

**3-DIMENSIONAL SEGMENTATION OF
MESORECTAL VOLUME FOR ASSESSMENT
OF COLORECTAL SURGERY**

SOH RONG'EN (A0002506Y)

BSc, Physics

A THESIS SUBMITTED FOR THE DEGREE OF MASTER OF
COMPUTING

DEPARTMENT OF COMPUTER SCIENCE
NATIONAL UNIVERSITY OF SINGAPORE

2018

Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Soh Rong'en (A0002506Y)

February 2018

Acknowledgments

I would like to thank my supervisor A/Prof. Leow Wee Kheng for his invaluable advice and support. I am also very grateful to Mr. Xie Shudong for his continuous mentorship and input on mesh deformation. In addition, I would like to thank Dr. Chong Choon Seng and Dr. Lim Tian Zhi, from the National University Hospital, for their advice.

Abstract

Due to the prevalence and severity of colorectal cancer, the segmentation of the mesorectum in MRI scans, to better determine treatment options and patient outcomes for colorectal cancer patients, is of particular interest.

However, medical MRI images are prone to significant variation and thus difficult to segment. Manual segmentation has issues with intra-operator and inter-operator reproducibility, and is so tedious and time-consuming to perform that is impossible to practice routinely in hospitals. Thus, ground truth is not available, so accuracy cannot be used as a measurement of performance. Alternative metrics will be used instead. In this study, a semi-automated algorithm that utilizes mesh deformation is proposed, and has been found to be robust, consistent and efficient.

This work has the potential to augment the work of medical staff in analyzing MRI scans, leading to better patient outcomes. Finally, the algorithm would enable other medical researchers easily measure mesorectum volumes and conduct further research in this area.

Keywords: segmentation, mesh deformation, MRI, mesorectum, medical, colorectal cancer.

Contents

List of Figures	vii
List of Tables	xi
1 Introduction	1
1.1 Motivation	1
1.2 Thesis Objective	3
2 Related Work	5
2.1 Image Segmentation	5
2.2 Laplacian Deformation	8
2.3 Thin Plate Spline	10
2.4 Choice of Segmentation Techniques	12
3 Proposed Algorithm	13
3.1 Proposed Algorithm Overview	13
3.2 Algorithm Step 1: Initialization	13
3.3 Algorithm Step 2: Finding landmarks	16
3.4 Algorithm Step 3: Mesh Deformation	22
3.5 Algorithm Step 4: Overlay of Mesh on MRI Slices	24
3.6 Volume of MRI Slice	25
4 Experiments	34
4.1 Overview	34
4.2 Experiment 1: Robustness	35
4.3 Experiment 2: Consistency	42

Contents	vi
4.4 Experiment 3: Efficiency	42
4.5 Summary	45
5 Conclusion & Future Work	47
5.1 Limitations	47
5.2 Conclusion	48
References	50

List of Figures

2.1	MRI scan of mesorectum. Yellow: segmentation of 2D slices stacked together.	8
3.1	Anatomical planes: axial plane (green), coronal plane (blue), median plane (red) and sagittal plane (yellow).	14
3.2	MRI slice, axial view. White arrows point to the mesorectum boundary. Mesorectum shape resembles a square.	15
3.3	MRI slice, axial view. White arrows point to the mesorectum boundary. Mesorectum shape resembles an ellipse.	15
3.4	MRI slice, axial view. Example of landmarks (represented by yellow dots) selected by the operator, in the case of $N = 4$.	16
3.5	Searching process (not to scale). With $N = 4$, blue diamonds at the corners represent landmarks, while the red circle at the center is the global centroid. The green bar represents the intensity profile measured along the bar, and the length of the bar corresponds to the window size. The size of the search space is parameterized by the angle ϕ .	18
3.6	Example of intensity profile along line connecting global centroid to landmark. The mesorectum boundary corresponds to one of the peaks, and the window size parameterizes the length of the intensity profile used as the ground truth.	19
3.7	Set of all points that were found by the algorithm within the search space.	21
3.8	Mesh deformed via Laplacian deformation.	23
3.9	Mesh deformed via Thin Plate Spline.	24
3.10	MRI slice #25, axial view. Overlaid with mesh deformed via TPS.	26

3.11 MRI slice #25, axial view. Overlaid with mesh deformed via Laplacian deformation.	26
3.12 MRI slice #24, axial view. Overlaid with mesh deformed via TPS.	27
3.13 MRI slice #24, axial view. Overlaid with mesh deformed via Laplacian deformation.	27
3.14 MRI slice #23, axial view. Overlaid with mesh deformed via TPS.	27
3.15 MRI slice #23, axial view. Overlaid with mesh deformed via Laplacian deformation.	27
3.16 MRI slice #22, axial view. Overlaid with mesh deformed via TPS.	28
3.17 MRI slice #22, axial view. Overlaid with mesh deformed via Laplacian deformation.	28
3.18 MRI slice #21, axial view. Overlaid with mesh deformed via TPS.	28
3.19 MRI slice #21, axial view. Overlaid with mesh deformed via Laplacian deformation.	28
3.20 MRI slice #20, axial view. Overlaid with mesh deformed via TPS.	29
3.21 MRI slice #20, axial view. Overlaid with mesh deformed via Laplacian deformation.	29
3.22 MRI slice #19, axial view. Overlaid with mesh deformed via TPS.	29
3.23 MRI slice #19, axial view. Overlaid with mesh deformed via Laplacian deformation.	29
3.24 MRI slice #18, axial view. Overlaid with mesh deformed via TPS.	30
3.25 MRI slice #18, axial view. Overlaid with mesh deformed via Laplacian deformation.	30
3.26 MRI slice #17, axial view. Overlaid with mesh deformed via TPS.	30
3.27 MRI slice #17, axial view. Overlaid with mesh deformed via Laplacian deformation.	30
3.28 MRI slice #16, axial view. Overlaid with mesh deformed via TPS.	31

3.29	MRI slice #16, axial view. Overlaid with mesh deformed via Laplacian deformation.	31
3.30	MRI slice #15, axial view. Overlaid with mesh deformed via TPS.	31
3.31	MRI slice #15, axial view. Overlaid with mesh deformed via Laplacian deformation.	31
3.32	MRI slice #14, axial view. Overlaid with mesh deformed via TPS.	32
3.33	MRI slice #14, axial view. Overlaid with mesh deformed via Laplacian deformation.	32
4.1	Plot of mesorectum volume (cm^3) against number of input landmarks per slice, N . Solid red line refers to volume when Laplacian deformation was used; dashed blue line refers to volume when TPS was used.	36
4.2	MRI slice #14, axial view. Overlaid with mesh deformed via Laplacian deformation, for $N = 5$.	37
4.3	MRI slice #14, axial view. Overlaid with mesh deformed via Laplacian deformation, for $N = 6$.	37
4.4	MRI slice #14, axial view. Overlaid with mesh deformed via Laplacian deformation, for $N = 7$.	38
4.5	MRI slice #14, axial view. Overlaid with mesh deformed via Laplacian deformation, for $N = 8$.	38
4.6	MRI slice #20, axial view. Overlaid with mesh deformed via Laplacian deformation, for $N = 5$.	38
4.7	MRI slice #20, axial view. Overlaid with mesh deformed via Laplacian deformation, for $N = 6$.	38
4.8	MRI slice #20, axial view. Overlaid with mesh deformed via Laplacian deformation, for $N = 7$.	39
4.9	MRI slice #20, axial view. Overlaid with mesh deformed via Laplacian deformation, for $N = 8$.	39

4.10 MRI slice #25, axial view. Overlaid with mesh deformed via Laplacian deformation, for $N = 5$.	39
4.11 MRI slice #25, axial view. Overlaid with mesh deformed via Laplacian deformation, for $N = 6$.	39
4.12 MRI slice #25, axial view. Overlaid with mesh deformed via Laplacian deformation, for $N = 7$.	40
4.13 MRI slice #25, axial view. Overlaid with mesh deformed via Laplacian deformation, for $N = 8$.	40

List of Tables

3.1	Possible cases of mesh vertices positioning relative to plane of intersection.	25
4.1	Computed volume of MRI slices for varying N (Mean \pm standard deviation).	35
4.2	p -values for pairwise comparison of differences in mesorectum volume across consecutive N .	36
4.3	p -values for comparison of differences in mesorectum volume across Laplacian deformation and TPS.	41
4.4	Percentage standard deviation in mesorectum volume measurements.	42
4.5	Computational time taken for proposed algorithm (Mean \pm standard deviation).	45

Introduction

1.1 Motivation

In Singapore, cancer is the top cause of death. In 2015, cancer accounted for almost 30% of deaths [oDO17]. Among the various types of cancer (e.g. brain, lung, bone cancer), colorectal cancer accounts for the most commonly diagnosed type of cancer in Singapore. Between 2011 and 2015, 9,807 new cases of colorectal cancer were diagnosed, and in the same period of time, 3,906 people died from colorectal cancer. Moreover, colorectal cancer, when measured against breast cancer and cervical cancer, tends to be detected at later stages, resulting in lower survival rates. Statistics from the National Registry of Diseases Office put the percentage of colorectal cancer cases detected at late-stage (i.e. stages III & IV) at 57.6% (male) 59.7% (female), while only 28.9% for breast cancer and 33.4% for cervical cancer. Thus, colorectal cancer is an important and urgent area where advances in technology can prove helpful.

The main treatment for colorectal cancer is total mesorectal excision (TME), where the whole mesorectum is surgically removed [HHR82]. The mesorectum refers to the region around the rectum, and is filled with vascular structures, lymphatic and blood vessels, nerves, lymph nodes and fatty tissue [THL⁺07]. Additionally, there may be chemotherapy or radiotherapy involved, depending on the case.

