IMPACT OF VARIED LOW RESOLUTION PHANTOMS ON INTENSITY MODULATED PROTON THERAPY DOSE DISTRIBUTION

A Thesis

Presented to the

Faculty of

California State University,

San Bernardino

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in

Computer Science

---

by

Aarohi Shyam Padhye

September 2013

IMPACT OF VARIED LOW RESOLUTION PHANTOMS ON INTENSITY MODULATED PROTON THERAPY DOSE DISTRIBUTION

A Thesis

Presented to the

Faculty of

California State University,

San Bernardino

---

by

Aarohi Shyam Padhye

September 2013

Approved by:

Dr. Keith Evan Schubert, Advisor, School of Computer Science and Engineering	Date
Dr. Ernesto Gomez	
Dr. Kay Zemoudeh	
Dr. Reinhard W. Schulte	

© 2013 Aarohi Shyam Padhye

ABSTRACT

In recent years, inversely planned intensity modulated proton therapy (IMpRT) has become the focus of research to further improve the dose distributions that can be obtained with proton therapy. IMpRT is best used to deliver a potent and precise dose of protons to the most complicated tumors like the ones embedded in the nooks and crannies of the head and neck or skull base. The starting point of each IMpRT dose calculation is a digital model of the patient volume of interest, e.g., the patient's head, usually provided by a computed tomography (CT) scan. A head CT scan consists of about 200 slices of 1-2 mm thickness and each slice is organized into a matrix of 512 $\times$ 512 image pixels. In 3D, this creates a digital space comprised of the order of 50 million voxels. Each voxel has material properties that are needed to calculate the proton dose delivered by a large number of proton pencil beams with different directions and intensities.

The calculation of the proton dose distribution has to take into account the material (tissue, bone, brain) in the treatment area of the patients body. Type and density of the different tissues are inferred from the x-ray attenuation displayed in the x-ray CT images.Some materials are easy to identify but something like an inner cortical bone which is very thin as well as gets blended with brain and soft tissue due to the limited spatial CT resolution is tough to identify. This thesis tries to resolve this problem using different image segmentation techniques to get a fast and consistent tissue assignment. This methodology is also useful for generation of standardize digital phantoms using segmentation of a high-resolution CT scan of a phantom with tissue-equivalent materials, but this phantom generation is not only costly and time-consuming. Developing advanced and fast segmentation

techniques to generate a high-resolution digital head phantom from the CT scan of a commercial pediatric head phantom (Model 715, CIRS) is the final goal of this thesis. These methods can later be used for fast segmentation of real patient CT data.

ACKNOWLEDGEMENTS

Completing my Masters is apparently one of the most crucial activities of first 26 years of my life. The best and worst times of this venture have been shared with many people. It has been a great experience to spend the last 2 years at the Department of Computer Science at California State University, San Bernardino, and I will always cherish this experience.

First and foremost, I would like to express my gratitude to my advisor, Dr. Keith Schubert. He was able to patiently provide the guidance and encouragement throughtout the span of work on my thesis. He has always been a strong and helpful advisor during my courses and thesis. He gave me the freedom and support to pursue work independently.

Additionally, I would like to thank my committee members, Dr. Ernesto Gomez, Dr. Kay Zemoudeh and Dr. Reinhard Schulte for their continuous support and guidance. Their suggestions have always served me well and I owe them my heartfelt gratitude.

I would also like to recognize here the efforts of Dr. Josephine Mendoza who in every step of the way from 2011 to date has helped me to ensure that I complete my coursework requirements and move into candidacy with success.

Special thanks go to Mrs. Monica Latimer, the best and one of the most supportive department secretary, for her help throughout my graduate work in the department.

Members of Loma Linda University Medical Center also deserve my sincere thanks, without whose assistance I could not have continued my work. I would also like to thank LLUMC for allowing me to be part of a very talented and professional group

of people.

I also would like to thank my parents for their constant love and support which provided me inspiration during my work and was an ongoing driving force. I owe them my success and I wish I could show them how much I love and appreciate their support. My husband, Ajit Marathe, whose love and backing allowed me to stay focused on my graduate work. He already has my heart so I will just give him a heartfelt thanks. I also want to thank my in-laws for their unconditional support.

DEDICATION

This thesis is dedicated to my husband who has always stood by me and dealt with my absence from many family occasions with a smile and my Professor Dr. Keith Schubert who has been my friend, guide and philosopher.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

The primary purpose of this thesis is to discuss the usefulness of image segmentation techniques in creating accurate proton dose distribution plans. The methods presented in this thesis were specifically implemented on the phantoms to identify the materials and regions but for the patient IMpRT actual patient CT scans would be used to overcome the limitations towards creating proton treatment dose distribution plans . Although there exist current systems that successfully create these plans to treat patients, these new segmented phantoms would provide a basis to create automatically identified regions which can be later used for the actual patient CT scans to create accurate digital models for calculating accurate dose distribution plans. The tasks accomplished during this thesis were implemented under the contiguous guidance of Dr. Reinhard W. M. Schulte, M.D., at Loma Linda University Medical Center (LLUMC).

This thesis provides analysis and solutions to the problems related to identification of materials and regions in the CT Scans. The following questions are addressed:

1. How do the proposed methods differ from past and present work, and where have been the enhancements made in this work?

2. What precision can be attained from rectified images? Does this meet or surpass the recognized objectives of the project?

1.1 Background

1.1.1 Principles of Intensity Modulated Proton Radiation Therapy

Intensity modulated proton radiation therapy and radiosurgery, short IMpRT and IMpRS, are evolving techniques for highly conformal dose delivery to tumor or other targets in close proximity to sensitive and critical organs at risk. IMpRT is delivered in several dose fractions, while IMpRS is delivered in as a single dose or a few (up to 5) dose fractions applying stereotactic techniques. The underlying principle of these techniques is to aim at the target from many different directions (either in 2D or 3D) with multiple narrow proton beams, or *pencil beams*, and to modulate the intensity (or fluence) of each beam, taking into account whether they pass through critical organs at risk or not. The most important characteristic of a proton beam is that it delivers a low dose in the initial part of the beam followed by a rapid increase of dose, leading to a dose peak (the Bragg peak) and a rapid distal dose fall-off to zero dose behind the Bragg peak. The Bragg peak is placed inside the target at a given *beam aiming point*. Note that several pencil beams sharing the same central axis can be "stacked" in beam direction, and this arrangement may be called a *beamlet*.

The starting point of each IMpRT/RS calculation is a digital model of the patient volume of interest (1.1), e.g., the patient's head, usually provided by a computed tomography (CT) scan. A head CT scan consists of about 200 slices of 1-2 mm thickness and each slice is organized into a matrix of 512 $\times$ 512 image pixels. In 3D, this creates a digital space comprised of the order of 50 million voxels. Each voxel has material properties that are needed to calculate the proton dose delivered by the

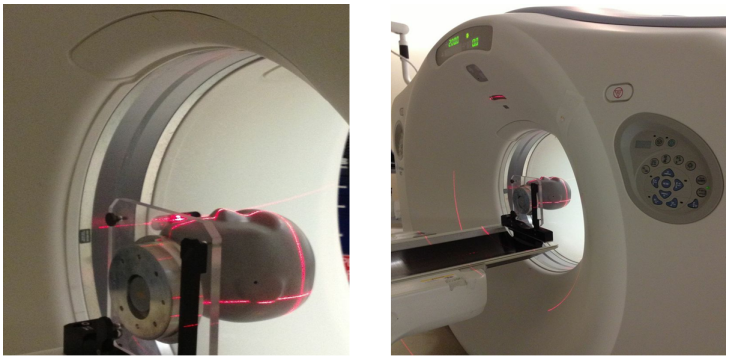

Fig. 1.1: Commercial Pediatric Head Phantom in the CT Scanner of the Department of Radiation Medicine at Loma Linda University Medical Center

different proton pencil beams.

