Sequential slice object labeling in tomographic data via trajectory estimation

Brandon Thomas Mikulis

Follow this and additional works at: https://scholarworks.rit.edu/theses

Recommended Citation

This Thesis is brought to you for free and open access by RIT Scholar Works. It has been accepted for inclusion in Theses by an authorized administrator of RIT Scholar Works. For more information, please contact ritscholarworks@rit.edu.
SEQUENTIAL SLICE OBJECT LABELING IN TOMOGRAPHIC DATA VIA TRAJECTORY ESTIMATION

by

Brandon Thomas Mikulis

A Thesis Submitted
in
Partial Fulfillment
of the
Requirements for the Degree of
MASTER OF SCIENCE
in
Electrical Engineering

Approved by:

PROF. Raghuveer Rao

PROF. S. A. Dianat

PROF. Vincent J. Amuso

PROF. Robert J. Bowman

DEPARTMENT OF ELECTRICAL ENGINEERING
COLLEGE OF ENGINEERING
ROCHESTER INSTITUTE OF TECHNOLOGY
ROCHESTER, NEW YORK
June, 2006
Thesis/Dissertation Author Permission Statement

Title of thesis or dissertation: Sequential Slice Object Labeling in Tomographic Data via Trajectory Estimation

Name of author: Brandon Mikulis
Degree: Master's of Science
Program: Electrical Engineering
College: College of Engineering

I understand that I must submit a print copy of my thesis or dissertation to the RIT Archives, per current RIT guidelines for the completion of my degree. I hereby grant to the Rochester Institute of Technology and its agents the non-exclusive license to archive and make accessible my thesis or dissertation in whole or in part in all forms of media in perpetuity. I retain all other ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

Print Reproduction Permission Granted:

I, Brandon Mikulis, hereby grant permission to the Rochester Institute of Technology to reproduce my print thesis or dissertation in whole or in part. Any reproduction will not be for commercial use or profit.

Signature of Author: Brandon Mikulis Date: 1/3/06

Print Reproduction Permission Denied:

I, ____________________________, hereby deny permission to the RIT Library of the Rochester Institute of Technology to reproduce my print thesis or dissertation in whole or in part.

Signature of Author: ____________________________ Date: __________
Sequential Slice Object Labeling in Tomographic Data via Trajectory Estimation

by

Brandon Thomas Mikulis

Submitted to the Electrical Engineering Department
College of Engineering on June 20, 2006
in Partial Fulfillment of the Requirements of the Degree of
Masters of Science in Electrical Engineering
at the Rochester Institute of Technology

ABSTRACT

The increasing usage of volumetric imaging modalities, such as magnetic resonance imaging (MRI) and computed tomography (CT) in areas such as medicine and nondestructive evaluation (NDE) has placed a great importance on 3D visualization techniques. This growth of volume data in the form of cross sections has created the need for object labeling in volume data sets for 3D visualization. A sequential, slice-to-slice approach is proposed that is less computational and memory intensive than 3D connected-component labeling while achieving better results than the 2D overlap sequential processing approach. Labeling occurs while tracking each object through the 3D volume via updated trajectory approximation.

This thesis is motivated by the desire to develop a labeling technique that captures key aspects of the human visual approach to the task. The proposed approach, while labeling 3D data, also provides a computationally efficient method by sequential processing of 2D slices instead of the whole 3D volume at once. Additionally, the
proposed trajectory tracking approach performs correctly in many cases where current 2D sequential labeling techniques fail.

Trajectory tracking for labeling is a new approach, representing the 3D objects as curves and performing 3D curve tracing to label the approximate trajectories of the objects. The labeled trajectories are then mapped back to the 3D objects to complete the labeling process. Development of the proposed labeling approach is discussed while multiple examples are presented. These examples are used to illustrate that the proposed approach performs correctly where the current overlap approach fails; examples are also used to show that the behavior of the proposed approach parallels that of the typical human approach to object tracking.

Thesis Supervisor: Raghuveer Rao, Ph.D.

Professor of Electrical Engineering, RIT
Acknowledgements

This thesis would not have been possible without the support and help of so many people. It has been a privilege to have Dr. Raghuveer Rao as my thesis advisor. I am deeply appreciative of his support, guidance, and most importantly his patience. Dr Rao has had a strong influence on my direction and work from the first time I took a graduate course with him. He has always motivated me by posing problems that sparked my interest and pushed my abilities. Many thanks are owed to Dr. Rao for the time spent reading and editing the drafts for my work and having the patience to work with me even after I moved to Boston and started to work full time.

Dr Sohail Dianat and Dr Vincent Amuso who in addition to the help they have provided with this work as well as my development as during my whole undergraduate and graduate experience at the Rochester Institute of Technology.

Thanks to Dr Daniel Phillips for opening my eyes to the world of possibilities for an electrical engineer in medicine. Additionally for many good discussions during the course of my studies that have been both insightful and educational.

Matt Holmes, the best partner for problem sets, tests, and labs. Your dedication has served as a great motivator and inspiration. As well as your ability to recognize the breaking point of fellow students and apply an infinite knowledge of methods to maintain sanity.

Heather Ganley, thank you for being my loudest supporter. Your encouragement, unfailing confidence, and gentle pushing have been a force as well as a gentle hand.
Gavin Page someone who never grew tired of talking about my thesis or his, a multitude of ideas where shared making both of our results stronger. Steve Miketa and Leslie Bull thank you for being all too eager to edit revisions of my work.

I am incredibly indebted to my support network in Boston, Dr. Deborah Burstein and Dr. Martha Gray, always interested in my progress, well being, and abilities to coordinate social activities. To Dr Jesse Little, Michelle Farley, Ashley Williams, Joe Samosky, Nitya Krishnan, and Benjamin Diop-Frimpong for many entertaining and enlightening conversations in the halls of HIM.

Will Hedglong, Johanna Varner, Chris Ward, Julie Park, Marin Kress, Liz Maynard, Malyka Ianni, Ashley Provencher for enriching the cities of Boston and Rochester.

Dale Cotton an unspoken depth of gratitude, your hard work, dedication, and unwavering ideals stand as a beacon of inspiration. Thank you for all your encouragement and friendship.

Finally, Dad, Mom, and Heather, thank you. Your love, support and belief means the world to me. You have been my true champions.

Brandon T Mikulis
June 16, 2006
Boston, MA
# Table of Contents

1 **INTRODUCTION**  
1.1 Importance of Labeling  
1.2 Motivation: Computer Vision  
1.3 Motivation: Applications  
1.4 Objectives and Specific Questions Answered in this Thesis  

2 **BACKGROUND**  
2.1 MRI  
2.2 CT  
2.3 Back Projection  
2.4 Segmentation  
2.4.1 Edge Detection  
2.4.2 Thresholding  
2.4.3 Morphological Operators  
2.4.4 Watershed Transform  
2.5 Labeling  
2.6 2D Curve Tracing  
2.7 3D Rendering  

3 **THE LABELING ALGORITHM**  
3.1 Existing Approach  
3.2 Proposed Approach  
3.3 Defining Trajectory Points  
3.4 Labeling Trajectory Points  
3.5 Re-labeling  

4 **METHODS**  
4.1 Data Set  
4.2 Segmentation  
4.3 3-D Visualization  
4.4 Interpolation of Cross Sectional Data  

5 **RESULTS**  
5.1 Foam Data Set  
5.2 Calculated Data Set  
5.3 Quadruple Helix Labeling  
5.4 Real CT Data: The Pasta Patient
Table of Figures

**Figure 2.1** Precession About Dipole In Line with an External Magnetic Field .......................... 18
**Figure 2.2** Longitudinal Relaxation with T1 and Transversal Relaxation with T2 ............... 19
**Figure 2.3** Projection at angle [13] ....................................................................................... 23
**Figure 2.4** Multiple Projections [13] ....................................................................................... 23
**Figure 2.5** Frequency Domain constructed from multiple projections .................................... 24
**Figure 2.6** Reconstruction from one, 4, 64, and 512 projections (top left clockwise). [13] .................................................................................................................. 25

**Figure 2.7** Bimodal Histogram Distribution, Arrow denotes local minimum for single a threshold ................................................................. 28