There have been several studies that investigated the significance of the mesorectum

volume and its possible correlation with factors that impact patient outcomes, such as [TB05, AGB07, THL⁺07, IOH⁺13, TRS⁺15]. [IOH⁺13] showed the significance of the mesorectum volume as a predictor of surgical difficulty in keyhole surgery for rectal cancer. This is clinically important because knowledge of the mesorectum volume can aid the surgeon in deciding which type of surgery (keyhole or open) is best for the patient. Additionally, [AGB07] showed that mesorectum volume does not correlate with body physique and body mass index (BMI), suggesting that mesorectum size is largely genetic, rather than environmentally influenced. This has implications for patient care, because some doctors may recommend a high-calorie diet prior to surgery on the assumption that doing so would increase the mesorectum volume and thus ensure that the tumor would be completely within the mesorectum; but this assumption may not be true, and a high-calorie diet might have introduced complications for no gain. Going further, [AGB07] suggested the mesorectum volume might be a predictor for a positive surgical margin, which refers to the margin of the excised tissue containing cancer cells. The ideal situation for a patient is for a negative surgical margin, where the margin of the removed tissue contains no cancer cells, giving confidence that all cancer cells had been removed. In addition, [TRS⁺15] found that patients with a greater mesorectum volume had a lower risk of a relapse occurring and a higher mortality risk than patients with a lesser mesorectum volume, although neither of these findings were statistically significant. [TRS⁺15] also called for a larger study to establish the significance of the mesorectal volume. These studies clearly show the importance of the mesorectum volume.

All of the studies cited above used manual segmentation to segment the mesorectum, where an expert is called upon to manually segment the mesorectum. However, manual segmentation has several downsides, which include issues with reproducibility, intra-operator and inter-operator variability and operator fatigue [MT96]. Intra-operator variability occurs when the same operator measures the mesorectum volume differently

across different MRI scans. In contrast, inter-operator variability occurs when different operators, who may differ also in years of experience and judgment, segment the mesorectum differently. Above all, manual segmentation is so tedious and time-consuming that it is impossible to practice routinely in hospitals. There are also no practical alternative methods to obtain the mesorectum volume.

Hence, this thesis explores a semi-automated segmentation technique, which can incorporate domain knowledge and computational efficiency, in order to achieve segmentation results that are reliable and useful. This semi-automated technique can augment doctors' and medical staffs' capability to measure the volume of the mesorectum quickly and reliably. Moreover, by making measurements of the mesorectum volume easy and quick, this greatly increases the ease of conducting further research into the impact of mesorectum volume, which in turn has clinical relevance and impact to patients' quality of life.

As discussed above, it is very tedious and time-consuming to manually segment the mesorectum. Although the treatment for colorectal cancer involves the removal of the entire mesorectum, such that the mesorectum (after removal) can be measured, the volume measured *in vitro* (inside the human body) and *ex vivo* (outside the human body) will be different as the mesorectum is largely composed of soft tissues that are influenced by the surrounding structure when inside the body. Consequently, ground truth is not available in this study, so accuracy cannot be used to measure the performance of the semi-automated segmentation method. Instead, other measurements will be discussed in Chapter 4.1 to investigate the performance of the method.

1.2 Thesis Objective

This paper proposes a robust semi-automated algorithm for segmenting the mesorectum in a quick and reliable manner, with the ultimate goals of providing doctors and medical

staff with more consistent and readily available information at a faster pace to make better surgical decisions, thereby minimising colorectal surgery risks and improving colorectal cancer patient outcomes.

CHAPTER 2

Related Work

2.1 Image Segmentation

There are many ways to perform image segmentation, and can be classified into three main categories: manual, semi-automated (also known as interactive) and automated. Manual segmentation, as the name suggests, refers to segmentation done solely by the manual efforts of an expert. For example, given a 3D MRI scan, a skilled operator (e.g. radiologist) would use some software to view the MRI slice by slice, i.e. in 2D. For each slice, if the object of interest is present, the operator uses the software to draw a closed polygon to demarcate the boundary of this object. This method has several limitations, including difficulty in obtaining reproducible measurements, is time-consuming, prone to intra-operator and inter-operator variability, and leads to operator fatigue [MT96]. Conversely, automated algorithms work without any human input, but there has not been much success, due to many reasons, including the significant variability in organs and soft tissue shapes across specimens. Semi-automated methods enjoy the most success, because they combine human input and efficiency of algorithms to speed up segmentation in an accurate manner. For example, a operator might be prompted to initialize certain parameters, and tweak these parameters if the output is not as expected. In this section, semi-automated methods will be discussed as the proposed algorithm is of this kind.

One basic method of image segmentation is thresholding, also known as binarization, which works on the assumption that different objects, or structures, have different intensities [SSW88]. Usually, this is done in a semi-automated fashion, where the operator experiments with the threshold value and visually inspects the output, although fully automated methods are possible.

Another common technique in image segmentation is region growing, where the operator places one or more seeds in a region, and the region is grown until certain criteria are met (e.g. when it reaches an edge)[PXP00]. This method is commonly used in conjunction with other methods such as thresholding, to segment basic structures (e.g. tumors). However, region growing is not particularly robust, where noise present in an image can cause the region to have “holes”.

Deformable models are another class of techniques that has demonstrated success in image segmentation is that of [MT96]. Deformable models are highly successful at segmentation because they are able to utilize constraints from the image, as well as incorporate prior information about the structure. This prior information may take the form of initial conditions or values, constraints on model parameters, data constraints or implemented as part of the model fitting procedure. By doing so, such models can handle the large variation in medical images when segmenting structures of interest.

One well-known deformable model is the snake [KWT88], which is a type of active contour model that is able to localize lines and edges near its initialization. This is achieved by using an energy minimization framework, premised on the assumption that boundaries are piecewise continuous or smooth. The snake is typically initialized in a semi-automated fashion where an operator places it near the object boundary to be delineated, although it is also possible for the snake to be placed in an automatic fashion if there exists a reliable way to do so. The snake deforms based on internal and external forces, where the internal forces impose a piecewise smoothness constraint

while external forces on the snake push it towards the object (e.g. boundary, line) of interest.

Deformable models have been successfully applied to medical MRI and computed tomography (CT) images, for example, to the segmentation of the inner wall of the left ventricle [BCA98], reconstruction of the cerebral cortex [DB96], and segmentation of spinal vertebra [NL98]. In one of the early studies of deformable models, Miller [MBL⁺91] initialized an icosahedron (as an approximation to a sphere) and geometrically deformed this model until the model surface matched the object surface in 3D CT data. The deformation was, as usual, formulated by the minimization of a cost function.

Many segmentation techniques have been successfully applied to 2-dimensional images. However, when the object of interest is 3-dimensional, if these 2-dimensional techniques were applied to each slice of this object, there necessitates a tedious process after segmentation in order to reconstruct the 3-dimensional boundary. During this reconstruction, there may be inconsistencies in the output. An example of this is shown in Fig. 2.1, where consecutive slices of an MRI scan were manually segmented using the *Horos* software. It can be observed that when all the segmented images are stacked together, the overall boundary is not very smooth. Thus, it is preferable to use a 3-dimensional deformable model to construct a smoother, more accurate, and more reliable segmentation.

Furthering the concept of model deformation, [Fen10] proposed using a 3D quadrilateral mesh to segment an object in the context of medical image segmentation. By evolving the mesh to register at the target object boundary, segmentation could be achieved with higher accuracy and efficient manner than other common segmentation approaches such as snakes.

In the next two sections, two methods of model deformation, namely Laplacian

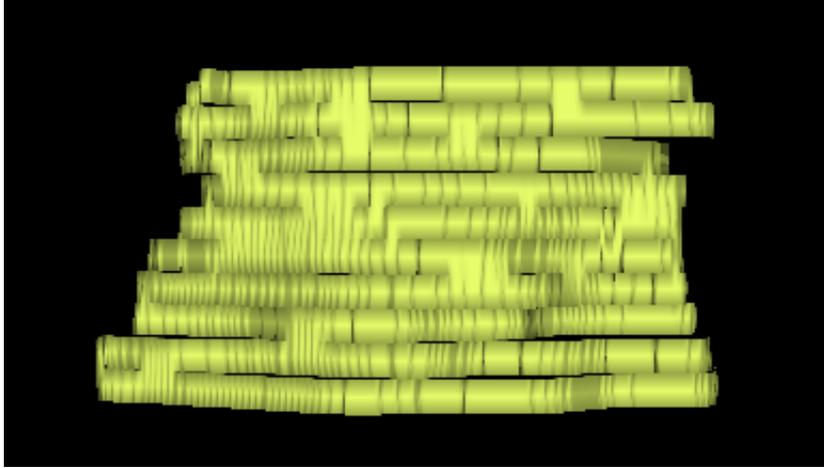


Figure 2.1: MRI scan of mesorectum. Yellow: segmentation of 2D slices stacked together.

deformation and Thin Plate Spline, will be covered in greater detail, as these will be used in the proposed algorithm subsequently.

2.2 Laplacian Deformation

Another technique based on deformable models is that of Laplacian deformation [SCOL⁺04], which is a non-rigid deformation that allows for a better fit. The *source* model is an initial model such as a sphere, that is deformed towards a *target*, or desired, output. In the context of 3D medical imaging, it can be envisioned as deforming a sphere to fit the boundary of an organ. As with other deformable models, Laplacian deformation is also a minimization problem, with two terms: the fitting term E_f which refers to the distance between the deformed source and target, and the distortion term E_d which minimizes the distortion to the source shape. This method has been shown to be effective and highly useful, which can be adapted for use in medical image segmentation. The Laplacian deformation works as follows: we denote the n number of source points as p_1, p_2, \dots, p_n , and the m number of target points as q_1, q_2, \dots, q_m . Then the Laplacian deformation becomes a question of finding all p'_i such that Eq. 2.1

is minimized, where p'_i are the source points after deformation.

$$E = E_f + E_d \quad (2.1)$$

The fitting term can be expressed as:

$$E_f = \sum_{i=1}^m \|p'_i - q_i\|^2 \quad (2.2)$$

Recall that the Laplacian operator L is

$$L(p_i) = p_i - \frac{1}{|N_i|} \sum_{j \in N_i} p_j \quad (2.3)$$

where N_i is set of neighboring vertices of p_i . Then the distortion term can be expressed as

$$E_d = \sum_{i=1}^n \|L(p'_i) - \delta_i\|^2 \quad (2.4)$$

where δ_i is the Laplacian at p_i before deformation.