In practical applications, one generates a generic pencil beam dose model for a unit-intensity proton beam in water and scales the distance between the entry point of a proton beam into the object and the beam aiming point by multiplying the intersection length of each voxel with the so-called relative stopping power (RSP) with respect to water. This information is provided by converting the numbers provided by the CT scan (Hounsfield units) to RSP, using a HU-to-RSP calibration curve. In the future, the RSP of voxels will be directly reconstructed from a proton CT (pCT) scan. Knowing the central beam axis dose as a function of depth in water, one can then assign the correct dose of the unit-intensity proton pencil beam to each voxel on the central beam axis. Similarly, knowing the lateral dose fall-off at each depth, one can calculate the correct dose for each off-axis voxel based on its orthogonal distance from the beam axis.

Given a distribution of the intensities of in the limit, continuously spaced proton pencil beams directed at the target, one can calculate the resulting dose distribution

in the voxels of the object using a proton dose operator $\mathfrak{D}$ that mathematically connects the two quantities. Often times, the chosen intensities do not result in a satisfactory dose distribution, that is, one that meets the dose constraints dictated by the radiosensitivity of the tumor and the organs at risk. In general, one wants the target dose to exceed some minimum value and the dose in organs at risk not to exceed a maximum value that can lead to serious complications. Therefore, it is better to "prescribe" a dose distribution selected from a subset in a continuum of possible dose distributions that meet the clinical requirements and then to find a fluence distribution that will lead to a dose distribution that is a member of this "solution" subset. As we will see below, the solution of such an "inverse" treatment planning problem can be found mathematically by formulating a discrete mathematical model of IMpRT that can be solved, in principle.

1.1.2 The Discrete Model of Intensity Modulated Proton Radiation Therapy

In the absence of a closed-form analytic representation of the proton dose operator $\mathfrak{D}$ that calculates the dose distribution given a the fluence of an continuum of proton pencil beams, and, therefore, the absence of such a presentation of its inverse operator $\mathfrak{D}^{-1}$, one must resort to a fully-discretized model of the problem. The term *full* in "fully-discretized model" refers to the fact that both the external proton radiation field and the patient volume are discretized, leading to a problem formulated in a finite-dimensional vector space. To do this we divide the beam's cross-section into a finite rectangular grid of squares and the beam angles into discrete angular steps separated by a constant interval, which may be chosen differently for each IMpRT

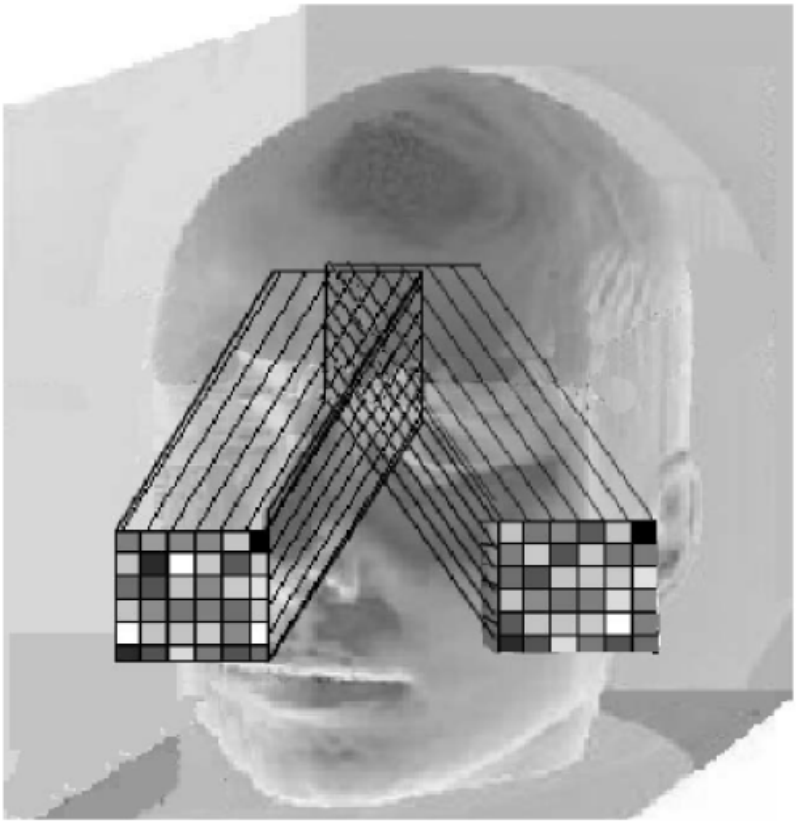

Fig. 1.2: Two IMpRT Beams from Different Directions. Variable Shades of Gray Correspond to Different Fluences (Number of Protons Per Area). Note that each Square in the Beam Cross Section can be Occupied by more than One Proton Pencil Beam, making up a Beamlet, each with a Different Bragg Peak Depth and Intensity.

treatment plan (see Figure 1.2). Further, we discretize the proton energy into steps, such that the proton Bragg peaks, i.e., the dose maximum of a proton pencil beam, are located at well-defined discrete aiming points within the patient volume. Each proton pencil beam is thus assigned a discrete direction and a discrete energy.

Figure 1.3(left) shows a representative two-dimensional (2D) cross-section through the object. In a contiguous set of cross-sections, the treatment planner defines a set of voxels that belong to the target. Other voxels sets may be defined that are assigned

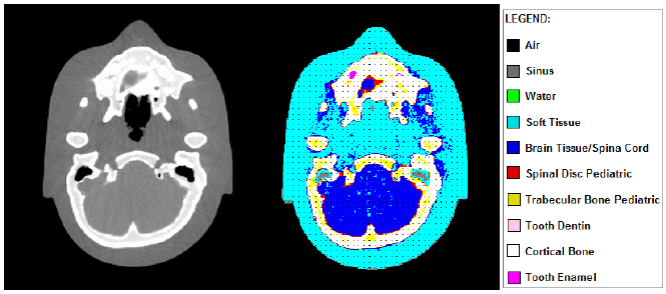

Fig. 1.3: Example of a CT Head Section Before (Left) and After Conversion to a Color-Coded Image that gives each Voxel a Tissue Assignment (Right)

to an organ at risk, e.g., the brainstem, or other normal tissue regions, such as brain and skull bone. In order to simplify the image segmentation process and to calculate the dose of unit-intensity beams, each image of the CT data set needs to be processed in order to assign a given tissue type to each voxel based on the CT (HU or RSP) value.This is shown in Figure 1.3(right).

Mathematical formulation of the discrete IMpRT model

The patient volume Ω is divided into a discrete grid of voxels the centers of which are the desired dose calculation points. These are represented by the family of triplets of 3D coordinates $\{(r_j) \mid j = 1, 2, \ldots, J\}$. Further, we define a discrete number of proton pencil beams by their entry direction unit vectors $\{v_i \mid i = 1, 2, \ldots, I\}$. and aiming point $\{(\hat{r}_i) \mid i = 1, 2, \ldots, I\}$.