**Figure 2.8** Binary Mask and Resulting Gray Scale Image after Application of Distance Transform ........................................................................................................ 31
**Figure 2.9** Topographical Model from Gray Scale in Figure 9 .................................................. 31
**Figure 2.10** Mislabeled Crossing Objects ................................................................................. 34
**Figure 2.11** 2D Curve Tracing ................................................................................................. 35
**Figure 3.1** Single Frame from a Random-Dot Kineomatogram .............................................. 38
**Figure 3.2** Center of Mass Points ............................................................................................ 41
**Figure 3.3** Trajectories of the Objects After Labeling using the Existing Approach .......... 45
**Figure 3.4** Trajectories of the Objects After Labeling using the Proposed Approach .......... 45
**Figure 3.5** Multiple Instances of an Object in one Cross Section ........................................... 46
**Figure 3.6** Correctly Labeled Object using Trajectory Tracking with Object Extension 46
**Figure 4.1** Scanned Slice 8 From Data Set #2 ........................................................................ 49
**Figure 4.2** Gray Scale Slice 8 From Data Set #2 ...................................................................... 49
**Figure 4.3** Histogram for the gray scale image in figure 12, threshold of 100 selected .......... 49
**Figure 4.4** Results of Thresholding with a threshold of 100 ................................................... 49
**Figure 4.5** Disk Structuring Element ...................................................................................... 51
**Figure 4.6** Result of Morphological Filtering .......................................................................... 51
**Figure 4.7** Segmented Slice #8 Data Set #2 .......................................................................... 51
**Figure 5.1** Foam Data Set ....................................................................................................... 56
**Figure 5.2** Foam Data Set with Slice Interpolation by four ...................................................... 56
**Figure 5.3** Synthetic Data Set .................................................................................................. 57
**Figure 5.4** Object Centroids From Synthetic Data Set ............................................................. 57
**Figure 5.5** Correct Labeling using Trajectory Tracking ......................................................... 57
**Figure 5.6** Labeling using the Overlap Approach .................................................................... 57
**Figure 5.7** Mislabeled Object from Overlap Approach ........................................................... 58
**Figure 5.8** Quadruple Helix Original Object ............................................................................ 59
**Figure 5.9** 10 Cross Sectional Slices Covering half the height .............................................. 60
**Figure 5.10** Centers of Mass Marked with Object Origination ................................................ 60
**Figure 5.11** Centers of Mass After Labeling ............................................................................ 61
**Figure 5.12** Objects After Labeling ......................................................................................... 61
**Figure 5.13** 10 Cross Sectional Slices Covering full the height ............................................... 62
**Figure 5.14** Centers of Mass Marked with Object Origination ................................................ 62
**Figure 5.15** Centers of Mass After Labeling ............................................................................ 62
**Figure 5.16** Objects After Labeling ......................................................................................... 62
**Figure 5.17** 3D Rendering of Spaghetti CT Data Set ................................................................. 63
Figure 5.18 Results of Labeling ................................................................. 64
Figure 5.19 Failure of Labeling ................................................................. 65
Figure 5.20 Trajectory Points at Failure ...................................................... 65
1 Introduction

1.1 Importance of Labeling

The advancement of tomographic imaging techniques such as MRI and CT has increased the complexity and quantity of data acquired [1, 2]. These tomographic imaging modalities capture data as cross sectional slices of the target object, which offer views different than more traditional modalities such as conventional X-ray. The advantage of these tomographic techniques is that volumes can be built by “stacking” the cross sections on top of each other. Traditional image processing techniques can be used to remove or suppress noise from the data or to segment objects of interest. Labeling is important at this step to in order to identify the same object in each slice for complex 3D renderings.

Labeling an object is completed by applying the same label or value to each object pixel in the volume that is part of that object, by using a unique label for each object a labeled data set is produced. Different labeling techniques currently exist that utilize different properties of the data in attempt to correctly label the objects. Techniques use the fact that there should be only small changes from one cross section to the next, based on this labeling is performed by using common information between cross sections. This can lead to error due to the distance between cross sectional slices and the assumptions made about the the interactions between objects. Other techniques operate by performing labeling on the whole 3D data set at one time. These techniques becomes memory intensive as the whole 3D data set needs to be loaded into memory at once, also the computations increase as calculations are being tracked in 3D dimensions. A new
technique needs to be developed that minimized computational and memory intensity while being less susceptible to the effects of large inter-cross sectional distances.

1.2 Motivation: Computer Vision

Computer Vision is an artificial intelligence (AI) technology that captures and processes visual information much in the way that normal human vision functions. Our daily interaction with the world attests to the power of human cognition and perception. The human visual pathway, an integral part of this cognition and perception enables us to drive a car, inspect mechanical parts for physical defects, and diagnose medical images such as Magnetic Resonance Imaging (MRI) scans and Computed Tomography (CT) scans. Replicating aspects of the human approach to different visual tasks can simplify and automate many complex daily tasks. Many times these tasks can be computationally intensive, thus limiting their applicability in real time situations.

Currently there are AI systems in development or in use that operate a motor vehicle, perform non-destructive evaluation in industrial settings, and diagnose medical images. These systems are improved by the implementation of fast algorithms that produce the same results as if a human were performing the task via replication of the human visual process. Novel approaches require further investigation to improve both computational complexity as well as to produce results that more closely emulate human perception.
1.3 Motivation: Applications

Approaches proposed in this thesis provide an original approach to a common task in volume visualization, object labeling. Volumetric imaging modalities have seen an increase in usage, such as magnetic resonance imaging (MRI) and computed tomography (CT) in medicine and industrial use of nondestructive evaluation (NDE) has placed a greater importance on 3D visualization techniques. These 3D visualization techniques have evolved to utilize computer vision systems that mimic human visual systems [3-5]. Techniques in image segmentation, volume rendering, and labeling all comprise technologies used in computer vision for object visualization.

Originally developed for medical usage, CT scans are being used for nondestructive evaluation outside of the hospital [6, 7]. A 3D model of the target object is created from a CT scan that provides an internal view without making physical changes to the object. X-ray projections from different angles produce slices of a finite thickness that represent the object’s radio-opacity within the volume enclosed by the slice, stacking the slices results in the 3D model of the target object. The ability to see inside of an object without cutting it open allows for the inspection of internal components in place as well as the interior of containers. Visualization of objects from the scans relies on computer vision to create easily manipulated 3D objects and to determine individual components.

While in medicine, computer vision research is underway within diagnostic imaging and surgery. Computer enhances visualizations from CT or MRI scans, presenting the data in a more accessible format for the radiologist and surgeon, this visualization reveals the internal structure of obstructing objects, such as viewing bone
structure through muscle. The research in using computer vision to produce accurate and useful visualizations also aims to produce minimally invasive surgical techniques. For example, inclusion of these technologies could eliminate arthroscopic surgery to determine integrity of cartilage prior to total knee replacement. With real time 3D visualization available, more invasive surgeries could become less invasive. The Surgical Planning Laboratory at Brigham and Women’s Hospital in Boston, MA researches all of these areas. They have developed an intraoperative MR unit within a functional operating room that allows for MR guided procedures as well as increasing visualization and control while minimizing the invasiveness of certain procedures [8, 9].

Non-Destructive Evaluation

In an industrial setting often one desires to evaluate a target object that is within another container or object. For instance one could avoid a long downtime for a generator by utilizing CT to check the blades on a turbine without disassembly; the use of CT enables the turbine and housing to be imaged and visualized [10]. As mentioned, the blades on the turbine are the primary concern of such an investigation, so not only does the scan need to be rendered into a 3D visualization, but each object needs to be individually viewed. Visualization in 3D enables looking at the objects from any point of view, but given that the turbine could not be properly evaluated using just 2D projections, then within the visualization there will be features that block the turbine such as the housing. Labeling the objects prior to creating the 3D visualization allows for the optional exclusion of objects in the visualization, such as the housing. Thus the 3D
visualization could be the entire generator, or the whole turbine, or even just an individual blade.

Medical Visualization

In the medical field research is underway to utilize 3D visualization for both diagnosis and surgical procedures. Both of these situations propose a similar problem as non-destructive evaluation: structures that are not of interest are included in the scan and need to be removed from the visualization. Additionally it might be desired to view the structure of a single object such as the tibial plateau from a MRI scan of the knee. Once again all the pixels must be assigned to the correct corresponding object so only the desired object’s pixels are used in the visualization. Other research uses diffusion tensor MRI in order to visualize synaptic connections within the brain based on labeling of these connections in 3D volumes [11].

The process to create these desired interactive 3D visualizations starts with segmentation of the objects in each slice; which identifies the individual objects within the slice. Once segmentation is performed on each slice, the objects must be labeled so that from slice to slice objects that are part of the same global object have the same label in each segmented slice. This to say if a CT or MRI scan contained two distinct objects, object ‘A’ and object ‘B’, then in each slice the segmented objects that were part of object ‘A’ would be labeled to designate this. Similarly, objects belonging to object ‘B’ would be labeled to designate this. After the objects are segmented and labeled, a 3D visualization of the individual objects of interest from the MRI or CT scan can be created.
Since the individual objects are labeled, the 3D visualization provides greater flexibility and be customizable, instead of making a volume or surface visualization of the whole scan, an individual object or multiple objects can be viewed. Visualization software exists that enables the scene to be rotated 360 degrees around multiple axes providing a complete view of the objects of interest. Utilizing properly labeled objects, a whole scene could be visualized and the user could select objects of interest to provide unobstructed views from any direction.

