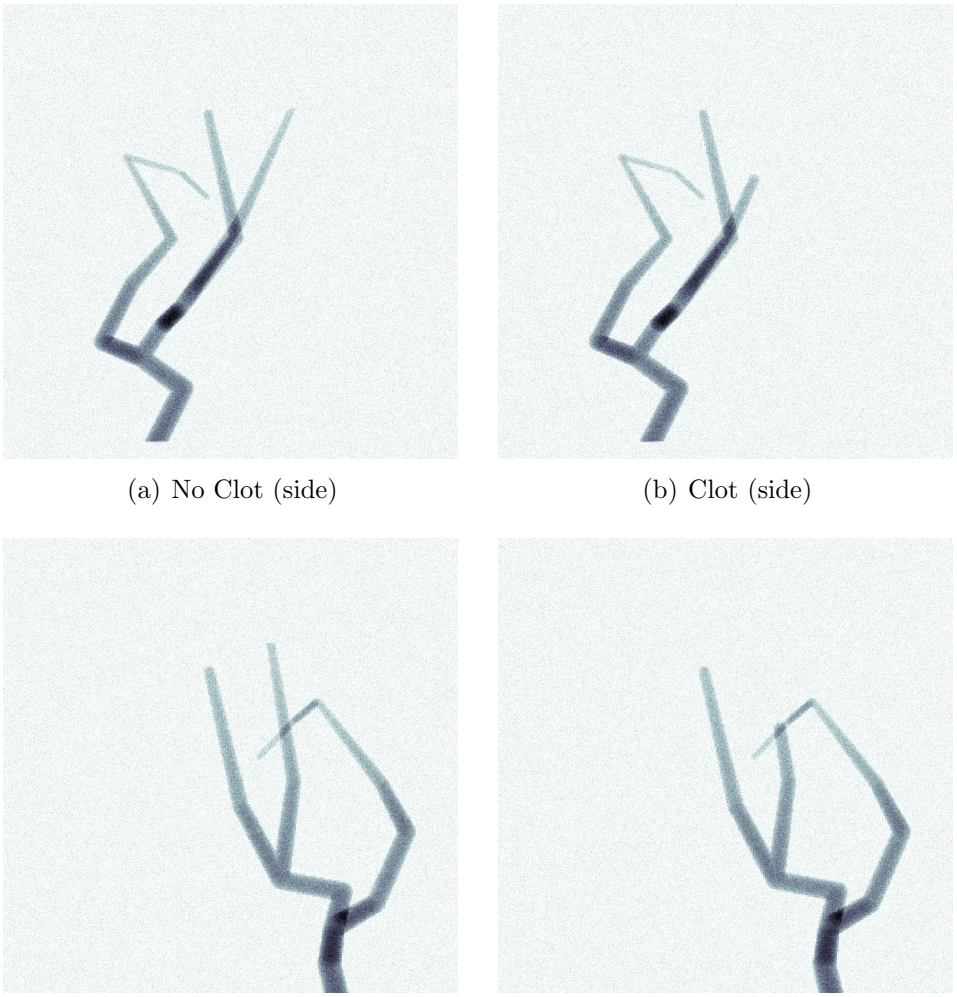
(b) Clot



(c) No Clot

(d) Clot

Figure 6-7: **3D arrival times for both before and after a simulated clot** – Visualization of the underlying phantom time series with (right) and without (left) a simulated clot from two views. Color is used denote the arrival time of contrast at the different points on the surface of the mesh. The units of time are in frames.



(a) No Clot (side)

(b) Clot (side)

(c) No Clot (front)

(d) Clot (front)

Figure 6-8: **Projections of the phantom time series both before and after a simulated clot** – Digital reconstructed radiographs of the underlying 3D phantom time series with (right) and without (left) a simulated clot at frame 10. The top panels are from the side view and the bottom panels are from the front view. Gaussian noise has been added to the projections to make them more realistic.

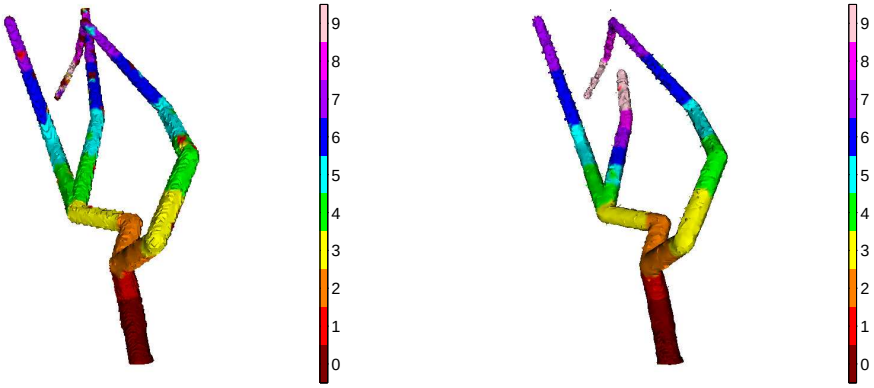
application where the results of the algorithm can be used to detect a simulated clot. In the next section, a similar example on clinical data is presented.

6.2 Angiographic Data

Figure 6-10 provides a visualization of the vasculature map for patient A that is used as an input to the spatio-temporal data fusion algorithm. The map is accurate for the larger vessels, but there are some noticeable errors in the smaller vessels. Two examples of the errors are the several snippets of vessels that are not connected to the main network of vessels and vessels that are chopped because they extend beyond the boundary of the reconstructed map. The inaccuracies in the map appear in the reconstruction, and therefore limit the quality of the reconstruction.

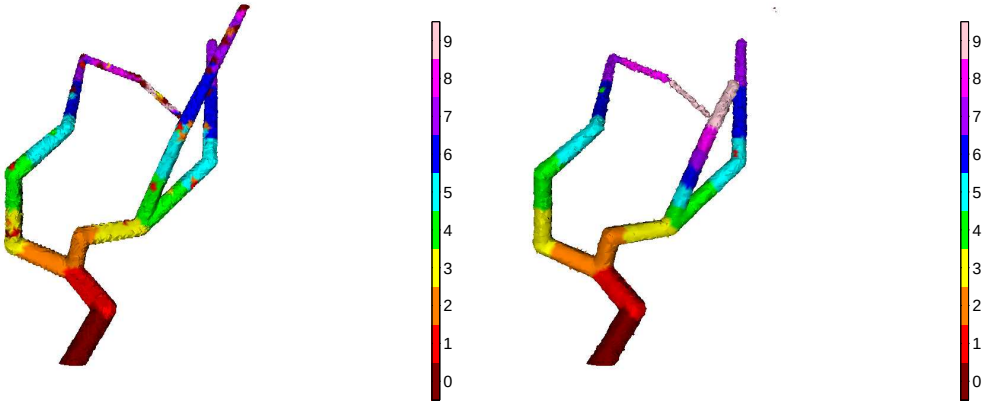
A sampling of the original (left panels) and reconstructed (right panels) 2D image sequences of patient A are shown in Figure 6-11 from the front view and in Figure 6-12 from the side view. The original projections provided the additional set of data needed for the spatio-temporal data fusion algorithm. As desired, the reconstructed image sequences are similar to the original measured image sequences. The few differences between the original and reconstructed time series are due to certain vessels not appearing within the original 3D map of Figure 6-10, the small delay between the acquisition times of the image pair (the front view was delayed slightly from the side view), and the regularization terms used in the reconstructions.

Figure 6-13 shows a visualization of the reconstruction of a 3D time series of real angiographic data at nine different times. The figure shows the result of rendering a mesh at an isosurface based on a particular threshold. These panels show the outward flow of blood as time progresses. This flow is consistent with the type of blood flow that is expected, as well as what is observed in the 2D time series. The reconstruction results in intensities in space and time that can be viewed by taking a slice from a particular plane as is shown in Figure 6-14. The panels show the increasing intensity in the main vessel and the appearance of contrast in other vessels as time progresses. Viewing the intensities in this way is useful but it does not provide a picture of



(a) No Clot

(b) Clot



(c) No Clot

(d) Clot

Figure 6-9: **3D arrival times of reconstructed phantom both before and after a simulated clot** – Visualization of the reconstructed phantom time series with (right) and without (left) a simulated clot from two views. Color is used denote the arrival time of contrast at the different points on the surface of the mesh. The units of time are in frames.

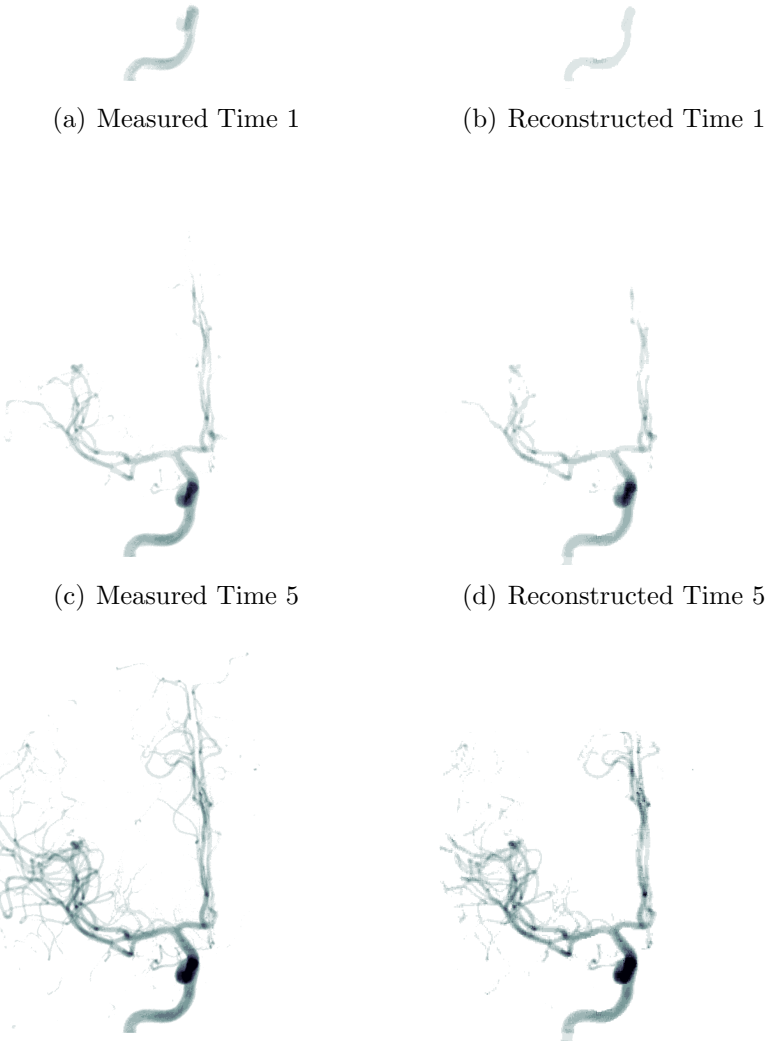
Figure 6-10: **3D map visualization of patient A** – The visualization of the 3D map generated from patient A that is used in Spatio-Temporal Data Fusion. Two separate views are shown.

time flow of blood in 3D. A better way of visualizing the 3D flow is using the color coded arrival times, as shown in Figure 6-15. The color coded arrival times shows the advancing front of the contrast. This allows the 3D flow of the blood to be accessed in a single surface from a multitude of views.