Let a_{ij} be the dose deposited at the jth grid point (r_j) in the patient volume Ω due to the ith pencil beam $(\hat{r}_i, v_i)$ of unit proton fluence and define the I-dimensional vector $a^j = (a_{ij})_{i=1}^I$ for $j = 1, 2, \ldots, J$. Let x_i denote the actual (yet unknown) fluence of the ith pencil beam $(\hat{r}_i, v_i)$ and define the I-dimensional vector $x = (x_i)_{i=1}^I$ which is

unknown vector of all pencil beams' fluences that should deliver the required dose to the patient volume Ω. Finally, let $\overline{d}_j$ and $\underline{d}_j$ be an upper-bound and a lower-bound, on the permitted or required, respectively, dose in the jth grid point (r_j) in the patient volume Ω.

With these notions we can define discrete forward and inverse problems of IMpRT as follows.

The discrete forward problem of IMpRT: Given a patient volume Ω, whose physical properties are known, and a discretized (into I proton pencil beams) external proton radiation field $\{(\hat{r}_i, v_i) \mid i = 1, 2, \ldots, I\}$, along with a proton pencil beams intensity vector x, find the discretized proton dose distribution function $D(r_j)$ for all $(r_j) \in \Omega$

This discrete forward problem can be solved if all I-dimensional vectors $a^j = (a_{ij})_{i=1}^I$ for $j = 1, 2, \ldots, J$, are known to us, e.g., by having been pre-calculated by a forward problem solver computer package. In that case, denoting $d_j = D(r_j, \theta_j)$ for all $j = 1, 2, \ldots, J$, we just need to calculate

$$\sum_{i=1}^{I} a_{ij} x_i = d_j, j = 1, 2...., J. \tag{1.1}$$

The J-dimensional vector $d = (d_j)_{j=1}^J$, whose components are the discretized proton dose distribution function $D(r_j)$ values, is called a dose vector.

The discrete inverse problem of IMpRT: Given are a patient volume Ω, whose physical properties are known, an upper-bound dose vector $\overline{d} = (\overline{d}_j)_{j=1}^J$ and a lower-bound dose vector $\underline{d} = (\underline{d}_j)_{j=1}^J$, on the permitted and required, respectively, doses at the grid points $\{(r_j, \theta_j) \mid j = 1, 2, \ldots, J\}$ in the patient volume Ω. Find a proton pencil beams fluence vector x such that

$$\underline{d}_j \leq \sum_{i=1}^{I} a_{ij} x_i \leq \overline{d}_j, \quad forall \quad j = 1, 2, ..., J \quad and \quad x_i \geq 0 \quad forall \quad i = 1, 2, ..., I (1.2)$$

This formulation of the discrete inverse problem of IMpRT does not aim at a proton pencil beams fluence vector x that will deposit a fixed prescribed dose in each voxel but rather calls for a solution of that is called in optimization theory the solution of a *linear feasibility problem*. The term "feasibility" refers here to the fact that no exogeneous objective function is set up for optimization but rather any point in the feasible set $\{x \in R^I \mid \underline{d}_j \leq \sum_{i=1}^{I} a_{ij} x_i \leq \overline{d}_j$, for all $j = 1, 2, \ldots, J\}$ will be "acceptable" by the treatment planner. This feasibility approach to setting up the discrete inverse problem has its roots in some early papers on radiation therapy treatment planning where the term IMpRT was even not used, see [5, 17, 16, 15].

The J individual linear feasibility constraints of (1.2)can be grouped according to volumes of interest in the patient volume Ω.

1.2 Significance

In 1990, LLUMC inaugurated Proton Treatment Center, it's first ever hospital-based facility for proton therapy. Due to this treatment's increasing popularity and efficiency, there has been a sudden increase in these proton treatment centers. Recently, more than 10 such proton treatment facilities have been launched world wide.

With the rising application of protons for the cure of patients, doctors want to use their unique ability to extract relevant information about the materials from the CT scans to create accurate treatment plans. There have been major improvements in generating medical images but there still exist issues like low contrast, noise, and

other imaging ambiguities which makes the identification difficult. Currently a physician manually defines the boundaries and regions of interest in individual scans to create the proton treatment plans. However, how to automate this task is not yet been explained. In this case, it brought me the idea to investigate the possibility of automating the regions and material identification process by using different image segmentation techniques. What are the unique characteristics of each materials and region and how can we use them for segmenting the dicom images. Image segmentation's primary goal is to locate objects and boundaries by the means of allotting a tag to every pixel in an image such that pixels with the same tags share some genuine visual attributes, which is exactly what the main purpose of this thesis is.

1.3 Purpose

For the forward dose calculation, it is essential to assign different regions in the CT images to different materials, in this case to different human tissues. The simplest way to do this is to define HU intervals and assign them to a specific tissue, as shown in table 1.1, which is the conversion table for a pediatric head phantom with 9 different tissue types. However, as can be seen in Figure, this assignment is not always perfect due to the presence of noise and artifacts in the CT images.

Automated image segmentation, whose main intent is to extricate the object boundary features automatically, plays a significant role in finding the boundaries between different tissue regions and assigning pixels inside these boundaries to the correct materials. A crucial issue is to segment regions with boundary inadequacies, which means either there are lacking edges and/or absence of texture contrast amidst

regions of interest (ROIs) and background. To concentrate on this issue, various segmentation methods have been recommended and implemented in this thesis, with many of them procuring rather acceding results.

Tab. 1.1: Tissue Categorization according to Hounsfield Unit Value

HU Interval	Tissue
$[-1000, -800)$	air
$[-800, -700)$	sinus
$[-700, 40)$	soft tissue
$[40, 90)$	brain
$[90, 150)$	spinal disc
$[150, 200)$	trabecular bone
$[200, 1000)$	cortical bone
$[1000, 2000)$	tooth dentin
≥ 2000	tooth enamel

2. IMAGE SEGMENTATION

2.1 Introduction

In computer vision, image segmentation is the process of partitioning a digital image into multiple segments (sets of pixels, also known as superpixels) [3]. Segmentation's primary aim is to segregate the pixels of an image into clusters so that it can greatly correspond with the objects in an image and this is essentially the beginning of any application related to automated computer vision. With time, the analysis on image segmentation has received an immense rate of concentration. There are jillions of segmentation techniques, but none of them can be acknowledged as suitable for numerous images, each method works particularly well for a specific type of image. There is no method approved extensively for image segmentation and so it remains a crucial problem to pin-point a specific method for different applications.

On the basis of 2 key attributes of intensity values, image segmentation methods can be broadly divided into two groups:

- Detecting Discontinuity - This group divides an image by identifying the sudden transitions in intensity. Edge detection technique falls in this group of image segmentation algorithms.
- Detecting Similarity This group divides an image into sections that are identical

based on a set of predefined norm. Thresholding, region growing, region splitting and merging techniques fall in this group of image segmentation algorithms.