Putting Eqs. 2.2 and 2.4 together yields

$$E = \sum_{i=1}^m \|p'_i - q_i\|^2 + \sum_{i=1}^n \|L(p'_i) - \delta_i\|^2 \quad (2.5)$$

which is of the form

$$\min_x \|\mathbf{Ax} - \mathbf{b}\|^2 \text{ subject to } \mathbf{Cx} = \mathbf{d} \quad (2.6)$$

Eq. 2.6 can be interpreted where \mathbf{A} is the $3n \times 3n$ matrix of Laplacian constraints, \mathbf{x} is a $3n \times 1$ vector that we want to solve for and has been sorted such that the top $3m$

elements are those with positional constraints, and \mathbf{b} is the $3n \times 1$ vector containing δ_i , \mathbf{C} a $3m \times 3n$ matrix of positional constraints, and \mathbf{d} is a $3m \times 1$ vector of target positions. This is an equality constrained least squares problem that can be solved exactly by QR factorization, with the solution:

$$\mathbf{x} = \begin{pmatrix} \mathbf{d} \\ \mathbf{v} \end{pmatrix} \quad (2.7)$$

where

$$\mathbf{v} = (\mathbf{A}_2^T \mathbf{A}_2)^{-1} \mathbf{A}_2^T (\mathbf{b} - \mathbf{A}_1 \mathbf{d}) \quad (2.8)$$

where \mathbf{A}_1 and \mathbf{A}_2 are simply derived by splitting the matrix \mathbf{A} into a $3n \times 3m$ matrix and a $3n \times 3(n - m)$ matrix.

2.3 Thin Plate Spline

The Thin Plate Spline is another method that describes deformations by using the thin plate spline as an interpolant [Boo89]. TPS is similar to Laplacian deformation where both source and target points are required to be specified. In contrast, there exists a closed-form solution for TPS, which entails solving a linear system of equations. TPS can be formulated as follows: given source points p_1, p_2, \dots, p_n and target points q_1, q_2, \dots, q_n , it outputs a deformation function $f(p)$ for every point p . TPS is also a minimization problem, as shown in Eq. 2.9 where the fitting term E_f carries the same meaning as in Laplacian deformation, but now E_d refers to the distortion of space, and there is an additional λ term referring to the degree of non-rigid warping allowed.

$$E = E_f + \lambda E_d \quad (2.9)$$

The fitting term can be expressed as:

$$E_f = \sum_{i=1}^n \|f(p_i) - q_i\|^2 \quad (2.10)$$

while the distortion term E_d for a 3D thin metal sheet can be expressed as:

$$E_d = \int \int \int \left(\frac{\partial^2 f}{\partial x^2} \right)^2 + \left(\frac{\partial^2 f}{\partial y^2} \right)^2 + \left(\frac{\partial^2 f}{\partial z^2} \right)^2 + 2 \left(\frac{\partial^2 f}{\partial xy} \right)^2 + 2 \left(\frac{\partial^2 f}{\partial xz} \right)^2 + 2 \left(\frac{\partial^2 f}{\partial yz} \right)^2 dx dy dz \quad (2.11)$$

Minimization of Eq. 2.11 takes the form:

$$f(\tilde{\mathbf{p}}) = \mathbf{a}^T \cdot \tilde{\mathbf{p}} + \sum_{i=1}^n \phi(\|\mathbf{p} - p_i\|) w_i \quad (2.12)$$

where $\tilde{\mathbf{p}}$ are the homogeneous coordinates of source points, $\phi(r)$ is some function that increases with r , for example, $r^2 \log(r)$, \mathbf{a} is an affine transformation matrix, and w_i the weights.

For TPS, the aim is to solve for \mathbf{w} and \mathbf{a} in Eq. 2.13, given source points \mathbf{p} and target points \mathbf{v} :

$$\begin{pmatrix} \mathbf{w} \\ \mathbf{a} \end{pmatrix} = \begin{pmatrix} \mathbf{K} & \mathbf{P} \\ \mathbf{P}^T & \mathbf{0} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{v} \\ \mathbf{0} \end{pmatrix} \quad (2.13)$$

where \mathbf{K} is an $n \times n$ matrix with entries $K_{ij} = U(\|p_i - p_j\|)$, and of course its diagonal entries are 0; and \mathbf{P} is an $n \times (d + 1)$ matrix of source points in homogeneous coordinates.

2.4 Choice of Segmentation Techniques

Based on literature, snakes and region-growing techniques were considered during preliminary studies, and experiments using these techniques were carried out using the software *ITK-SNAP*. However, the results from these preliminary studies were not promising, due mainly to two reasons. First, the MRI images are highly noisy; and second, the mesorectum boundary is very faint and on many occasions, not visible at all. Consequently, of the techniques mentioned in this chapter, Laplacian deformation and TPS deformation were chosen to be used in the proposed algorithm.

Proposed Algorithm

3.1 Proposed Algorithm Overview

There are 4 main steps in the algorithm. The first is initialization of data and parameters, including operator input of initial landmarks. The second step is for the algorithm to search within those MRI slices with landmarks, for more points that delineate the mesorectum boundary. The third step is to utilize the initial landmarks, along with the points found along the mesorectum boundary, to deform an initial spherical mesh. For this third step, two different techniques are attempted: Laplacian deformation and TPS. In the fourth and final step, the mesh, after deformation, is overlaid onto the MRI slices to delineate the boundary of the mesorectum. Then, the mesorectum volume for each slice is calculated, which are summed up to give the total mesorectum volume.

3.2 Algorithm Step 1: Initialization

The full 3D MRI comes in three views: axial, coronal and sagittal. Each view provides information along an orthogonal axis, and these are illustrated in Fig. 3.1 [Ric14]. To visualize this, we use a human standing up as the frame of reference. The axial view shows MRI slices of a person standing up as planes parallel to the ground, while the

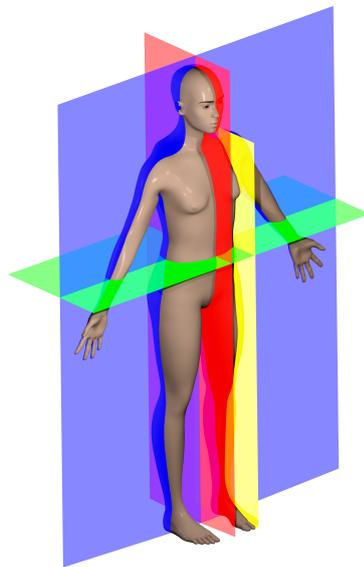


Figure 3.1: Anatomical planes: axial plane (green), coronal plane (blue), median plane (red) and sagittal plane (yellow).

coronal view shows MRI slices of a person with the planes running from the front to back, and the sagittal view runs across from the left to the right.

For the purposes of segmenting the mesorectum, anonymized axial MRI images of the mesorectum were loaded into *Mathematica* version 11.1.1.0, and dimensions of these MRI slices were recorded in both pixels (512 by 512) and millimeters (180 by 180). Both units are carefully distinguished because subsequent operations are performed on the images, but the final output (e.g. MRI volume) should correspond to physical reality.

As part of the initialization, the operator specifies three slices corresponding to the top, middle and bottom of the mesorectum. This has multiple functions: (1) specification of the ends of the mesorectum constrains the ends of the mesh that will be deformed to fill the volume; and (2) it gives a better fit because the shape of the mesorectum varies significantly along the z -axis. For example, Figs. 3.2 and 3.3 illustrates the variation in shape of the mesorectum from squarish to an ellipse respectively. Another



Figure 3.2: MRI slice, axial view. White arrows point to the mesorectum boundary. Mesorectum shape resembles a square.



Figure 3.3: MRI slice, axial view. White arrows point to the mesorectum boundary. Mesorectum shape resembles an ellipse.

MRI slice might show the mesorectum being roughly triangular too.

