Automated Perfusion-Weighted MRI Metrics via Localized Arterial Input Functions

by

Cory Lorenz


Submitted to the department of Electrical Engineering and Computer Science in Partial Fulfillment of the Requirements for the Degree of

Masters of Engineering in Electrical Engineering and Computer Science at the

Massachusetts Institute of Technology

June 2004

© 2004 Cory Lorenz. All rights reserved.

The author hereby grants to MIT permission to reproduce and distribute publicly paper and electronic copies of this thesis document in whole or in part.

Author

Department of Electrical Engineering and Computer Science

May 12, 2004

Certified by

A. Gregory Sorensen
Associate Professor of Radiology, Harvard Medical School
HST Affiliated Faculty, Thesis Supervisor

Accepted by

Arthur C. Smith
Chairman, Departmental Committee on Graduate Students
Automated Perfusion-Weighted MRI Metrics via Localized Arterial Input Functions
by Cory Lorenz

Submitted to the Department of Electrical Engineering and Computer Science on May 12, 2004, in Partial Fulfillment of the Requirements for the Degree of Masters of Engineering in Electrical Engineering and Computer Science

ABSTRACT

This thesis describes and validates a new method for calculating perfusion-weighted MRI (PWI) metrics, a non-invasive technique for calculating cerebral blood flow by tracking a bolus of contrast agent. Past methods to do this calculation require human intermediaries and can lead to errors in the presence of delay and dispersion of the contrast bolus, situations which occur commonly in the pathological conditions which require PWI. The new method described calculates perfusion metrics by defining an arterial input function (AIF) for every voxel in the brain based upon the voxels in close proximity to it. This allows for automated calculation of perfusion metrics, and the localized nature of the AIFs creates an implicit regard for delay and dispersion. This thesis demonstrates that this local AIF method is indeed able to correct flow misestimations due to delay and dispersion, and that it is also more useful for predicting tissue outcome post-stroke.

Thesis Supervisor: A. Gregory Sorensen, M.D., Ph.D.
Title: Associate Professor of Radiology, Harvard Medical School, HST Affiliated Faculty
5.2.1 Patient Selection ................................................................. 41
5.2.2 MRI Acquisition ................................................................. 42
5.2.3 AIF Algorithms ................................................................. 43
5.2.4 Generalized Linear Model .................................................. 43
5.2.5 MTT Lesion Volumes .......................................................... 44
5.3 Results .................................................................................. 44
  5.3.1 ROC analysis ................................................................. 44
  5.3.2 MTT Lesion Volume Analysis ........................................... 47
  5.3.3 Example Cases ............................................................... 47
5.4 Discussion ............................................................................ 50
5.5 Conclusions ........................................................................ 52

Chapter 6: Conclusions .............................................................. 53
  6.1 Summary ............................................................................ 53
  6.2 Suggestions for Further Study ............................................. 54

Appendix A: References .............................................................. 57
Appendix B: Local AIF Algorithm Code ........................................ 59
Acknowledgements

First and foremost, I’d like to thank my thesis advisor, Greg Sorensen, for giving me this opportunity and for all his support. His good advice and positive outlook were always helpful and appreciated.

Secondly, I’d like to thank Thomas Benner (a man who needs no acronym) for all of his advice, coding help, proofreading, brainstorming sessions, and for his good sense of humor about things.

At the MGH-NMR center I’d like to thank Chloe Lopez for all of her help with the GLM, showing me how to use various programs, as well as making and linking lots of perfusion maps for me. I’d also like to thank Nina Menezes and Hakan Ay for their proofreading and suggestions on manuscripts, as well as answering my questions about submitting abstracts, attending conferences and many of the other things that I had to do for the first time this year. Mark Vangel also has my gratitude for his statistical consultations. In addition, I’d like to thank Tara Ullrich for helping me out with all of my paperwork and getting me setup at the lab.

Of course, I have to thank the many friends that I have made at MIT, who have made my life here so much better. First, I’d like to thank Kevin Pipe, for being a role-model of sorts to me. His wisdom, good advice and jump shot (as long as I didn’t have to defend it) were always valuable. I also extend my gratitude to my roommates Nathan Mahn and Louis Lopez, for being not only great roommates, but also great friends. Thanks for putting up with my ranting and complaining, as well as my mess, (even though I didn’t always make it) but also for all of the food and good times. In addition, I’d like to thank my many other close friends: Aditi Garg for her advice and great friendship; Rob Kyker, for his encouragement and idealism; Andrew Brooks, for always having his door open when I needed advice; Alex Firshein and Tanya Burka for being loyal and helpful friends; Matt Brooks and Tristan Hayeck for being there when I needed them. I will (or already do) miss all of you.

Last, and certainly not least, I’d like to thank my parents. For supporting me all these years, for always believing in me, and for putting up with me when things were most difficult, thank you.
Chapter 1: Introduction

1.1 Motivation

Perfusion-weighted magnetic resonance imaging (PWI) is a common imaging technique used in the clinical treatment of patients with brain pathologies such as stroke or cancer (1,2). PWI allows physicians to characterize blood perfusion in the brain non-invasively as part of a patient’s standard MR protocol. Perfusion weighted images are obtained by injecting a bolus of gadolinium chelate into a patient’s bloodstream and imaging as it passes through the brain. The gadolinium acts a contrast agent due to its T2 and T2* effects, which cause a drop in transverse relaxation time (3). This signal drop can then be used to easily calculate the concentration of the agent in a given volume over time. This concentration-time data is then used with standard tracer kinetic models to calculate perfusion metrics such as blood volume, blood flow, and mean transit time.

Due to its non-invasiveness and its relative ease of acquisition, PWI has many applications for both research and clinical purposes. Post-stroke, PWI can be used as a tool to direct a patient’s care by identifying whether tissues are therapeutically treatable or not. This is usually accomplished by determining which tissues are receiving an adequate blood supply, and therefore may be salvageable (4). In stroke research, PWI can be used to evaluate the effectiveness of novel drugs or other therapeutic treatments (such as hyperoxia). The cerebrovascular dynamics of stroke and the autoregulation mechanisms involved can also be better understood with PWI (5). In addition, perfusion metrics can be useful for cancer diagnosis as tumors require increasing amounts of blood flow to sustain growth. Thus, classification of brain tumors is possible, and the different stages of their growth and development can be tracked and better understood (6).

However, the calculation of perfusion metrics in MR is not entirely straightforward. Solving the tracer kinetic model equations to calculate blood flow requires the deconvolution of an arterial input function (AIF) from the concentration-time curves (7,8). Since this function is not known explicitly, the AIF must be estimated from the
data. In current practice, the AIF is estimated by a trained specialist who examines the data and selects a single AIF for the entire brain.

While simple, this approach is problematic in the case of severe pathologies. For instance, many of the physiologic conditions that occur in stroke contradict the assumptions that the single AIF selection is based upon. It has been shown that these contradictions can lead to significant flow misestimations (9,10,11). Also, the time and effort required to manually select the AIF can make the technique inconvenient or impractical in an emergency situation. In addition, the time window for some therapies can be narrow, so this time spent searching out an AIF can lead to a patient losing more of their brain tissue than was necessary. Thus, a technique capable of automatically calculating perfusion metrics without a human intermediary would be valuable, as it could be directly implemented on the MRI scanning console to produce perfusion metrics in minutes. This would then allow patients to receive potentially beneficial therapies.

1.2 Overview
The goal of this work is to correct these accuracy and operational problems by developing an algorithm that can automatically determine an AIF for each voxel in the brain based upon the voxels local to it. This algorithm can then be coupled with established deconvolution techniques to yield PWI metrics that can be calculated automatically on the MRI scanner console. In addition, the locally-defined nature of the AIFs should be able to correct some of the misestimations due to pathology since the locality should endow the AIFs with an implicit regard for delay and dispersion.

While previous work on semi-automatic (12), automatic (13,14) and local (15) AIF generation has already been proposed, the purpose of this work is to combine the approaches as well as to extend the work done on local AIFs. Furthermore, this work will validate the local AIF approach and show its usefulness in the setting of acute stroke.

The organization of this thesis is therefore as follows:
Chapter 2 presents the relevant background in standard tracer kinetic theory and deconvolution by singular value decomposition that the techniques of PWI are based upon. This chapter will also describe the properties of AIFs and their selection for use in PWI along with the problems of the current method.

Chapter 3 describes the local AIF algorithm and the motivation behind its methodology. Examples of the application of the algorithm are given and limitations of the algorithm are also discussed.

Chapter 4 is an analysis of the application of the algorithm. This analysis will investigate the effect of using the algorithm on clinical datasets in comparison to the global AIF method.

Chapter 5 investigates the predictive value of the locally defined perfusion metrics as compared to that of the globally defined ones. This analysis is performed with a previously developed general linearized model (GLM) for estimating a tissue’s risk of death. MTT lesions for the different methods are also compared.

Chapter 6 concludes the thesis with a general overview of the results contained, as well as preliminary work which suggests further study.
Chapter 2: Background

2.1 Tracer Kinetic Theory

In magnetic susceptibility MR imaging, when a gadolinium contrast agent is present in a patient’s brain, it produces a change in the transverse relaxation time ($\Delta R_2$) proportional to its concentration (7). When a bolus of this contrast agent is injected into a patient’s blood stream, the magnetic structure of the gadolinium induces a dephasing of the magnetic dipoles, decreasing the transverse relaxation time. The relationship between this signal drop and the contrast agent concentration has been shown to be (3):

$$S(t) = S_0 e^{-\frac{TE}{\Delta R_2}}$$

where $TE$ is the echo time and $S_0$ is the baseline MR intensity before the contrast agent arrives. Solving this equation for $\Delta R_2$ and using the linear relationship to concentration gives:

$$C(t) \propto \Delta R_2(t) = \frac{1}{TE} \ln \frac{S(t)}{S_0}$$

Once these concentration-time curves have been calculated, standard tracer kinetic models for intravascular agents can be used to calculate cerebral blood flow (CBF), cerebral blood volume (CBV) and tracer mean transit time (MTT). The standard tracer kinetic model entails characterizing each voxel in the brain with a transport function $h(t)$, an arterial input concentration $C_a(t)$ and a venous output concentration $C_v(t)$, as shown in Figure 2.1.1.

![Figure 2.1.1 Standard kinetic model of tracer concentration for a given volume](image-url)
The transfer function \( h(t) \) can be modeled as the probability density function of the transit time for an individual particle passing through the volume. Being a probability density function, \( h(t) \) has the following property:

\[
\int_{0}^{\infty} h(t) dt = 1
\]  

(2.1.3)

The relationship between the input and output concentration is then given by:

\[
C_j(t) = C_a(t) * h(t)
\]

(2.1.4)

where * represents the convolution operator.

Furthermore, we can calculate the fraction of injected tracer still present in the volume by defining \( R(t) \), the residue function, to be:

\[
R(t) = 1 - \int_{0}^{t} h(\tau) d\tau
\]

(2.1.5)

By the definition of \( h(t) \) as a probability density function, \( R(0) = 1 \) and \( R(t) \) is a positive, decreasing function of time. Using this new definition \( R(t) \), the concentration of injected tracer in a given volume can be written as:

\[
C(t) = F_j \cdot C_a(t) R(t) = F_j \int_{0}^{t} C_a(t) R(t - \tau) d\tau
\]

(2.1.6)

The blood flow within the volume can then be calculated by taking the first term of the residue function after deconvolving the arterial input function from the concentration-time curve. However, the arterial input function is not known for every voxel, and thus is typically estimated directly from MR images themselves. In current clinical practice, a single AIF is used for deconvolution. Since the concentration of contrast agent for this AIF is not known, true quantitative CBF is not possible. Thus, only relative CBF maps are possible with this technique.