2.1.1 Applications of Image Segmentation

Some of the real-world applications of image segmentation [3] are:

- Medical imaging
  - Locate tumors and other pathologies
  - Measure tissue volumes
  - Diagnosis, study of anatomical structure
- Object Detection
  - Locate objects in satellite images (roads, forests, etc.)
  - Pedestrian detection
  - Face detection
  - Brake light detection
- Recognition Tasks
  - Face recognition
  - Iris recognition
  - Fingerprint recognition
- Traffic control systems
- Machine vision

- Agricultural imaging crop disease detection
- Content-based image retrieval

There are numerous algorithms and techniques, which have evolved over the years for image segmentation. But to use them efficiently, these methods should be integrated with the field's specific knowledge so as to resolve the respective segmentation problems accurately.

2.2 Image Segmentation Methods

2.2.1 Edge Detection Methods

Edge detection pertains to the routine of classifying and detecting the sudden breaks in an image. These breaks or gaps are unexpected shifts in pixel intensity which differentiate the boundaries of objects in a scene. Traditional edge detection methods concern convolution of an image with a operator, which is designed to be susceptible to huge gradients in the image while restoring values of zero in uniform regions.

Below mentioned are some of the variables which are involved in determining the edge detection operator:

- Edge orientation: This variable is responsible for determining the native direction in which it is most responsive to edges. This operator can be optimized as per the needs of the application to search for diagonal, horizontal or vertical edges.
- Noise environment: Noise as well as edges contain high-frequency content so edge detection gets difficult in noisy images. If we try to reduce the noise, the result is an image with blurred and distorted edges. So to detect edges in noisy

images, the operators selected are normally larger in extent which can equate enough data to deduct localized noisy pixels.

- Edge structure: Edge detection relies on the step change in intensity but in some cases edges might not have this. Effects such as refraction or poor focus can return objects with boundaries determined by a successive fluctuation in intensity. The operator should be selected in a way that it is susceptible to a successive change in those cases.

Edge detection can be achieved in numerous ways. However, other techniques may be classified into two categories:

- Gradient: This technique detrmines the edges by searching for the largest and least in the first derivative of the image.

- Laplacian: To find edges, laplacian technique looks for zero crossings in the second derivative of the image. The shape of the edge is like a 1-D shape of a ramp and computing the derivative of the image can accentuate its location.

Roberts Cross Operator

The Roberts Cross operator is an efficient way to compute, 2-D spatial gradient measurement on an image. At individual points in the output, pixel values describe the pedicted absolute magnitude of the spatial gradient of the input image at that point. The operator comprises of a couple of 22 convolution kernels and one kernel is obtained by rotating the other by 90 degrees. This is very much the same as the Sobel Operator. These kernels are optimized maximally for edges running at 45

degrees to the pixel grid with one kernel each representing for the two perpendicular orientations. You can apply these kernels to the input image either separately which would create distinct calculations of the gradient component in each orientation or in combined form which would find the absolute magnitude of the gradient at each point and the orientation of that gradient.

The gradient magnitude is given by:

$$|G| = \sqrt{G(x^2) + G(y^2)} \tag{2.1}$$

but in majority cases an approximate magnitude is computed using:

$$|G| = |Gx| + |Gy| \tag{2.2}$$

which is much faster to compute.

Prewitt's Operator

Prewitt operator is identical to the Sobel operator and is good for determining the vertical and horizontal edges in images.

Laplacian of Gaussian (LoG)

The Laplacian is a 2-D isotropic measure of the 2nd spatial derivative of an image. The primary reason for using Laplacian for edge detection is its ability to focus on the regions of rapid intensity change. In majority scenarios, a smoothed image which has been processed using a Gaussian Smoothing filter is fed to the Laplacian to lower its receptivity to noise. This operator usually takes one graylevel image as input and produces another graylevel image as output.

Canny's Edge Detection Algorithm

The Canny edge detection algorithm is known to many as the optimal edge detector. The number one and most evident reason for its popularity is the low error rate. The first criterion is that edges appearing in images are not hidden and that it generates a NO response to non-edges. The next criterion is that the edge points need to be well localized, which means, there should be minimal distance between the edge pixels found by the detector and the actual edge in the image. A third criterion is to have a single response to each individual edge. The major reason behind implementation of this criterion was the inability of first 2 in completely eliminating the possibility of multiple responses to an edge. Based on these criterions, canny edge detector starts by smoothing the image so as to eliminate the noise. The next step is to find the image gradient to accentuate the regions with high spatial derivatives. Once these regions are highlighted, canny walks along these regions and eliminates the pixels which are is not at the maximum. The gradient array is further compressed using hysteresis. Hysteresis is a technique which traces along the leftover pixels that have not been removed earlier. Hysteresis utilizes 2 thresholds: low threshold which sets the magnitude to zero (non-edge) if its lower than this value and a high threshold which sets the magnitude as an edge if its above this value. Also if the magnitude is between the low and high thresholds, it is set to zero unless there is a path from this pixel to a pixel with a gradient above higher threshold.

Below mentioned are the steps to implement this algorithm:

1. Smooth the image with a Gaussian filter

2. Measure the gradient magnitude and orientation using finite-difference approxi-

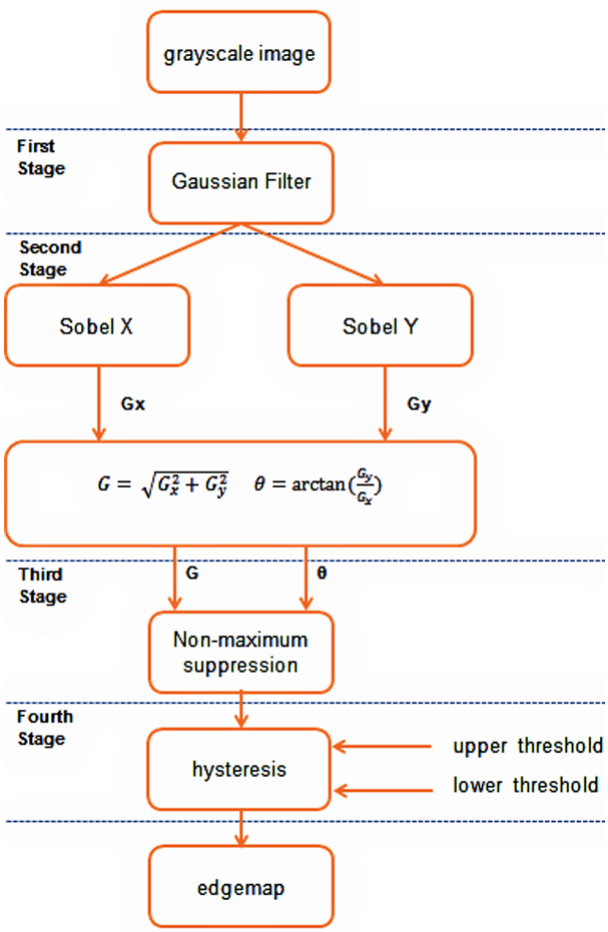

Fig. 2.1: Canny's Edge Detection Algorithm

mations for the partial derivatives

3. Assign non-maxima suppression to the gradient magnitude

4. Use the double thresholding algorithm to detect and link edges

Some of the factors which can affect the efficiency and computation time of Canny edge detector are:

- Size of the Gaussian filter: Selecting a smoothing filter in the initial stage of the algorithm is very important as it directly affects the results. A small filter

results in minor blurring and detection of small, sharp lines i.e. the weak edges. On the contrary, a large filter leads to more blurring by smudging out the given pixel's value over considerable area of the image. This kind of blurring might not be useful for weak or small edges but provide acceptable results in detecting larger, smoother edges like for the edges of a rainbow.