Figure 6-16 shows the impact of varying the threshold used in the mesh generation algorithm used to visualize the reconstruction at time 8. As the threshold is lowered more of the blood vessel can be seen, but at the same time more artifacts appear. Because of this tradeoff an appropriate threshold needs to be carefully chosen. The figure also gives a visualization of where the large values are in the reconstruction. This appearance is consistent with the reconstructions within the 2D images in Chapter 5.

Figure 6-17 shows the visualization of the reconstruction at time 8 after different numbers of iterations. After 10 iterations, not all of the blood vessels are apparent and by 30 or 40 iterations, the reconstruction is stable and does not change significantly with the addition of more iterations. This suggests that the algorithm converges rapidly.

Because intensities are produced in the reconstructions, projections or DRRs can

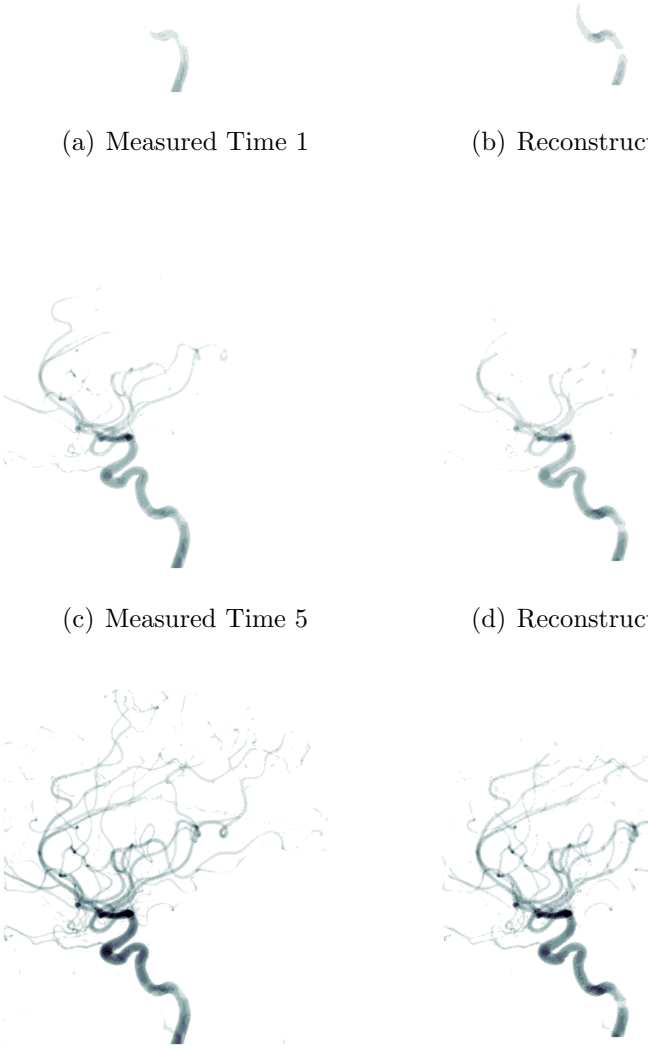


(a) Measured Time 1 (b) Reconstructed Time 1

(c) Measured Time 5 (d) Reconstructed Time 5

(e) Measured Time 10 (f) Reconstructed Time 10

Figure 6-11: **Both original and reconstructed time series projections of patient A (front view)** – Original angiographic sequence (left) and digital reconstructed radiographs (right) of the reconstructed 3D time series for patient A. Projections are shown at three separate times.

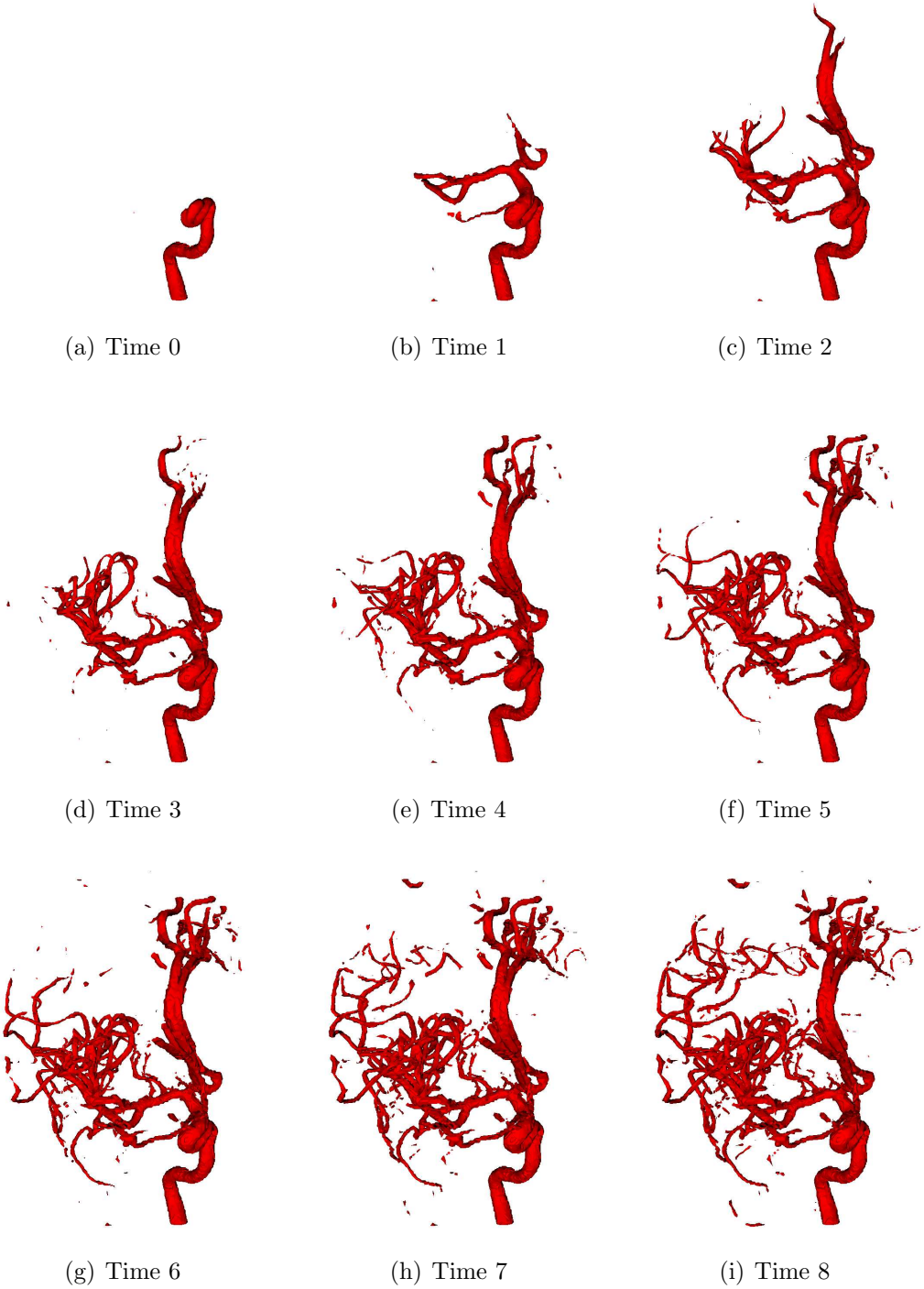


(a) Measured Time 1 (b) Reconstructed Time 1

(c) Measured Time 5 (d) Reconstructed Time 5

(e) Measured Time 10 (f) Reconstructed Time 10

Figure 6-12: **Both original and reconstructed time series projections of patient A (side view)** – Original angiographic sequence (left) and digital reconstructed radiographs (right) of the reconstructed 3D time series for patient A. Projections are shown at three separate times.