Different problems exist in the process of creating visualizations, and one issue is that while it can be faster than manual segmentation, it is very difficult to achieve reliable results using fully automated segmentation. Conversely, hand segmentation provides the ability to acquire a properly segmented image, but it can be very time consuming. The next issue is with labeling; MRI and CT data is collected in slices and global volume labels need to be applied to the objects in each slice. There are multiple algorithms for labeling, and each one addresses a different problem resulting from the interaction of objects between slices. Each approach has its disadvantage, from being computationally and memory intensive, to being less memory intense but with higher error rates. The last problem occurs in the final step of creating the actual visualization, dealing with the loss of information between slices. In order to create a surface visualization, the surface between slices needs to be interpolated in a manner that creates a realistic representation of the object. If the cross sectional slices are close enough, this is less of a problem as there is sufficient data to produce a volume, but in sparse cross sectional data the volume is less smooth and visually choppy. There are two different approaches for dealing with
this missing information; one involves interpolating additional slices while the other involves creating a surface using polygons from the available slices.

1.4 Objectives and Specific Questions Answered in this Thesis

The issue of development of a rapid, computationally efficient object labeling algorithm applied before 3D visualization is addressed in this thesis. Focusing in particular on the situation of labeling 3-D objects in cross-sectional slices such as those obtained in CT. The novel labeling approach presented represents objects with curves in 3D and then performs labeling by applying 3D curve tracing techniques and mapping the labels back to the 3D objects. Thus the approach mimics aspects of the human visual approach to the task.

The proposed algorithm was implemented and multiple examples are presented to illustrate and evaluate performance. A data set is labeled using a current algorithm resulting in object labels that differ from how a human would label them. The data set is then re-labeled using the proposed algorithm to produce results that more closely resemble human labeled data. The proposed algorithm addresses mislabeling that occurs from sequential slice labeling of 3D volumetric data, containing two or more real continuous objects that cross each other or pass relatively close to each other. Due to the relative closeness of the objects, the current algorithm mislabels them at the vertex of their ‘intersection.’ After labeling with the proposed algorithm, 3D models are created allowing a view of the objects and verification of the labeling process.
2 Background

Pertinent background information to the development and implementation of the proposed approach is disclosed within this chapter. Starting with cross sectional data acquisition and formation from CT and MRI modalities in 2.1, 2.2, and 2.3. Section 2.4 reviews traditional techniques of object segmentation for both volume and edge segmentation. Object labeling is discussed in section 2.5 detailing both volumetric labeling and sequential slice labeling. The 2D curve tracing algorithm developed by Raghupathy and Parks is explained in section 2.6 as it pertains to the proposed labeling algorithm. Finally, 3D volume rendering approaches are discussed in section 2.7.

2.1 MRI

Magnetic resonance imaging produces visualizations of the magnetic properties of the sample being imaged. The two most commonly measured properties are longitudinal relaxation and transversal relaxation. There are two ways for the magnetic dipoles to align in the magnetic field, one way aligns the dipole moment with the direction of the external field as shown in figure 2.1, or in the opposite direction. The two alignments have different energies; the lower energy state, with the dipole aligned in the direction of the magnetic field and the high energy state, when the dipole aligns in the opposite direction. Most of the magnetic dipole moments are canceled out by opposing ones aligned in the opposite direction, except for the few extra aligned with the external magnetic field. A net moment is created by the individual moments that are not negated, this net moment points in the same direction of the external field.
Figure 2.1 Precession About Dipole In Line with an External Magnetic Field

The atoms precess around the magnetic dipole as shown in figure 2.1. The frequency of precession is given by:

\[ w_0 = \gamma B_0 \]

\( B_0 \) is the field strength in Tesla, \( w_0 \) is the precession frequency in Hertz, known as the Lamor frequency, and \( \gamma \) is the gyromagnetic ratio. The gyromagnetic ratio is a constant specific to each type of nucleus; Hydrogen, the most common atom in used in medical MRI has a gyromagnetic ratio of 42.6 MHz/Tesla.

The net magnetic force in the \( z \) direction, known as longitudinal magnetization \( (M_z) \), cannot be measured, as it is in the same direction of the external magnetic field and cannot be isolated. In order to measure the magnetic field produced by the nuclei, the force of the external field must be excluded. The best way to measure this magnetization would be if it were transverse to external magnetic field, containing no component of the external field, known as transversal magnetization. Energy is transferred to the precessing nuclei with a radio frequency (RF) pulse. The frequency of the RF pulse must be the
same as the Lamor frequency of the nuclei for the energy transfer to occur, this phenomenon is known as resonance.

As mentioned earlier the two most commonly measured properties are longitudinal relaxation and transversal relaxation. Longitudinal relaxation is the curve produced by plotting longitudinal magnetization as it increases back to its original state after application of the RF pulse. The time constant is the value used to describe the exponential decay; this time constant for longitudinal relaxation is called $T_1$, defined as the amount of time it takes the longitudinal magnetization to reach 63% of its original value. $T_2$ is the time constant for the decay for transversal magnetization after application of the RF pulse. It is defined as the time it takes for the transversal magnetization to decrease to 37% of its original value. Figure 2.2 shows longitudinal and transversal relaxation with their corresponding time constants. Longitudinal relaxation and transversal relaxation are independent from each other. Since the system uses resonant frequencies, the antenna used to produce the RF pulse is also able to capture the signal created by the relaxation of transversal and longitudinal relaxation.

![Figure 2.2 Longitudinal Relaxation with T1 and Transversal Relaxation with T2](Adapted from [12].)
2.2 CT

CT operates by measuring the radio-opacity of the sample by the attenuation of X-rays. X-Ray CT projections are formed by transmitting a beam of x-ray photons through the volume to be imaged and measuring the amount of photons that are transmitted through it. Attenuation of X-rays through the volume occurs due to the photoelectric and Compton effects. The photoelectric effect occurs when all of a photon's energy is transferred to an electron within an atom, freeing the electron, and the Compton effect occurs when only part of the energy is transferred to an electron resulting in the photon losing some energy and traveling off in a different direction.

In a homogeneous volume, the projection is proportional to the distance the X-rays traveled through the sample. This proportion is the coefficient of attenuation. If the number of photons projected at a volume is \( N \) and the change in photons as a result of attenuation is \( \Delta N \) then the attenuation coefficient \( \mu \) is defined by:

\[
\frac{\Delta N}{N} \cdot \frac{1}{\Delta x} = -\mu
\]

where \( \Delta x \) is the distance over which the attenuation occurred, in this case the length of the volume through which the photons were projected. Rewriting equation 2 as a differential equation by taking the limit as the distance goes to zero:

\[
\frac{1}{N} dN = -\mu dx
\]
Solving the differential equation yields:

\[ \int_{N_0}^{N} \frac{1}{N} dN = -\mu \int_{0}^{x} dx \Rightarrow \ln N - \ln N_0 = -\mu x \quad \text{eq 4} \]

where \( N_0 \) is the number of projected photons into the object and \( N \) is the number after distance \( x \) in the volume. The problem with this result is that it assumes a homogenous volume which is not true in real life situations. In a real volume of interest, the attenuation coefficient would change as a function of position in the volume, so equation 4 is rewritten as:

\[ \ln N - \ln N_0 = -\int_{0}^{x} \mu(x) dx \Rightarrow \ln \frac{N}{N_0} = -\int_{0}^{x} \mu(x) dx \quad \text{eq 5} \]

Once again as in MRI, back projection is needed to determine the spatial function that produced these projections.

### 2.3 Back Projection

As discussed earlier, applying a gradient magnetic field to the static magnetic field in MRI produces line integrals across the slice. The same was discussed for CT imaging; the data collected is the line integral resulting in the transmission of x-ray photons through the object of interest. These line integrals are known as projections, and unfortunately these projections contain little data, much like shining a light on one’s hand can project a silhouette that resembles the head of a rabbit, there is data lacking as hands are not normally confused for rabbits. If projections were taken from multiple
angles, the object could be reconstructed since the occluded parts in one projection are revealed in another.

Given a projection, \( P_\theta(t) \) at any angle \( \theta \), where \( t \) is the axis perpendicular to the lines over which the integral is performed. Defining the x-y plane, \( \theta \), and \( t \) through the following equation:

\[
x \cos \theta + y \sin \theta = t
\]

Thus \( P_\theta(t) \) can be defined in terms of the object \( f(x,y) \) by use of the Radon transform:

\[
P_\theta(t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x,y) \delta(x \cos \theta + y \sin \theta - t) \, dx \, dy
\]

Figure 2.3 shows the projection \( P_\theta(t) \) of the object \( f(x,y) \) at the angle \( \theta \), with the line \( x \cos \theta + y \sin \theta = t \) producing the projection, \( P_\theta(t) \).