- Thresholds: Using 2 thresholds with hysteresis provides more extensibility to the single-threshold approach, but the conventional issues of the thresholding approaches are still applicable. If chosen a threshold that is too high, would result in missing important information. While setting a threshold that is too low will lead to false identification of irrelevant information like noise to be important. It is tough to identify a common threshold that would work effectively on all images. There doesn't yet exist a proved and verified approach to this problem.

2.2.2 Thresholding Method

Threshold based image segmentation is considered as a plain yet adept approach, especially for segmenting images that have light objects on dark background. This method depends on imagespace regions i.e. on the attributes of image. Thresholding techniques selects an appropriate threshold T, which separates the image pixels into numerous regions and isolates the objects from background. Let's consider a pixel (x, y) whose intensity if higher than or equal to threshold value i.e., a(x, y) T would be counted as a part of object, if not then the pixel is considered as a part of the background. Depending on how we choose the thresholding value, thresholding methods can be categorized into 2 groups: global and local thresholding. When the threshold

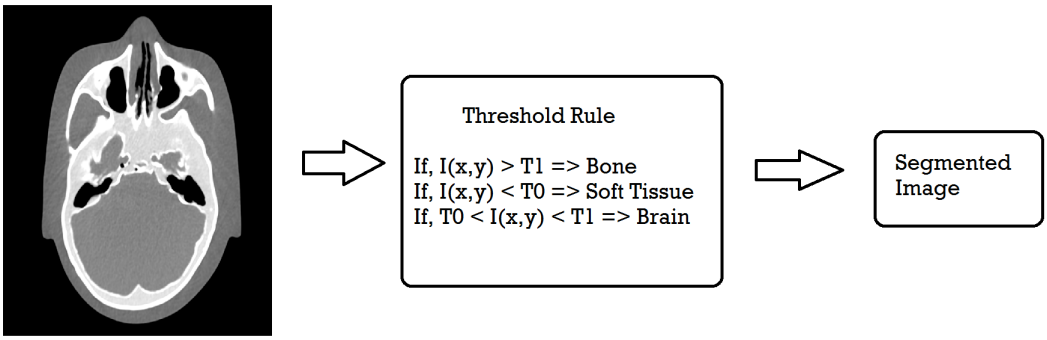

Fig. 2.2: Threshold Based Segmentation Algorithm

value T is consistent, the method is called global thresholding else it is considered as local thresholding. Primary limitation of global thresholding methods is the failure to give accurate results in case of uneven background illumination. This limitation is not visible in local thresholding due to the use of multiple thresholds. Normally, selection of thresholds is done interactively but development of automated threshold selection algorithms is also achievable. Drawback of this threshold based method is that, we can create only 2 classes so it cannot be used in segmenting multichannel images. Moreover, due to its sensitivity to noise, thresholding does not consider the spatial characteristics of an image, as both of these features lead to the corruption of the image histogram making the division more difficult.

For example, given the histogram of a 2-D medical image, for example I(x,y) , we can define a simple threshold rule to classify bone and soft tissue or a compound threshold rule to classify brain:

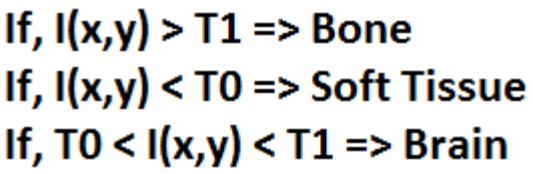

Fig. 2.3: Threshold Example

2.2.3 Independent Component Analysis

It is one of the most dexterous techniques used to isolate the independent sources blended linearly in several sensors. For example, let's consider the recording electroencephalograms (EEG) on the scalp, this technique has the ability to separate the artifacts encapsulated in the EEG data and this is possible as these components are normally independent of each other. There are numerous algorithms implementing ICA but FastICA has proved to be one of the most apt and prominent algorithm for independent component analysis invented by Aapo Hyvrinen at Helsinki University of Technology [10]. FastICA is implemented using fixed-point iteration scheme which makes use of maximizing non-Gaussianity as a criterion for checking the statistical independence. However, it is recommended to perform some preprocessing on the data before applying this ICA algorithm. Next section explains in brief about this algorithm and the preprocessing techniques, see [4] used to make the ICA estimation simpler and better conditioned.

FastICA Algorithm: Preprocessing for ICA

1. Centering: The most fundamental and essential preprocessing steps is centering $\mathbf{x}$, i.e. subtract the mean $E\{\mathbf{x}\}$ from the observed variable $\mathbf{x} = As$ so it has

zero mean. By doing so, the sources $\mathbf{s}$ also become zero mean because $E\{\mathbf{x}\} = \mathbf{A}\ E\{\mathbf{s}\} = 0$. When the mixing matrix $\mathbf{A}$ is available, $E\{\mathbf{s}\}$ can be estimated to be $\mathbf{A}^{-1}E\{\mathbf{x}\}$.

Centering is performed purely to clarify the ICA algorithms: It should not be assumed that the mean could not be approximated. After approximating the mixing matrix $\mathbf{A}$ with centered data, the approximation by adding the mean vector of $\mathbf{s}$ back to the centered estimates of $\mathbf{s}$ can be completed. The mean vector of $\mathbf{s}$ is calculated by $\mathbf{A}^{-1}\mathbf{m}$, where $\mathbf{m}$ is the mean that was subtracted in the preprocessing.

2. Whitening:

Second step in preprocessing rule of ICA is whitening the observed variables. To be more precise, before applying the ICA algorithm and after centering, the observed vector $\mathbf{x}$ is transformed linearly to attain a new vector $\tilde{\mathbf{x}}$ which is white, i.e. its components are independent and their variances equal unity. In other words, the covariance matrix of $\tilde{\mathbf{x}}$ equals the identity matrix:

$$E\{\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T\} = \mathbf{I}.$$

The whitening transformation is consistently achievable. One of the well-known whitening method is the eigen-value decomposition (EVD) of the covariance matrix $E\{\mathbf{x}\mathbf{x}^T\} = \mathbf{E}\mathbf{D}\mathbf{E}^T$, where $\mathbf{E}$ is the orthogonal matrix of eigenvectors of $E\{\mathbf{x}\mathbf{x}^T\}$ and $\mathbf{D}$ is the diagonal matrix of its eigenvalues, $\mathbf{D} = \mathrm{diag}(d_1, ..., d_n)$. Note that $E\{\mathbf{x}\mathbf{x}^T\}$can be approximated in a typical manner from the available sample $\mathbf{x}(1), ..., \mathbf{x}(T)$. Whitening can now be done by

$$\tilde{\mathbf{x}} = \mathbf{E}\mathbf{D}^{-1/2}\mathbf{E}^T\mathbf{x}$$

where the matrix $\mathbf{D}^{-1/2}$is computed by a simple component-wise operation as $\mathbf{D}^{-1/2} = \text{diag}(d_1^{-1/2}, ..., d_n^{-1/2})$. It is simple to verify now $E\{\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T\} = \mathbf{I}$. Whitening transforms the mixing matrix into a new one, $\tilde{\mathbf{A}}$:

$$\tilde{\mathbf{x}} = \mathbf{E}\mathbf{D}^{-1/2}\mathbf{E}^T\mathbf{A}\mathbf{s} = \tilde{\mathbf{A}}\mathbf{s}$$