Then, for each MRI slice chosen (i.e. top, middle and bottom), the operator selects N number of points along the mesorectum boundary, which are called “landmarks”, by clicking on the image as shown in Fig. 3.4. N is held constant across slices for a chosen value of N . For a given slice, these landmarks form a set of x-y coordinates forming the set S_i :

$$S_i = \{\{x_{i,1}, y_{i,1}\}, \{x_{i,2}, y_{i,2}\}, \dots, \{x_{i,N}, y_{i,N}\}\} \quad \text{for } i = 1, 2, 3 \quad (3.1)$$

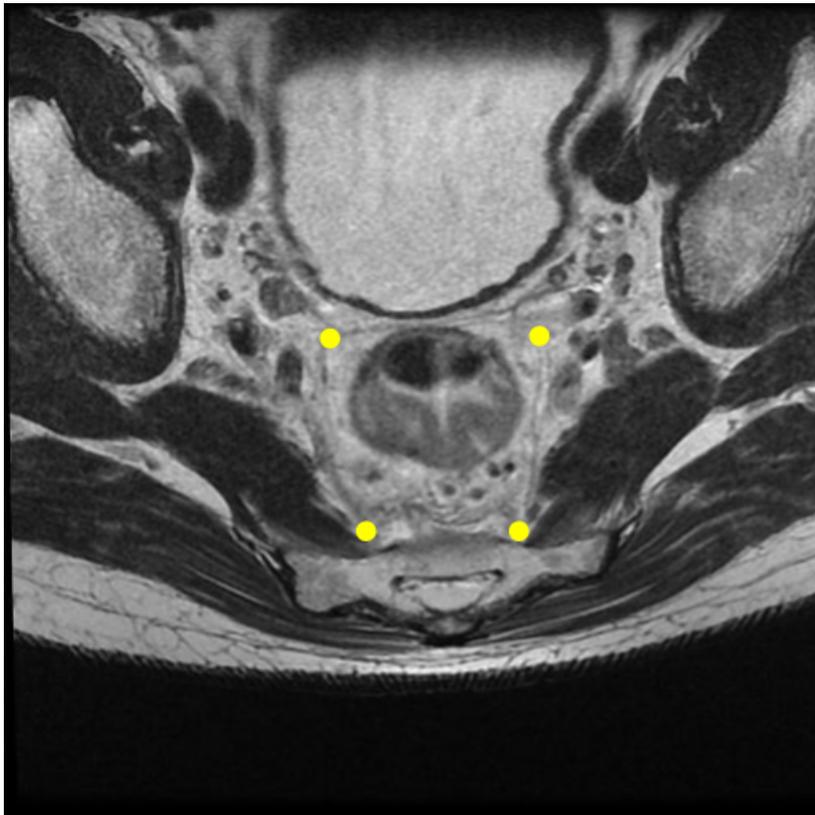


Figure 3.4: MRI slice, axial view. Example of landmarks (represented by yellow dots) selected by the operator, in the case of $N = 4$.

3.3 Algorithm Step 2: Finding landmarks

As MRI images are prone to noise, a curvature flow filter is applied to smoothen the image without losing information about the edges. Furthermore, since the mesorectum boundary is actually darker than the surrounding areas, the pixel intensity of the image is inverted, i.e. the value v of each pixel intensity is replaced with $(1 - v)$. This operation is so that when the intensity profile was studied, the mesorectum boundary appears as a peak instead of a trough, and is easier for the human eye to recognize.

The first objective is for the algorithm to find a set of points along the mesorectum boundary, with the least amount of input from the operator. The algorithm needs a starting point, however, as the mesorectum boundary is difficult for an automated

algorithm to characterize. To complicate matters, a single starting point may not be always useful because at points along the mesorectum boundary may have different characteristics. Thus, the minimum N was chosen to be 4, because 4 points are needed to define the corners of a square, which is the shape the mesorectum corresponded to in Figure 3.4. Moreover, the number of operator-specified landmarks needs to achieve a balance between the amount of work the operator has to do, and the amount of initializing data for the algorithm. This can be intuitively understood where having more points (i.e. higher N), the algorithm has more initial landmarks to work with and be in a better position to choose similar points that correspond to the mesorectum boundary, but at the cost of additional effort from the operator.

With these N landmarks, the global centroid c_g for that slice can be computed as:

$$c_g = \left\{ \frac{x_1 + x_2 + \dots + x_N}{N}, \frac{y_1 + y_2 + \dots + y_N}{N} \right\} \quad (3.2)$$

Fig. 3.5 illustrates the process the algorithm goes through to find points along the mesorectum boundary. The coordinate system will also be transformed from the xy -coordinate system to a radial coordinate system with the global centroid as the origin.

First, the intensity profile of the landmark along the line connecting the global centroid and one of the N landmarks is measured, as shown in Fig. 3.6. A subset of the intensity profile will be used as the ground truth when comparing against other intensity profiles in the region, and is centered on the landmark. The intensity profile is parameterized by the window size, which refers to the size of the intensity profile. Preliminary experiments show that a small window size of about 11 pixels is optimal for MRI slices when there are other structures (e.g. the rectum) in close proximity to the mesorectum boundary, while a larger window size of 41 pixels is better for MRI slices where there

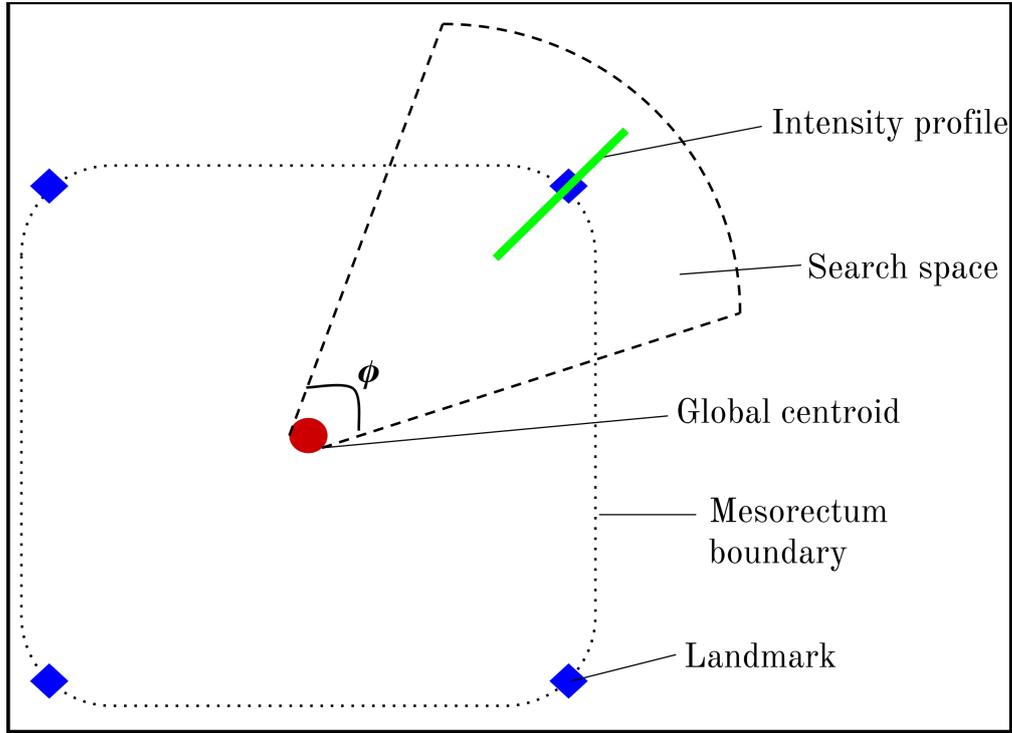


Figure 3.5: Searching process (not to scale). With $N = 4$, blue diamonds at the corners represent landmarks, while the red circle at the center is the global centroid. The green bar represents the intensity profile measured along the bar, and the length of the bar corresponds to the window size. The size of the search space is parameterized by the angle ϕ .

are other structures are further away.

Working in radial coordinates, each landmark is used to identify other points that lie on the mesorectum boundary, only in its immediate region, where this region is termed the “search space”. The size of the search space ϕ for any given landmark is determined by taking the minimum of the angle between itself and the two adjacent landmarks, where each i^{th} landmark has a corresponding angle θ_i relative to the global centroid c_g :

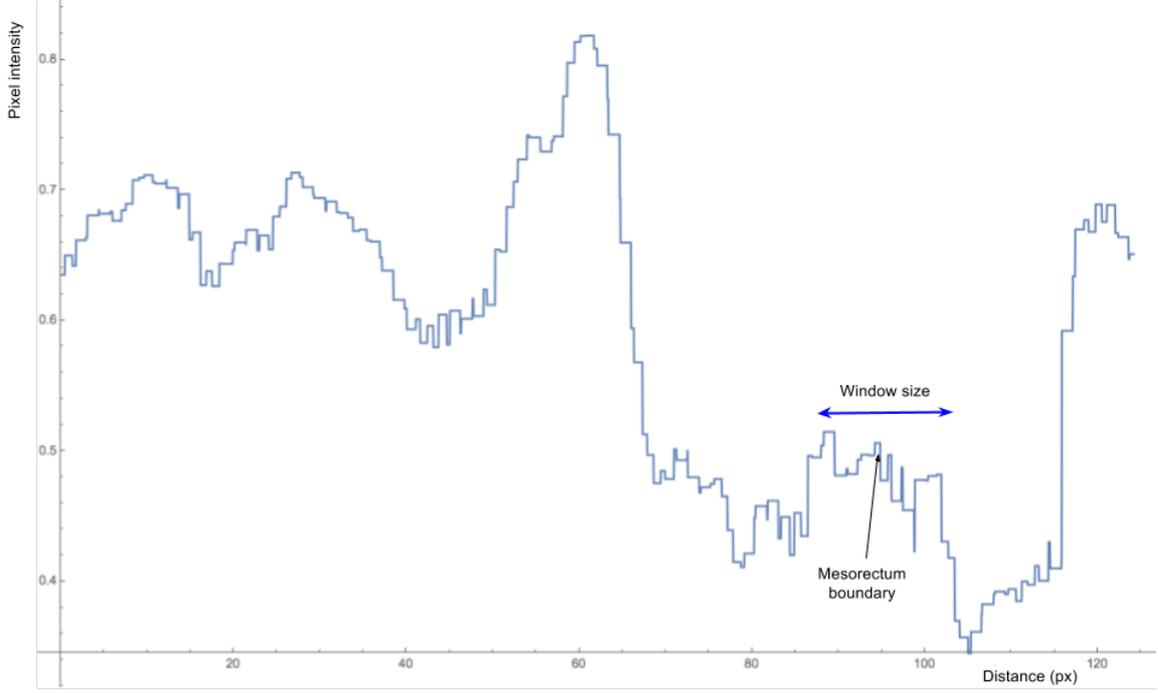


Figure 3.6: Example of intensity profile along line connecting global centroid to landmark. The mesorectum boundary corresponds to one of the peaks, and the window size parameterizes the length of the intensity profile used as the ground truth.

$$\phi_i(\theta_i, N) = \begin{cases} \min\{|\theta_i + 2\pi - \theta_{i-1}|, |\theta_{i+1} - \theta_i|\}, & \text{for } i = 1 \\ \min\{|\theta_i - \theta_{i-1}|, |\theta_{i+1} - \theta_i|\}, & \text{for } 1 < i < N \\ \min\{|\theta_i - \theta_{i-1}|, |\theta_1 + 2\pi - \theta_i|\}, & \text{for } i = N \end{cases} \quad (3.3)$$

The line connecting the origin and the i^{th} landmark is treated as $\theta = 0^\circ$ and $r = r_0$. Then, at every step of searching, the angle θ is incremented by 1° up to the limit of $\frac{\phi_i}{2}$; and in the radial direction, the algorithm searches along an interval corresponding to the window size centered on r_0 . The previous radius is used as a starting point for the next radius.