Image reconstruction is performed using the Fourier Slice Theorem. The two dimensional Fourier transform of the object \( f(x,y) \) is written as:

\[
F(u,v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x,y) e^{-j2\pi(ux+vy)} \, dx \, dy
\]

The projection at angle \( \theta \), being only one in dimension has a Fourier transform of:

\[
S_\theta(u) = \int_{-\infty}^{\infty} P_\theta(t) e^{-j2\pi ut} \, dt
\]
The Fourier Slice Theorem states that the Fourier transform of the projection is equal to a slice of the two dimensional Fourier transform of the original object at the same angle. So given multiple projections at different angles as shown in figure 2.4, the frequency domain can be approximated as in figure 2.5.

Figure 2.3 Projection at angle [13]  
Figure 2.4 Multiple Projections [13]
Using enough projections at different angles, the object’s two dimensional Fourier transform is approximated by interpolating of the missing data in the frequency domain, and then applying the inverse Fourier transform to the approximation results in the reconstructed object. The Fourier Slice Theorem is proven in [13] which also discusses faster and computationally more efficient methods of forming an image from the projections. The simplification of the method consists of taking the projections and then “smearing” them across the image at their angle of projection, building up the final image with more projections. This process is illustrated in figure 2.6.
2.4 Segmentation

Image segmentation is an important part of image processing applications in all fields, as it enables the identification of regions of interest or objects of interest. Segmentation separates foreground objects from background objects as well as foreground objects from one another. The importance of segmentation has led to many different techniques; most approaches result in a binary mask that identifies the objects from the background. One approach for 3D data sets is to use thresholding followed by the morphological operations of erosion and dilation to extract the objects [14-16]. The thresholding sets up the basic object boundaries while morphological operations help disconnect objects that could be bridged by a few pixels due to similarities in pixel intensities near boundaries between two objects. This 3D segmentation is followed by a 3D labeling technique such as connected-component labeling. Being based on 3D volume operations, this approach is both computationally intensive and memory intensive. This

Figure 2.6 Reconstruction from one, 4, 64, and 512 projections (top left clockwise). [13]
thesis’ concern of producing a computationally efficient labeling algorithm is met by utilizing data slices that have been segmented in 2D, thus eliminating the need for the whole 3D volume to be loaded at any one time. The 3D binarization technique of thresholding and morphological operators can be applied to 2D data, additionally connected-component labeling can be applied to 2D data. Common binarization techniques in 2D usually fall within one of two main categories; edge detection or volume detection. Edge detection works by trying to determine the edges of a region or object. With edges identified, the enclosed volume is the region of object of interest. Volume detection determines the pixels in the region of interest through techniques like thresholding or region growing.

2.4.1 Edge Detection

The simplest form of edge detection is using a gradient operator. When used with a gray scale image, the gradient operator provides a mapping of the changes of the relative gray scale levels. The gradient in the x direction and y direction are taken separately with the resulting images combined to create an edge map. While this technique is fast and simple to implement the results are not always reliable. The gradient operator is very susceptible to noise and often results in discontinuous edges. Another problem with edge detection is that enclosed areas produced using this technique are foreground objects separated from the background, and additional steps are required to produce an object mask that identifies the pixels of each individual object for automated labeling rather than object outlines.
2.4.2 Thresholding

Thresholding and region growing techniques in addition to being important image processing techniques are also the basis of many popular 2D and 3D volume detection techniques, as discussed earlier. Simple thresholding involves the use of a single threshold; all points with a gray scale value greater than the threshold is an object point, otherwise it is a background point. The threshold value picked when using a single threshold is usually a local minimum in the histogram, especially for a bimodal distribution such as the one in figure 2.7. Multilevel thresholding uses more than one threshold to create more than two different object classes. In addition to single level or multilevel implementation of the thresholding, there are the options of applying thresholds globally or adaptively. Global thresholding applies the same threshold to the whole image based on estimates for the whole image, while adaptive thresholding breaks the image first into blocks and then uses different thresholds for each block as estimated on a block by block basis. Results from thresholding are label masks that classify volumes into different object classes, or if only a single threshold is used, a binary mask, with labeling occurring through the use of a separate algorithm.
2.4.3 Morphological Operators

In image processing, morphological operators are used in many different ways: pre or post processing, boundary extraction, and noise removal. Morphological operators use mathematical set theory to perform the desired operations and in the scope of this thesis, are applied during the segmentation phase of the image. The two main building blocks of morphological processing are dilation and erosion [17]. Dilation is defined as

\[ A \oplus B = \left\{ z \left| (\hat{B})_z \cap A \subseteq A \right. \right\} \tag{10} \]

The dilation of A by B is the set of all points, z so that the intersection of A with the reflection of B translated by z is a subset of A, where A is the set of points that define the image and B is a structuring element. Thus if the reflected and translated structuring element overlaps an object by one or more pixels, that point is added to the set. The result
of dilation is that object's boundaries are enlarged or the object becomes dilated. Dilation is used to close gaps; the size and shape of the structuring element determines the width and orientation of the gaps that are closed by the dilation.

The second morphological operator building block is erosion, this erodes the objects in the set. Erosion is defined as

\[ A \ominus B = \{ z \mid (z + \hat{B}) \subseteq A \} \quad \text{eq 11} \]

The erosion of A by B is the set of all points, z so that the reflection of B translated by z is a subset of A. Once again, A is the set of points that define the image, and B is the structuring element. The reflected and translated structuring element must lay entirely within the set A for that point to be included in the set. This produces an erosion of object boundaries resulting in smaller objects and larger openings.

Combinations of dilation and erosion can also be used to produce many different results. Opening and closing are two useful, common morphological operations that are built from combinations of erosion and dilation. Opening defined as:

\[ A \circ B = (A \ominus B) \oplus B \quad \text{eq 12} \]

This is the erosion of A by B followed by the dilation by B. Opening eliminates thin parts of an object and enlarges small cracks and holes. The opposite is closing which is defined as:
\[ A \ast B = (A \oplus B) \odot B \]  

the dilation of A by B followed by the erosion by B. Closing fills small holes and cracks while thickening narrow bridges.

Morphological operators can be applied to binary or grayscale images, but in the context of this thesis they are applied to binary segmentation masks. Segmentation techniques do not always produce perfect or ideal segmentation, so the use of morphological operators can help with cleaning up segmentation masks. Morphological operators applied to a binary segmentation mask can remove holes in objects and small groups of background pixels that are incorrectly classified. Additionally, object contours can be smoothed while small bridges between different objects are dissolved with the use of morphological processing.

### 2.4.4 Watershed Transform

The watershed transform performs segmentation by using the gray scale data as topographical information [18, 19], but can also be used on binary masks to further separate out foreground objects. In the case of a binary mask, a distance transform is applied to the mask to create a gray scale image as shown in figure 2.8. Using a gray scale image or a binary image converted to gray scale, objects are created by “flooding” the topographical model such as the one in figure 2.9 from the gray scale image in figure 2.8. The local minima, being the lowest points fill up first. As the level increases, regions fill up and the point at which two regions overflow into each other a “dam” is built. When the whole image is “flooded,” the “dams” enclose the segmented volumes. This method in segmentation leads to more complete segmentation with continuous boundaries. The
ability to handle binary masks means that an image that used a single threshold level to segment foreground objects from the background can then use the watershed to segment out the individual foreground objects, especially if there is occlusion or objects touching.

Figure 2.8 Binary Mask and Resulting Gray Scale Image after Application of Distance Transform

Figure 2.9 Topographical Model from Gray Scale in Figure 9
2.5 Labeling

Labeling in 3D data sets can occur in two methods, first the whole 3D data set is operated on segmentation and labeling can occur in 3D at once. This is both computationally intensive and requires a significant amount of memory. The second approach is the one used within this thesis, consisting of segmenting and labeling each individual slice and then carrying common 2D labels throughout the 3D volume [20, 21]. This approach while being less computationally intensive, allows for the consideration of the objects’ interactions during labeling.

In each slice, there are multiple foreground objects all segmented from each other. A binary mask has been created to illuminate the foreground objects, but greater control and information is obtained by having access to individual objects, achieved by producing a label mask. Connected-component labeling can be performed in 2D as well as the 3D application discussed earlier. The binary mask objects are labeled based on connectivity or pixel neighbors. All pixels that are neighbors are part of the same object and are labeled accordingly. The most basic neighborhood for pixel connectivity are the four neighbors of pixel $p$. For pixel $(x, y)$, the four neighbors are given by the following coordinates:

$$\{(x+1,y),(x-1,y),(x,y+1),(x,y-1)\}$$  eq 14

Another common connectivity neighborhood is the 8-neighbors, which includes the pixels of the four neighbors as well as:
Each group of pixels determined to be connected is assigned a unique label producing a label mask for that particular slice.