The whitening efficacy vests in the fact that the new mixing matrix $\tilde{\mathbf{A}}$ is orthogonal. This can be seen from

$$E\{\mathbf{x}'x'^T\} = \mathbf{I} = \mathbf{A}'E\{\mathbf{s}s^T\}\mathbf{A}'^T = \mathbf{A}'A'^T$$

From the above equation, it is clear that the number of parameters to be estimated are reduced due to the whitening process. Instead of having to assess n2 parameters, i.e. elements of the original matrix $\mathbf{A}$, we only have to assess the new, orthogonal mixing matrix $\tilde{\mathbf{A}}$. An orthogonal matrix consists of $n(n-1)/2$ degrees of freedom. For example, in 2-D an orthogonal transformation is determined by a single angle parameter. In larger dimensions, an orthogonal matrix consists only of about half the number of parameters of an arbitrary matrix. Thus it can be deduced that whitening solves the problem of ICA atleast partially. Due to this method's simplicity and standard procedure which is simpler than any ICA algorithms, it is an excellent idea to reduce the complexity of the problem. It would also be a good idea to reduce the dimension of the data when we are applying whitening. Then we look at the eigenvalues dj of $E\{\mathbf{x}\mathbf{x}^T\}$ and

discard those that are too small, which is normally done in the statistical technique of principal component analysis. This also helps sometimes in reducing noise. Moreover, reducing dimensions also solves the issue of overlearning, which can sometimes be observed in ICA.

FastICA Algorithm

FastICA is based on a fixed-point iteration scheme for finding a maximum of the nongaussianity of $\mathbf{w}^T\mathbf{x}$, as measured in preprocessing [4]. This nongaussianity can also be obtained by using an approximative Newton iteration. The derivative of the nonquadratic function G is designated by g; for example the derivatives of the functions are:

$g_1(u) = \tanh(a_1 u)$, $g_2(u) = u\exp(-u^2/2)$

where $1 \leq a_1 \leq 2$ is some suitable constant, often taken as a1=1. The basic form of the FastICA algorithm is as follows:

1. Select a primary (e.g. random) weight vector $\mathbf{w}$.
2. Let $\mathbf{w}^+ = E\{\mathbf{x}g(\mathbf{w}^T\mathbf{x})\} - E\{g'(\mathbf{w}^T\mathbf{x})\}\mathbf{w}$
3. Let $\mathbf{w} = \mathbf{w}^+/\|\mathbf{w}^+\|$
4. If not converged, go back to 2.

It should be observed here that, the original and latest values of $\mathbf{w}$ point in the similar path, in case of convergence, i.e. their dot-product is (approximately) equal to 1. Taking into consideration that $\mathbf{w}$ and $-\mathbf{w}$ represent the same path, it is not compulsory for the vector to converge to an unique point,. The reason for this is the

restriction of defining the independent components only up to a multiplicative sign. The assumption of prewhitened data should also be registered here.

FastICA can be derived in following mentioned way. It should be observed here that the maxima of the approximation of the negentropy of $\mathbf{w}^T\mathbf{x}$ are acquired at definite optima of $E\{G(\mathbf{w}^T\mathbf{x})\}$. According to the Kuhn-Tucker conditions the optima of $E\{G(\mathbf{w}^T\mathbf{x})\}$ under the constraint $E\{(\mathbf{w}^T\mathbf{x})^2\} = \|\mathbf{w}\|^2 = 1$ are obtained at points where

$$E\{\mathbf{x}g(\mathbf{w}^T\mathbf{x})\} - \beta\mathbf{w} = 0$$

We will attempt to resolve this equation by Newton's method. Designating the function on the left-hand side by F, we attain its Jacobian matrix $JF(\mathbf{w})$ as

$$JF(\mathbf{w}) = E\{\mathbf{x}\mathbf{x}^T g'(\mathbf{w}^T\mathbf{x})\} - \beta\mathbf{I}$$

To clarify the inversion of this matrix, we choose to approximate the first term. Since the data is sphered, a reasonable approximation seems to be $E\{\mathbf{x}\mathbf{x}^T g'(\mathbf{w}^T\mathbf{x})\} \approx E\{\mathbf{x}\mathbf{x}^T\}E\{g'(\mathbf{w}^T\mathbf{x})\} = E\{g'(\mathbf{w}^T\mathbf{x})\}\mathbf{I}$. Thus the Jacobian matrix becomes diagonal, and can easily be inverted. Thus we acquire the following approximative Newton iteration:

$$\mathbf{w}^+ = \mathbf{w} - [E\{\mathbf{x}g(\mathbf{w}^T\mathbf{x})\} - \beta\mathbf{w}]/[E\{g'(\mathbf{w}^T\mathbf{x})\} - \beta]$$

This algorithm can be further simplified by multiplying both sides by:

$$\beta - E\{g'(\mathbf{w}^T\mathbf{x})\}$$

After algebraic simplication this gives the FastICA iteration.

To be practical, estimates should be placed in place of expectations in FastICA. The natural estimates are of course the corresponding sample means. To be optimal,

we should use all available data, but typically this is not regarded as a very neat choice because the computations may emerge as very challenging. Smaller samples can be used to estimate the averages, but its size might produce a reasonable effect on the accuracy of the final estimates. It is recommended to choose sample points individually at every iteration. If the convergence is not acceptable, the sample size can be increased.

The computational complexity required to reach a given source extraction quality is put forward as a natural objective measure of convergence speed for BSS/ICA algorithms, see [7]. This FastICA technique is acknowledged as a gradient-based algorithm with constant step size. Its speed relies essentially on prewhitening and occassionally on initialization.

3. RESULTS

3.1 Canny's Edge Detection

The primary purpose of edge detection is to significantly reduce the amount of data in an image, while maintaining the structural properties to be used for further image processing [1]. Canny Edge detection algorithm has proved to be an optimal technique to achieve this goal. As explained in previous section there are many edge detection operators out there but Canny's edge detection algorithm has proved to be providing better performance than any of them under almost all scenarios. Even under noisy conditions, Canny has exhibited better results than LoG, Prewitt, Sobel and Robert's Operator. Canny is computationally more expensive compared to these techniques but also gives excellent results in detecting the weakest edges.

The performance of the Canny algorithm relies massively on the pliable param-

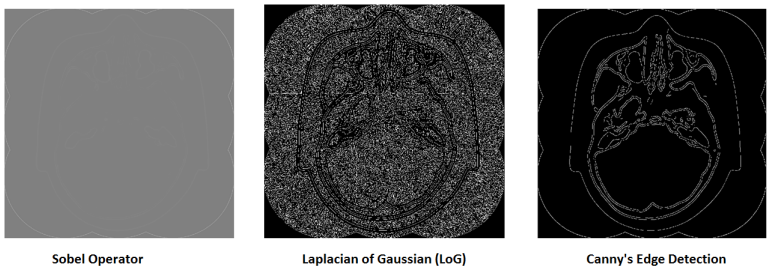

Fig. 3.1: Edge Detection Techniques Comparison - Result Images

Sobel Operator	Prewitt Operator	Laplacian of Gaussian (LoG)	Canny Edge Detection
More sensitive to the diagonal edge than to the horizontal and vertical edges	More sensitive to horizontal and vertical edges.	Determine if the pixels of image are in the dark areas or bright area of the known edge	Not susceptible to noise interference enables its ability to detect true weak edges
Computational Time: 4.9 sec		Computational Time: 2 sec	Computational Time : 4.3 sec