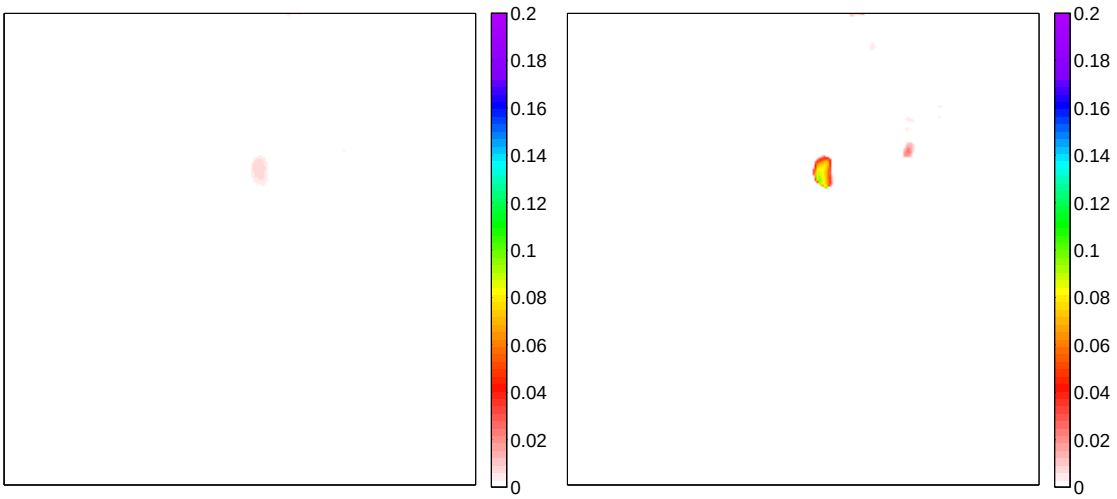


(a) Time 0 (b) Time 1 (c) Time 2

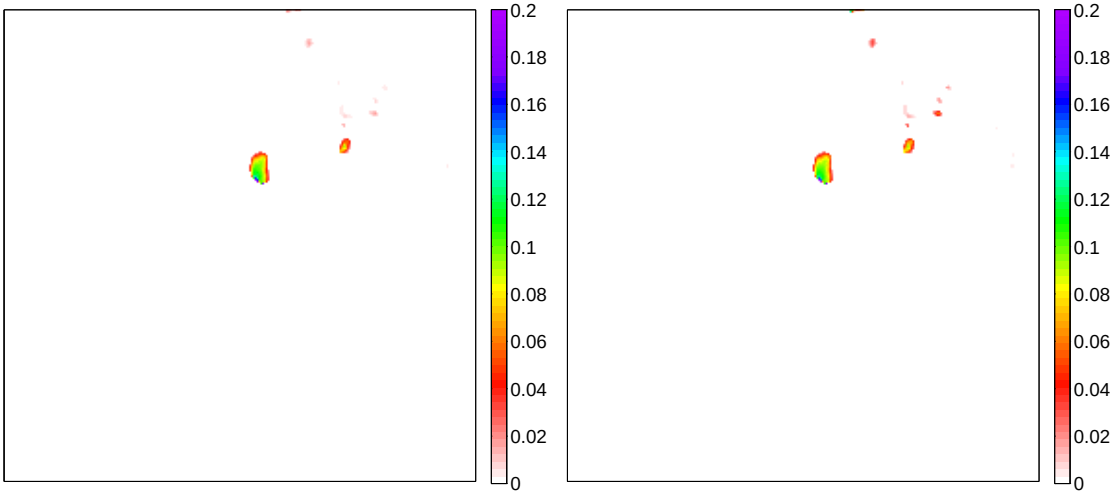
(d) Time 3 (e) Time 4 (f) Time 5

(g) Time 6 (h) Time 7 (i) Time 8

Figure 6-13: **Visualization of reconstructed 3D time series for patient A** – Visualizations of the reconstructions at time 0 through 8. The total observed sequence took one second. The threshold of 0.01 was used to generate the isosurfaces.



(a) Time 1 (b) Time 2



(c) Time 3 (d) Time 4

Figure 6-14: **Slices from reconstructed volumes of patient A** – Slices from the reconstructed time series of Patient A through the plane $z = 80$.

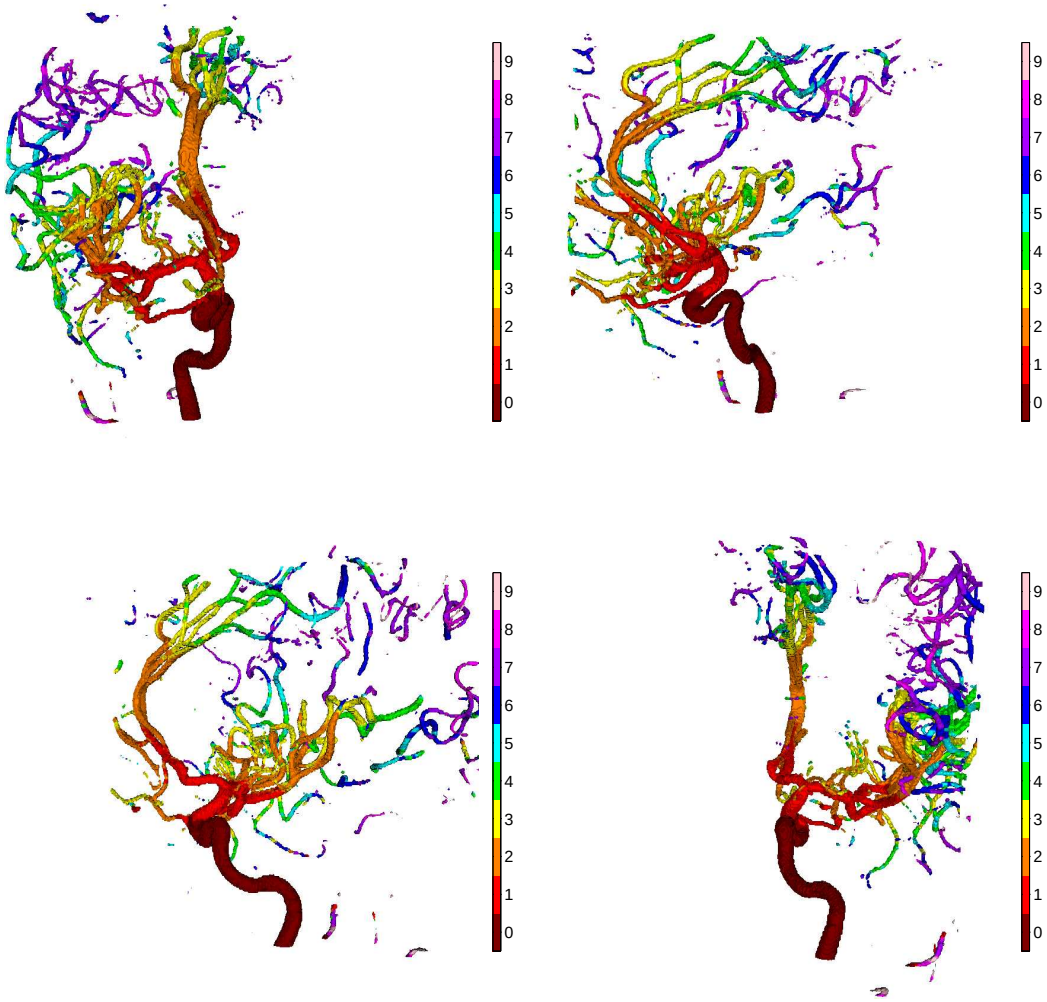
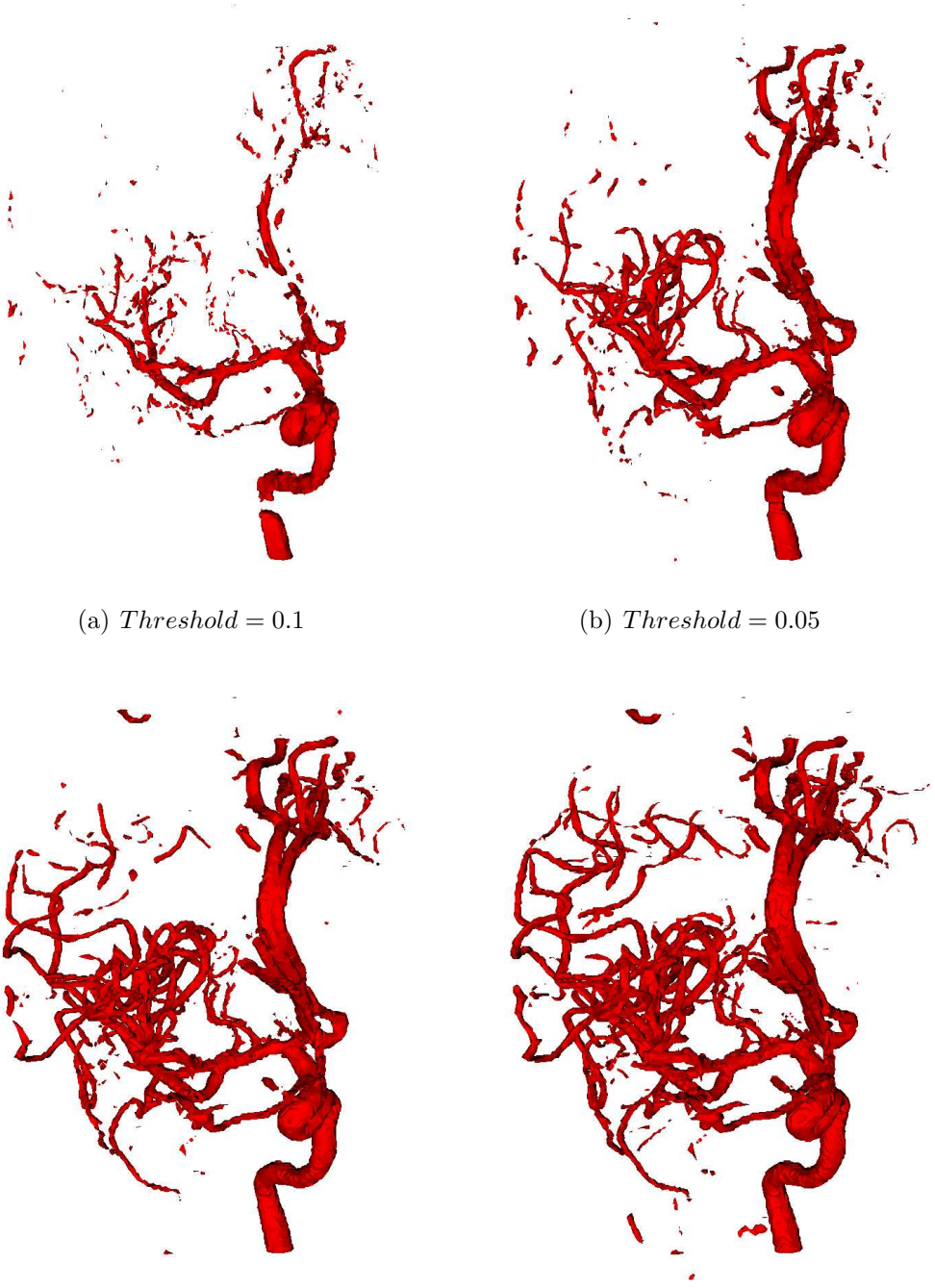


Figure 6-15: **3D time of arrivals for patient A** – The visualization of the 3D time series using color to denote the arrival time of contrast at the different points on the surface of the mesh. This visualization is shown for four separate views. The units of time are in frames sampled at eight hertz.