Once all the objects in each of the slices are labeled, the objects need to be labeled throughout the 3D data set. This way, any given object will have the same label applied to it throughout the slices. Labeling three-dimensional objects from cross sections overcomes the computational and memory constraints created by volume operations such as 3D connected-component labeling, but is not without problems. One method for object labeling in cross sections is using information between two consecutive slices for labeling. The objects are segmented and the labeling is performed using the logical AND operator on the binary segmentation masks. If an object in the first slice overlaps an object in the second slice, then the objects in the two separate slices are considered to be cross sections of the same object. If two objects in slice 1 overlap the same object in slice 2 then the number of overlapped pixels is used as the tie breaker [22, 23]. This can result in errors if two objects are close to or cross each other and the data are captured on sparse cross sections.

This approach of using overlap is based on the assumption that there is only a small change in the object cross-sections from one slice to the next. The use of overlap is limited by the distance between cross-sections; if they are relatively far apart, the cross sectional data become sparse and the likelihood of error in the overlap approach increases. As shown by the example in figure 2.10, with sparse cross-sectional data, the objects can be incorrectly labeled by the overlap approach.


2.6 2D Curve Tracing

Raghupathy and Parks address the problem of extracting and linking curve points in an image. In this algorithm curve points are classified by using the property that a curve point has a vanishing first derivative and the amplitude of the second derivative is highest in the direction perpendicular to the curve [24]. Curve points are linked by comparing neighboring curve points with similar orientations. Incorrect linking at junctions is corrected by using the curve orientation to determine the trajectory and look for a suitable match after the junction. The results of this algorithm are illustrated in figure 2.11, the first picture is the original image, the second picture is of the labeled curves linked using subpixel distance and orientation linking, and the final picture is the labeled curves using trajectory projection. From these results it is seen that using object trajectories in tracking help produce results that are similar to those of humans.
Figure 2.11a Image of curves to be traced [24]

Figure 2.11b Curve tracing using curve pixel orientation and subpixel distance

Figure 2.11c Curve tracing using trajectory projection with orientation and subpixel distance

2.7 3D Rendering

There are many approaches for creating volume visualizations from cross sections. One approach is to use numerous thin cross sections that are very close together; in this ideal situation, the cross sections can be “stacked” on top of one another creating a volume that is representative of the object of interest. Data within these ideal requirements is not always available. When cross sections are sparser, one approach would be to interpolate new cross-sections in-between the existing cross sections to provide more data and smoother visualizations. Another approach is to use a polygon [25, 26], such as triangles, to connect points on cross sectional contours. These polygons estimate the surface between cross sections, the more points per contour used the smoother the visualization.
3 The Labeling Algorithm

Given the shortcomings of traditional labeling approaches discussed in 3.1, a new algorithm is developed in 3.2 that reduces the volume data to three dimensional curve points representing the object trajectories. Curve tracing is performed in 3D to label each trajectory, and then object labeling occurs by projecting the trajectories back onto the segmented objects.

3.1 Existing Approach

Labeling of cross sectional volume data using sequential slice processing approaches are more simple than 3D volume labeling techniques. Implementation is made easier as processing is performed on a limited number of slices at one time, where 3D volume labeling requires the whole volume to be loaded into memory at the time of processing. By processing only a few slices at a time, computational complexity is decreased and the memory requirements are also decreased. The proposed algorithm, while using sequential slice processing, overcomes errors seen in current sequential slice labeling approaches.

The existing sequential slice volume labeling approach in [22, 23], uses object projections from a labeled mask to label the objects in an unlabeled mask. Each object is labeled by projecting the labeled mask onto the unlabeled mask using the logical AND operator since the masks are binary the result is the overlap between objects. If an object in the unlabeled mask overlaps a labeled object from the labeled mask, the unlabeled object is labeled with the same label as the object in the labeled mask [23], and if two unlabeled objects overlap the same labeled object from the labeled mask the object with
the larger portion of the labeled projection receives the label. Once all the labeled objects are projected onto the unlabeled mask, any remaining unlabeled objects are new objects starting in that mask. New objects are each assigned a label by assigning one of the objects the current maximum label value, then incrementing the maximum label value to label the next unlabeled object. After this, all objects in the unlabeled mask are now either assigned to a previous object, or designated as a new object, thus completing the labeling of this slice. The mask that was the unlabeled mask is now used as the base labeled mask for the next unlabeled mask, this is repeated until all masks are labeled.

### 3.2 Proposed Approach

The flaw in the overlap algorithm for labeling is that this is not intuitive to how human vision interprets the data. One component of the human approach to identifying a whole object in 3D space is to begin at a portion of the object and “track” it in 3D space. This tracking in 3D space correlates to object tracking in motion sequences. A 3D data set can be viewed as a motion sequence by viewing the sequence in order along one of the axes. Re-labeling one of the spatial axes as a temporal axis, the 3D data set, viewed as a movie, becomes a fly through of the object being imaged. The slope of an object in the data set then translates directly to the velocity of its trajectory in the motion sequence.

It is debated how humans track multiple identical objects in a motion sequence [27, 28], it is known that motion perception in human vision uses inferences based on changes from one scene to the next. The abilities of these inferences and processing in our neural pathways have been illustrated with different behavioral experiments. One such experiment or demonstration is the random-dot kineomatogram [29]. Random-dot kineomatograms are used to illustrate an inference perception known as structure from
motion. These kineomatograms are constructed of a series of frames, in each frame there are dots that appear to be randomly placed to the observer as in figure 3.1. From frame to frame the dots move as if they were placed on a moving object, but the structure of the object is only revealed to the observer through the playing of the successive frames. This structure from motion is the result of the inferences that occur in processing along the visual pathway and within the visual cortex of the brain.

![Figure 3.1 Single Frame from a Random-Dot Kineomatogram](image)

Computer vision techniques while trying to mimic the mechanics of human vision also repeat the inferences made in the visual pathways. The random-dot kineomatogram only works if the dot motion is less than a certain amount between frames, else the sequence will only look like randomly flashing dots. Motion estimation algorithms make the assumption that motion between frames is small, thus replicating this constraint inherent in the visual pathway. Other computer vision algorithms exist that use methods similar to those in the visual pathway in order to produce similar results. The curve
tracing algorithm developed by Raghupathy and Parks discussed earlier operates under similar principles [24].

The algorithm developed in this thesis, assumes from slice to slice that although the object might change in fine details on the surface, the overall mass or volume will change very little, for this reason the center of mass serves as the trajectory. So in each slice the center of mass for each of the segmented objects is a single point on that object’s trajectory. Trajectory tracking is performed by estimating center of mass points for a slice as a linear function of the points from the previous slices. A center of mass point belongs to the trajectory that predicted the point closest to it; while the linear function for prediction is updated every slice using the last two slices to determine the new predictor. Iterating through the slices results in labeled trajectories that are based on predictors that are updated every slice. Then, with calculated trajectories, object labeling is performed by mapping the trajectory labels back to the original object.

### 3.3 Defining Trajectory Points

The developed algorithm operates by simplifying the 3D objects to trajectories in 3D space, each slice of the 3D volume is a point on the trajectory. The trajectory of an object is defined as the path through that object’s center of mass at each slice. Once all the objects are segmented the centroids are calculated for each object in each slice. The centroid of an object is the center of mass, or the place at which it would balance if it was placed on a pin. For a two dimensional plane with a density function of $\rho(x, y)$, the mass is $M = \iint \rho(x, y) dA = \int \int \rho(x, y) dxdy$ and the resulting centroid coordinates are found using:
The data sets used are digital images and therefore are not continuous, thus the centroid coordinate equations are rewritten in their discrete form:

\[
\bar{x} = \frac{\sum_{i=1}^{n} m_i x_i}{M} \quad \text{eq 18}
\]

\[
\bar{y} = \frac{\sum_{i=1}^{n} m_i y_i}{M} \quad \text{eq 19}
\]

where \( m_i \) is the "mass" at the location \( x_i \) or \( y_i \). Since all pixels have the same weighting in the binarized segmentation masks, \( m_i \) is number of object pixels at location \( x_i \) or \( y_i \). 

\( M \) is the "mass" of the object which is the total number of pixels in the object. Figure 3.2 shows center of mass points calculated for each object in each slice of one the sample data sets, these are like sampled versions of the objects' trajectories.
3.4 Labeling Trajectory Points

At any slice, \( k \) the mean squared error is calculated between the predicted centroid locations and the actual centroid locations in that slice. Centroid locations are predicted...
from the trajectory estimate, a first order bivariate polynomial. The centroid that
minimizes the error between the centroid and the predicted centroid location is
considered the continuation of the centroid trajectory, with some rules.

1. Origination:
   If a centroid $c$ is not the centroid that minimizes error for any of the trajectory
   projections $p$ then $c$ belongs to a new object starting in the current slice $k$.

2. Termination:
   If a trajectory projection $p$ does not have a centroid that minimizes the error,
   then the corresponding trajectory is terminated in the previous slice $k-1$.