### 2.2 Deconvolution by Singular Value Decomposition

The method used to deconvolve the arterial input function is also important for this technique. Model independent methods using Fourier Transforms, regularization, and
singular value deconvolution (SVD) have been investigated, as well as a model dependent method based upon modeling the residue function as an exponential decay. Using Monte-Carlo simulations, it was shown that the SVD method gave the most accurate estimation of flow independent of the underlying vasculature structure and volume (7). These simulations also showed that the model dependent approaches were in accurate when the modeling assumptions were violated (which is often the case with pathophysiology). In addition, the Fourier technique was found to be very sensitive to noise and underestimated high flow values. Furthermore, the regularization approach was found to be sensitive to the underlying vascular volume. Thus, the SVD based technique is most often used in common clinical practice, and was also used for this work.

The SVD technique begins by discretizing Eqn. 2.1.6:

\[ C(t_j) = \Delta t \cdot F_t \sum_{i=0}^{t_j} C_a(t_i)R(t_j - t_i) \]  

(2.1.7)

Expanding this equation into matrix notation, this deconvolution can be rewritten as a system of linear equations:

\[
\begin{bmatrix}
C(t_0) \\
C(t_1) \\
\vdots \\
C(t_{N-1})
\end{bmatrix}
= \Delta t
\begin{bmatrix}
C_a(t_0) & 0 & \cdots & 0 \\
C_a(t_1) & C_a(t_0) & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
C_a(t_{N-1}) & C_a(t_{N-2}) & \cdots & C_a(t_0)
\end{bmatrix}
\begin{bmatrix}
R(t_0) \\
R(t_1) \\
\vdots \\
R(t_{N-1})
\end{bmatrix}
\]  

(2.1.8)

which can be written as:

\[ c = F_t \cdot A \cdot b \]  

(2.1.9)

The elements of R(t) can then be obtained by solving for b. However, since A is typically close to singular, the inverse of A is calculated using a singular value decomposition. An SVD decomposes A as:

\[ A = U \cdot S \cdot V \]  

(2.1.10)

where U and V are orthogonal matrices and S is a non-negative square diagonal matrix.

The inverse of A is then calculated as:

\[ A^{-1} = V \cdot W \cdot U^T \]  

(2.1.11)
where \( W = S^{-1} \). In order to eliminate singular values and enforce a stable solution, values of \( W \) corresponding to values where \( S \) is less than 20% of its maximum are set to zero. The residue function scaled by the flow, \( b \), is then calculated as:

\[
b = F_i \cdot V \cdot W \cdot U^T \cdot c
\]  

(2.1.12)

The rCBF value is then taken as the maximum value of \( b \), the estimated \( R(t) \). The relative CBV is also calculated by integrating Eqn. 2.1.2 with respect to time:

\[
rCBV = \int_0^\infty C(t) dt
\]  

(2.1.13)

by assuming no recirculation or consumption of the contrast agent.

Finally, the MTT is calculated using the Central Volume Theorem:

\[
MTT = \frac{rCBV}{rCBF}
\]  

(2.1.14)

### 2.3 AIF Selection and Properties

In current clinical practice, a single AIF is selected manually by a trained technician and is used as the AIF for the entire brain. This AIF is normally selected to be the average of some voxels located near a major artery e.g. middle cerebral artery (MCA), interior cerebral artery, or posterior cerebral artery (16) as it has been found that selection of voxels in large vessels do not follow Eqn. 2.1.1 due to flow effects (3). This technique makes the assumptions that the contrast agent reaches all parts of the brain at nearly the same time and that the contrast agent does not disperse (i.e. \( C(t) \) spreads in time) very much on its path from the major arteries to the brain tissue.

Given these assumptions, the ideal AIF to select is as early and narrow as possible, indicating that it has been minimally dispersed and delayed on its arrival to the brain tissue. Typical quantities to characterize an AIF include: a measure of how fast the bolus arrives (such as slope from baseline to peak), its peak value, a measure of how wide the bolus is (such as the full width at half of the maximum value, or FWHM), and a measure of the bolus arrival time (or similarly, the first moment or time to peak). A typical idealized AIF is shown in Figure 2.2.1 with these properties indicated as well as other
recognizable components of concentration-time curves such as the recirculation effect and baseline region.

Figure 2.2.1 A typical AIF concentration versus time graph. Letters indicate important components of the curve.

A: The baseline region of the signal before the contrast arrives, so it should be near zero.
B: Arrival time of the contrast agent.
C: Slope from peak to baseline. This is indicative of how fast the bolus arrives, and hence its level of dispersion as a dispersed bolus will arrive slower.
D: Peak concentration value. Minimally dispersed AIFs will have a higher peak as the concentration will be constrained to a smaller time frame.
E: Full width at half maximum value. This is also a measure of how dispersed the bolus is as dispersion widens the duration of the signal.
F: Recirculation effect caused by the portion of the bolus which has not been eliminated recirculating through the vascular system.

2.4 Limitations of the Global AIF Technique

It has been shown (9,10,11) that delay and dispersion of the contrast agent can lead to severe flow misestimations. This occurs because the assumptions underlying the single AIF method are no longer valid. Unfortunately, delay and dispersion often occur in the pathologies that are studied with PWI. Specifically in the case of acute stroke, occluded
blood vessels can lead to a significant difference in the arrival time of the contrast agent between two sides of the brain. In addition, dispersion of the contrast agent can occur from collapsed or partially occluded vessels. Thus, the delay and dispersion that is common in stroke can have a detrimental effect on the blood flow estimates, and thus patient treatment.

Furthermore, these are not the only setbacks to the single AIF method. As mentioned in the introduction, the need for training in order to select the AIF can be impractical. But more importantly, in emergency situations such as stroke, the time that must be spent having a technician pick out the AIF can be inconvenient, as well as detrimental to the patient’s outcome since therapeutic windows can be missed. However, the dependence on human decision alone can also lead to a bias in the results. It has been shown in some cases, particularly when there is significant delay or dispersion between the two hemispheres of the brain, that the resulting CBF maps can be very sensitive to which side of the brain the AIF was selected from (11,16).

It has been shown that one solution to the problem of delay is to perform a circular deconvolution via a block-circulant matrix SVD. While this technique can reduce flow misestimation in the CBF due to delay (11), it does not address the problem of dispersion. Thus, a technique that can automatically and deterministically calculate perfusion metrics as well as account for delay and dispersion could vastly improve clinical diagnosis and therapy.
Chapter 3: Technical Development

3.1 Description of Local AIF Algorithm

The local AIF algorithm can be divided into three stages: a preparatory stage, a searching stage, and a deconvolution stage. The preparatory stage began by converting the signal intensity curves into concentration-time curves. This proceeded by assuming a linear relationship between concentration and the change in transverse relaxation (7). The baseline intensity for each voxel was obtained by first finding an average peak time (APT) for the whole brain by averaging the time to peak for every voxel in the brain. Following that, the baseline intensity was calculated on a voxel by voxel basis to be the average of the intensities from time zero to twelve seconds before the APT. The concentration-time curves were then calculated and the cerebral blood volume (CBV) was calculated as the area under the concentration curve for each voxel.

Next, a process of weeding out easily identifiable voxels unfit for AIF selection began. The first step was to exclude voxels which had a CBV which was too low, (<5% of the maximum CBV) or a CBV which was too high (>60% of the maximum CBV). Following this, the first moment of each voxel was calculated, a 30 second arrival window centered on the APT was defined, and commonsense checks were performed in order to exclude voxels that were corrupted by noise, vessel pulsation, patient motion, susceptibility artifacts or other sources of artificial signal drops. These exclusion criteria are summarized below and examples are shown in Figure 3.1.1. A voxel was excluded if it had:

- A first moment that was more than 7.5 seconds before the APT. (Having a majority of the concentration before the bolus arrives is indicative of noise.)
- A maximum value outside of the arrival window. (Having maximum values well before or after the bolus arrival time is indicative of artificial signal drops.)
- Two points at 60% of its maximum separated by more than 26 sec. (26 sec was found to be enough time for bolus passage, so voxels with this property were typically noise.)
- A “negative concentration” with an absolute value more than half of its
maximum. (Since noise is usually distributed evenly around zero, this will exclude noisy voxels.)

- A negative concentration change (ΔC) > 40% of its maximum ΔC before the arrival window. (Large -ΔC were expected only after the bolus arrives, so this excluded voxels with artificial signal drops.)

- A positive ΔC > 60% of its maximum ΔC after the arrival window. (Similarly, large ΔC were expected only when the bolus arrives, so this will exclude voxels with artificial signal drops.)

- A mean concentration after the arrival window that was not higher than 2 standard deviations above the mean concentration before the arrival window. (Valid AIFs contain a recirculation artifact which will increase the mean concentration post-bolus.)

Finally, four different statistics were calculated for each voxel as the criteria upon which the AIF was searched for. The first criterion was the first moment (FM) calculated above, which is ideally as early as possible. The second criterion was the average slope (AS) from the last baseline point to the peak. The third criterion was the peak value (PV). Ideally, these last two values were as high as possible, since they indicate that an AIF has been minimally dispersed. The fourth and last criterion was the full width half maximum (FWHM) which was ideally small, as a minimally dispersed AIF will be very narrow.
Figure 3.1.1: Examples of different AIFs that would fail the exclusion criterion as compared to a given valid AIF. Plot A shows an idealized AIF with its single sharp peak and recirculation artifact. Plot B shows a potential AIF that violates properties 1 and 2 since it has an early first moment as well as an early peak, as compared to the ideal AIF. Plot C shows a potential AIF that violates property 3, since it has high concentration values that occur for longer than a typical bolus passage time. Plot D shows a potential AIF that violates property 4, since the large negative peak is illogical and indicates noise corruption. Plot E shows a potential AIF that violates properties 5 and 6 with the large concentration changes before and after the bolus. Plot F shows a potential AIF that violates property 7 since it does not have a recirculation peak, which is indicative of a motion artifact.

The second stage was the search stage, where the best AIF voxels were searched out and an AIF was determined for every voxel. This stage began by making a search cube of roughly (27mm)^3 centered on each voxel in the brain. An example of the search cube size is shown in Figure 3.1.2. For each search cube, the corresponding criterions calculated earlier were normalized by subtracting out the minimum value of each statistic in the cube followed by dividing by the maximum value of each statistic in the cube. Thus, each criterion was constrained to between 0 and 1. Finally, these four criteria were converted into a score, S, defined to be:

\[ S = FM - PV - AS + FWHM \] (3.1.1)
such that each voxel in the cube had a score between -2 and 2, with the ideal voxels having the lowest scores. The three voxels with the lowest score were then selected and interpolated with a 3-D 27mm FWHM Gaussian kernel. This result was stored as the AIF for the voxel centered in the cube.

![Figure 3.1.2 Example of the relative size of the (27mm)^3 search cube (blue) for a given voxel (yellow) in the center image. The slices to the left and right of center are the slices above and below the center image, respectively. The resolution of the images is 1.72 mm^2 with a slice thickness of 6mm and a gap between the slices of 1mm.](image)

The final stage of the algorithm was the deconvolution stage which began by smoothing the local AIFs (for continuity) with the same 3-D Gaussian kernel used for interpolation. However, in order to prevent a bias on the edges of the brain, the AIF was extended out by 14mm in all directions at the brain periphery in order to avoid edge effects during the smoothing process. This was accomplished by adding extra slice above and below the brain, as well as using morphological operators to replicate the values on the edge of the brain out an extra 14mm. Finally, the local AIFs, C_a(t), were deconvolved from the concentration curves, C(t). The SVD approach described in Chapter 2 was used to do this deconvolution on a voxel by voxel basis. Following deconvolution, the MTT map was obtained by dividing the CBV map by the calculated CBF map. The entire local AIF algorithm, its stages and their important features are summarized in Figure 3.1.3 below. In addition, an implementation of the code using Matlab is given in Appendix B.
Figure 3.1.3 Flowchart Overview of the Local AIF algorithm
3.2 Example Outputs of the Local AIF Algorithm

Figure 3.2.1 shows four CBF slices for a single patient that resulted from using the local AIF algorithm (top row) as well as from using the standard global AIF algorithm (bottom row). The local AIF algorithm is leading to flow increases in the ipsilateral hemisphere (i.e. the hemisphere of the brain which shows the perfusion defect). Particularly in the region pointed to by the green arrows, the local AIF algorithm is making a large difference.