For each angle $\theta = \pm 1, 2, \dots, \frac{\phi_i}{2}$, the local intensity profile $i_{x,0}$ is compared against the

intensity profile within the search space i_x . The loss function L_x is chosen to be the absolute difference in intensities at each pixel, and the cost function C is the sum of all losses along the intensity profile. The reason for choosing an absolute loss over a mean squared loss is that MRI images tend to be noisy and full of outliers, and the effect of outliers is magnified by the mean squared loss, thus the absolute loss is more likely to give better results. In doing so, the point which has the most similar intensity profile to that of the landmark would be chosen.

$$L_x = |i_{x,0} - i_x| \quad (3.4)$$

$$C = \sum_x L_x \quad (3.5)$$

However, just because two points share a similar intensity profile, need not necessitate that these two points are on the mesorectum boundary. The resulting points found from the above method may not actually lie on the mesorectum boundary, as there could be other points that have a closer intensity profile similarity to the landmark. An additional piece of information is thus included, which is the position of the previous two landmarks found. Thus, instead of using a point found at face value, the mean of that point and the previous two points is used. Evidently, this cannot be the case for the points immediately adjacent to the landmark, in which case the mean of the landmark found and the operator-specified landmark is used. This helps to smoothen the points found, and is a way of encoding prior information (that all the points found should form a line) into the algorithm.

After testing the algorithm, it was observed that for lower slices (i.e. slices numbered #20 and below), due to the shape of the mesorectum, it was highly likely that if a landmark was above the global centroid, then the boundary should not be far above it. This additional information was then encoded back into the algorithm, in the form

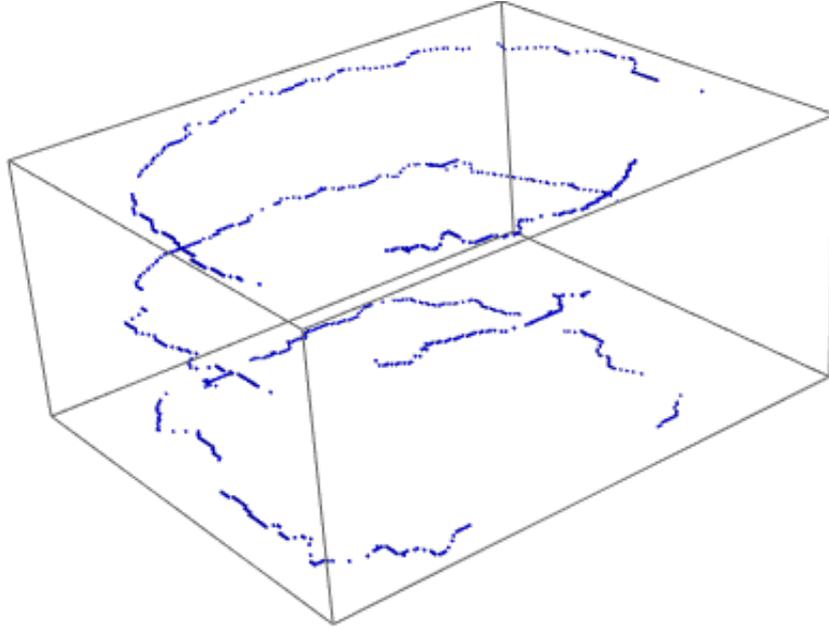


Figure 3.7: Set of all points that were found by the algorithm within the search space.

of a hard constraint, such that if the algorithm erroneously predicts such points, these are removed from consideration.

This search for points is repeated for each of the N landmarks, and for each of the 3 MRI slices. Then, outliers such as single points or tiny clusters that are not near the main curves (one for each landmark), are removed. Finally, the output of the algorithm at this stage is a set of points shown in Fig. 3.7, where the points distinctly outline 3 boundaries, as expected. For ease of terminology, let us call all these points landmarks as well. There are some discontinuities in the curves, and this is expected because the algorithm was not able to find suitable points there, due to the fact that the mesorectum boundary is not easily completely delineated in the first place, as well as the variety of signal changes around the boundary [BJPB07]. Nevertheless, as will be shown later, Laplacian deformation and TPS are both robust against these discontinuities.

Depending on the choice of initial landmarks, it is possible that the output is not a

good fit to the mesorectum boundary. The advantage of an interactive algorithm is that these inputs or parameters may be changed from their default values, to achieve a good fit:

- initial landmarks;
- window size of the intensity profile;
- size of the search space; and
- threshold value for the hard constraint on y-values of fitted points;

Throughout the experiments conducted, it was observed that a suitable choice of initial landmarks tends not to require any modification of other parameters.

3.4 Algorithm Step 3: Mesh Deformation

The next stage is to utilize the landmarks found earlier, and deform a mesh to fit these points. Thus far, we have been working in coordinate space (i.e. 512 px by 512 px). However, the desired mesh should be in real space, thus the coordinates of the landmarks are transformed via a scaling of the x-coordinates and y-coordinates to actual coordinates (i.e. 180 mm by 180 mm).

The mesh deformation procedure works as follows: first, the initialization of the mesh is done by placing a tiny sphere of radius 1 mm at the centre of all the fitted points. This ensures that the sphere is completely within the fitted points, and can only deform outwards. The number of mesh vertices forming initial sphere can be increased arbitrarily by imposing a maximum size on the tiny triangles that form the mesh. In this example, a size of 10^{-5} is used as it gives a sufficiently high number of mesh vertices (approximately 40,000) and can be computed quickly, under 5 seconds.

For each landmark found, a vector can be projected from the center of the initial

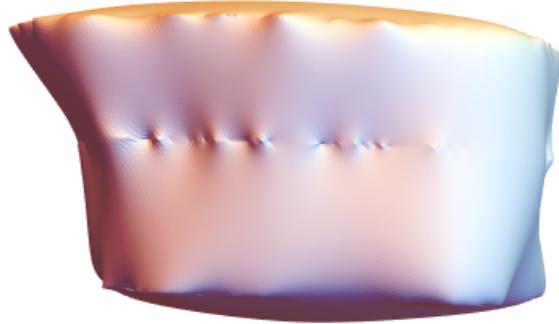


Figure 3.8: Mesh deformed via Laplacian deformation.

spherical mesh to that landmark, and the intersection of this vector on the surface of the spherical mesh is calculated. We denote these points of intersection as the source points, which are the points that deform.

In the case of TPS, there is an additional step of downsampling by taking the every d^{th} landmark and discarding the rest, where d is the downsampling factor. This requirement comes about because when there are many points (on the order of a few hundred or more), the TPS algorithm is unable to find a correspondence, or even if it does, does not produce a smooth mesh. There is thus a need to downsample the landmarks from hundreds of such points, down to tens. During the downsampling, the operator-specified landmarks might be dropped, so these landmarks are explicitly added back in to the array of points. The exact downsampling factor is not known in advance and has to be trialed and obtained by visual inspection. There is no such requirement of downsampling for the Laplacian deformation.

When the meshes are deformed via Laplacian deformation and TPS, the overall shape is as expected, as shown in Figs. 3.8 and 3.9.

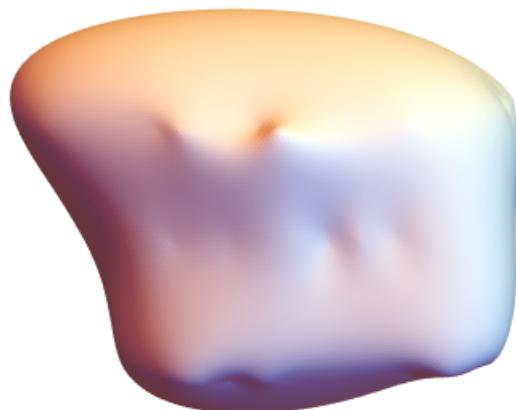


Figure 3.9: Mesh deformed via Thin Plate Spline.

3.5 Algorithm Step 4: Overlay of Mesh on MRI Slices

In order to observe the fitting of points to the mesorectum boundary, especially on MRI slices where there were no operator-specified landmarks, after the 3-dimensional mesh has been obtained, the intersection of the mesh with a plane with z -coordinates of each slice are calculated, and overlaid onto the MRI slice. This enables the operator to perform a visual inspection for the goodness of fit. The landmarks previously found on those slices where the operator had specified the landmarks, are also included.

Each mesh is composed of tiny triangles, and each vertex of the triangle must lie in some combination of above, on or below the plane of intersection. The calculations for the mesh intersection with the plane can be done by considering the following cases:

The intersection can be solved by using the similar triangles formula in 3D. We denote that intersection to have vertices (x_0, y_0, z_0) where z_0 is known because that is the z -coordinate of the plane. To illustrate the calculation, 2 of the points which form a

Table 3.1: Possible cases of mesh vertices positioning relative to plane of intersection.