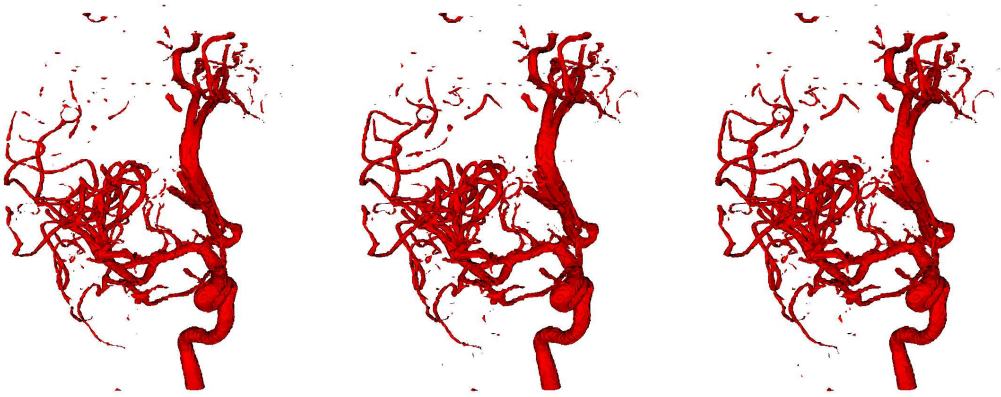
(a) $Threshold = 0.1$

(b) $Threshold = 0.05$

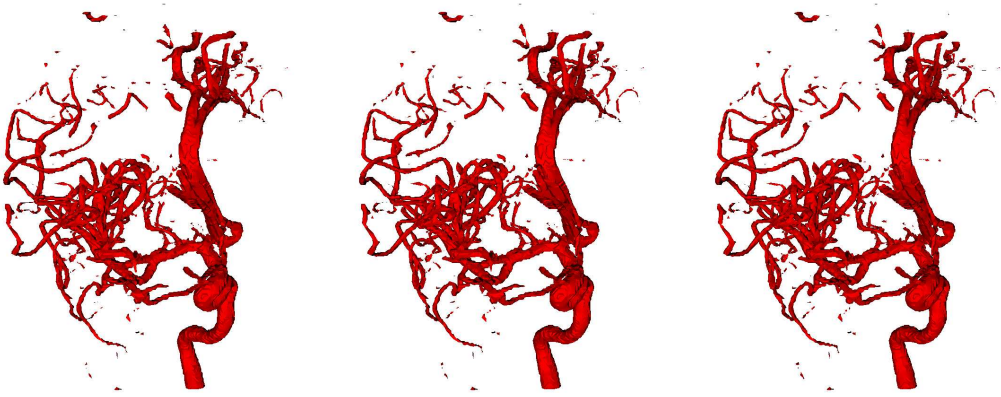
(c) $Threshold = 0.015$

(d) $Threshold = 0.005$

Figure 6-16: **Effect of thresholds on visualization** – The visualization of the reconstruction for patient A at time 8 using four separate thresholds.



(a) Iterations = 10 (b) Iterations = 20 (c) Iterations = 30



(d) Iterations = 40 (e) Iterations = 50 (f) Iterations = 60

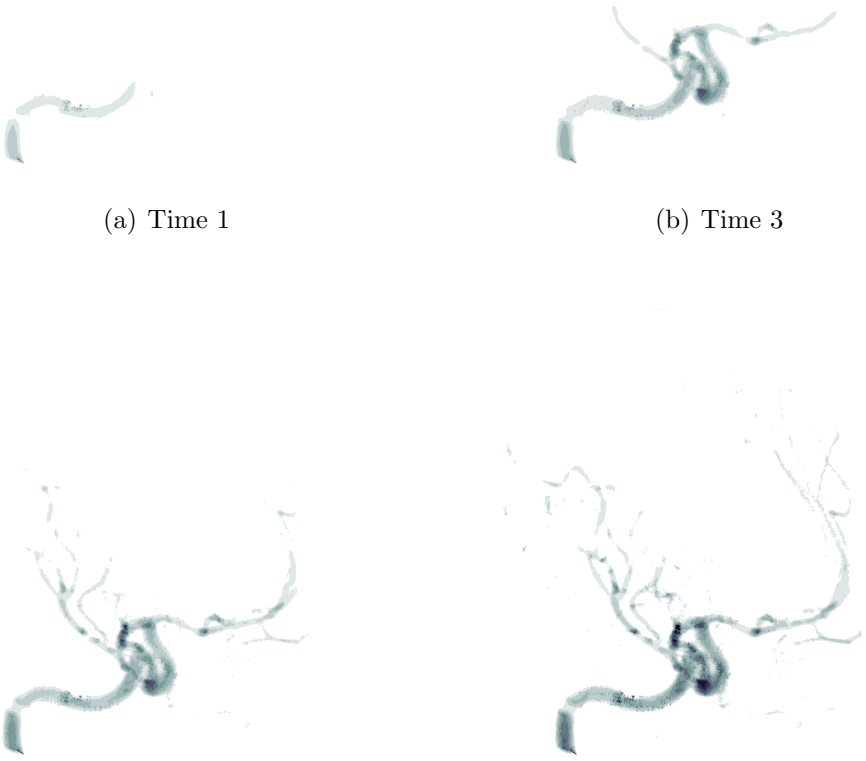
Figure 6-17: **Reconstruction after different iterations** – The visualization of the reconstruction for patient A at time 8 using a threshold of 0.01 after six different iterations have been completed.

be produced from unobserved views. Figure 6-18 shows such a projection from an alternative view based on the 3D series of data captured on patient A. This process allows surgeons to find the best view for observing an area of interest such as one containing a clot or an aneurysm. This allows angiographic time sequences taken from different views to be compared directly. Additionally, views from unobservable views, such as from the top of the patients head, are also possible.

Figure 6-19 provides a visualization of the vasculature map for patient B that is used in the spatio-temporal data fusion algorithm. Example frames of the 2D images both before (left panels) and after (right panels) a clot developed in patient B are shown in Figure 6-20 from the side view and in Figure 6-21 from the front view. The clot occurred in the Middle Cerebral Artery (MCA).

The data sets for patient B were reconstructed and visualized using the color 3D arrival time from different views as shown in Figures 6-22 and 6-23. The panels on the left are for the normal flow while the panels on the right are for flow in the presence of the clot. The arrival times of the blocked artery are delayed from 1-3 frames from what they were in the baseline case. These visualization allow for a straightforward comparison to be made between the two cases. Such a visualization would alert the surgeon to the presence of a problem that can be further assessed and acted on if necessary.

As was mentioned in Section 5.8, it is necessary for a quantitative assessment of the reconstructions before further processing can be done. As in other fields, it is necessary to know the statistics or the distributions of measurements before the measurements can be combined or compared.



(a) Time 1

(b) Time 3

(c) Time 5

(d) Time 7

Figure 6-18: **Reconstructed time series projections of patient A from unobserved view** – Digital reconstructed radiographs of the reconstructed 3D time series for patient A from a view that was not observed in the original angiographic sequences. Projections are shown at four different times.

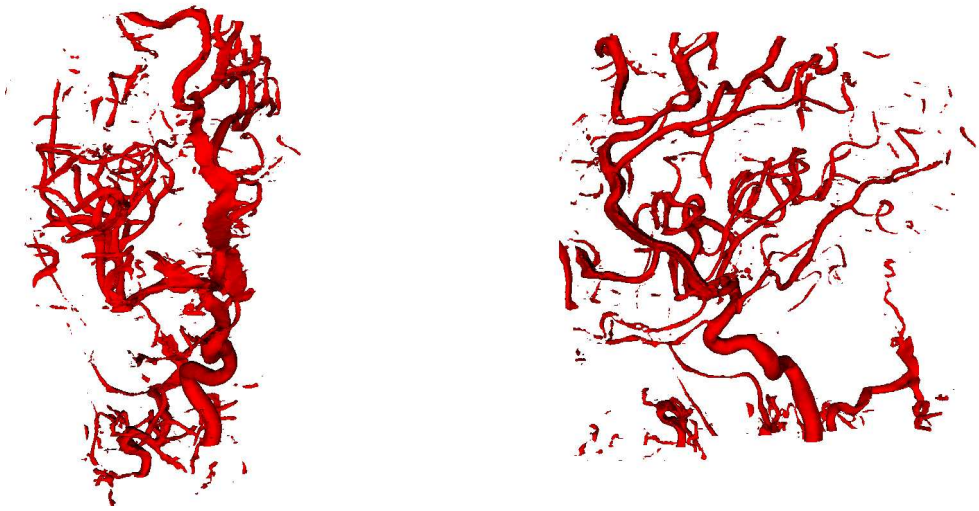
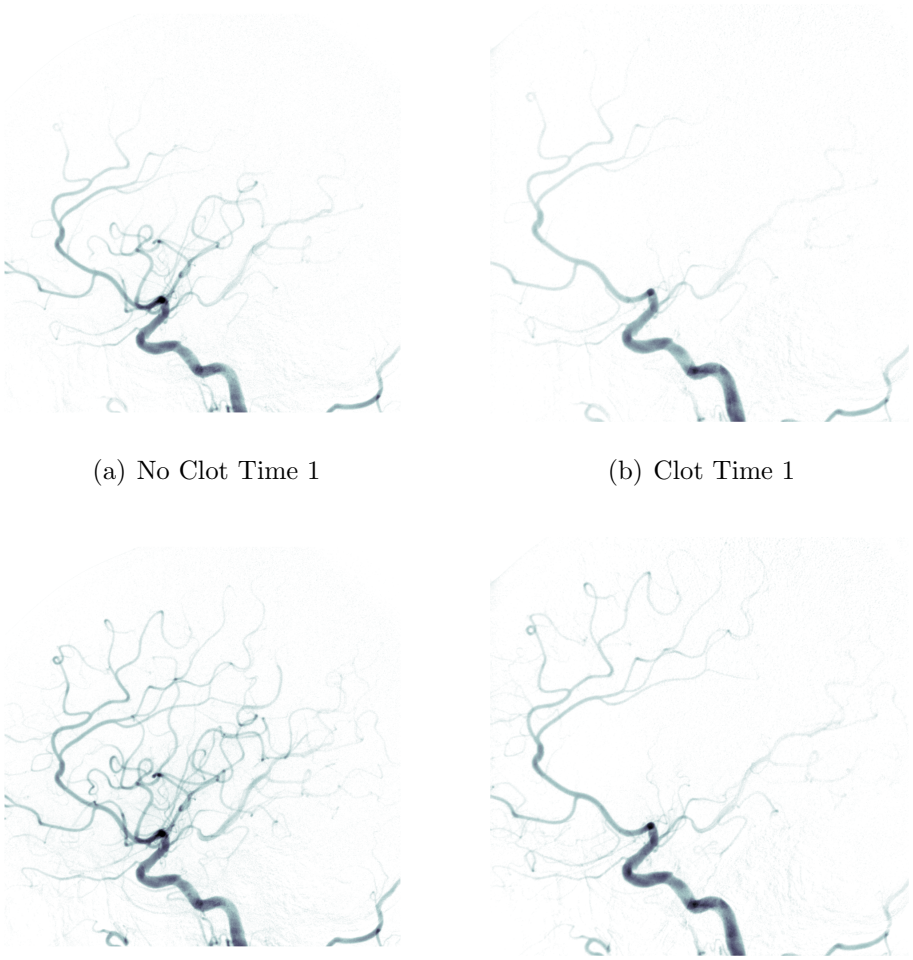


Figure 6-19: **3D map visualization of patient B** – The visualization of the 3D map generated from patient B that is used in Spatio-Temporal Data Fusion.