![Figure 3.2.1 Example CBF outputs for the local AIF (top row) and global AIF (bottom row) methods. The local AIF algorithm is increasing flow estimates in the ipsilateral hemisphere, as indicated with green arrows. Most likely, the decreased flow values on the global CBF are due to delay or dispersion of the contrast bolus.](image)

Additionally, Figure 3.2.2 compares four MTT slices produced by the local AIF (top row) and global AIF (bottom row) methods. The local MTT is showing much less abnormality in the ipsilateral hemisphere, as indicated by the green arrows. The global MTT indicates that most of the ipsilateral hemisphere has an increased MTT, while the global method isolates it to a particular region. Thus, the local method appears to be more specific to the actual flow defect.
Figure 3.2.2 Example MTT outputs for the local AIF (top row) and global AIF (bottom row) methods. As compared to the global MTT, the local MTT maps seem to be much more specific as to the actual region of flow defect. Particularly in the region indicated by the green arrows, the local AIF methods estimates a normal transit time, as opposed to the global method, which has nearly the entire ipsilateral hemisphere as abnormal.

3.3 Limitations of the Local AIF Algorithm

The beneficial effects of the local AIF algorithm are dependent on an uncorrupted dataset. As shown in Figure 3.3.1 below, the algorithm selects AIFs from all over the brain. Since it selects so many AIFs, such things as large patient motions (even after motion-correction), susceptibility artifacts or vessel pulsation can adversely affect the local AIF performance as they can lead to artificial signal drops which can cause regional discolorations in the perfusion maps in the areas where they are chosen as the AIF. This is because they lead to artificially high concentrations that lower the CBF estimate when they are deconvolved from the concentration-time curves. An example of the effect of patient motion is shown in Figure 3.3.2. In addition, a low contrast to noise ratio (CNR) can also be detrimental to local AIF performance. The high noise variance can also lead to artificial signal drops, causing effects similar to those described above. Furthermore, the low contrast can make finding valid AIFs extremely difficult, particularly in regions of low flow, which can occur in stroke.
Figure 3.3.1 Example of the locations and frequency of the voxels selected by the local AIF algorithm for a typical slice. The locations are overlaid (with the color representing the number of times a voxel was selected) on an image slice at the time of peak contrast concentration. Note that the selected voxels are highly concentrated around the vessels (i.e. the dark bands).

Figure 3.2.2 Example of the misestimations that can arise due to artificial signal drops, in this case from a large patient motion. The misestimations can be seen as the lower intensity on the right side of the image, curving down from the top (green arrows). The hyperintense bands on the edges of the brain are also due to the patient motion, but these also occur with the single manual AIF method.

However, without care, auto or semi-automatic global AIF selection could choose a flawed AIF in any of these situations as well and produce worse perfusion metrics than the local approach since the error would not be isolated to a certain region of the brain. Thus, with the local AIF algorithm (and most automated techniques), scanner maintenance, motion correction, and proper contrast agent administration are important factors in getting accurate blood flow estimates.
3.4 Clinical Implementation

When converted to the C programming language, the local AIF algorithm can be used to calculate perfusion metrics for a 128x128x11x43 dataset in about 6 minutes on a 2GHz computer. Since an SVD must be calculated for every voxel in the brain, the deconvolution accounts for roughly two thirds of the running time. Since most MRI scanners are programmed in a C derivative language that can run C programs, this would correspond to the running time on a recent MRI scanner. Unfortunately, six minutes is a long time to wait in a busy emergency room, although it is a significant improvement as compared to manual calculation. However, while this execution time may be slightly infeasible with current MRI systems, this execution time could easily be cut in half with either a dual processor system or faster CPU. Thus, with improved processing technology, this algorithm could become a valuable tool for clinical MRI systems.
Chapter 4: Effects of Using Localized AIFs

4.1 Introduction
The “gold-standard” technique for calculating perfusion metrics in animals is to inject microspheres into the bloodstream, euthanize the animal and then count the number of spheres present in each volume of the brain. This technique, for obvious ethical reasons, is not an option in human perfusion research. Thus, without a “gold-standard” quantitative technique for calculating in vivo perfusion metrics to compare directly, other techniques are needed to validate the performance of the local AIF algorithm. One way to do this is to check that the new technique is actually fulfilling its potential. One of the important theoretical benefits of using local AIFs is that they should have a delay and dispersion that is approximately the same as the tissues near them. Thus, the local AIF algorithm should produce less flow misestimations due to delay and dispersion of the contrast bolus. Proving that the local AIF technique actually achieves this flow estimation improvement is then very important to the validation of the method.

We propose to validate the local AIF method by first examining its performance under normal and pathological conditions in clinical datasets. To accomplish this, CBF values of the local and global AIF methods will be compared for normal and abnormal full width half max (FWHM) values as well as normal and abnormal time to peak (TTP) values. Furthermore, the locations where large CBF ratios occurred will be mapped and compared. Finally, the average underlying tissue concentration curves will be calculated and compared for these large ratio voxels in order to understand where the method is making improvements.

4.2 Methods
4.2.1 Patient Selection
This study included 53 patients with diffusion and perfusion-weighted images obtained within the first 12 hours of symptom onset and a follow-up imaging on day 5 or later. The study population was selected from a retrospective database into which patients with ischemic stroke from 2 sites were registered between years 2000 and 2002. Fourteen of
the 53 patients were excluded because of motion or scanner artifacts. Three additional patients with completely normal PWI due to early reperfusion were also excluded. Of the remaining 36 patients, 21 were male and 15 were female. The mean age was 65 ranging from 26 to 83 years. There were 24 patients from site 1 and 12 patients from site 2. The cerebral infarction occurred within the territory of basilar artery in 1, posterior cerebral artery in 1, and middle cerebral artery in 34. The stroke mechanism was large vessel atherosclerosis in 10, cardiac embolism in 12, other rare causes in 4, small vessel disease in 1, and cryptogenic in 9 patients. The standard stroke treatment included anticoagulants and/or antiplatelet agents and no patients were treated by intravenous or intraarterial thrombolysis or experimental drugs.

4.2.2 MRI Acquisition

All data sets from site 1 were axial single-shot gradient-echo EPI images acquired during the first pass of 0.2 mmol/kg of a gadolinium-based contrast agent injected 10 s after the start of imaging at a rate of 5 ml/s, followed by a comparable volume of normal saline injected at the same rate. These studies were performed on a 1.5 T General Electric system with an MRI-compatible power injector (Medrad, Pittsburgh, PA). They had a TR/TE = 1500/65 ms, a field of view (FOV) = 220 x 220 mm², and an acquisition matrix = 128 x 128. The datasets consisted of 11 slices with a slice thickness of 6 mm and a gap of 1 mm, collected over 46 time points.

The data sets from site 2 were single-shot spin-echo EPI images obtained from a 1.5 T Siemens system with an MRI-compatible power injector (Spectris, Medrad, Pittsburgh, PA). They had a TR/TE = 1200/78, an FOV of 26 x 26 cm², and an acquisition matrix = 116 x 256 interpolated to 256 x 256. These studies contained 7 slices with a slice thickness of 6.5 mm and a gap of 1 mm, collected over 40 time points. All data analysis was performed retrospectively, with approval from our institution’s committee for human subject research.

4.2.3 AIF Algorithms

For all patients, the global AIF perfusion maps were created by an experienced technician.
by first selecting a number of voxels from the MCA in the contralateral hemisphere of the brain and taking their average as the AIF for the entire dataset, as described in section 2.3. This AIF and the dataset were then run through a common SVD based deconvolution algorithm to create the global CBF maps. In addition, the local AIF algorithm described in Chapter 3 was used to create the local CBF maps.

4.2.4 CBF Difference Analysis

In addition to the CBF calculation, the FWHM and TTP of every voxel in the brain were also calculated in order to get a measure on the delay and dispersion of each voxel. The difference ($D_{lg}$) of the local-defined CBF (lCBF) and the global CBF (gCBF) was then calculated for every voxel in the brain after normalizing by the average flow value in a normal gray matter ROI. The mean of the ratios for voxels with “normal” and delayed or dispersed TTP and FWHM voxels were then calculated on a patient by patient basis. More specifically, the “normal” TTP was defined to be one time point before the most frequent TTP found in the brain ($T_{MF}$) and the “normal” FWHM was defined to be a one TR window centered on the most frequent FWHM ($F_{MF}$). Furthermore, the “delayed” TTP was defined to be 4 time points after the “normal” TTP and the “dispersed” FWHM was defined to be a one TR window centered at 4.5 seconds after the “normal” FWHM.

In summary, for each patient, we calculate:

$$\mu_{Fn} = \text{mean}(D_{lg}) | F_{MF} - \frac{TR}{2} < \text{FWHM} < F_{MF} + \frac{TR}{2}$$  \hspace{1cm} (4.2.4.1)

$$\mu_{Fd} = \text{mean}(D_{lg}) | F_{MF} + 4.5 - \frac{TR}{2} < \text{FWHM} < F_{MF} + 4.5 + \frac{TR}{2}$$  \hspace{1cm} (4.2.4.2)

$$\mu_{Tn} = \text{mean}(D_{lg}) | TTP = T_{MF} - 1$$  \hspace{1cm} (4.2.4.3)

$$\mu_{Td} = \text{mean}(D_{lg}) | TTP = T_{MF} + 3$$  \hspace{1cm} (4.2.4.4)

Finally, for each patient the difference of these means were calculated for both TTP and FWHM. (i.e. $\mu_{Fd} - \mu_{Fn}$ and $\mu_{Td} - \mu_{Tn}$).

4.2.5 Voxel Location Analysis

Knowing where the lCBF values differ greatly from the gCBF values can also give us an indication of the benefits of the local AIF approach. To accomplish this, the mean and
standard deviation of both the ratio of ICBF and gCBF ($R_{lg}$) and its inverse ($R_{gl}$, or the ratio of gCBF to ICBF) were calculated over the entire brain for each patient. Next, all voxels with an $R_{lg}$ greater than 1.75 standard deviations beyond the mean of $R_{lg}$ were found and their locations where highlighted on the CBF maps for that patient. Similarly, all voxels with an $R_{gl}$ greater than 1.75 standard deviations beyond the mean of $R_{gl}$ were found and highlighted in a similar manner. (For the rest of this work, having a ratio greater than 1.75 standard deviations beyond the mean will be considered a “significant difference”). Finally, the average tissue concentration curves for the voxels with a significant $R_{gl}$ or $R_{lg}$ difference were calculated.

4.3 Results

4.3.1 CBF Difference Analysis

A stem plot of $\mu_{Td} - \mu_{Tn}$ is shown in Figure 4.3.1.1 on a patient-by-patient basis. In most patients (31 out of 36), $\mu_{Td}$, the mean difference of ICBF and gCBF (i.e. $D_{lg}$) with a delayed TTP is higher than $\mu_{Tn}$, the mean of $D_{lg}$ with a TTP that is typical for the brain. This implies that the local AIF method is increasing flow values for voxels with a delayed tissue-concentration curve. It has been shown (9,11) that using an AIF that arrives too early compared to the underlying tissue-concentration can create flow underestimations. Thus, the local AIF method appears to be correcting these flow misestimations in most cases.