3. Tie Breaking:
   If centroid $c$ of an object in slice $k$ produces the minimum MSE for two or
   more trajectory projections, it is associated with that projection that produces
   the lowest MSE. The MSE for the centroids are re-calculated for the other
   trajectory projections while omitting the common centroid $c$, that has already
   been assigned to a trajectory.

4. Simultaneous Termination and Origination:
   If MSE between a trajectory projection, $p$ and the minimizing centroid $c$ is
greater than some threshold $t$, then centroid $c$ is not an element of the
trajectory that projected $p$. Thus the trajectory is terminated in slice $k-1$,
while a new trajectory to which $c$ belongs, originates in slice $k$. 

42
The polynomial estimate of the trajectory is reduced to a vector in 2D space using the knowledge that the distance between slices is constant. So, the predicted centroid position in slice $k$ is:

$$\vec{T} = \vec{Q} + \vec{P} = \begin{bmatrix} Q_x + P_x \\ Q_y + P_y \end{bmatrix}$$  \hspace{1cm} \text{eq 20}$$

where $\vec{T}$ is the expected centroid position in slice $k$, $\vec{Q}$ is the location of the object centroid in slice $k-1$, and $\vec{P}$ is the predicted trajectory estimate. Once a centroid is assigned to a trajectory, the trajectory estimate is updated using the two most recent points added to the trajectory, defining $\vec{P}$ as:

$$\vec{P} = \vec{Q} - \vec{R} = \begin{bmatrix} Q_x \\ Q_y \end{bmatrix} - \begin{bmatrix} R_x \\ R_y \end{bmatrix} = \begin{bmatrix} Q_x - R_x \\ Q_y - R_y \end{bmatrix}$$  \hspace{1cm} \text{eq 21}$$

Once again $\vec{Q}$ is the location of the object centroid in slice $k-1$, while $\vec{R}$ is the location of the object centroid in slice $k-2$.

All object centroids in a given slice are either assigned to a current trajectory or are used as the start of a new object. Then current trajectories that did not continue are terminated before moving onto the next slice. Initialization of objects is handled in the first two slices of a new object. In any slice $k$, a new object can be created, in slice $k+1$ there is only one prior point in the object centroid trajectory, a location cannot be predicted for the object centroid in slice $k+1$, so the object centroid in slice $k+1$ closest to the object centroid in slice $k$ is assigned to the object.
3.5 Re-labeling

Each centroid in the centroid location matrix corresponds to an object within a slice with a unique label. These objects need to be re-labeled so that all of the segmented objects that belong to the same object are labeled so. A MATLAB function was written to implement the proposed labeling algorithm. The $l \times 2k$ centroid location matrix is passed to the function, which returns a sorted $l \times 2q$ matrix where $q$ is the number of objects in the whole data set. The sorted matrix contains the centroid locations arranged in columns that contain the $x$ and $y$ locations of the centroids for a particular object, with the rows corresponding once again to the slices. Slices in which a particular object is not present is represented with zeros as the centroid location for that object in that slice.

Using the sorted matrix, the objects in the slices are relabeled so that all objects in the slices that belong to the same global object have the same label. This is done with another MATLAB function that determines the number of objects in the data set based on the sorted matrix size, then for each slice it locates the object corresponding to each centroid location and labels them based on their centroid object classification. The object trajectories are visualized in figure 3.3 and 3.4 for the calculated data set discussed later.
Some times an object can double back on itself, or the object trajectory can change quickly enough that it is labeled as multiple separate objects as shown in figure 3.5. To prevent this, for each object endpoint the distance is calculated between that endpoint and the projected extension of all objects that terminate or originate in the slice prior to or the slice subsequent to that endpoint. Where the projected extension of an object is predicted centroid position for the slice before origination and the slice subsequent to termination of an object. If this distance between an object’s endpoint and the projected extension of another object is less than a predetermined threshold, then the trajectory points are continuations of the same trajectory, and the two labeled objects are the same object and thusly re-labeled as one object.
The addition of endpoint extension to the trajectory tracking approach further increased the versatility of the proposed labeling algorithm to handle data that doubles back on itself. As shown in figure 3.6.
4 Methods

Specific applications of the techniques discussed earlier are covered as they apply to the data within this thesis. Section 4.1 details the segmentation of the constructed data sets used for the evaluation of the developed algorithm. Section 4.3 and 4.4 discuss the 3D volume rendering techniques used to visualize the data, as well as issues encountered in forming the 3D visualizations.

4.1 Data Set

Since this algorithm was developed as a proof of concept to make an object identification technique that operates in a similarly to a human would identify objects; a data set was developed specifically to illustrate this feature. The first data sets were made using a combination of different types of foam insulation. Black foam pipe insulation was used to form two objects, each one a cylinder. These objects were then placed in a box in a manner trying to produce the desired data set to illustrate the confusion encountered by other algorithms but resolved with the proposed algorithm. The box was then filled with expanding foam insulation. Once the foam set it was sliced to produce the data slices.

4.2 Segmentation

As CT and MRI imaging techniques only produce gray scale images, any color data sets such as the example in figure 4.1 were converted to grayscale producing the image shown in figure 4.2. Conversion to gray scale was done by copying only the luminance information from the image. The red green blue (RGB) image was converted to National Television System Committee (NTSC) color space using the following transform:
\[
\begin{bmatrix}
Y \\
I \\
Q
\end{bmatrix} =
\begin{bmatrix}
0.299 & 0.587 & 0.114 \\
0.596 & -0.274 & -0.322 \\
0.211 & -0.523 & 0.312
\end{bmatrix}
\begin{bmatrix}
R \\
G \\
B
\end{bmatrix} \tag{22}
\]

where Y is the luminance channel while I and Q are the two chrominance channels, the gray scale image is created by using just the luminance channel. Labeling only needs to be performed on the segmented image and can be later applied to the gray scale image, so the scanned images were converted to binary images. Conversion to binary images was the first step of the segmentation used, this separates the objects from the background. Binary conversion was done using single level thresholding; the threshold for each slice was manually selected based on the gray scale histogram for the slice, the histogram for slice 8 from data set #2 from the example is shown in figure 4.3. Most of the slices used a threshold between 70 and 100 on the gray scale from 0 to 255. The thresholds used all fell within 60 to 110 on the gray scale from 0 to 255, the example was thresholded with a threshold of 100, the results are shown in figure 4.4.
From figure 4.4 it is seen that thresholding did not produce complete segmentation, there are small clusters of misclassified pixels. Morphological filtering was then applied to slice images. Each slice had the open operation applied with the disk shaped structuring element shown in figure 4.5. From the results of morphological filtering shown in figure
4.6 it is seen that there are still groups of misclassified pixels; these were manually fixed. Then all the slices were cropped to remove most of the irregular edges, producing the segmented slice shown in figure 4.7.
The binary images of the data slices have more than one object in them, but since they were only thresholded, there is no discerning information stored to separate the objects in the slice. Connected-component labeling was applied to each 2D slice to segment and identify the individual objects.

### 4.3 3-D Visualization

Wang uses a simple and effective rendering technique for high resolution CT scans [23], determining the pixel intensity based on its distance from the observer. This simulates illuminating the object from the direction of the observer while assuming uniform illumination. Implementation results in slow processing and is limited to only
one point of view at a time. Since the technique relies on rendering only the pixels that are seen as a result of a perspective projection, so for each viewing angle a new projection needs to be performed. So for any one viewing angle a 3D visualization can be reduced to a two dimensional array, but to produce an interactive display either a large number of these 2D arrays need to be stored or the visualization needs to be done in three dimensions.

Advancements in medical imaging and even imaging in general have resulted in many software packages to assist in image processing and visualization. MATLAB was used to do all of the computational processing and already contains functions to assist in volume visualization. The isosurface function is included in the standard installation of MATLAB version 7 (R14) and most likely earlier versions is what was used for the final visualizations. The isosurface function and graphics handling of MATLAB enables fully interactive 3D modeling, where the model can be rotated about 3 axes to provide continuous visualization from any point of view. Additionally the lighting type, intensity, and location can all be adjusted as well as the surface properties can be adjusted to change color and reflectivity.

An iso-surface is a surface corresponding to a constant value within a data set. Since the data set used for visualization is a binary set, this iso-surface is the surface of the object or objects of interest. Iso-surfaces do not interpolate data; they are only a rendering of data that is present. For this reason, the data being represented should be smooth to produce ideal visualizations. The data used to exemplify the labeling approach is representative of sparse cross sections, resulting in non-smooth data. To produce
smooth data and therefore, a more visually appealing 3D visualization, cross sectional data needed to be interpolated resulting in more cross sections.