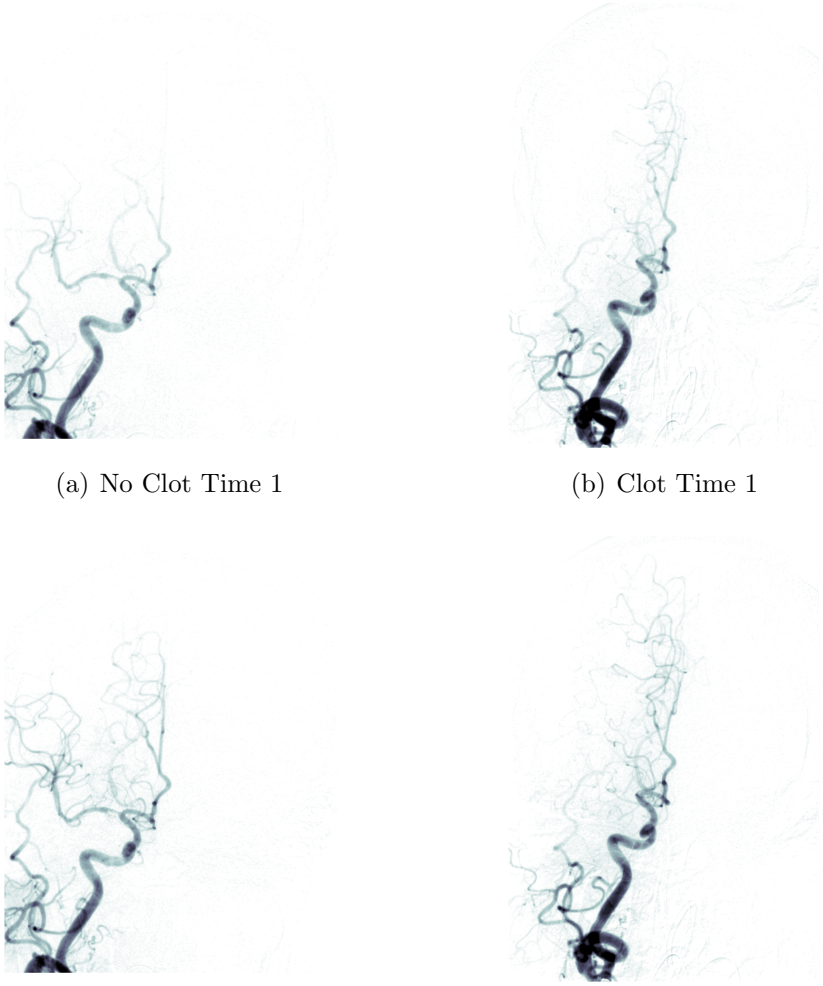
(a) No Clot Time 1

(b) Clot Time 1

(c) No Clot Time 2

(d) Clot Time 2

Figure 6-20: **Angiographic images for patient B both before and after a clot developed (side view)** – Two Angiographic images of patient B before (left) and after (right) a clot developed. These images were taken from the side view.



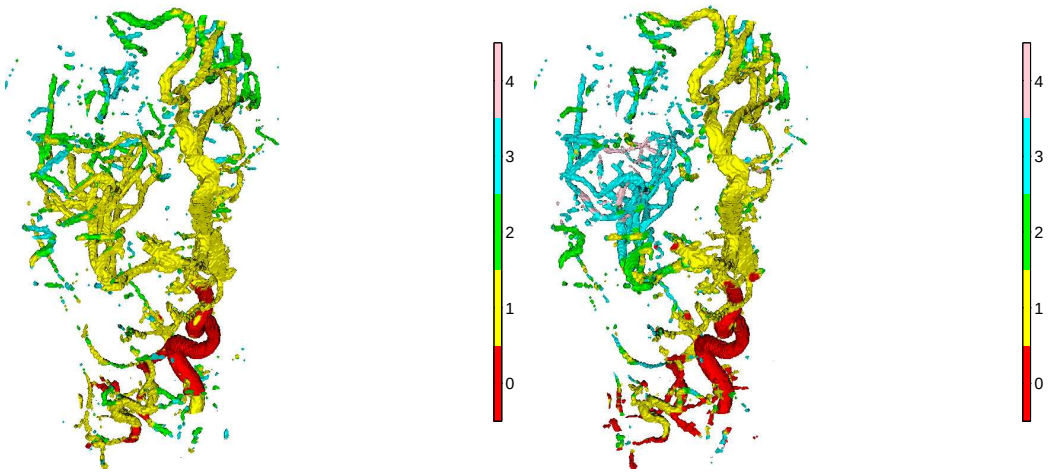
(a) No Clot Time 1

(b) Clot Time 1

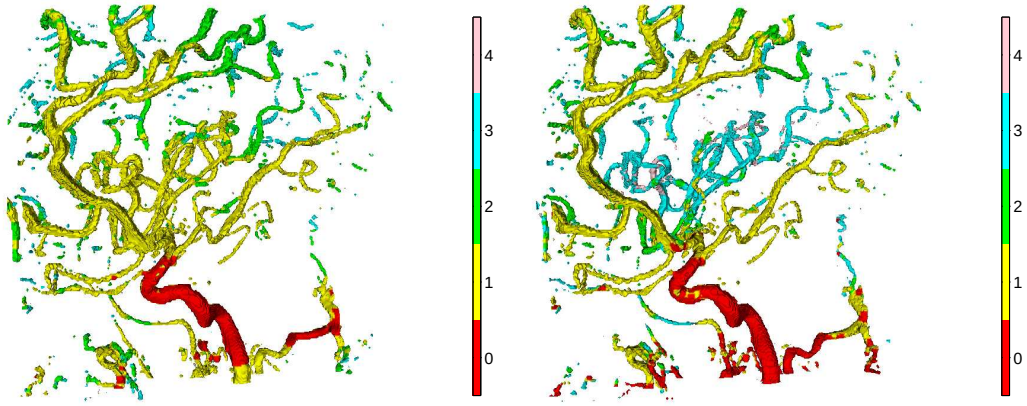
(c) No Clot Time 2

(d) Clot Time 2

Figure 6-21: **Angiographic images for patient B both before and after a clot developed (front view)** – Two Angiographic image of patient B before (left) and after (right) a clot developed. These images were taken from the front view.

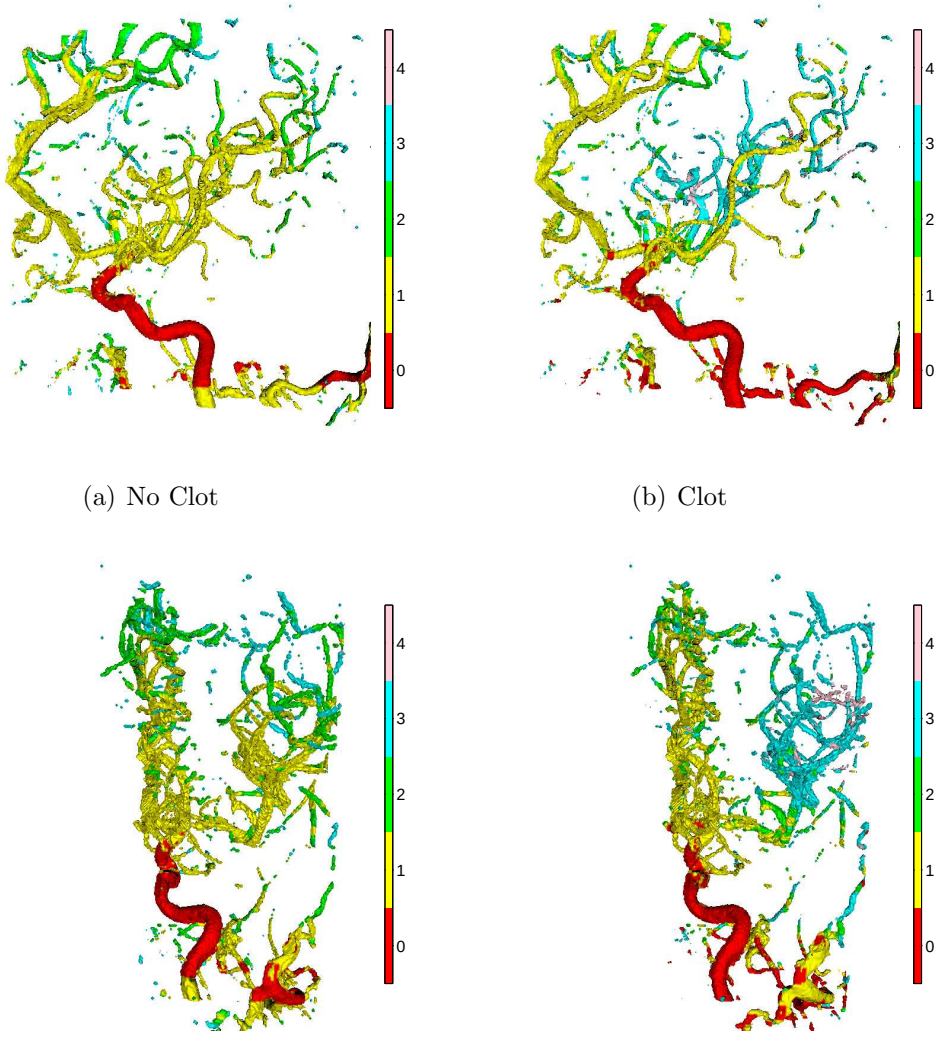


(a) No Clot (b) Clot



(c) No Clot (d) Clot

Figure 6-22: **3D arrival times for patient B both before and after a clot developed** – Visualization of the underlying reconstructed time series of patient B before (left) and after (right) a clot developed for from views. Color is used denote the arrival time of contrast at the different points on the surface of the mesh. The arrival times of the blocked artery are delayed from 1-3 frames from what they were in the baseline case (green, blue, or pink instead of yellow). The units of time are in frames sampled at 3 hertz.