Similarly, a stem plot of $\mu_{Fd} - \mu_{Fn}$ is shown in Figure 4.3.1.2 on a patient-by-patient basis. In a majority of patients (23 out of 36), $\mu_{Fd}$, the mean of $D_{lg}$ with an increased FWHM, is higher than $\mu_{Fn}$, the mean of $D_{lg}$ with a FWHM that is typical for the brain. It has also been shown (9) that dispersion of the concentration-time curves can also lead to flow underestimations. Thus, it appears that the local AIF method can correct these flow misestimations in some cases as well.
4.3.1.1 Mean difference of ICBF and gCBF for voxels with underlying “delayed” and “normal” TTP values. The difference is positive for “delayed” TTP values in a vast majority of patients, which suggests that the local AIF method is increasing flow estimates for voxels with a significant delay in the contrast bolus.

4.3.1.2 Mean difference of ICBF and gCBF for voxels with underlying “dispersed” and “normal” FWHM values. The difference is higher for “dispersed” FWHM values in a majority of patients, which suggests that the local AIF method is increasing flow estimates for voxels with a dispersed time profile.

4.3.2 Voxel Location Analysis

Figure 4.3.2.1 below shows the difference of the percentage of total brain voxels considered significantly different for Rtg and Rgl. Thus, we see that for most patients, the local AIF algorithm is increasing flow values in more voxels than it is decreasing them. Accordingly, across all patients, the local AIF algorithm was significantly increasing
flow values for an average of 7.3% of voxels, while it was decreasing them for an average of 4.3% of voxels.

Figure 4.3.2.1 Percentage of voxels that the local AIF method significantly increases flow estimates minus the percentage of voxels that the local AIF method significantly decreases flow estimates. For most patients, the local AIF method is leading to more increased flow values.

Upon visual inspection of the voxel locations of significant lCBF increase, it was found that in 7 of the 36 patients, the voxels were highly concentrated in the ipsilateral hemisphere. For these same patients, there was no striking concentration of flow decrease in the ipsilateral hemisphere. An example of this phenomenon is shown in Figure 4.3.2.2, which shows blood flows with the locations of flow increase and decrease overlayed on all slices for one patient. Single slices from the remaining six patients with the same overlays are shown in Figure 4.3.2.3. In addition, there were 8 patients which had the locations of significant local flow increase concentrated in the ipsilateral hemisphere. However, these patients also had a high number of the local flow decrease voxels located in the ipsilateral hemisphere. Example slices from these patients are shown in Figure 4.3.2.4 with the significant voxel locations overlayed. Finally, there was one patient that only had a high concentration of significantly decreased local flow voxels in the ipsilateral hemisphere.
Figure 4.3.2.2 Voxel locations of significant local CBF increase (blue) and decrease (green) for a single patient. Note the high concentration of the blue voxels in the ipsilateral hemisphere (the half of the brain that contains the lesion, indicated in red).

Figure 4.3.3.3 Example slices from patients with increased ICBF voxels (blue) highly concentrated in the ipsilateral hemisphere while the decreased ICBF voxels (green) are not concentrated to a hemisphere. For reference, the follow-up lesion is indicated in red.
When the average concentration-time curves were calculated for the increased ICBF voxels, it was found that in 31 out of the 36 patients the underlying tissue curves had the shape of a generalized concentration-time curve. On the other hand, only 13 out of the 36 patients exhibited a generalized shape for the decreased ICBF voxels. These 13 were a subset of the 31 patients mentioned above, and they typically had a much lower magnitude than the average curve for increased ICBF. Furthermore, when the 31 average curves with increased ICBF were compared to a typical AIF, it was found that they were delayed in 25 patients and dispersed in 17 patients with 16 patients exhibiting both delay and dispersion. A few examples of these average curves and the typical AIFs are plotted in Figure 4.3.2.5.

When these concentration-time curves were cross-referenced with the voxel location analysis described above, it was found that of the 8 patients with voxels of significant ICBF increase and decrease in the ipsilateral hemisphere, 6 of them had average concentration-time curves of ICBF increase which were of the generalized form.
Conversely, these patients had average concentration-time curves of ICBF decrease which were either very low magnitude or just noise. In addition, for the one patient which showed the concentration of significantly lower ICBF voxels in the ipsilateral hemisphere, it was found that this average tissue curve was just noise as well. Furthermore, of the 7 patients that had the significant ICBF increase voxels concentrated in the ipsilateral hemisphere with no concentration of ICBF decrease voxels, 6 of these patients exhibited average underlying curves with the shape of generalized delayed or dispersed concentration-time curves.

![Figure 4.3.2.5 Examples of underlying tissue concentration-time curves and corresponding AIFs for two patients (arranged by row). Plots A and C show an AIF (solid line) compared to the average concentration-time curve for voxels with a significant ICBF increase (dashed line) for two different patients. Note the delay of almost 6 seconds between the AIF and the average curves in both of these patients, as well as the widening of the curve (i.e. dispersion) in plot C. Similarly, plots B and D show the corresponding scaled AIF (solid line) as well as the average concentration-time curve for voxels with a significant ICBF decrease (dashed line). Note the much lower magnitudes of the local flow decrease signals in both cases.](image)
4.4 Discussion

We have shown that the local AIF method is capable of increasing flow estimates for voxels with underlying FWHM or TTP values which are delayed or dispersed. As was shown in Figure 4.3.1.1, for most patients, the flow estimates are increased for voxels with a longer underlying TTP. It must be noted that there are three cases which seem to not fit into this classification though. When the original raw perfusion data is examined however, the reasons for these contrary results are easily seen. In two cases, the region of perfusion abnormality is small, so nearly the entire brain sees no delay at all. In the other case, the contrast agent arrived normally to everywhere but the lesion, where it did not lead to a noticeable signal drop. Thus, in these three cases, there did not seem to be any considerable regions that could have been classified as having abnormal delay. Thus, the regions with these larger TTP values were probably noisy or nonsense voxels for which the flow ratios are erratic and unimportant.

Looking at the FWHM results in Figure 4.3.1.2, we see that the results are much noisier as compared to the TTP analysis, but the local AIF method still seems to be increasing the flow values in dispersed voxels on average. Upon examination of the four datasets yielding the lowest negative values in the plot, we see that similar phenomena to the TTP analysis above are influencing the FWHM calculation as well. The two patients with small lesions that led to contrary results in the TTP calculation also led to contrary results in the FWHM calculation, for most likely the same reason. With a small lesion, there was little to no dispersion, so the voxels with increased FWHM values were most likely noise. The other two patients were patients that seemed to not have been given quite enough contrast agent (i.e. had a low contrast to noise ratio, or CNR). Thus, the FWHM values could more easily be extended or shortened by the noise fluctuations and voxels that were not dispersed could be classified as dispersed, or voxels that were dispersed could be classified as not dispersed if a large noise fluctuation occurred at an opportune moment. Since the TTP calculation is not as sensitive to noise fluctuations that could explain why the result is more conclusive than the FWHM comparison. In addition, delay seems to be a much more common problem than dispersion. Nevertheless, new methods to show the
effect of the local AIF method on dispersed voxels should be attempted and this result should be taken only as provisional.

We also showed that the local AIF algorithm is creating significant flow increases in more patients than it is creating significant flow decreases (Figure 4.3.2.1). However, we see two outliers in this calculation as well. Upon inspection of these datasets, we find that one patient was not given enough contrast agent, while the other seems to have been given too much. The low CNR resulting from a deficit of contrast agent could make the local AIF calculation less accurate as it will be more difficult to find suitable voxels, and they could also be corrupted by noise. In addition, when too much contrast agent was used, the two methods disagreed on the flow voxels in or immediately near the large vessels. Since these values are so large, differences in these voxels are not really that important in terms of clinical diagnosis or treatment.

Furthermore, we showed that the voxels where the local AIF method was calculating significant ICBF increases were highly concentrated in the ipsilateral hemisphere in 15 patients (Figure 4.2.3.2-4). In addition, we looked at the average underlying concentration-time curves for these voxels and found that in most cases, these were normal looking curves (albeit reduced) that often showed some measure of delay and/or dispersion (Figure 4.2.3.5). Of the 15 patients mentioned above, 12 patients had the generalized tissue curve shape, indicating that the flow improvement was on voxels that actually had some degree of flow in them. In some cases, the voxels of ICBF decrease were also found in significant numbers in the ipsilateral hemisphere (Figure 4.3.4). However, in all of these cases, the average underlying curve for these voxels was either of negligible flow or was just noise. Thus, the ICBF decrease in these voxels appears to be insignificant, as these voxels are either nonsense or correspond to lesion tissue which has already infarcted. This seems to suggest that the groupings in Figure 4.3.3.3 and 4.3.3.4 were unnecessary as the decreased ICBF voxels in the ipsilateral hemisphere were mostly from dead or noisy tissue.
It must be noted however that there were a number of voxels for which the local AIF method increased flow estimates that still went on to infarction. While this is not necessarily a big problem, it certainly brings to light the question of this new method’s usefulness in clinical diagnosis. Thus, demonstrating that the changes to flow estimates that the local AIF method makes are clinically relevant is important to the validation of this technique, and is the topic of the next chapter.

4.5 Conclusions

The main theoretical benefit of using localized arterial input functions is that by using them, delay and dispersion of the contrast bolus are implicitly accounted for. In this work, by examining the voxels of significant flow differences between the local and global AIF methods as well as the average flow change for underlying levels of delay and dispersion, we have provided evidence that the local AIF method is actually increasing flow values for voxels which are delay and dispersed. Since it has been shown that delay and dispersion lead to flow underestimations, this suggests that the local AIF method is fulfilling its theoretical promise as a technique that is less sensitive to delay and dispersion.
Chapter 5: General Linearized Model Analysis

5.1 Introduction

Chapter 4 showed that the local AIF method is making flow improvements in voxels with delay and dispersion, but the usefulness of this improvement was not explored. Since a "gold-standard" perfusion metric is unavailable, other methods of validation are needed. Since PWI is used largely in clinical situations for diagnosis or research post-stroke, another way to quantify performance is to compare how well the locally defined perfusion metrics can predict patient outcome as opposed to the older methods.

A general linearized model (GLM) is one way to do such an analysis (17). A GLM can assign a tissue risk of infarction to every voxel in the brain, and these risk maps can then be cross referenced with follow-up scans of patient outcome to calculate statistics on the performance of the different techniques. These statistics can then be compiled into a receiver operating characteristic (ROC) curve and the performance of the different methods can be compared on the basis of their ROC curves. As an alternative, MTT lesion volumes can be outlined by an experienced neuroradiologist, and the results can also be cross-referenced to the follow-up lesion volumes. While this method may not be as sensitive since it is predominately based on only one imaging modality, it is at least a way to reinforce the results of the GLM by showing that the images obtained are useful to the human viewer and not just some mathematical model.

5.2 Methods

5.2.1 Patient Selection

This study included 53 patients with diffusion and perfusion-weighted images obtained within the first 12 hours of symptom onset and a follow-up imaging on day 5 or later. The study population was selected from a retrospective database into which patients with ischemic stroke from 2 sites were registered between years 2000 and 2002. Fourteen of the 53 patients were excluded because of motion or scanner artifacts. Three additional patients with completely normal PWI due to early reperfusion were also excluded. Of the remaining 36 patients, 21 were male and 15 were female. The mean age was 65 ranging
from 26 to 83 years. There were 24 patients from site 1 and 12 patients from site 2. The cerebral infarction occurred within the territory of basilar artery in 1, posterior cerebral artery in 1, and middle cerebral artery in 34. The stroke mechanism was large vessel atherosclerosis in 10, cardiac embolism in 12, other rare causes in 4, small vessel disease in 1, and cryptogenic in 9 patients. The standard stroke treatment included anticoagulants and/or antiplatelet agents and no patients were treated by intravenous or intraarterial thrombolysis or experimental drugs.