Fig. 3.2: Edge Detection Technique Comparison

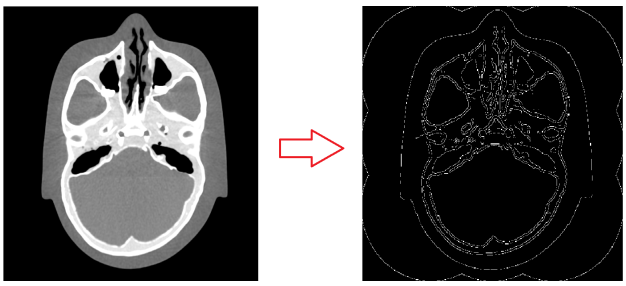

Fig. 3.3: Canny Edge Detection Result: Slice 94, Thickness 1.25mm, Computational Time: 4.3 sec

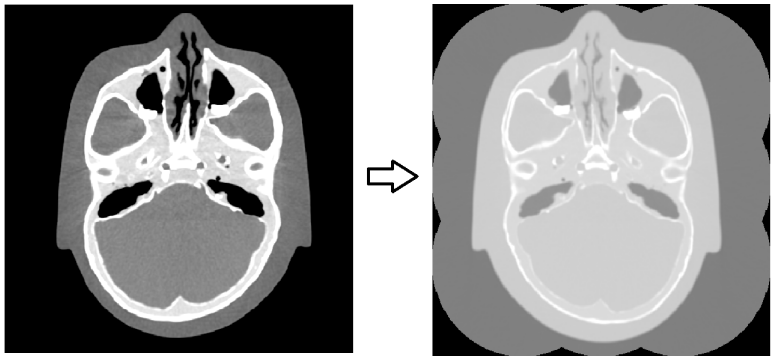

Fig. 3.4: FastICA Result: Slice 94, Thickness 1.25mm, Computational Time: 6 sec

eters, σ, which is the standard deviation for the Gaussian filter, and the threshold values, $T1$ and $T2$. σ is directly proportional to the size of the Gaussian filter, as the value for σ increases so does the Gaussian filter. Thus a user can alter the algorithm by tuning these parameters to comply to different environments. But the least computational complexity that we can achieve in Canny Edge Detection algorithm is $O(n^2)$.

3.2 Independent Component Analysis (ICA)

The computational complexity required to reach a given source extraction quality depends on the measure of convergence speed for ICA algorithms. FastICA uses fixed-point iteration scheme that has been found in independent experiments to be 10-100 times faster than conventional gradient descent methods for ICA. FastICA method's speed relies heavily on prewhitening and sometimes on initialization. Taking this into consideration, the least complexity that can be achieved using FastICA is quadratic.

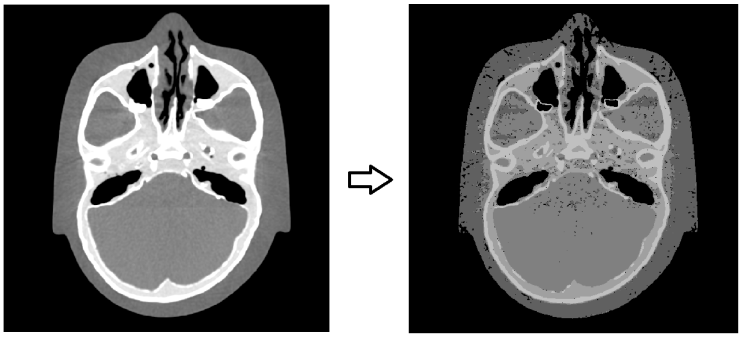

Fig. 3.5: Threshold Result: Slice 94, Thickness 1.25mm, Computational Time: 0.3 sec

3.3 Threshold based Image Segmentation

This method was implemented using pre-defined classes of thresholds which are constant for the respective materials like for e.g. outer cortical bone falls in the range of 1500-2220 HU (Hounsfield unit). With the initial thresholding, I was able to get clear enough image, where all regions and edges were easily separable. In these images, there were some regions which had minute patches of different materials. Some patches were big and some were as small as one pixel. The initial task was to remove the single pixel/salt and pepper non-uniformities, at the end of which I was able to get better, cleaner picture. Below are the results for the same:

Due to the unique ability of canny edge detection method to detect weak edges and low computational time, it was used as a combination with Threshold based method. Due to the this hybrid technique, I was able to clean-up some more minute patches present near the nose and sinus regions. Also the inner cortical bone edges became more continuous and clear. Below is the result of this hybrid technique:

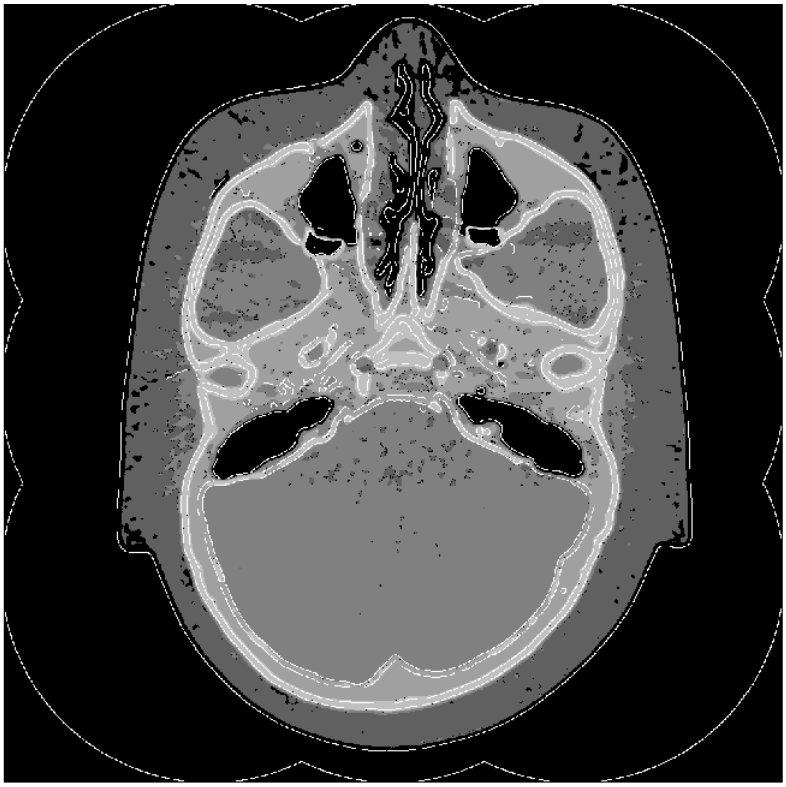

Fig. 3.6: Canny + Threshold Result: Slice 94, Thickness 1.25mm, Computational Time: 4.6 sec

4. FUTURE WORK

With current implemented methods, there are still some minute artifacts present in few regions which don't belong to the respective materials in that area. For. example there are small patches of brain being displayed near the soft tissue, which is totally incorrect as there is no soft tissue surrounding the brain but just the inner cortical bone. Due to these minute patches, regions are showing some nonuniformities. In the final thresholded image, I was able to remove the salt and pepper noise which helped to clean-up the single pixel non-uniformities. The future tasks here would be to clean-up the bigger patches so we can establish solid regions and extend the incomplete edges.