(a) No Clot

(b) Clot

(c) No Clot

(d) Clot

Figure 6-23: **3D arrival times for patient B both before and after a clot developed (continued)** – Visualization of the underlying reconstructed time series of patient B before (left) and after (right) a clot developed from two additional views. Color is used denote the arrival time of contrast at the different points on the surface of the mesh. The arrival times of the blocked artery are delayed from 1-3 frames from what they were in the baseline case (green, blue, or pink instead of yellow). The units of time are in frames sampled at 3 hertz.

Chapter 7

Contributions and Suggestions for Future Work

7.1 Contributions

This thesis provides two major and several other minor contributions to the field of Cerebral Angiography. The major contributions of this thesis are the following:

1. *Reconstruction of 3D Time Series* – a new variational reconstruction algorithm for tomography that provides a reconstructed 3D time series.

2. *Contrast Arrival in 2D* – a new algorithm for determining the arrival time of contrast in 2D.

Reconstruction of 3D Time Series – The first contribution is the variational algorithm (see Equations 5.37-5.42) that achieves improved reconstructions by incorporating a 3D vasculature map in addition to using the 2D time sequences that were used in [64, 85, 116] without a 3D map. The formulation allows for the use of sparsity constraints, as in [35], to concentrate the reconstructed contrast in compact regions. In addition, the formulation introduces temporal constraints that approximate an implicit fluid flow model. The implementation of the algorithm is designed to exploit the space-time structure of the problem. The resulting formulation is a further reg-

ularized algorithm that requires fewer projection than the algorithms do in [4] and [39].

Contrast Arrival in 2D – The second contribution is the 2D arrival time algorithm that uses a Cusum-based algorithm to provide a more accurate measurement of the 2D arrival times than any of the previous methods. The other algorithms depend on the shapes of the time series at different locations being either the same but shifted versions of one another as in [96, 99, 100] or being known a priori as in [51]. The 2D arrival time algorithm presented in this thesis provides more accurate results because it does not make as many assumptions about the shape of the time series as the other algorithms.

Minor Contributions – The methods described in this thesis will enable physicians to better visualize and measure the dynamics of cerebral blood flow. This will improve the physicians ability to diagnoses and their awareness during procedures. To achieve these goals, two new registration algorithms have been developed: a new groupwise registration algorithm to prepare the 2D time series and a new algorithm for 2D-3D registration of the 3D volume relative to the time series. In addition, a new variational reconstruction algorithm for a simplified 2D example that can be extended into 3D.

The groupwise registration algorithm is a generalized algorithm that can extend any two image registration algorithm into a groupwise algorithm that uses an iteratively calculated representative image. This algorithm is distinct from the algorithms of [75, 113] because it registers a single image with the representative image of the set instead of registering an image relative to all other images in each iteration. This produces a more efficient groupwise algorithm of similar accuracy.

The 2D-3D registration algorithm builds upon the many existing meshing algorithms to produce a fast way of calculating Digital Reconstructed Radiographs from an arbitrary pose. This algorithm, which uses the mesh from the isosurface to generate projections, is most similar to the algorithms in [19, 38], where a skeleton was instead used to calculate the projections. The new mesh-based algorithm has been further sped up by using a mesh reduction algorithm to reduce the number of triangles in the mesh.

The 2D variational algorithm combines all of the constraints, including the new constraint map, in a 2D energy functional framework as is done in the 3D variational framework. The 2D framework allows for simple algorithm development without the computational demands of the full 3D version. These concepts may be extended to other situations where sparse observations with different numbers of dimensions are available.

7.2 Suggestions For Future Work

The use of the variational approach provided a direct way to impose all of the different constraints. One choice for future work is to examine different ways of imposing these constraints in the reconstruction of the 3D time series. An alternative approach is to use linear programming for the optimization instead of Calculus of Variations, see [15] for a good starting point. These methods do not have a straightforward way to apply the edge term and it is difficult to add support for flow models, but they do provide a good way of applying the map constraint and sparsity constraints. Another option is to use the level set methods [113, 39] that might provide better performing algorithms. They were not used here because applying the sparsity constraints may be impractical and 3D geometries of the projection data are more difficult to impose than in the 2D work done in [39]. Should these obstacles be overcome, a much faster algorithm might be possible.

The map that was generated in this work provided good results, but it was clear that if a better map was available the results could be improved. This suggests that the use of more accurate segmentation methods might lead to better reconstructions. A different approach that uses a probabilistic model for the constraint map instead of a hard threshold might provide superior results. Such an approach might place contrast in the reconstruction based on the probability of each point appearing within the vasculature. A simple approach is to redefine the function $\eta(\mathbf{x})$ in Equation 5.6 to be equal to the probability that contrast is at the point $\mathbf{x}$. An open question remains of how to best use these probabilities.

Another option for generating the 3D map, is to use other imaging modalities such as MRA or CTA. These methods are often less invasive and may not expose the patient to dosages of radiation. Other modalities might also provide additional information about the location and composition of the vasculature. If a segmentation of the vasculature can be obtained from the other modalities, the segmentation can be used directly in spatio-temporal data fusion.

A method that inserts a fluid flow model into reconstruction, such as in Equation 5.13, might achieve improved results. Another such method is to include the flow model in determining the probabilities of a point in the constraint map containing contrast at the current time. This method could further restrict the regions where contrast can appear by estimating the velocities of the blood flow. However, unless faster sampling rates are made available for the time series, it is unclear that this model will be useful. This is because coarsely sampled time series introduce too much uncertainty for any flow models to be useful.

As was mentioned in Section 5.8, the reconstruction results in the Maximum A Posteriori estimates of the intensities and edge functions corresponding to the conditional distribution of the intensities and edge functions given noisy observations. In order to use the reconstructions to make assertions, it is necessary to know more about the distribution of the reconstructions. Further analysis of these distributions using techniques such as Monte-Carlo as suggested in [78] are another open area for future research.

Appendix A

Calculus of Variation

A.1 Euler Lagrange Equation

From Gelfand and Fomin [41]

$$J(f) = \int_a^b F(x, f, f')dx \tag{A.1}$$

In order for $J(f)$ to have an extremum at $f = f(x)$

$$\delta J(y) = \int_a^b (F_f h + F_{f'} h')dx = 0 \tag{A.2}$$

for the increment $h(x)$. This yields the Euler Lagrange equation

$$F_f - \frac{d}{dx} F_{f'} = 0. \tag{A.3}$$

If F does not depend on x or

$$J(f) = \int_a^b F(f, f')dx \tag{A.4}$$

the Euler Lagrange equation is of the form $F - f' F_{f'} = C$. In the multivariate case

the functional is

$$J(f) = \int_{R^N} F(x_1, x_2, \ldots, x_N, f, f_{x_1}, f_{x_2}, \ldots, f_{x_N}) d_{x_1} d_{x_2} \ldots d_{x_N}.c \tag{A.5}$$

The Euler Lagrange equation is

$$F_f - \sum_{i=1}^{N} \frac{\partial}{\partial x_i} F_{f_{x_i}}. \tag{A.6}$$

Bibliography

[1] Levenberg-marquardt algorithm. `http://en-wikipedia.org/wiki/Levenberg-Marquardt_algorithm`.

[2] D. Adalsteinsson and J. A. Sethian. The fast construction of extension velocities in level set methods. *Journal of Computational Physics*, 148(1):2–22, January 1999.

[3] M. Akra, L. Bazzi, and S. Mitter. Sampling of images for efficient model-based vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(1):4–11, 1999.

[4] C. V. Alvino. *Multiscale Active Contour Methods in Computer Vision with Applications in Tomography*. PhD thesis, Georgia Institute of Technology, April 2005.

[5] L. Ambrosio and V. Tortorelli. Approximation of functionals depending on jumps by elliptic functionals via Gamma convergence. *Commun. Pure Appl. Math.*, 43(8):999–1036, 1990.

[6] L. Ambrosio and V. Tortorelli. On the approximation of free discontinuity problems. *Bolletino U.M.I.*, 6B:105–123, 1992.

[7] W. F. Ames. *Numerical Methods for Partial Differential Equations*. Computer Science and Applied Mathematics. Academic Press, 2nd edition, 1977.

[8] A. H. Andersen and A. C. Kak. Simultaneous algebraic reconstruction technique (SART): A superior implementation of the art algorithm. *Ultrasonic Imaging*, 6:81–94, 1984.

[9] R. Bajcsy and S. Kovacic. Multiresolution elastic matching. *Vis., Graph., Image Process.*, 46:1–21, 1989.

[10] R. N. Bracewell and A. C. Riddle. Inversion of fan-beam scans in radio astronomy. *Astrophys. J.*, 150:427–434, Nov. 1967.

[11] R. P. Brent. *Algorithms for Minimization without Derivatives*. Prentice-Hall, Englewood Cliffs, New Jersey, 1973.

[12] W. L. Briggs, V. E. Henson, and S. F. McCormick. *A Multigrid Tutorial*. SIAM, 2nd edition, 2000.

[13] P. J. Burt and E. H. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, COM-31(4):532–540, April 1983.

[14] E. J. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489 – 509, February 2006.

[15] E. J. Candes and T. Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, December 2005.

[16] E. J. Candes and T. Tao. Near-optimal signal recovery from random projections: universal encoding strategies? *IEEE Transactions on Information Theory*, 52(12):5406–5425, December 2006.

[17] M. Çetin and W. C. Karl. Feature-enhanced synthetic aperture radar image formation based on nonquadratic regularization. *IEEE Transactions on Image Processing*, 10(4):623–631, April 2001.