5.2.2 MRI Acquisition
All data sets from site 1 were axial single-shot gradient-echo EPI images acquired during the first pass of 0.2 mmol/kg of a gadolinium-based contrast agent injected 10 s after the start of imaging at a rate of 5 ml/s, followed by a comparable volume of normal saline injected at the same rate. These studies were performed on a 1.5 T General Electric system with an MRI-compatible power injector (Medrad, Pittsburgh, PA). They had a TR/TE = 1500/65 ms, a field of view (FOV) = 220 x 220 mm², and an acquisition matrix = 128 x 128. The datasets consisted of 11 slices with a slice thickness of 6 mm and a gap of 1 mm, collected over 46 time points.

The data sets from site 2 were single-shot spin-echo EPI images obtained from a 1.5 T Siemens system with an MRI-compatible power injector (Spectris, Medrad, Pittsburgh, PA). They had a TR/TE = 1200/78, an FOV of 26 x 26 cm², and an acquisition matrix = 116x256 interpolated to 256 x 256. These studies contained 7 slices with a slice thickness of 6.5mm and a gap of 1mm, collected over 40 time points.

All 36 subjects also had EPI diffusion scans performed along with the perfusion studies. Apparent diffusion coefficient (ADC), average diffusion-weighted images (DWI), and averaged non-diffusion-weighted images (LowB) maps were calculated from the diffusion data sets. All data analysis was performed retrospectively, with approval from both institutions' committees for human subject research.
5.2.3 AIF Algorithms

For all patients, the global AIF perfusion maps were created by an experienced technician by first selecting a number of voxels from the MCA in a single hemisphere of the brain and taking their average as the AIF for the entire dataset, as described in section 2.3. This AIF and the dataset were then run through a common SVD based deconvolution algorithm to create CBV, CBF and MTT maps. These maps were created twice: once for AIFs selected from the ipsilateral and once from the contralateral hemisphere, with the maps resulting from the contralateral hemisphere considered as the globally defined perfusion metrics unless otherwise noted. In addition, the local AIF algorithm described in Chapter 3 was used on all datasets, thus leaving three different sets of data to compare.

5.2.4 General Linearized Model

The GLM computed risk probabilities for every voxel using the following six parameters: LowB, ADC, DWI, CBV, CBF, and MTT in a method identical to that used by Wu, et al (18). Just as in that work, the training of the model was accomplished via jackknifing (19). Jackknifing is a technique used to avoid the bias of running a model on the same data that it was trained on. This is done by letting the model coefficients for each patient be those obtained by training the model on all of the other patients. Separate training and evaluation steps were performed for the global ipsilateral and contralateral AIF methods as well as the local AIF method, based on outlines drawn by an experienced neuroradiologist.

These risk probabilities were then used to calculate the following statistics about the model’s performance over the entire brain (based upon the above mentioned outlines): the true positives (TP, the number of voxels predicted to infarct that actually did); the false positives (FP, the number of voxels predicted to infarct which did not); the true negatives (TN, the number of voxels predicted to not infarct that did not infarct); and the false negatives (FN, the number of voxels predicted to not infarct that actually infarcted). From these statistics, the true positive ratio (TPR), or sensitivity, and the true negative ratio (TNR), or specificity, were calculated as: TPR=TP/(TP+FN) and TNR=TN/(TN+FP). The ROC curves were then plotted as the TPR (sensitivity) versus
the false positive ratio (FPR, or 1 - specificity). Finally, the area under the curve (AUC), which has been shown to represent the probability that an image will be correctly ranked normal or abnormal (20), was calculated via trapezoidal integration. Statistical comparison was performed by a Wilcoxon signed-rank test performed on the AUCs for each patient. Differences were deemed significant for p-values < 0.01.

5.2.5 MTT Lesion Volumes
Lesion volumes were outlined by an experienced neuroradiologist for the MTT maps generated by the contralateral, ipsilateral, and local AIF methods. These lesion volumes were then compared to the follow-up lesion volume and the mean squared error was calculated for the three methods. In addition, volumes were compared on a patient-by-patient basis with volumes classified as different only if they had a 20% volume disagreement. This was done in order to reduce bias associated with possible interexaminer disagreement in the outlines.

5.3 Results
5.3.1 ROC Analysis
Figure 5.3.1.1A shows the ROC curves generated from running the multivariate GLM three times: once with each of the globally generated parameter sets and also with the locally generated set with only the CBF and MTT differing as input to the GLM. The local AIF approach has a higher ROC curve than either one of its global AIF counterparts. Noticeably, the local AIF metrics show an increased sensitivity in areas of high specificity (FPR < 0.3) as compared to both of the global AIF metrics, while areas of low specificity show a similar sensitivity. This means that in the region of high specificity, given an acceptable probability of incorrectly predicting that a tissue will infarct, the locally defined metrics will correctly predict a tissue to infarct with higher probability. This would make the local metrics preferable in the clinical setting, as it is preferable to not classify healthy tissue as unhealthy with high probability (particularly since it leads to minimal gains in accurately labeling tissue as unhealthy).

Comparing the AUCs on a patient by patient basis, it was found that the locally generated
perfusion metrics had an AUC value that was greater than the AUC for the two globally generated metrics in 31 out of the 36 cases ($p < 0.001$ by binomial distribution). As shown in Figure 5.3.1.1B, the largest margin by which the local AUC was higher than the contralateral global AUC was 0.1020, with a mean of 0.037. However, the largest margin by which the contralateral global AUC was higher than the local AUC was 0.050, with a mean of 0.017. Similarly, as shown in Figure 5.3.1.1C, the largest margin that the local AUC improved as compared to the global ipsilateral metrics was 0.1420 with a mean of 0.036. On the other hand, the global ipsilateral AUC was higher than the local AUC with a maximum margin of 0.045 and mean 0.021. In addition, the differences between the two globally generated metrics are shown in Figure 5.3.1.1D. The ipsilateral AUC values were higher for 18 out of 36 patients with 3 patients having equal AUC values. The average margin by which the ipsilateral was higher than the contralateral was 0.0170, while the contralateral was higher with a mean of 0.0166.

Table 5.3.1.1 shows the mean and standard deviation of the GLM coefficients over all 36 patients used in this study. The ADC, DWI, and CBV, and coefficients decreased for the local AIF model, while the LowB, CBF, an MTT coefficients increased as compared to the global AIF models.
Figure 5.3.1.1 Plot A shows the ROC curves for the local and global AIF approaches pooling results from 36 patients. The local AIF method shows a significantly higher AUC compared to both of the global AIF methods. This can be seen in the regions of high specificity, where the local AIF method shows an increased sensitivity. Plots B-D show the stem plots of AUC differences on a patient by patient basis for local versus contralateral AIF, local versus ipsilateral AIF and contralateral versus ipsilateral AIF. The local AUCs were higher than the global AUCs in 31 out of 36 patients, while the ipsilateral AUC was higher than the contralateral AUC in 18 out of 36 patients.

### Table 5.3.1.1. Estimate and Error of the GLM Coefficients for all 36 subjects

<table>
<thead>
<tr>
<th></th>
<th>LowB</th>
<th>ADC</th>
<th>DWI</th>
<th>CBF</th>
<th>CBV</th>
<th>MTT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local AIF</td>
<td>0.71±0.14</td>
<td>0.11±0.17</td>
<td>4.22±0.12</td>
<td>-1.17±0.04</td>
<td>-0.27±0.04</td>
<td>0.67±0.01</td>
</tr>
<tr>
<td>Contra AIF</td>
<td>0.30±0.15</td>
<td>0.63±0.18</td>
<td>4.46±0.13</td>
<td>-2.61±0.05</td>
<td>0.74±0.03</td>
<td>0.01±0.003</td>
</tr>
<tr>
<td>Ipsi AIF</td>
<td>0.31±0.14</td>
<td>0.55±0.17</td>
<td>4.53±0.13</td>
<td>-2.63±0.05</td>
<td>1.03±0.04</td>
<td>0.02±0.002</td>
</tr>
</tbody>
</table>
5.3.2 MTT Lesion Volumes
As compared to the follow-up lesion volumes, the local MTT lesion volumes were closer (i.e. considered different and were nearer to the follow-up volume) than the contralateral volumes in 23 patients, and closer than the ipsilateral volumes in 19 patients. Overall, the local method had a mean-square error (MSE) of $5.11 \times 10^{3} \text{cm}^6$. On the other hand, the ipsilateral volumes were closer than the local volumes in 8 patients, while the contralateral volumes were never closer than the local volumes. Overall, the contralateral method had a MSE of $1.58 \times 10^{4} \text{cm}^6$, and the ipsilateral had a MSE of $9.03 \times 10^{3} \text{cm}^6$, a result consistent with (16).

5.3.3 Example Cases
Figure 5.3.3.1 shows the input and resulting output of the GLM for patient 19. This was a 70 years old male who presented with acute onset of left sided facial droop and left hemiparesis. He was discovered to have a completely occluded right internal carotid artery upon admission and rushed into the scanner 90 minutes after the symptom onset. There is a noticeable increase in the flow estimates in the right (ipsilateral) hemisphere of the brain which leads to compelling differences in the MTT map in the same hemisphere. This difference is reflected in the tissue outcome maps produced by the GLM for the two different inputs. When compared to the F/U T2, the tissue outcome maps that used CBF maps calculated from local AIFs showed a visually apparent decrease in risk of infarction for the regions which did not go on to infarction and a higher risk of infarction in regions that did infarct.
Figure 5.3.3.1. Sample inputs and GLM outputs for patient 19, a 70 year-old patient with sudden onset left facial droop and left hemiparesis due to right internal carotid artery occlusion. The acute MRI was done 90 minutes after the symptom onset. The local AIF approach leads to increased flow estimates (ICBF) in the right (ipsilateral) hemisphere when compared to the flow estimates produced by the global AIF approach (gCBF). The use of local AIF approach produces substantial changes in the GLM tissue outcome map, as it assigns higher risks of infarction to regions that eventually infarcted (as seen on follow-up T2 on day 8 of stroke onset) as well as lowering risks of infarction in regions that did not infarct.

Figures 5.3.3.2 and 5.3.3.3 provide additional examples of the beneficial effects of the local AIF approach. Fig. 5.3.3.2 shows the time course of the contrast agent in patient 3, a 71 year-old male with right middle cerebral artery syndrome. There was severe stenosis of the right internal carotid artery. In his initial study performed 4 hours and 40 minutes after the symptom onset, a large delay was observed between the right and left hemispheres of the brain. This difference is clearly picked up by the local AIF algorithm, as shown in Fig. 5.3.3.2, where the delay in the right hemisphere is observed to be about 6 seconds.
Figure 5.3.3.2. Comparison of the global and local AIFs in a patient with acute infarction within the territory of the right middle cerebral artery due to severe ipsilateral internal carotid artery stenosis. A large difference between the two functions can be seen in the right (ipsilateral) hemisphere of the brain, as the input function to the right hemisphere is delayed and more dispersed as compared to the left hemisphere. This delay and dispersion can lead to flow misestimations, as shown in Figure 5.3.3.3.