One of the methods to achieve this goal would be traversing and backtracking algorithm. To give a brief explaination, say you want to create the brain as one solid region so you have no pixels inside it belonging to any different material or region. Start by looking at the 4 neighbors of the current location/pixel, if you find a pixel that belongs to the brain, jump there and make it your new position to check in 4-neighborhood. If you don't find a pixel that belongs to the region, return and start searching again from your previous location. Thus there are 3 steps to this algorithm: check, jump, and return. To implement the checking part, we can follow a pre-specified order to verify the 4-neighborhood. Say the order is right, down, left and

top. First check right, if it is the material you are searching for, jump to that location, if not check down and so on. To return back to your previous location, you need to keep track of the location, wherein Backtracking comes into picture. Before you jump to the next location, you need to push the location of the pixel/square you are in and about to leave so you can return. You also need to keep track of the locations you have already traversed so that you don't get stuck in a loop. For this you can create a map where you can create chalk marks on the locations you have already traversed to. For return, if you complete all four checks and there is no where to go, you read the last entry on the stack (pop it) and set it to the current coordinates. To implement backtracking, stack is the best way to go but Matlab doesn't have one. In this case one of the options is to import the java library $java.util.*$ and create a stack object as $stack = Stack$, which has in-built push-pop functions and can be used directly.

APPENDIX A

CODE FOR THE IMAGE SEGMENTATION TECHNIQUES

Canny Edge Detection

```
A = dicomread('DICOM Image for Edge Detection');

tic;
% Canny edge detection for A
[C1, Ct1] = edge(A,'canny',[],1.0);

% Recompute lowering both automatically computed
% thresholds by fraction k
k = 0.75;
C1 = edge(A,'canny',k*Ct1,1.0);

canny = figure('Name','Canny');
iptsetpref('ImshowBorder','tight');
imshow(C1,'InitialMagnification',100);
dicomwrite(C1,'Name of dcm file you want to create');
toc;
```

Threshold Based Image Segmentation

```
function picOut = Thresholds(picture)
corticalthresh1 =1500;
corticalthresh2=2220;
incorticalthresh1 = 1300;
incorticalthresh2=1500;
trabecularthresh1 = 1200;
trabecularthresh2= 1300;
brainthresh1 = 1060;
brainthresh2= 1200;
softtissuethresh1 = 1030;
softtissuethresh2= 1060;
sinusthresh1= 200;
sinusthresh2= 700;

tic;

corticalLayer = picture > corticalthresh1
                & picture < corticalthresh2;
innercortical = picture > incorticalthresh1
                & picture < incorticalthresh2;
trabecularLayer = picture > trabecularthresh1
                  & picture < trabecularthresh2;
brainLayer = picture > brainthresh1
             & picture < brainthresh2;
softtissueLayer = picture > softtissuethresh1
                  & picture < softtissuethresh2;
sinus = picture < sinusthresh2
        & picture > sinusthresh1;

layer1 = corticalLayer * 192;
layer2 = innercortical * 192;
layer3 = trabecularLayer * 160;
layer4 = brainLayer * 128;
layer5 = softtissueLayer * 96;
layer6 = sinus * 0;

allLayers=layer6+layer5+layer3+layer4+layer2+layer1;

figure;imshow(uint8(allLayers));
toc;
picOut=uint8(allLayers);
```

Hybrid: Canny Edge Detection + Threshold

```
function picOut = Hybrid_CannyEdge_Thresholds(picture)
corticalthresh1 =1500;
corticalthresh2=2220;
incorticalthresh1 = 1300;
incorticalthresh2=1500;
trabecularthresh1 = 1200;
trabecularthresh2= 1300;
brainthresh1 = 1060;
brainthresh2= 1200;
softtissuethresh1 = 1030;
softtissuethresh2= 1060;
sinusthresh1= 200;
sinusthresh2= 700;

tic;

corticalLayer = picture > corticalthresh1
                & picture < corticalthresh2;
innercortical = picture > incorticalthresh1
                & picture < incorticalthresh2;
trabecularLayer = picture > trabecularthresh1
                  & picture < trabecularthresh2;
brainLayer = picture > brainthresh1
             & picture < brainthresh2;
softtissueLayer = picture > softtissuethresh1
                  & picture < softtissuethresh2;
sinus = picture < sinusthresh2
        & picture > sinusthresh1;

layer1 = corticalLayer * 192;
layer2 = innercortical * 192;
layer3 = trabecularLayer * 160;
layer4 = brainLayer * 128;
layer5 = softtissueLayer * 96;
layer6 = sinus * 0;
layer7='Edge Detected Image';

allLayers=layer6+layer5+layer3+layer4+layer2+layer1;

figure;imshow(uint8(allLayers));
toc;
picOut=uint8(allLayers);
```

REFERENCES

[1] Canny edge detection. http://www.cse.iitd.ernet.in/~pkalra/csl783/canny.pdf.

[2] Fastica package for matlab. http://research.ics.aalto.fi/ica/fastica/.

[3] Image segmentation. http://en.wikipedia.org/wiki/Image_segmentation.

[4] Independent component analysis: A tutorial. http://cis.legacy.ics.tkk.fi/aapo/papers/IJCNN99_tutorialweb.

[5] M.D. Altschuler and Y. Censor. Feasibility solutions in radiation therapy treatment planning. In *Proceedings of the Eighth International Conference on the Use of Computers in Radiation Therapy*, 1984.

[6] John Canny. A computational approach to edge detection. In *Pattern Analysis and Machine Intelligence, IEEE Transactions*, 1986.

[7] Mike E. Davies. *Independent Component Analysis and Signal Separation*. Springer, 2007.

[8] Hai-jun Sun Dong Chen Xiu-Ling Liu Hong-Rui Wang, Jian-li Yang. An improved region growing method for medical image selection and evaluation based

on canny edge detection. In *Management and Service Science (MASS), 2011 International Conference*, 2011.

[9] Xian-Chuan Yu Hui He, Ting Zhang and Wang-Lu Peng. Application of fast independent component analysis on extracting the information of remote sensing imagery. In *Machine Learning and Cybernetics, 2006 International Conference*, 2006.

[10] Aapo Hyvarinen and Erkki Oja. Independent component analysis: Algorithms and applications. *Neural Networks Research Centre, Helsinki University of Technology*, 2012.

[11] The MathWorks Inc. Image processing toolbox for matlab, 2012.

[12] Xiu Sun and Xiaoxiao Wang. Study of edge detection algorithms for lung ct image on the basis of matlab. In *Control and Decision Conference (CCDC), 2011 Chinese*, 2011.

[13] Chen Xiao-juan and Li Dan. Medical image segmentation based threshold svm. In *Biomedical Engineering and Computer Science (ICBECS), 2010 International Conference*, 2010.

[14] Wu Bingrong Xie Mei, Zhen Zheng and Li Guo. The edge detection of brain tumor. In *Communications, Circuits and Systems, 2009. ICCCAS 2009. International Conference*, 2009.

[15] M.D. Altschuler Y. Censor and W.D. Powlis. A computational solution of the inverse problem in radiation-therapy treatment planning. 1988.

[16] M.D. Altschuler Y. Censor and W.D. Powlis. On the use of cimmino's simultaneous projections method for computing a solution of the inverse problem in radiation therapy treatment planning. 1988.

[17] W.D. Powlis Y. Censor and M.D. Altschuler. On the fully discretized model for the inverse problem of radiation therapy treatment planning. In *Proceedings of the Thirteenth Annual Northeast Bioengineering Conference*, 1987.