[18] Y. Censor and T. Elfving. Block-iterative algorithms with diagonally scaled oblique projections for the linear feasibility problem. *SIAM J. Matrix Anal. Appl.*, 24(1):40–58, 2002.

[19] H. Chan, A. C. Chung, S. C. Yu, and W. M. W. III. 2D-3D vascular registration between digital subtraction angiographic (DSA) and magnetic resonance angiographic (MRA) images. In *Proceedings of the IEEE International Symposium on Biomedical Imaging: Macro to Nano.*, pages 708 – 711, April 2004.

[20] T. Chan and J. Shen. *Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2005.

[21] T. Chan and L. Vese. An active contour model without edges. In *Scale-Space Theories in Computer Vision*, pages 141–151, 1999.

[22] L. Chang. Planning and scheduling of concurrent high-level activities for UUV mission operations. Master's thesis, Massachusetts Institute Of Technology, June 2007.

[23] P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud. Deterministic edge-preserving regularization in computed imaging. *IEEE Transactions on Image Processing*, 6(2):298–311, February 1997.

[24] S.-Y. J. Chen and J. D. Carroll. Kinematic and deformation analysis of 4-D coronary arterial trees reconstructed from cine angiograms. *IEEE Transactions Medical Imaging*, 22(6):710–721, 2003.

[25] T.-T. Chen and M. Adams. A sequential failure detection technique and its application. *IEEE Transactions on Automatic Control*, 21(5):750–757, October 1976.

[26] A. Collignon, F. Maes, D. Delaere, D. Vandermeulen, P. Suetens, and G. Marchal. Automated multi-modality image registration based on information the-

ory. In Y. Bizais, editor, *Proceedings of Information Processing in Medical Imaging Conference*, pages 263–274. Kluwer Academic Publishers, 1995.

[27] A. D. Copeland. Robust motion estimation in the presence of fixed pattern noise. Master's thesis, Massachusetts Institute Of Technology, 2003.

[28] E. Coste, C. Vasseur, and J. Rousseau. 3D reconstruction of the cerebral arterial network from stereotactic DSA. *Medical Physics*, 26(9):1783–1793, September 1999.

[29] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, INC., New York, 1991.

[30] B. Davis, P. Lorenzen, and S. Joshi. Large deformation minimum mean squared error template estimation for computational anatomy. In *Proceedings of the IEEE International Symposium on Biomedical Imaging*, pages 173–176, April 2004.

[31] D. Delaere, C. Smets, P. Suetens, G. Marchal, and F. Van de werf. A knowledge-based system for the 3D reconstruction of blood vessels from two angiographic projections. *Journal Medical and Biomedical Engineering and Computing*, 29:NS27–36, November 1991.

[32] M. Desai, D. Kennedy, R. Mangoubi, J. Shah, W. Karl, A. Worth, N. Makris, and H. Pien. Model-based variational smoothing and segmentation for diffusion tensor imaging in the brain. *Neuroinformatics*, 4(3):217–233, September 2006.

[33] M. Desai, R. Mangoubi, J. Shah, W. Karl, H. Pien, A. Worth, and D. Kennedy. Functional MRI activity characterization using response time shift estimates from curve evolution. *IEEE Transactions on Medical Imaging*, 21(11):1402–1412, November 2002.

[34] M. N. Desai, J. C. Deckert, and J. J. Deyst Jr. Dual-sensor failure identification using analytic redundancy. *Journal Guidance and Control*, 2(3):213–220, May-June 1979.

[35] D. Donoho. For most large underdetermined systems of linear equations the minimal l^1-normal solution is also the sparsest solution. Technical Report 10, Department of Statistics, Stanford University, CA, 2004.

[36] D. Donoho and M. Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via l^1 minimization. *PNAS*, 100(5), 2003.

[37] L. A. Feldkamp, L. C. Davis, and J. W. Kress. Practical cone-beam algorithm. *Journal of the Optical Society of America A*, 1:612–619, June 1984.

[38] J. Feldmar, G. Malandain, N. Ayache, S. Fernández-Vidal, E. Maurincomme, and Y. Trousset. Matching 3D MR angiography data and 2D x-ray angiograms. In *CVRMed-MRCAS '97: Proceedings of the First Joint Conference on Computer Vision, Virtual Reality and Robotics in Medicine and Medial Robotics and Computer-Assisted Surgery*, pages 129–138, London, UK, 1997. Springer-Verlag.

[39] H. Feng, W. Karl, and D. Castanon. A curve evolution approach to object-based tomographic reconstruction. *IEEE Transactions on Image Processsing*, 12(1):44–57, January 2003.

[40] M. Garland and P. S. Heckbert. Surface simplification using quadric error metrics. *Computer Graphics*, 31(Annual Conference Series):209–216, 1997.

[41] I. Gelfand and S. V. Fomin. *Calculus of Variations*. Prentice-Hall, Inc., Englewood Cliffs, N.J., 1963.

[42] S. Geman and D. Geman. Stochastic relaxation, gibbs distribution, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, November 1984.

[43] R. Gordon, R. Bender, and G. T. Herman. Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and x-ray photography. *J. Theor. Biol*, 29:471–481, 1970.

[44] P. Grangeat. Mathematical framework of cone beam 3D reconstruction via the first derivative of the Radon transform. In G.T.Herman, A. Louis, and F. Natterer, editors, *Mathematical Methods in Tomography*, number 1497 in Lecture Notes in Mathematics, pages 66–97. Springer Verlag, 1991.

[45] N. Guggenheim, P. Doriot, P. Dorsaz, P. Descouts, and W. Rutishauser. Spatial reconstruction of coronary arteries from angiographic images. *Physics in Medicine and Biology*, 36(1):99–110, January 1991.

[46] A. B. Hamza, Y. He, H. Krim, and A. Willsky. A multiscale approach to pixel-level image fusion. *Integrated Computer-Aided Engineering*, 12(2):135–146, 2005.

[47] G. E. Healey and R. Kondepudy. Radiometric CCD camera calibration and noise estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(3):267–276, March 1994.

[48] J. Hipwell, G. Penney, R. McLaughlin, K. Rhode, P. Summers, T. Cox, J. Byrne, J. Noble, and D. Hawkes. Intensity-based 2-D - 3-D registration of cerebral angiograms. *IEEE Transactions on Medical Imaging*, 22(11):1417–1426, November 2003.

[49] H. P. Hiriyannaiah. X-ray computed tomography for medical imaging. *IEEE Signal Processing Magazine*, 14(2):42–59, March 1997.

[50] K. Hoehne, M. Boehm, and G. Nicolae. The processing of X-ray image sequences. In P. Stucki, editor, *Advances in digital image processing: Theory, application, implementation*, pages 147–163. Plenum, 1979.

[51] I. Horiba, A. Iwata, N. Suzumura, Y. Yamagishi, Y. Miyagi, T. Uwatoko, and Y. Mizuno. Blood flow phase detection on digital subtraction angiography. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 11, pages 1757–1760, Tokyo, 1986.

[52] B. Horn. Fan-beam reconstruction methods. *Proceedings of the IEEE*, 67(12):1616–1623, December 1979.

[53] B. Horn. *Robot vision*. MIT Press, Cambridge, MA, USA, 12 edition, 1986.

[54] B. Horn and B. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[55] E. Ising. Beitrag zur theorie des ferromagnetismus. *Zeitschrift Physik*, 31:253–258, 1925.

[56] M. Jiang and G. Wang. Convergence of the simultaneous algebraic reconstruction technique (SART). *IEEE Transactions on Image Processing*, 12(8):957–961, August 2003.

[57] A. C. Kak and M. Slaney. *Principles of Computerized Tomographic Imaging*. IEEE Press, New York, electronic copy edition, 1999.

[58] S. Katawa and O. Nalcioglu. Constrained iterative reconstruction by the conjugate gradient method. *IEEE Transactons on Medical Imaging*, MI-4(2):65–71, June 1985.

[59] J. Kim, J. W. Fisher III, A. J. Yezzi, M. Çetin, and A. S. Willsky. A nonparametric statistical method for image segmentation using information theory and curve evolution. *IEEE Transactions on Image Processing*, 14(10):1486–1502, 2005.

[60] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(459):671–680, 1983.

[61] A. Klug and R. Crowther. Three-dimensional image reconstruction from the viewpoint of information theory. *Nature*, 238:435–440, August 1972.

[62] S. R. Kulkarni, S. Mitter, and T. J. Richardon. *An existence theorem and lattice approximations for a variational problem arising in computer vision*, pages 189–210. Springer-Verlag, London, UK, 1990.

[63] J. Kybic, T. Blu, and M. Unser. Variational approach to tomographic reconstruction. In K. M. H. Milan Sonka, editor, *Proceedings of SPIE*, volume 4322 of *Medical Imaging*, pages 30–39, San Diego, February 2001.

[64] L. Launay, E. Maurincomme, P. Bouchet, J.-L. Mallet, and L. Picard. 3D reconstruction of cerebral vessels and pathologies from a few biplane digital angiographies. In *VBC '96: Proceedings of the 4th International Conference on Visualization in Biomedical Computing*, pages 123–128, London, UK, 1996. Springer-Verlag.

[65] E. G. Learned-Miller. Data driven image models through continuous joint alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(2):236–250, February 2006.

[66] J. Lee, C. Chen, J. Tasai, Y. Sun, and C. Mao. 3-D image reconstruction of brain vessels from angiograms. In *TENCON '96. Proceedings. 1996 IEEE TENCON. Digital Signal Processing Applications*, volume 2, pages 547–552, November 1996.

[67] T.-C. Lee, R. L. Kashyap, and C.-N. Chu. Building skeleton models via 3-D medial surface/axis thinning algorithms. *CVGIP: Graph. Models Image Process.*, 56(6):462–478, 1994.

[68] M. Li, H. Yang, and H. Kudo. An accurate iterative reconstruction algorithm for sparse objects: application to 3D blood vessel reconstruction from a limited number of projections. *Physics in Medicine and Biology*, 47(15):2599–2609, 2002.

[69] J. S. Lim. *Two-Dimensional Signal and Image Processing*. Prentice Hall PTR, 1999.

[70] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *SIGGRAPH '87: Proceedings of the 14th annual*

conference on Computer graphics and interactive techniques, pages 163–169, New York, NY, USA, 1987. ACM Press.