The results of accurately modeling the delay in this patient are shown in Fig. 5.3.3.3, where the globally generated CBF and MTT maps are compared to the locally generated CBF and MTT maps, as well as with the follow-up T2-weighted images obtained 6 days after stroke onset. This figure also demonstrates that the local AIF approach can overcome possible discrepancies with single manual AIF selection. The perfusion metrics calculated with a contralateral AIF indicate larger areas of tissue at risk, with low flow and a high transit time in the anterior right hemisphere as compared to the ipsilaterally calculated metrics. However, the perfusion metrics obtained via the local AIF algorithm clearly resolve the difference in the anterior right hemisphere, indicating normally perfused tissue and also indicate increased flow in the posterior right hemisphere. When compared to the follow-up T2 scan, these improvements occurred in areas that did not infarct. Upon further comparison, the local MTT map seems an accurate indicator for final infarction volume.
Figure 5.3.3.3. Example of bias resulting from manual AIF selection. (A) and (D) show the CBF and MTT maps that result from selecting a global AIF from the ipsilateral MCA, while (B) and (E) show the CBF and MTT maps that result from selecting a global AIF from the contralateral MCA. Notice the difference between these two maps in the anterior right hemisphere of the brain. The CBF and MTT maps that result from using the local AIF algorithm are shown in (C) and (F). Note the local AIF algorithm’s improved performance in both the anterior and posterior areas of the right hemisphere when compared to (G), the follow-up T2 image.

5.4 Discussion

We have presented an automated method for calculating perfusion weighted MR metrics via locally defined arterial input functions. We then compared the local metrics to the global metrics produced from both the ipsilateral and contralateral hemispheres in terms of their predictive value. Using a GLM, we found that the local metrics outperformed the global metrics in terms of pooled AUC as well as in AUC differences on a patient by patient basis. In addition, these results were reinforced by examining MTT lesion volumes, with the local volumes having the least MSE and being closer to the follow-up in a majority of patients. While the results were not as strong as the GLM, this can most likely be explained by the fact that they were strongly based upon one metric, while the GLM incorporates six.
By examining individual patients, we showed that the local algorithm was able to improve the resulting perfusion metrics in terms of both predictive value (Fig. 5.3.3.1) as well as bias elimination (Fig. 5.3.3.3). The local AIF method was able to make these improvements presumably due to its implicit regard for dispersion and delay (Fig. 5.3.3.2). However, these are preliminary results that should be confirmed with larger studies, it appears that by using this local algorithm, perfusion metrics that are less sensitive to human variability and pathological conditions can be automatically generated.

Looking at the plot of AUC differences on a patient by patient basis in Figures 5.3.1.1B&C, we see patient 2, for which the global methods seem very preferable to the local method. Upon inspection, this dataset reveals a potential limitation of the local AIF method: low image quality. Low contrast to noise ratio (CNR) can make choosing an AIF in regions of low flow difficult because large noise fluctuations can deteriorate a genuine AIF or be mistaken for a concentration increase, resulting in poor performance. The global AIF method would perform better since it requires a single AIF while the local AIF method typically chooses on the order of a hundred over the entire dataset. Thus, it is easier to find a single valid AIF as opposed to a hundred.

In addition, large patient motions (even after motion-correction), susceptibility artifacts or vessel pulsation can adversely affect the local AIF performance as they can lead to artificial signal drops that create artificially high concentrations. These appear as regions of lower CBF estimates upon deconvolution from the concentration-time curves. Then again, without care in, auto or semi-automatic global AIF selection could choose a flawed AIF as well and produce worse perfusion metrics than the local approach since the error would not be isolated to a certain region of the brain. However, with proper scanner maintenance and contrast agent administration, these effects can be minimized and localized AIFs can be safely used.

This study is limited by the relatively small number of cases and we did not compare the
interobserver variability of global AIF generation. Other studies have shown that using different AIF locations can have a significant impact on the PWI lesion size and prenumbra volume (G), so we plan to do similar studies including local AIF metrics in the future. Nevertheless, if these results uphold in larger studies, adoption of this local AIF-based approach could provide two potentially important clinical benefits. First, the apparently superior quality perfusion maps, could improve the utility of PWI in the management of stroke patients. Second, the automatic generation of perfusion maps could remove the need for human interaction. Thus, with further optimization, the local AIF approach could be implemented directly on the MRI scanner console, and the maps could be available minutes after acquisition. In our experience, this time can make the difference between having accurate CBF and MTT maps being used clinically versus not.

5.5 Conclusions

Based upon these results, we conclude that fully automated CBF calculation using localized AIFs is feasible and appears to produce more useful perfusion metrics when applied to reasonable-quality datasets. As the example cases showed, this is most likely due to the ability of the local AIFs to account for the delay and dispersion of the contrast bolus. In addition, the automated nature of the local AIF method could improve stroke treatment by giving superior perfusion metrics quickly. Thus, we conclude that this method merits further investigation for the evaluation of acute stroke patients in terms of accurate prediction of tissue outcome.
Chapter 6: Conclusions

6.1 Summary
This thesis developed and validated an algorithm for calculating PWI metrics automatically using localized arterial input functions. Chapter 2 described the method and shortcomings of the standard technique of PWI calculation via a manually selected global AIF. It also described the basis for the technique, as well as the properties of AIFs. Chapter 3 described the localized AIF algorithm, its justification, and some of its limitations.

Chapter 4 demonstrated that this local AIF algorithm was fulfilling its theoretical potential in stroke patients by improving flow estimates in situations of delay and dispersion of the contrast bolus. This was done by examining the performance of the local method as compared to the global method for normal and abnormal levels of underlying tissue delay and dispersion. In addition, it was found that the local method was estimating many significant increases in flow in the hemisphere of the brain which contained the stroke lesion. Furthermore, the underlying tissue concentration curves for these improved voxels displayed elevated levels of delay and dispersion. Since delay and dispersion of the contrast bolus relative to the AIF has been shown to create flow underestimations, this suggested that the local AIF method was correcting some of these problems.

Chapter 5 demonstrated that the local AIF algorithm, when used to predict final tissue outcome in stroke, was more accurate than global AIF algorithms using AIFs selected from either the ipsilateral or contralateral hemisphere. This prediction was tested in two ways: by using a general linearized model (GLM) and also by having a neuroradiologist outline lesion volumes on the resulting mean transit time maps. The GLM made use of six different image acquisitions to predict tissue risk of infarction, which were compared to lesions outlined on follow-up T2 images to generate a receiver operating characteristic (ROC) curve. The local AIF method had a higher ROC curve than either of the global metrics, implying that it is a better predictor for final infarct. 
addition, the MTT lesion volumes outlined on the locally defined metrics had the minimum mean squared error over all three methods when compared with the follow-up T2 lesion volumes.

These results suggest that indeed the local AIF algorithm presented can be a very useful technique for calculating perfusion metrics with MRI due to its decreased sensitivity to delay and dispersion, which allows it to make better predictions of tissue outcome. Furthermore, it is a valuable tool since it is automated and can be directly implemented on an MRI scanning console to produce perfusion metrics in a short time with just the click of a few buttons.

6.2 Suggestions for Further Study

These results were obtained on a population of only 36 patients, so larger studies are needed to completely verify the benefits of the local AIF algorithm. Different field strengths, temporal resolutions, or other scanning parameters may make it necessary to modify or customize the algorithm for different situations. Most likely, these changes would only require changes to some of the parameters of the algorithm. For use with lower SNR or other non-ideal datasets, new ways of excluding corrupted voxels from AIF consideration could also make the algorithm perform better, although these would most likely become unnecessary at higher SNR.

A simple extension of the local AIF algorithm can be explored with the use of a new technique that calculates both spin and gradient echo EPI images nearly simultaneously. Since these two methods of image acquisition are more sensitive to small and large vessels, respectively (6), the local AIF algorithm could be used to find the locations for AIF selection on the gradient echo acquisition and apply those locations to the spin echo acquisition. Doing this could potentially allow for even more useful perfusion metrics. While this algorithm has been implemented, its application has not been studied or evaluated.
Furthermore, a new technique using arterial spin labeling (ASL) has been shown to be capable of generating quantitative CBF non-invasively (21). The correlation between ASL and PWI values could then be compared, and this could potentially be another way to validate the results of the local AIF algorithm. Currently however, this comparison is difficult to make. The SNR of the ASL is lower than that of the PWI, and getting the ASL at the 128x128 spatial resolution of PWI has some difficulties associated with it. Figure 6.2.1 shows scatter plots of ASL CBF versus local and global CBF values for spin and gradient echo. It has been reported that there is a near linear relationship between the two methods (22). However, such a relationship is not evident in Figure 6.2.1. With more work, higher field strengths, or modified pulses sequences, this relationship could be reproduced and ASL could be another way of verifying the local AIF method.

Figure 6.2.1 Scatter plots of PWI CBF versus ASL CBF. Plots A and B are the plot of local gradient and spin echo relative CBF versus the ASL CBF. Similarly, plot C and D are the plots of global gradient and spin echo relative CBF versus the ASL CBF. (The ASL CBF is the same in every plot.) Obviously, there are important differences in these plots and a linear relationship is not apparent. Understanding these differences and using ASL for validation of the local AIF method is an excellent extension of this thesis.
Appendix A: References


Appendix B: Local AIF Algorithm Code

Main Function

function [AIF,cbv,CF,B,MTT] = LocalAIFvl_3(inDir,outDir,Prefix,Nslice,Postfix,Type,TR,TE,thick,thick,Gap,FOVx,FOVy,smooth,exclude,postex,kk);

LocalAIFvl_3 Computes a local AIF, CBV, CBF, and MTT given a PWI data set

Where:

- inDir is the directory where the input data is located.
- outDir is the directory where the output data will be stored.
- Prefix is the prefix before the dot, without slice numbers.
- Nslice is the number of slices.
- Postfix is after and including the dot.
- Type specifies whether the data is in one big chunk or in slices.
- TR is the repetition time (time between samples)
- TE is the echo time.
- Thick is the slice thickness, in mm
- Gap is the spacing between the slices, in mm
- FOVx is the Field of View in the x dimension in cm
- FOVy is the Field of View in the y dimension in cm
- Smooth = 0 means do not smooth input signal in time.
- Exclude is the number of initial time points to exclude
- Post_ex is the number of end points to exclude
- AIF is the locally defined AIF, (not written)
- CBV is the Cerebral Blood Volume (written to OutDir)
- CBF is the Cerebral Blood Flow (written to OutDir)
- MTT is the Mean Transit Time (written to OutDir)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Start
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Read in the Dataset in one of two ways.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
cd(inDir);
if (Type == 0)
   i = 0;
   [temp,nr,nc,nsamp,e] = read_mgh_tb(strcat(Prefix, '00', Postfix));
   Sig = zeros(nr,nr,Nslice,nsamp-exclude-post_ex);
   Sig(:,i+1,:) = temp(:,1+exclude:nsamp-post_ex);
   for i = 1:Nslice-1,
      if (i < 10)
         num = strcat('0',num2str(i));
      else
         num = num2str(i);
      end
   [temp,nr,nc,nsamp,e] = read_mgh_tb(strcat(Prefix, num, Postfix));
   Sig(:,i+1,:) = temp(:,1+exclude:nsamp-post_ex);
   end
else
end
[Sig,nr,nc,total,e] = read_mgh_tb(strcat(Preffix,Postfix));
[Sig, nsamp] = temporal(Sig,Nslice);
Sig = Sig(:,:,1:exclude:nsamp-post_ex);
end

clear temp;

%% Extract out the brain and calculate some important stats for later
[nr,nc,nsl,nsamp] = size(Sig);

%% Noise Threshold in STDs
% This noise threshold worked very well! kk = 8;
% We gain information on the noise by looking at the corner of the set
mMean = mean(Sig(nr-floor(nr/2),nc-5,nsl-1,:));
mSTD = std(Sig(nr-floor(nr/2),nc-5,5,:));

%Find the AIFs Using an approximately 28x28x28 mm^3 search.
i = floor((nr/FOVx)*28/2)*2 + 1;
j = floor((nc/FOVy)*2.8/2)*2 + 1;
k = floor(12/(thick+gap))*2+1;
ext = max(ni,nj);