# of vertices above plane	# of vertices below plane	# of vertices on plane	Action
2	1	0	Solve for intersection
1	2	0	Solve for intersection
1	1	1	Solve for intersection
1	0	2	Record points of intersection
0	1	2	Record points of intersection
2	0	1	Record points of intersection
0	2	1	Record points of intersection

line intersecting the plane can be denoted as (x_1, y_1, z_1) and (x_2, y_2, z_2) . Then, we just need 2 equations to solve for the 2 unknowns x_0, y_0 :

$$\frac{y_2 - y_1}{z_2 - z_1} = \frac{y_2 - y_0}{z_2 - z_0} \quad \&\& \quad \frac{z_2 - z_1}{x_2 - x_1} = \frac{z_2 - z_0}{x_2 - x_0} \quad (3.6)$$

When solving for the intersection between mesh and plane, there will be 2 points in each solution. Together with the recorded points of intersection, these points trace out the mesorectum boundary based on the deformed mesh.

3.6 Volume of MRI Slice

These mesh overlays on the MRI slices can be used to calculate the area of each slice. The overlay demarcates the boundary of the mesorectum for each slice, and upon morphological dilation by a factor of 3, delineates clearly the mesorectum from the rest of the image. The morphological dilation has a negligible effect on the area of the

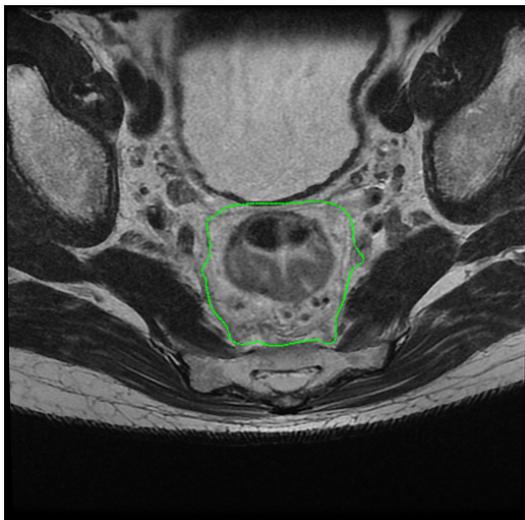


Figure 3.10: MRI slice #25, axial view. Overlaid with mesh deformed via TPS.

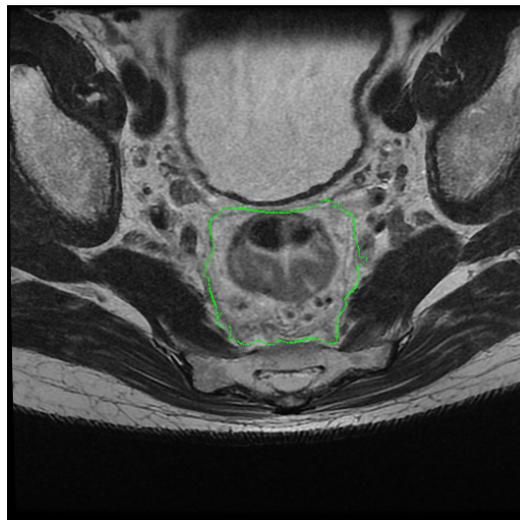


Figure 3.11: MRI slice #25, axial view. Overlaid with mesh deformed via Laplacian deformation.

mesorectum segment, below 1%, and is a necessary step to turn the boundary into a solid line, allowing a count of the area the mesorectum occupies in a given slice. The volume is calculated by multiplying this slice area by the slice height, as per current practice [TB05].

The results of the overlays, for $N = 4$, are shown in Figs. 3.10 - 3.33.

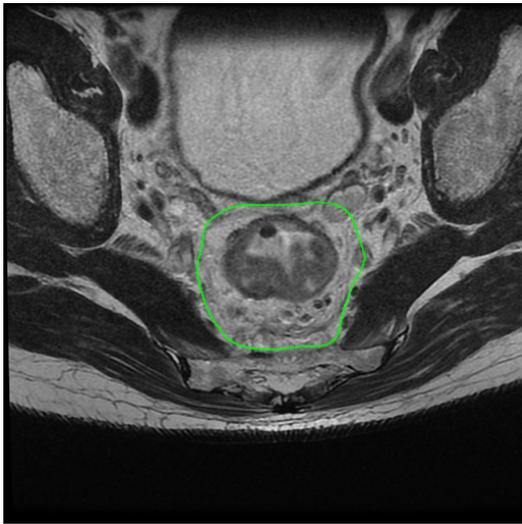


Figure 3.12: MRI slice #24, axial view. Overlaid with mesh deformed via TPS.

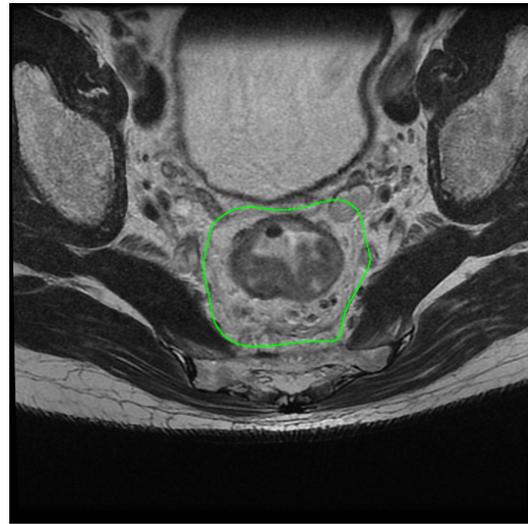


Figure 3.13: MRI slice #24, axial view. Overlaid with mesh deformed via Laplacian deformation.

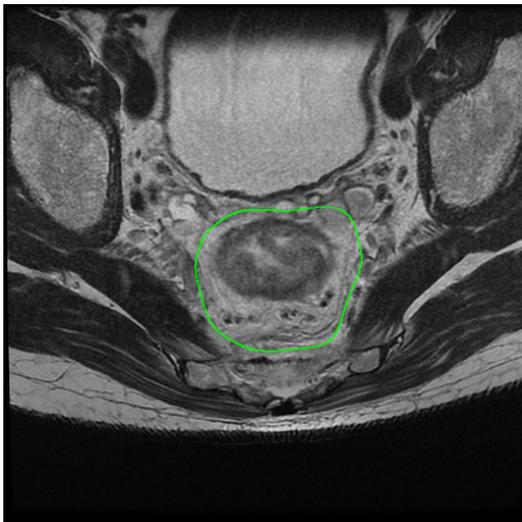


Figure 3.14: MRI slice #23, axial view. Overlaid with mesh deformed via TPS.

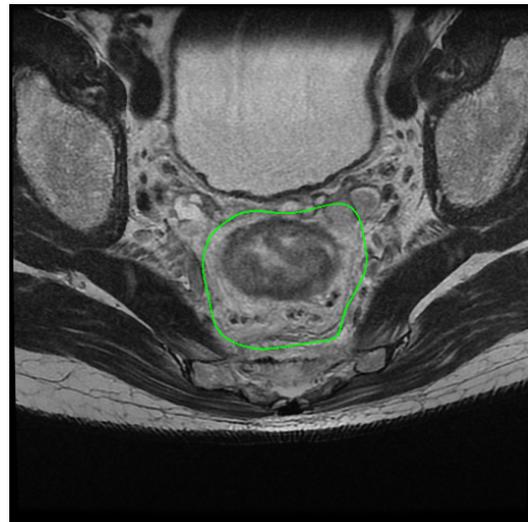


Figure 3.15: MRI slice #23, axial view. Overlaid with mesh deformed via Laplacian deformation.

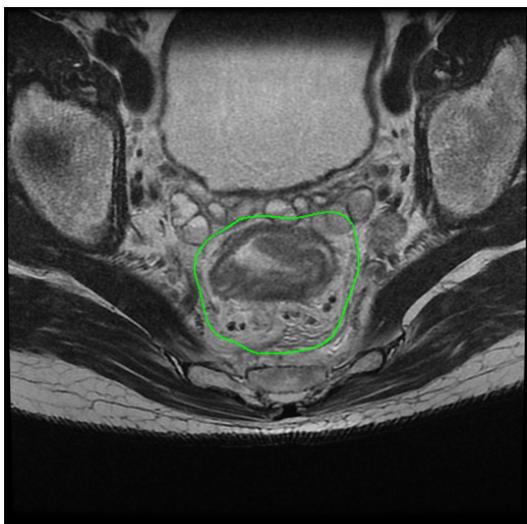


Figure 3.16: MRI slice #22, axial view. Overlaid with mesh deformed via TPS.



Figure 3.17: MRI slice #22, axial view. Overlaid with mesh deformed via Laplacian deformation.

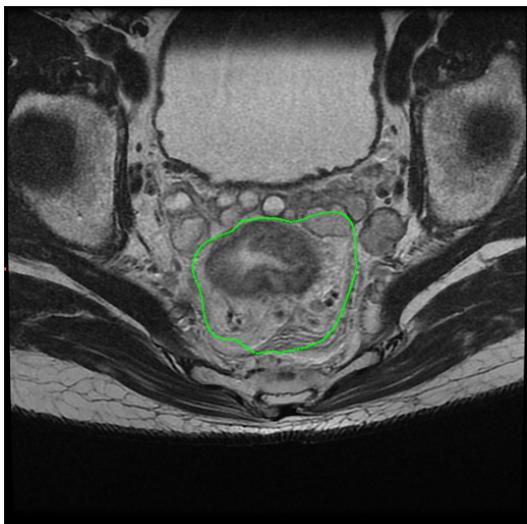


Figure 3.18: MRI slice #21, axial view. Overlaid with mesh deformed via TPS.

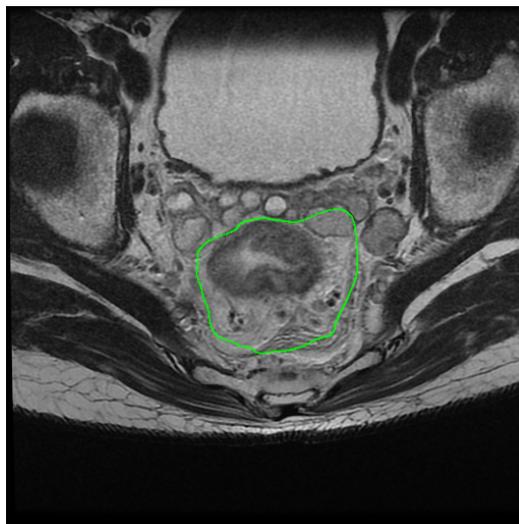


Figure 3.19: MRI slice #21, axial view. Overlaid with mesh deformed via Laplacian deformation.