[71] J. Marroquin. Surface reconstruction preserving discontinuities. Technical Report LIDS-P-1402, MIT Labratory for Information Decision Systems, Cambridge, MA, August 1984.

[72] The Mathworks, Natick, MA. *Matlab Function Reference*, 7th edition, September 2006.

[73] R. Medina, M. Garreau, J. Toro, J.-L. Coatrieux, and D. Jugo. Three-dimensional reconstruction of the left ventricle from two angiographic views: an evidence combination approach. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 34(3):359–370, 2004.

[74] E. H. W. Meijering, W. J. Niessen, and M. A. Viergever. Retrospective motion correction in digital subtraction angiography: A review. *IEEE Transactions on Medical Imaging*, 18(1):2–21, January 1999.

[75] E. G. Miller. *Learning from One Example in Machine Vision by Sharing Probability Densities*. PhD thesis, M.I.T., February 2002.

[76] E. G. Miller, N. E. Matsakis, and P. A. Viola. Learning from one example in machine vision by sharing probability densities. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2000.

[77] G. Minerbo. MENT: A maximum entropy algorithm for reconstructing a source from projection data. *Computer Graphics and Image Processing*, 10:46–68, 1979.

[78] S. K. Mitter. Markov random fields, stochastic quantization and image analysis. In R. Spigler, editor, *Applied and Industrial Mathematics*, pages 101–109. Kluwer Academic Publishers, Venice-1, 1991.

[79] A. A. Montillo. Shape priors in medical image analysis: Extensions of the level set method. Technical Report MS-CIS-02-08, University of Pennsylvania, Department of Computer and Information Science, November 2000.

[80] D. Mumford and J. Shah. Boundary detection by minimizing functionals. *Proc. IEEE Conference on Computer Vision and Patter Recognition*, pages 22–26, 1985.

[81] D. Mumford and J. Shah. Optimal approximation by piecewise smooth functions and associated variational problems. *Comm. Pure Appl.Math.*, 17:577–685, 1989.

[82] S. Osher and R. Fedkiw. *Level set methods and dynamic implicit surfaces.* Springer-Verlag, New York, 2003.

[83] S. Osher and J. A. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.

[84] P. Oskoui-Fard and H. Stark. Tomographic image reconstruction using the theory of convex projections. *IEEE Transactions on Medical Imaging*, 7(1):45–58, March 1988.

[85] C. Pellot, A. Herment, M. Sigelle, P. Horain, H. Matre, and P. Peronneau. A 3D reconstruction of vascular structures from two X-ray angiograms using an adapted simulated annealing algorithm. *IEEE Transactions Medical Imaging*, 13(1):48, Mar. 1994.

[86] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, July 1990.

[87] P. Radeva, R. Toledo, C. Von Land, and J. Villanueva. 3D vessel reconstruction from biplane angiograms using snakes. *Computers in Cardiology*, 25:773–776, September 1998.

[88] J. Radon. On the determination of functions from their integrals along certain manifolds. The Radon Transform and Some of its Applications, Appendix A, (translation of Radon's 1917 paper by R. Lohner), 1983.

[89] G. N. Ramachandran and A. V. Lakshminarayanan. Three dimensional reconstructions from radiographs and electron micrographs: Application of convolution instead of fourier transforms. *Proc. Nat. Acad. Sci.*, 68:2236–2240, 1971.

[90] T. J. Richardson. *Scale Independent Piecewise Smooth Segmentation of images via variational methods*. PhD thesis, Massachusetts Institute of Technology, January 1990.

[91] T. J. Richardson and S. K. Mitter. A variational formulation-based edge focussing algorithm. *Sadhana*, 22(4):553–574, 1997.

[92] E. Ritenour, T. Nelson, and U. Raff. Applications of the median filter to digital radiographic images. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 9, pages 251– 254, March 1984.

[93] K. Rohr, M. Fornefett, and H. S. Stiehl. Approximating thin-plate splines for elastic registration: Integration of landmark errors and orientation attributes. *Lecture Notes in Computer Science*, 1613:252–265, 1999.

[94] A. Rosenfeld and L. Davis. Image segmentation and image models. In *Proceedings of the IEEE*, pages 764–772, May 1979.

[95] D. Russakoff, T. Rohlfing, K. Mori, D. Rueckert, A. Ho, J. Adler, and C. Maurer. Fast generation of digitally reconstructed radiographs using attenuation fields with application to 2D-3D image registration. *IEEE Transactions on Medical Imaging*, 24(11):1441–1454, 2005.

[96] A. Santos, S. Furuie, and M. Gutierrez. Estimation of coronary blood flow by contrast propagation using simulated x-ray angiography. In *Computers in Cardiology 1999*, Hannover, Germany, 1999.

[97] A. Sarwal and A. Dhawan. Three dimensional reconstruction of coronary arteries from two views. In *Engineering in Medicine and Biology society, 1997. Proceedings of the 19th Annual International Conference of the IEEE*, volume 2, pages 565–568, Chicago, October 1997.

[98] A. Sarwal, A. P. Dhawan, and Y. S. Chitre. 3-D reconstruction of coronary arteries using estimation techniques. *SPIE*, 2434:361–369, MAY 95.

[99] H. Schmitt, M. Grass, V. Rasche, O. Schramm, S. Hähnel, and K. Sartor. An x-ray based method for the determination of the contrast agent propagation in 3D vessel structures. *IEEE Trans. Med. Imaging*, 21(3):251–262, 2002.

[100] H. Schmitt, M. Grass, R. Suurmond, T. Kohler, V. Rasche, S. Hahnel, and S. Heiland. Reconstruction of blood propagation in three-dimensional rotational x-ray angiography (3D-RA). *Computerized Medical Imaging and Graphics*, 29(7):507–520, October 2005.

[101] J. A. Sethian. *Level Set Methods and Fast Marching Methods: Evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science.* Cambridge University Press, New York, Second edition, 1999.

[102] J. Shah. A common framework for curve evolution, segmentation and anisotropic diffusion. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, 1996.

[103] G. Shechter, F. Devernay, E. Coste-Maniére, A. Quyyumi, and E. R. McVeigh. Three-dimensional motion tracking of coronary arteries in biplane cineangiograms. *IEEE Transactions on Medical Imaging*, 22(4):493–503, April 2003.

[104] L. A. Shepp and B. F. Logan. The fourier reconstruction of a head section. *IEEE Trans. Nucl. Sci.*, NS-21:21–43, 1974.

[105] A. Shiryaev. The problem of the most rapid detection of a disturbance in a stationary process. *Soviet Math. Dokl.*, 2:795–799, 1961.

[106] C. Studholme, D. Hill, and D. Hawkes. Multiresolution voxel similarity measures for MR-PET registration. In Y. Bizais, editor, *Proceedings of Information Processing in Medical Imaging Conference*, pages 287–298. Kluwer Academic Publishers, 1995.

[107] R. Szeliski. Image mosaicing for tele-reality applications. Technical report, Digital Equipment Corporation, 1994.

[108] Z. S. G. Tari, J. Shah, and H. Pien. Extraction of shape skeletons from grayscale images. *Computer Vision Image Understanding*, 66(2):133–146, 1997.

[109] P. Thèvanaz, U. E. Ruttimann, and M. Unser. A pyramid approach to subpixel registration based on intensity. *IEEE Transactions on Image Processing*, 7(1):27–41, January 1998.

[110] Q. Tian and M. N. Huhns. Algorithms for subpixel registration. *CVGIP*, 35:220–233, 1986.

[111] P. Toft and J. J. Jensen. A very fast implementation of 2D iterative reconstruction algorithms. *Nuclear Science Symposium*, 3:1742–1746, 1996.

[112] A. Tsai. *Curve Evolution and Estimation-Theoretic Techniques for Image Processing*. PhD thesis, Massachusetts Institute of Technology, 2000.

[113] A. Tsai, A. Yezzi Jr., W. Wells III, C. Tempany, D. Tucker, A. Fan, W. E. Grimson, and A. Willsky. Model based curve evolution technique for image segmentation. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1:463–468, 2001.

[114] A. Tsai, A. Yezzi Jr., W. Wells III, C. Tempany, D. Tucker, A. Fan, W. E. Grimson, and A. Willsky. A shape-based approach to the segmentation of medical imagery using level sets. *IEEE Transactions on Medical Imaging*, 22(2):137–154, February 2003.

[115] C. Twining, S. Marshland, and C. J. Taylor. Groupwise non-rigid registration: The minimum description length approach. In *Proceedings of the British Machine Vision Conference*, volume 1, pages 417–426, 2004.

[116] L. van Tran, R. Bahn, and J. Sklansky. Reconstructing the cross sections of coronary arteries from biplane angiograms. *IEEE Transactions on Medical Imaging*, 11(4):517–529, December 1992.

[117] P. A. Viola. *Alignment by Maximization of Mutual Information*. PhD thesis, M.I.T., June 1995.

[118] A. Wald. *Sequential Analysis*. Wiley, New York, 1947.

[119] J. Weese, R. Goecke, G. P. Penney, P. Desmedt, T. M. Buzug, and H. Schumann. Fast voxel-based 2D/3D registration algorithm using a volume rendering method based on the shear-warp factorization. In K. M. Hanson, editor, *Proc. SPIE, Medical Imaging 1999: Image Processing*, volume 3661, pages 802–810, May 1999.

[120] Z. Wu and J.-Z. Qian. Real-time tracking of contrast bolus propagation in x-ray peripheral angiography. In *WBIA '98: Proceedings of the IEEE Workshop on Biomedical Image Analysis*, page 164, Washington, DC, USA, 1998. IEEE Computer Society.

[121] S. Zokai and G. Wolberg. Image registration using log-polar mappings for recovery of large-scale similarity and projective transformations. *IEEE Transactions on Image Processing*, 14(10):1422–1434, October 2005.

[122] L. Zöllei, E. Grimson, A. Norbash, and W. Wells III. 2D-3D rigid registration of X-ray fluoroscopy and CT images using mutual information and sparsely sampled histogram estimators. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages II–696 – II–703, 2001.

[123] L. Zöllei, E. Learned-Miller, E. Grimson, and W. Wells. Efficient population registration of 3D data. *ICCV*, 2005.