% Initializes a simple mask to sort out the noise.
pre_mask = zeros(nr,nc,nsl);
pre_maskp = pre_mask;
extra_mask = zeros(nr,nc,nsl);
warning off;
for i = 1:nsl,
    % Looks at the first image to determine brain or not.
    temp = Sig(:,:,i,1);
    % This is a simple threshold based on user input.
    Q = find(temp <= mMean + kk*mSTD);
    temp = temp(Q);
    % Now morphological operators are used to extract brain
    temp2 = ones(nr,nc); temp2(Q) = 0;
    temp2 = erode(temp2, ones(3,3));
    temp2 = erode(temp2, ones(3,3));
    temp2 = erode(temp2, ones(3,3));
    temp2 = erode(temp2, ones(3,3));
    temp2 = erode(temp2, ones(3,3));
    temp2 = erode(temp2, ones(3,3));
    temp2 = erode(temp2, ones(3,3));
    temp2 = erode(temp2, ones(3,3));
    temp2 = erode(temp2, ones(3,3));
    temp2 = erode(temp2, ones(3,3));
    temp2 = erode(temp2, ones(3,3));
    temp2 = erode(temp2, ones(3,3));
    temp2 = erode(temp2, ones(3,3));
    temp2 = erode(temp2, ones(3,3));
    temp2 = erode(temp2, ones(3,3));
    temp2 = erode(temp2, ones(3,3));
    temp2 = erode(temp2, ones(3,3));
    temp2 = erode(temp2, ones(3,3));
    temp2 = erode(temp2, ones(3,3));
    temp2 = erode(temp2, ones(3,3));
    temp2 = erode(temp2, ones(3,3));
    temp3 = dilate(temp2,ones(ext,ext));
    pre_mask(:,:,i) = temp2;
    shell(:,:,i) = temp2 - erode(temp2,ones(3,3));
    extra_mask(:,:,i) = temp3-temp2;
end

% This is then an indication of what is brain and what isn't.
Relevant = find(pre_mask);

% Smooth the input dataset in space
scon = 3*22/128;
dimpri = max(nr,nc)/max(FOVx,FOVy)*scon;
sdim = 2*floor(dimpri/2) + 1;
for i=1:nsamp,
    Sig(:,:,i) = Gsmooth(Sig(:,:,i),sdim,.8*floor(dimpri/2));
end

% Find where largest signal drop is, but only in brain.
[Y, I] = min(Sig,[],4);
Relevant_I = I(Relevant);

% This calculates average arrival time. (in samples)
arrive_time = mean(Relevant_I(:));
display(arrive_time);
clear Relevant_I Relevant;

% Calculate baseline to be twelve seconds before the arrival time
Base = mean(Sig(:,:,1:round(arrive_time-12/TR)),4);

% Calculate the Concentration and subjects it to the masking.
% Replicate Baseline
Baseline = padarray(Base, [0 0 0 nsamp-1], 'replicate', 'pre');
% Make sure to divide by zero
Q = find(IMG_in==0);
Q2 = find(Baseline==0);
Q3 = union(Q,Q2);
IMG_in(Q3)=1;
Baseline(Q3) = 1;
% Calculate concentration
Conc = -log(Sig./Baseline);
clear Sig, Baseline;

% Get rid of extraneous non-brain voxels
mask = padarray(pre_mask,[0 0 0 nsamp-1], 'replicate', 'pre');
Conc = Conc./TE .* mask;
clear Base;

% Here the CBV is calculated.
cbv = trapz(Conc,4);
Q = find(cbv <0);
cbv(Q) = 0;
cbv = cbv .* mask(:,:,1);

if not(smooth == 0)
    temp_Conc = Conc;
    Conc2 = zeros(size(temp_Conc));
    Conc3 = zeros(size(temp_Conc));
    Conc1 = 2*temp_Conc/3;
    Conc2(:,:,2:nsamp) = temp_Conc(:,:,1:nsamp-1)/6;
    Conc3(:,:,1:nsamp-1) = temp_Conc(:,:,2:nsamp)/6;
    temp_Conc = Conc1 + Conc2 + Conc3;
    Conc = temp_Conc;
end
clear Concl temp_Conc Conc2 Conc3;
end

Q = find(Conc < 0);
Conc(Q) = 0;
cbv_orig = CBV(Conc);
display('CBV Computed');

% Here we define the high and low Vols for the selection of AIFs.
max_vol = max(max(max(cbv)));
t_low = 0.05 * max_vol;
t_high = 0.6 * max_vol;

% To avoid dividing by zero.
Q = find(cbv_orig == 0);
cbv_orig2 = cbv_orig;
cbv_orig2(Q) = Inf;

% Here we begin to compute the moments.
nums = zeros(nc, nr, nsl, nsamp);
for i = 1:nsamp,
    % This normalizes the Conc for computation of the first moment
    Conc(:,:,i) = Conc(:,:,i)./cbv_orig2;
    nums(:,:,i) = i*ones(nc, nr, nsl);
end
clear mask;

Concp = Conc .* nums;
clear nums cbv_orig cbv_orig2;

moment = zeros(nr, nc, nsl);
% Computes the first moment
moment = trapz(Concp,4);
clear Concp;

% Go through series of checks to throw out bad voxels
% Set the moments to infinity which do not fit in the volume range.
Q = find(cbv < t_low);
Q2 = find(cbv > t_high);
Q = union(Q, Q2);
moment(Q) = Inf;

% This will make the "bad" voxels Inf, and leave the "good" voxels alone.
Q = find(moment == 0);
moment(Q) = Inf;

% Now I get rid of points whose moments are obviously too low.
% (i.e. Before the bolus arrived.)
Q2 = find(moment < arrive_time - 7.5/TR);
moment(Q2) = Inf;

% This is a function which checks for the 'AIFness' of each voxel and
% throws out voxels which are noisilike.
Lp = Check_TR(Conc, arrive_time, TR);
Qc = find(Lp == 0);
moment(Qc) = Inf;
Compute 4 different Criterion

Q = find(moment == Inf);

First Criterion is First Moment

criterion1 = moment;

criterion2 = ones(nr,nc,nsl)*Inf;

criterion3 = ones(nr,nc,nsl)*Inf;

criterion4 = ones(nr,nc,nsl)*Inf;

Third Criterion is highest peak

[Y,I] = max(Conc,[],4);
criterion3 = Y;
criterion3(Q) = Inf;

Second Criterion is Average slope.
run = zeros(nr,nc,nsl);
rise = zeros(nr,nc,nsl);

for i=1:nr*nc*nsl,
if not(criterion1(i) == Inf)
    if (I(i) < round(18/TR))
        before = I(i);
    else
        before = round(18/TR);
    end
    % This computes the average Slope
    l = (I(i)-before)*nr*nc*nsl:nr*nc*nsl:(I(i)-1)*nr*nc*nsl);
    look = squeeze(Conc(i+l));
    Qp = find(look < .05);
    if not(isempty(Qp))
        [Y2,I2] = max(Qp);
        run(i) = before-Y2;
        rise(i) = Y(i)-Conc(i+nr*nc*nsl*(I(i)-run(i)));
        % Fourth Criterion is the FWHM.
        l = (0:nr*nc*nsl:(nsamp-l)*nr*nc*nsl);
        criterion4(i) = fwhm(squeeze(Conc(i+l)));
    end
end

warning off;
criterion2 = rise./run;
warning on;
criterion2(Q) = Inf;
criterion4(Q) = Inf;

clear rise run;

Create the matrix which will accomplish the interpolation and filtering.

sigma = sqrt(.5 * ((ni-1)/2)^2/log(2));
G2 = fspecial('gaussian', [ni,nj],sigma);
% This also gives a half max at 28mm
alph = (nk/2-1/2)^2/log(2);
\( G_1 = 1:1:nk; \) \( G_1 = G_1-(nk/2+1/2); \)
\( G_1 = \exp(-G_1.^2/\alpha); \)
\( G_{2p} = \text{padarray}(G_2, [0 0 \text{nk-1}], 'replicate', 'pre'); \)
\% Replicates \( G_2 \) in 3rd Dim
\( G_3 = \text{zeros}(ni,nj,nk); \)
\text{for} \( i = 1:ni, \)
\text{for} \( j = 1:nj, \)
\( G_3(i,j,:) = G_1' \cdot \text{squeeze}(G_{2p}(i,j,:)); \)
\text{end}
\text{end}
\( G_3 = G_3*1000; \)
\( G_4 = \text{padarray}(G_3, [0 0 \text{nsamp-1}], 'replicate', 'pre'); \)
\% Replicates \( G_3 \) in 3rd Dim
\( \text{div} = 3; \)
\% Replicates \( G_3 \) in 3rd Dim
\%\% Now we finally calculate the AIF.
\( \text{AIF}_{\text{int}} = \text{zeros}(nr,nc,n_{\text{sl}},\text{nsamp}); \)
\( h_1 = \text{floor}(ni/2); \)
\( h_j = \text{floor}(nj/2); \)
\( h_k = (nk-1)/2; \)
\% Converts the criterion into scores.
\%\% Now we finally calculate the AIF.
\( \text{freq}_{\text{map}} = \text{zeros}(nr,nc,n_{\text{sl}}); \)
\%\% Large Search Loop
\text{for} \( i = \text{ceil}(ni/2):nr-\text{floor}(ni/2), \)
\text{for} \( j = \text{ceil}(nj/2):nc-\text{floor}(nj/2), \)
\text{for} \( k = 1:n_{\text{sl}}, \)
\text{if} \( \text{pre}_{\text{mask}}(i,j,k) == 1 \) \% Don’t calculate for voxels not in brain.
\% These test for the extreme values which cannot do a full search
\text{if} \( k-h_k < 1 \) \% First slice
\( \text{addk} = h_k-k+1; \)
\( \text{crit}_1 = \text{criterion}_1(i-h_i:i+h_i,j-h_j:j+h_j,1:k+h_k,:); \)
\( \text{crit}_2 = \text{criterion}_2(i-h_i:i+h_i,j-h_j:j+h_j,1:k+h_k,:); \)
\( \text{crit}_3 = \text{criterion}_3(i-h_i:i+h_i,j-h_j:j+h_j,1:k+h_k,:); \)
\( \text{crit}_4 = \text{criterion}_4(i-h_i:i+h_i,j-h_j:j+h_j,1:k+h_k,:); \)
\( \text{conc}_{\text{cube}} = \text{Conc}(i-h_i:i+h_i,j-h_j:j+h_j,1:k+h_k,:); \)
\text{elseif} \( k+h_k > n_{\text{sl}} \) \% Last Slice
\( \text{addk} = 0; \)
\( \text{crit}_1 = \text{criterion}_1(i-h_i:i+h_i,j-h_j:j+h_j,k-h_k:n_{\text{sl}}); \)
\( \text{crit}_2 = \text{criterion}_2(i-h_i:i+h_i,j-h_j:j+h_j,k-h_k:n_{\text{sl}}); \)
\( \text{crit}_3 = \text{criterion}_3(i-h_i:i+h_i,j-h_j:j+h_j,k-h_k:n_{\text{sl}}); \)
\( \text{crit}_4 = \text{criterion}_4(i-h_i:i+h_i,j-h_j:j+h_j,k-h_k:n_{\text{sl}}); \)
\( \text{conc}_{\text{cube}} = \text{Conc}(i-h_i:i+h_i,j-h_j:j+h_j,k-h_k:n_{\text{sl}}); \)
\text{else} \% Middle Slices.
\( \text{addk} = 0; \)
\( \text{crit}_1 = \text{criterion}_1(i-h_i:i+h_i,j-h_j:j+h_j,k-h_k:k+h_k); \)
\( \text{crit}_2 = \text{criterion}_2(i-h_i:i+h_i,j-h_j:j+h_j,k-h_k:k+h_k); \)
\( \text{crit}_3 = \text{criterion}_3(i-h_i:i+h_i,j-h_j:j+h_j,k-h_k:k+h_k); \)
\( \text{crit}_4 = \text{criterion}_4(i-h_i:i+h_i,j-h_j:j+h_j,k-h_k:k+h_k); \)
\( \text{conc}_{\text{cube}} = \text{Conc}(i-h_i:i+h_i,j-h_j:j+h_j,k-h_k:k+h_k,:); \)
\text{end}