Figure 3.20: MRI slice #20, axial view. Overlaid with mesh deformed via TPS.

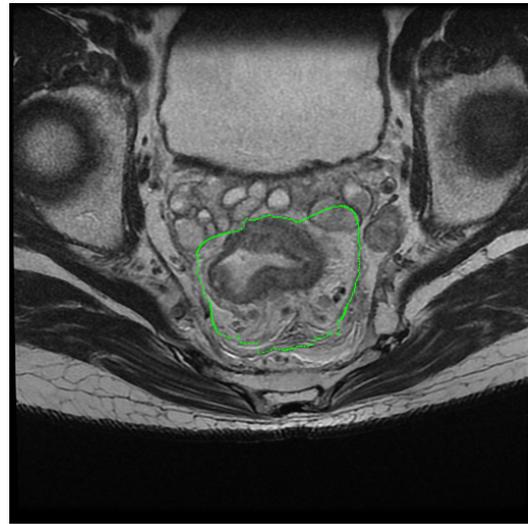


Figure 3.21: MRI slice #20, axial view. Overlaid with mesh deformed via Laplacian deformation.

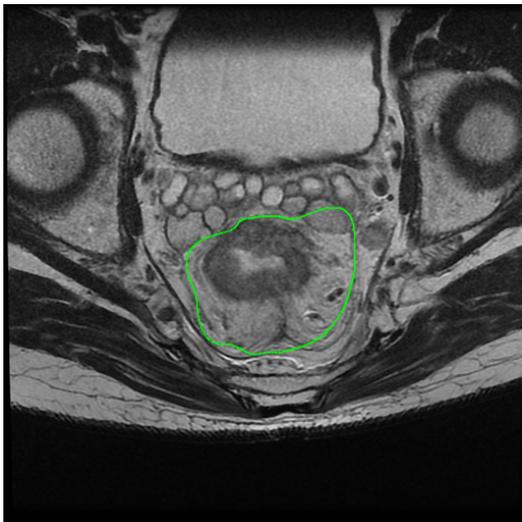


Figure 3.22: MRI slice #19, axial view. Overlaid with mesh deformed via TPS.

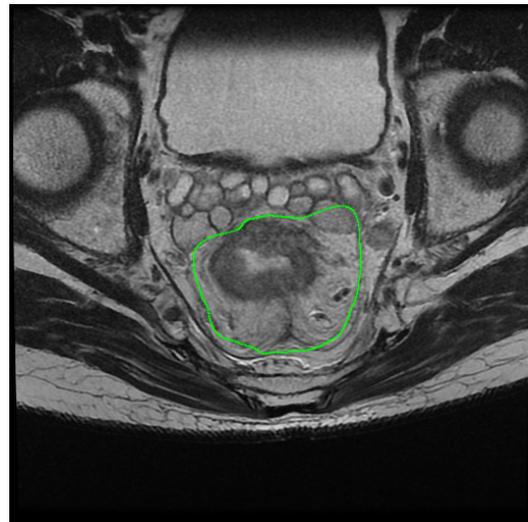


Figure 3.23: MRI slice #19, axial view. Overlaid with mesh deformed via Laplacian deformation.

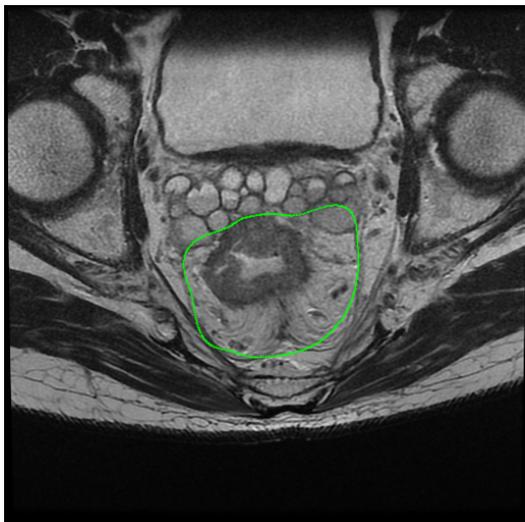


Figure 3.24: MRI slice #18, axial view. Overlaid with mesh deformed via TPS.

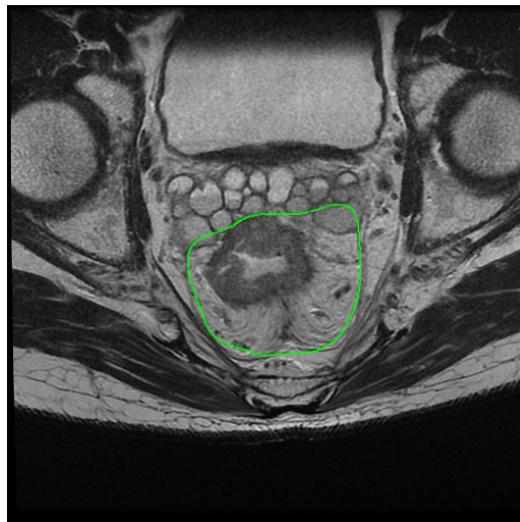


Figure 3.25: MRI slice #18, axial view. Overlaid with mesh deformed via Laplacian deformation.

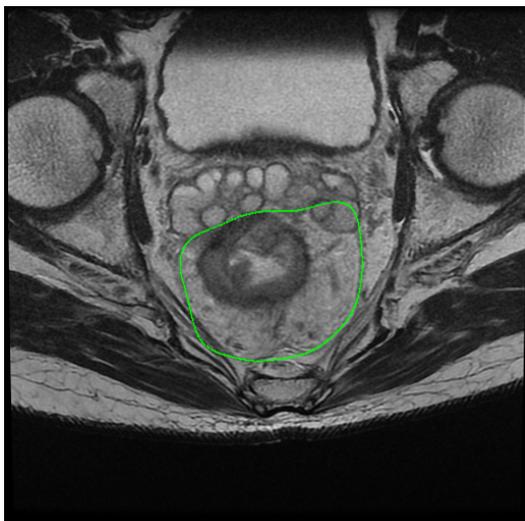


Figure 3.26: MRI slice #17, axial view. Overlaid with mesh deformed via TPS.

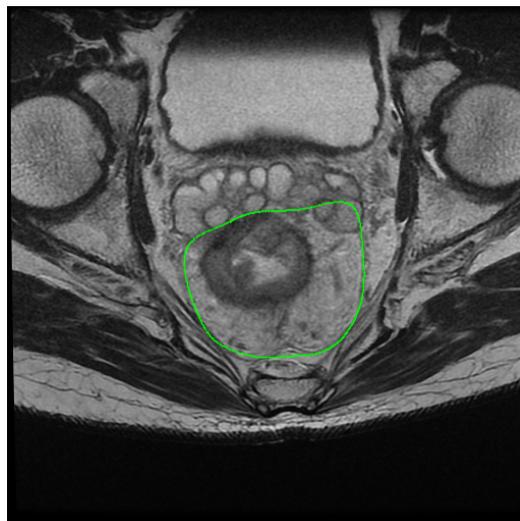


Figure 3.27: MRI slice #17, axial view. Overlaid with mesh deformed via Laplacian deformation.

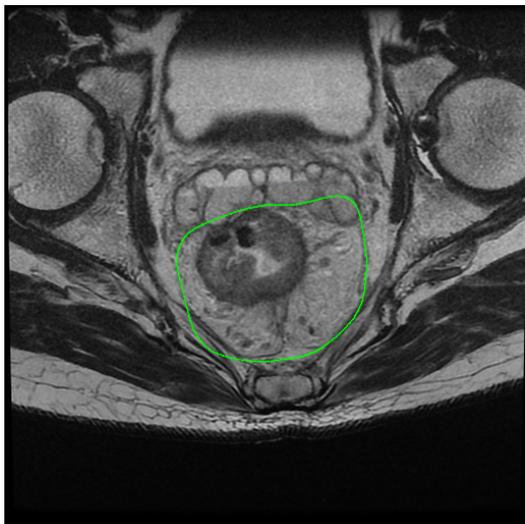


Figure 3.28: MRI slice #16, axial view. Overlaid with mesh deformed via TPS.

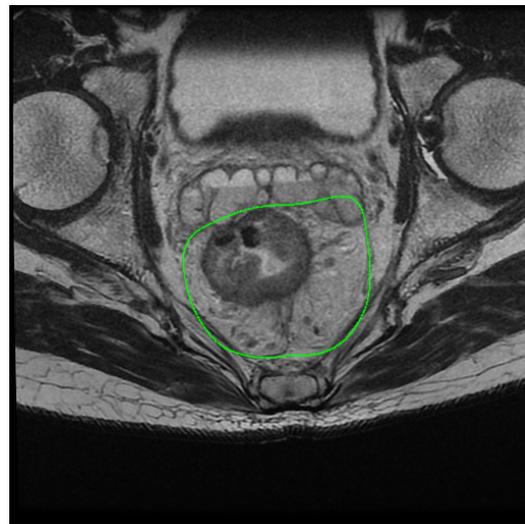


Figure 3.29: MRI slice #16, axial view. Overlaid with mesh deformed via Laplacian deformation.