### 4.4 Interpolation of Cross Sectional Data

Many techniques and algorithms are available for interpolation of gray scale images, but binary data does not work when these are applied. Binary data can be first converted to a grayscale image, and then grayscale values can be interpolated for the missing slices before being converted back to binary. As with the watershed transform discussed earlier, a binary image can be converted to a gray scale image through a distance transform. Two different distance transforms were investigated for the sake of interpolation of new slices; one is a standard Euclidian distance transform for binary data, the other is a distance transform developed by G.T. Herman et al. for shape based interpolation.

The Euclidian distance transform calculates the distance for each pixel to the closest non-zero pixel. So the value stored at each pixel location is its distance to the closest non-zero pixel. The resulting matrix can be displayed as a gray scale image which provides a basis for interpolation. Linear interpolation cannot be applied directly to a binary image, as interpolation results would be not be binary. The distance transform provides a smooth function to interpolate, interpolating slices from the distance transform results in new slices with the distance transform for the objects at those slice locations. After interpolation the slices are thresholded producing binary interpolated slices.

G.T. Herman et al. develop a different distance transform based on the observation that use of linear interpolation on data that has been segmented using thresholding results in the “staircase artifact” [30]. The “staircase artifact” is when the
object boundary remains in the same location from slice to slice and then shifts suddenly which defeats the purpose of interpolation, as the added slices provide no additional information. The proposed algorithm involves initializing the binary image so pixels within the object have a really large positive value, while the pixels outside have a really large negative value, they use 99 and -99 respectively. Edge pieces are handled by using 5 for inside pixels sharing an edge with an outside pixel, and -5 of outside pixels sharing an edge with an inside pixel. Templates are then used to chamfer these values to arrive at Euclidean or city-block distance transform estimates. Two passes are used to chamfer the images, the first one goes from across the rows left to right while going down from top to bottom. The second pass with the template flipped vertically and horizontally goes across the rows from right to left and up them from bottom to top.

The procedure for updating pixels during chamfering goes as follows:

1. If the central pixel under the template is a 5 or -5 it is left as is.

2. If the central pixel under the template is positive, then each pixel of the template is added to the image pixel under it and the central pixel is replaced with the smallest of the sums.

3. If the central pixel under the template is negative, then each pixel of the template is subtracted from the image pixel under it and the central pixel is replaced with largest difference.
5 Results

The results of testing the devolved algorithm on multiple data sets are presented within this chapter. Section 5.1 covers data sets physically constructed with foam and then scanned to produce digital data. Additionally, synthetic data sets are constructed digitally in 5.2 to illustrate desired object interactions for labeling comparisons. A more complex example of a quadruple helix is presented in section 5.3 to illustrate the relationship between when the algorithm fails and human mislabeling of data.

5.1 Foam Data Set

The first data sets were made using a combination of different types of foam insulation. Black foam pipe insulation was used to form two objects, each one a cylinder. These objects were then placed in a box in a manner trying to produce the desired data set to illustrate the confusion encountered by other algorithms but resolved with the proposed algorithm. The box was then filled with expanding foam insulation. Once the foam set it was sliced to produce the data slices.

Due to the thickness of the foam pipe and the difficulties securing the foam pipe while the foam insulation cured, it was not possible to construct a physical data set with the desired properties. Both the proposed labeling approach and the comparison approach of using overlap correctly labeled the data set.

The data set did prove useful in the evaluation of the technique for interpolating new cross sectional data. Figure 5.1 shows the rendering of the foam objects from the physical data set, while figure 5.2 shows the 3D rendering after interpolating the cross sectional data so that there were four times as many slices. From figure 5.2 it is seen that
the technique developed by G.T. Herman et al. is an adequate tool for interpolating new cross sectional slices in this case with foam pipe insulation objects.

![Figure 5.1 Foam Data Set](image1)

![Figure 5.2 Foam Data Set with Slice Interpolation by four](image2)

### 5.2 Calculated Data Set

Given that the constructed physical data set did not produce the desired interactions to test the proposed labeling algorithm, a synthetic data set was constructed in MATLAB to provide greater control over the slope and sampling of the objects. The synthetic data set is shown in figure 5.3; it is seen that this data set contains two distinct objects that pass close to each other, the object centroids are shown in figure 5.4. Results of the proposed labeling approach are shown in figure 5.5, labeling the objects as would be done by a human manually while the overlap approach produces the labeling seen in figure 5.6 which is incorrect labeling by labeling parts of two separate objects as one and then labeling the remainders producing three objects instead of two. Labeling of the segmented data using the overlap approach was performed in 10ms with the MATLAB
code written during the course of this research, the overlap approach took 110ms to label the objects with code written for comparison purposes.

The proposed algorithm correctly labeled the two objects, shown in figure 5.5, enabling them to be rendered together or separately. Problems were encountered with interpolating new slices. Either the sparseness of the cross sections or the steepness of
the slope for the objects resulted in poor interpolation results. Both the Euclidian distance transforms and the technique developed by G. T. Herman et al. resulted in the “staircase artifact” which produces choppy visualizations. Different interpolation techniques besides linear where also tried, but due to computational complexity memory errors were encountered with the purported cubic spline method. Best visualization results were achieved using a combination of linear interpolation of gray scale distance transforms and a three-dimensional smoothing function.

The three-dimensional smoothing function was used as an attempt to remove part of the “staircase artifact” this technique does not improve the data, or interpolate new data. This smoothing function was only used as an attempt to create a more appealing visualization. It is seen in figure 5.7 that this interpolation does not produce realistic results for the given data.

![Figure 5.7 Mislabeled Object from Overlap Approach](image-url)
5.3 Quadruple Helix Labeling

A more complex example was developed to illustrate and explore the similarities between the labeling for the proposed algorithm and the expected results of manual labeling. Figure 5.8 shows four helixes rotating around a central axis, cross sectional data was taken from these helixes and labeled using the proposed approach.

![Figure 5.8 Quadruple Helix Original Object](image)

The helixes were formed by rotating four circles around a central axis, the circles rotated 5.7143° per increase in pixel height. This corresponds to two complete rotations of the helixes in a height of 126 pixels. First the data set was sampled by taking ten cross-sectional data slices covering half of the height (one complete rotation), figure 5.9 shows the cross sectional slices from this sampling. The objects were then labeled using the proposed trajectory tracking approach. Figure 5.10 displays the center of mass for each object in each slice color coded to correspond to the object from which that center of
mass originates from, while figure 5.11 shows the centers of mass after being labeled by trajectory tracking, it is seen that the trajectory tracking approach correctly tracked the trajectories. The labels from the trajectories were mapped back to the cross-sectional data as shown in figure 5.12 with the labels correctly corresponding to the initial objects.

Figure 5.9 10 Cross Sectional Slices Covering half the height

Figure 5.10 Centers of Mass Marked with Object Origination
Using the same original data for the quadruple helixes the data set was sampled by taking ten cross-sectional data slices covering the full height of the helixes (two complete rotations), figure 5.13 shows the cross sectional slices from this sampling. The objects were then labeled using the proposed trajectory tracking approach. Figure 5.14 displays the center of mass for each object in each slice, color coded to correspond to the object from which that center of mass originates from, while figure 5.15 shows the centers of mass after being labeled by trajectory tracking. From the color coded centers of mass, it is seen that the trajectory tracking incorrectly labels the objects, but it incorrectly labeled as a human would label the objects without any prior knowledge of the helixes. The labels from the trajectories were mapped back to the cross-sectional data as shown in figure 5.16 once again illustrating the fact the trajectory tracking fails in the same manner a human would.
5.4 Real CT Data: The Pasta Patient

The final test for the algorithm was to use real data from one of the imagine modalities that was expected be used with the algorithm, either CT or MRI. Cooked spaghetti was scanned on a MicroCT scanner (GE Healthcare eXplore Locus), cooked spaghetti was selected as a data set because the overlapping and interweaving of the
individual pieces of pasta would be ideal to illustrate the features of the designed algorithm. Figure 5.17 shows a 3D rendering (MRICro Chris Rorden, University of South Carolina) of the spaghetti, from the rendering it is seen that individual strands of spaghetti cross each other and lay in different orientations.

![Figure 5.17 3D Rendering of Spaghetti CT Data Set](image)

As discussed before the data set was segmented using thresholding and morphological filtering. Each individual slice was labeled using connected component labeling, segmentation and labeling was then verified by hand to ensure that the data set was properly segmented prior to 3D labeling. The 3D CT data set was then labeled using the developed algorithm, the color coded labeling results are shown in figure 5.18.
In the real CT data set it is seen that the algorithm generally works well but fails with slow curving objects at the apex of the curve, if the tangent of the apex lies in the plane of the cross sections as shown in figure 5.19. The trajectory of the centroids in these objects has a discontinuity where the two sides of the curve meet, as shown in figure 5.20, resulting in the labeling of separate objects instead of one continuous object. This discrepancy cannot be corrected by increasing the threshold in the endpoint extension since different objects that end near each other would then end up being mislabeled with the same label applied to them.
Figure 5.19 Failure of Labeling

Figure 5.20 Trajectory Points at Failure
6 Conclusion

The processing of sparse cross sections from MRI or CT scans presents many difficulties. Techniques and methods for each stage of the processing required to produce 3D visualizations was discussed and explored within this research. Objects where first segmented within the cross-sectional data slices then labeled followed by interpolating additional slices for the final 3D visualization. The final result was a labeled 3D data set that could be used to produce interactive visualizations of the objects in the data set. These accomplishments were secondary to the focus of this research which was to produce a labeling approach that was computationally efficient and memory efficient while produce the same labeling results as those done by a human via mimicking the same inferences made in the human visual pathway. Due to possible complications introduced by the orientation of the cross-sectional slices, curve linkage is added as the final step in trajectory tracking to ensure even objects with rapid trajectory changes are correctly labeled.