\% Find voxel with minimum score, get index and then exclude
\text{search}_{\text{cube}} = \text{score}_{\text{3}}(\text{conc}_{\text{cube}},\text{crit}_1,\text{crit}_2,\text{crit}_3,\text{crit}_4,\text{TR});
\[Y,I]\ = \text{min} (\text{search}_{\text{cube}}(:));
\text{search}_{\text{cube}}(I) = \text{Inf};
if (Y==Inf | isempty(I)) % make sure we find something.
ex1 = 0; ex2 = 0; ex3 = 0; I2 = 1; I3 = 1; div = 1;
else

  % Calculate the exact coordinate of the found voxel
  Ip = I-1;
  Ik = floor(Ip/(ni*nj))-hk+addk;
  I2p = Ip-(Ik+hk-addk)*ni*nj;
  Ij = floor(I2p/ni)-hj;
  Ii = I2p-(Ij+hj)*ni-hi;
  
  % Get its time series and Gaussian weight it.
  ex1 = Conc(i+Ii,j+Ij,k+Ik,:).*G4(I);
  freq_map(i+Ii,j+Ij,k+Ik) = freq_map(i+Ii,j+Ij,k+Ik)+1;

  % Find Next lowest score value, get index and then exclude
  [Y,I2] = min(search_cube(:));
  search_cube(I2) = Inf;

  if (Y == Inf) % make sure we find something.
    ex2 = 0; ex3 = 0; I3 = 1; div = 1;
  else
    % Calculate the exact coordinate of the found voxel
    Ip = I2-1;
    Ik = floor(Ip/(ni*nj))-hk+addk;
    I2p = Ip-(Ik+hk-addk)*ni*nj;
    Ij = floor(I2p/ni)-hj;
    Ii = I2p-(Ij+hj)*ni-hi;
    
    % Get its time series and Gaussian weight it.
    ex2 = Conc(i+Ii,j+Ij,k+Ik,:).*G4(I2);
    freq_map(i+Ii,j+Ij,k+Ik) = freq_map(i+Ii,j+Ij,k+Ik)+1;

    % Find last lowest score value, and its index
    [Y,I3] = min(search_cube(:));
    if (Y == Inf)
      ex3 = 0;
    else
      % Calculate the exact coordinates pf voxel
      Ip = I3-1;
      Ik = floor(Ip/(ni*nj))-hk+addk;
      I2p = Ip-(Ik+hk-addk)*ni*nj;
      Ij = floor(I2p/ni)-hj;
      Ii = I2p-(Ij+hj)*ni-hi;
      
      % Get its time series and Gaussian weight it.
      ex3 = Conc(i+Ii,j+Ij,k+Ik,:).*G4(I3);
      freq_map(i+Ii,j+Ij,k+Ik) = ...
                                freq_map(i+Ii,j+Ij,k+Ik)+1;
      end
      end

  end

  % Sum up time series, normalize and store
  AIF_int(i,j,k,:) = (ex1+ex2+ex3)/(G4(I)+G4(I2)+G4(I3));
end
end
end

clear G4;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Here we extend the AIF out so that it is not tailing off at the edges.

65
AIF_ext = zeros(nr,nc,nsl,nsamp);

extp = (extp+1)/2;
% Shell defines the outside shell of the brain, we extend this in time
shelled = padarray(shell,[0 0 0 nsamp-1], 'replicate', 'pre');
% Get this outside shell from AIF above.
shelled = shelled.* AIF_int;
dime = 2*extp+1;
for i = 1+extp:nr-extp,
    for j = 1+extp:nc-extp,
        for k = 1:nsl,
            if extra_mask(i,j,k) == 1
                % Find all elements in shell within a certain box around the voxel
                Q = find(shelled(i-extp:i+extp,
                    j-extp:j+extp,k,floor(arrive_time)));
                num = length(Q);
                for h = 1:num,
                    % Find where these elements exist
                    Qp = Q(h)-1;
                    Ij = floor(Qp/dime)-extp; Ii = Qp-(Ij+extp)*dime-extp;
                    % Take their average as the resulting edge AIF
                    AIF_ext(i,j,k,:) = AIF_ext(i,j,k,:) + ...
                        shelled(i+Ii,j+Ij,k,:)/num;
                end
            end
        end
    end
end
% Add in actual AIF to get final AIF.
AIF_ext = AIF_ext + AIF_int;
clear AIF_int;

% Extend AIF to an extra slice above and an extra slice below to avoid a bias
% on the slices.
AIF_int3 = zeros(nr,nc,nsl+2*hk,nsamp);
AIF_int3(:,:,l+hk:nsl+hk,:) = AIF_ext;
AIF_int3(:,:,l:hk,:) = ...
    padarray(AIF_ext(:,:,l,:),[0 0 0 nsamp-1], 'replicate', 'pre');
AIF_int3(:,:,nsl+hk+l:nsl+2*hk,:) = ...
    padarray(AIF_ext(:,:,nsl,:),[0 0 0 nsamp-1], 'replicate', 'pre');
clear AIF_ext;

% Now we do a spatial filtering of the AIF.
AIF_intf = zeros(nr,nc,nsl+2*hk,nsamp);
for i = 1:nsamp,
    AIF_intf(:,:,i,:) = imfilter(AIF_int3(:,:,i,:), G3);
end
AIF = AIF_intf(:,:,l+hk:nsl+hk,:);
clear AIF_int3 AIF_intf;
AIF = AIF.* padarray(pre_mask,[0 0 0 nsamp-1], 'replicate', 'pre');
display('Filtering Done');
%% Now we calculate the CBF.
%CBF_Ai = CBF_Deconv(Conc, AIF, pre_mask); Old Function call

\begin{verbatim}
warning off;
CBF_Ai = zeros(nr,nc,nsl);

%% Gets indices for Diagonal elements.
W = eye(nsamp);
Q = find(W);
R = zeros(1,nsamp);
for i=1:nr
    for j=1:nc,
        for k=1:nsl,
            if pre_mask(i,j,k) == 1 %% Don't calculate for non-brain
                tempAIF = squeeze(AIF(i,j,k,:));

                % this linearizes the data.
                AIF2 = zeros(nsamp,1);
                AIF3 = zeros(nsamp,1);
                AIF1 = 2*tempAIF/3;
                AIF2(2:nsamp) = tempAIF(1:nsamp-1)/6;
                AIF3(1:nsamp-1) = tempAIF(2:nsamp)/6;
                tempAIF = AIF1 + AIF2 + AIF3;

                % R is used to create the block-matrix for the SVD
                R(l) = tempAIF(l);
                [U,S,V] = svd(toeplitz(tempAIF,R));

                % Q is defined earlier.
                W(Q) = 1./S(Q);

                %% This finds the diagonal entries which are less than some threshold
                % and then sets those to zero.
                Hi = max(S(Q));
                Ex = find(S(Q) < Hi*.2);
                W(Q(Ex)) = 0;

                %% Multiply everything out to get final deconvolution
                CBF_Ai(i,j,k) = max(V*W*(transpose(U)*...
                                squeeze(Conc(i,j,k,:))));
            end
        end
    end
end

warning on;
\end{verbatim}
MTT = cbv./CBF_Ais;
warning on;

% Remove bad values and scale MTT.
Q = find(isnan(MTT)); MTT(Q) = 0;
Q = find(MTT == Inf); MTT(Q) = 0;
mMTT = max(MTT(:));
MTT = 50 * MTT/mMTT;

% Normalize CBF by its median.
Q = find(CBF_Ai > 0);
mF = median(CBF_Ai(Q));
CBF = CBF_Ai./mF;

% Write Final Output
cd(outDir);
write_mgh_mcc('UCBV.bfloat',cbv);
write_mgh_mcc('CBF.bfloat',CBF);
write_mgh_mcc('MTT.bfloat',MTT);

display('Finished');

Support Functions
Score3
function score_cube = ...
    score3(conc_cube,criterion1,criterion2,criterion3,criterion4,TR);
    % Score3 computes the score for a given set of 4 criterions.
    % Specifically, it just normalizes then to be between 0 and 1.
    %
    % score_cube is the score for every voxel.
    % criterion1 is the first moment
    % criterion2 is the average slope
    % criterion3 is the highest peak
    % criterion4 is the FWHM
    % TR is the repetition time.

    warning off MATLAB:divideByZero;
    [ni,nj,nk,nsamp] = size(conc_cube);
    nQ = find(criterion1 < Inf);
    score_cube = ones(ni,nj,nk)*Inf;

    if not(isempty(nQ))

        % Now we normalize all of these.
        mc1 = min(criterion1(:));
        criterion1 = criterion1 - mc1;
        mc2 = min(criterion2(:));
        criterion2 = criterion2 - mc2;
        mc3 = min(criterion3(:));
        criterion3 = criterion3 - mc3;
        mc4 = min(criterion4(:));
        criterion4 = criterion4 - mc4;
        Mc1 = max(criterion1(nQ));
        if (Mc1 == 0)
            criterion1(nQ) = criterion1(nQ) + 1;
        else

68
criterion1 = criterion1/Mc1;

Mc2 = max(criterion2(nQ));
if (Mc2 == 0)
    criterion2(nQ) = criterion2(nQ) + 1;
else
    criterion2 = criterion2/Mc2;
end

Mc3 = max(criterion3(nQ));
if (Mc3 == 0)
    criterion3(nQ) = criterion3(nQ) + 1;
else
    criterion3 = criterion3/Mc3;
end

nQ = find(criterion4 < Inf);
if isempty(nQ)
    score_cube(nQ) = criterion1(nQ) - criterion2(nQ) - criterion3(nQ);
else
    Mc4 = max(criterion4(nQ));
    if (Mc4 == 0)
        criterion4(nQ) = criterion4(nQ) + 1;
    else
        criterion4 = criterion4/Mc4;
    end

    score_cube = ones(n1,nj,nk)*Inf;
    score_cube(nQ) = criterion1(nQ) - criterion2(nQ) - ... criterion3(nQ) + criterion4(nQ);
    Q = find(criterion4 > (15*1.5/TR));
    score_cube(Q) = Inf;
end
end

warning on MATLAB:divideByZero;

Gsmooth
function IMG = Gsmooth(IMG_in,N, sigma);

%Gsmooth Takes a 4-D MRI Dataset and Smooths it with an NbyN Gaussian kernel.
% [IMG] = Gsmooth(IMG_IN,N, SIGMA)
% Where IMG_IN is the 4-D Dataset, N is the dimension of the
% Gaussian kernel, and SIGMA its std dev.
% Cory Lorenz <clorenz@nmr.mgh.harvard.edu> June 2003

G = fspecial('gaussian',[N N], sigma);
[nr,nc,nsl,nsm] = size(IMG_in);
for i = 1:nsl,
    for j=1:nsm,
        IMG(:,:,i,j) = filter2(G,IMG_in(:,:,i,j));
    end
end

FWHM

function fwhmax = fwhm(data);
% FWHM computes the FWHM of a time series.
% fwhmax = fwhm(data);
% Where:
% data = the time series
% fwhmax is the FWHM value

mdata = max(data);
I = find(data > mdata/2);
L = length(I);

if (L == 0)
fwhmax = Inf;
elseif (I(L) == length(data) | I(1) == 1)
fwhmax = Inf;
else
width = I(L)-I(1);
b0 = data(I(1));
a0 = data(I(1)-1);
x0 = (mdata/2-a0)*(1/(b0-a0));

b1 = data(I(L));
a1 = data(I(L)+1);
x1 = (mdata/2-b1)*(1/(a1-b1));

fwhmax = width + x1-x0+1;
end