The investigation underlying this thesis has successfully accomplished the following:

1. Synthetic data was created to represent possible object interactions in materials, containers, or anatomical regions that could be imaged using MRI or X-ray CT. This data was created in a way that it could easily be re-configured or scaled. Also the original configuration of the data could be seen in order to provide explicit knowledge and assist in evaluation of the labeling approaches. The data set was successfully segmented using thresholding and morphological filters.
2. The proposed labeling approach labels the objects throughout the 3D data set using inferences and constraints similar to those in the human visual pathway to produce labeling that is consistent with manual human labeling of the objects.

3. This proposed labeling approach is less computationally and intensive and more memory efficient than current 3D volume labeling approaches.

4. Trajectory tracking correctly labels objects in test sets that are incorrectly labeled by current 2D labeling approaches such as the overlap approach as well as performing labeling faster than the overlap approach.

5. Use of the proposed labeling approach allowed for 3D visualizations of the objects by selecting a single object label, or all of the labeled objects.

6. The proposed labeling approach was tested on a real CT data set, producing labeled 3D renderings.

The technique developed in this thesis can potentially be improved as follows:

1. The segmentation technique of 2D connected-component segmentation was not completely automated; some human interaction was required to produce proper segmentation. Thresholding and morphological filtering were used for their ease of implementation, more robust and complex segmentation techniques should be evaluated to produce better segmentation results. In medical and industrial applications, there are very few instances where segmentation is completely automated unless a technique is written for a very specific application. So a single robust segmentation algorithm for general applications is very unlikely.
2. Interpolation techniques for sparse-cross sections need to be further developed. The technique used on the data set from the foam slices was successful in producing more data slices for rendering. Unfortunately this technique did not have the same positive results with the calculated data set. This is most likely because of the steep object trajectories. A possible solution would stack use the object trajectory that is already calculated during labeling to reposition the objects on top of each other for interpolation; the objects would then be positioned back to their original location while the trajectory would be used to place the interpolated slices at their proper position.

3. Using the approach described a higher order labeling technique could be constructed to produce more complete labeling. The higher order system would use the centroid tracking with another perspective to gain more information on object interactions and compensate for errors. By re-slice the 3D volume along a different plane and performing curve tracing in the new direction, curves that were mislabeled due to their curve apex being tangential to the cross sectional plane, no longer lie in the cross sectional plane and are correctly labeled. A technique to identify mislabeled curves in one perspective needs to be developed to combine these results and produce correctly labeled objects. This would increase the computational complexity of the overall approach, but the payoff would be an improvement in labeling results.

4. An additional method of acquiring more data on the interaction of the objects would be to use a higher order predictor for trajectories, this would prove to be
more computationally expensive and not fix the issue with slow curving objects, but might be needed for more complex objects.

5. Investigations should be made into other curve point extraction mechanisms. It was seen that while the centroid calculation was simple and fast, the trajectories produced by this approach had discontinuities.
References


Appendix A: MATLAB Code

function [z] = getCent(t)
%-----------------------------------------------------------------------
%Returns the centroid for each labeled object in t
%If t is a 3D image, returns the centroids in dim 3
%Brandon Mikulis
%4/19/2005
%-----------------------------------------------------------------------
slices=size(t,3);
z=[];
for i=1:slices
    cent=regionprops(t(:,:,i),'centroid');
    c=struct2cell(cent);
    mCent=cell2mat(c);
    j=find(~isnan(mCent));
    labeledCent=mCent(j);
    if (size(labeledCent,2)>size(z,2))
        aSize=size(labeledCent,2)-size(z,2);
        aZeros=zeros([size(z,1) aSize]);
        z=[z,aZeros];
    elseif (size(labeledCent,2)<size(z,2))
        aSize=size(z,2)-size(labeledCent,2);
        aZeros=zeros([1 aSize]);
        labeledCent=[labeledCent, aZeros];
    end
    z=[z; labeledCent];
end
function [z] = CurveTrace (s)

%Labels a curve point matrix such as returned by getCent
%Curves points are labeled by being placed in the same x-y columns.
%
%Brandon Mikulis
%7/07/2005
%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Threshold=25;
i=find(s(1,:));
z=s(1,i);

odd=mod(length(i),2);
if (odd)
  Slice1Objects=ceil(length(i)/2);
else
  Slice1Objects=length(i)/2;
end

i=find(s(2,:));
odd=mod(length(i),2);
if (odd)
  Slice2Objects=ceil(length(i)/2);
else
  Slice2Objects=length(i)/2;
end
d=[];
for j=1:Slice1Objects  %forms a matrix of distances
  for y=1:Slice2Objects
    d(j,y)=sqrt((s(1,2*j-1)-s(2,2*y-1))^2+(s(1,2*j)-s(2,2*y))^2);   
  end
end

[m,y]=min(d,[],2);   %Terminated b/c Threshold
[r,c]=find(m>=Threshold);

r=unique(r);
Terminated=zeros(size(d));
Terminated(r,:)=d(r,:);
d(r,:)=inf;
%d(r,:)=Terminated(r,:)

[minDist, Object]=nonRepMin(d);   %Terminated b/c of Proximity and Continuing Objects
NewObjects=setdiff([1:Slice2Objects],Object);  %New Objects

Slice2=inf*ones([1 (2*Slice1Objects)]);
cont=find(Object<inf);
for j=1:length(cont)
  Slice2(2*cont(j)-1:2*cont(j))=s(2,2*Object(cont(j))-1:2*Object(cont(j)));
end

for j=1:length(NewObjects)
  Slice2([Slice2, s(2,2*NewObjects(j)-1:2*NewObjects(j))]);
end

if (length(Slice2) > length(z))
    append=inf*ones([1,length(Slice2)-length(z)]);
    z=[z,append];
end
z=[z;Slice2];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%END SLICE 1 and 2
% START ITERATIONS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

sliceKObj=Slice2Objects;

for sliceN = 2:size(s,1)-1
    sliceK=sliceN+1;
    sliceNObj=sliceKObj;
    
i=find(s(sliceK,:));
    odd=mod(length(i),2);
    if (odd)
        sliceKObj=ceil(length(i)/2);
    else
        sliceKObj=length(i)/2;
    end
	pred=[]; pts=[];
    for j=1:size(z,2)
        temp=[z(sliceN-1,j);z(sliceN,j)];
        pts=[pts,temp];
    end
    pred=2*pts(2,:)-pts(1,:);
    srt=fmd(pts(1,:)==inf);
    pred(srt)=pts(2,srt);
    spt=find(pts(2,:)==inf);
    pred(spt)=inf;
end

d=[];
for j=1:(size(z,2)/2)
    %forms a matrix of distances
    for y=1:sliceKObj
        d(j,y)=sqrt((pred(2*j-1)-s(sliceK,2*y-1))^2+(pred(2*j)-s(sliceK,2*y))^2);
    end
end

[m,y]=min(d,[],2);
[r,c]=find(m>Threshold);
    %Terminated b/c Threshold
r=unique(r);
Terminated=zeros(size(d));
Terminated(r,:)=d(r,:);
d(r,:) = inf;

[minDist, Object] = nonRepMin(d); % Terminated b/c of Proximity and Continuing Objects
[r2,c2] = find(minDist>=Threshold); % Terminated b/c of Threshold post Proximity test
minDist(r2,:) = inf;
Object(r2,:) = inf;
NewObjects = setdiff([1:sliceKObj], Object); % New Objects

newSlice = inf*ones([1,1:size(z,2)]);
cont = find(Object<inf);
for j = 1:length(cont)
    newSlice(2*cont(j)-1:2*cont(j)) = s(sliceK,2*Object(cont(j))-1:2*Object(cont(j)));
end

if (length(NewObjects) >= 1)
    for j = 1:length(NewObjects)
        newSlice = [newSlice, s(sliceK,2*NewObjects(j)-1:2*NewObjects(j))];
    end
end

if (length(newSlice) > size(z,2))
    append = inf*ones([size(z,1),length(newSlice)-size(z,2)]);
    z = [z, append];
end
z = [z; newSlice];
end