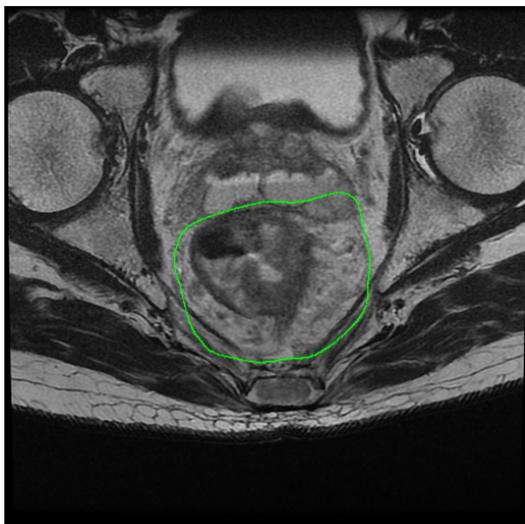


Figure 3.30: MRI slice #15, axial view. Overlaid with mesh deformed via TPS.

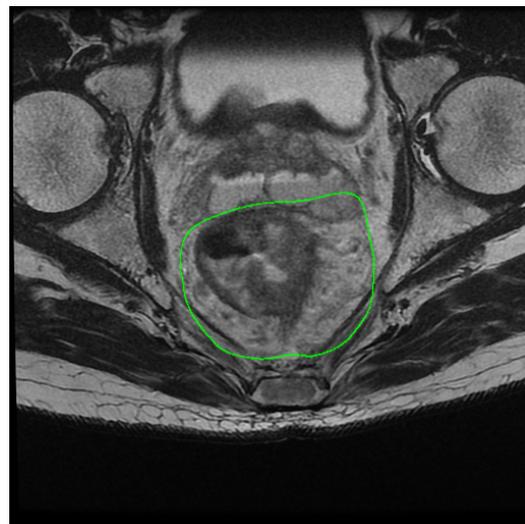


Figure 3.31: MRI slice #15, axial view. Overlaid with mesh deformed via Laplacian deformation.

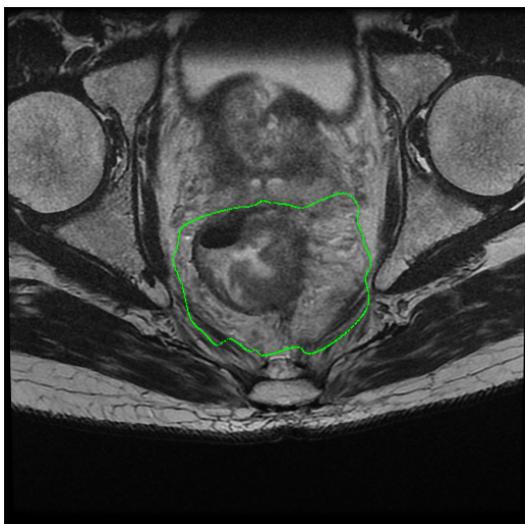


Figure 3.32: MRI slice #14, axial view. Overlaid with mesh deformed via TPS.

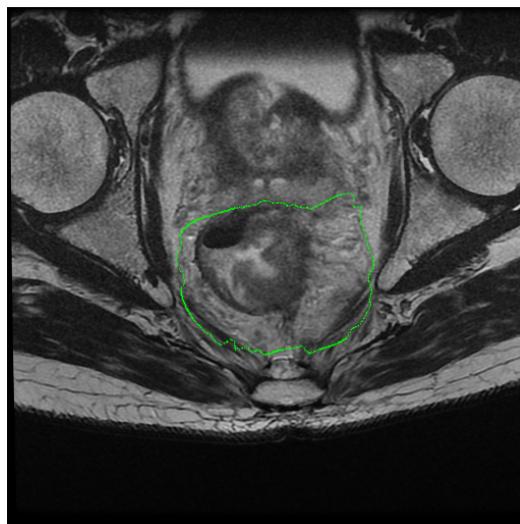


Figure 3.33: MRI slice #14, axial view. Overlaid with mesh deformed via Laplacian deformation.

In using the methods of Laplacian deformation and TPS to produce these 3D meshes, it is evident that the meshes fit exactly to the target points that were specified earlier. This is because these two methods consider the correspondence as hard constraints, thus have no error with regard to the corresponding target points.

CHAPTER 4

Experiments

4.1 Overview

Assessing the performance of an algorithm on image segmentation tasks is not easy. As mentioned in Chapter 1.1, accuracy cannot be used as a metric because the ground truth does not exist. Instead, this study uses the concept of robustness as one measure of the performance of the algorithm. Additionally, [UH99] offers two metrics to validate the algorithm's performance: consistency (also called *reliability*) and efficiency. Consistency is measured by repeating the measurement process and noting the variation using common statistical methods. Efficiency corresponds to the practical viability of the technique, for example, in the number of MRI scans that can be processed in an hour. The efficiency can be categorized into computational time and operator time (i.e. time spent by the operator), where computational time is said to be negligible provided it is below a practical threshold (e.g. 5 min), while the operator time is considered to be the crucial factor. In this chapter, 3 experiments are outlined, where the robustness, consistency and efficiency of the algorithm are assessed.

Table 4.1: Computed volume of MRI slices for varying N (Mean \pm standard deviation).

N	Volume (cm^3)(<i>Laplacian</i>)	volume (cm^3)(<i>TPS</i>)
4	93 \pm 1	94.5 \pm 0.9
5	98 \pm 1	98 \pm 3
6	96.0 \pm 0.8	96 \pm 1
7	96 \pm 2	95 \pm 2
8	95 \pm 2	95 \pm 1

4.2 Experiment 1: Robustness

Experiment 1 is to measure the robustness of the proposed algorithm. Here, robustness is measured by the variability in resultant mesorectum volume when the number of input landmarks N per slice varies. Intuitively, this can be thought of as when $N \rightarrow \infty$, the resultant mesorectum volume tends to a fixed value. In practice, N cannot be too large, else this process is too cumbersome for the operator, so the question becomes for what small values of N the algorithm will output a stable mesorectum volume. In this study, N was initialized at 4, and incremented until the total mesorectum volume measured is observed to stabilize at a fixed value. For slices #14 and #20, as N increased, additional landmarks were placed by the operator near the top boundary of the mesorectum, as the boundary at the top (also known as the anterior) is rather diffuse and blurs with other organs and structures, whereas the boundary at the bottom (also known as the posterior) is relatively better defined. This setup was replicated 5 times, and the results of mean mesorectum volume, as well as the standard deviation, for each value of N , are shown in Table 4.1. The data from Table 4.1 is plotted in Figure 4.1.

As suggested by [FC05], statistical methods should be used to analyze the mean of

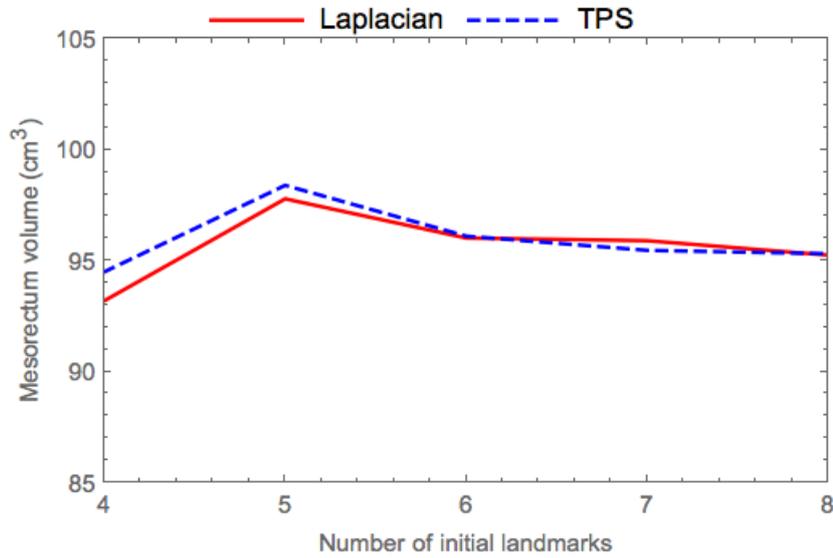


Figure 4.1: Plot of mesorectum volume (cm^3) against number of input landmarks per slice, N . Solid red line refers to volume when Laplacian deformation was used; dashed blue line refers to volume when TPS was used.

Table 4.2: p -values for pairwise comparison of differences in mesorectum volume across consecutive N .

Comparison between N_i & N_j	p -value (Laplacian)	p -value (TPS)
4 & 5	<.05	.05
5 & 6	.05	.19
6 & 7	.89	.48
7 & 8	.58	.86

segmented volumes, since segmentation is subject to variability. The Welch's t -test was chosen because unlike the standard student's t -test, there is no assumption that both populations have equal variances. Here, the significance level is set at 0.05.

From Table 4.2, it can be seen that there is a statistically significant difference in means of mesorectum volumes when comparing $N = 4$ and $N = 5$ for both Laplacian deformation and TPS, as well as when comparing $N = 5$ and $N = 6$ for Laplacian

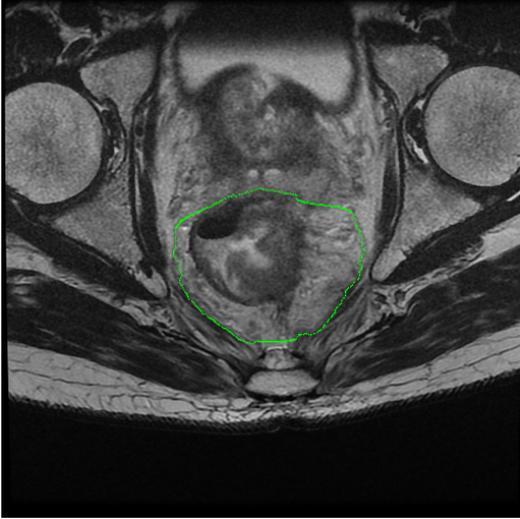


Figure 4.2: MRI slice #14, axial view. Overlaid with mesh deformed via Laplacian deformation, for $N = 5$.

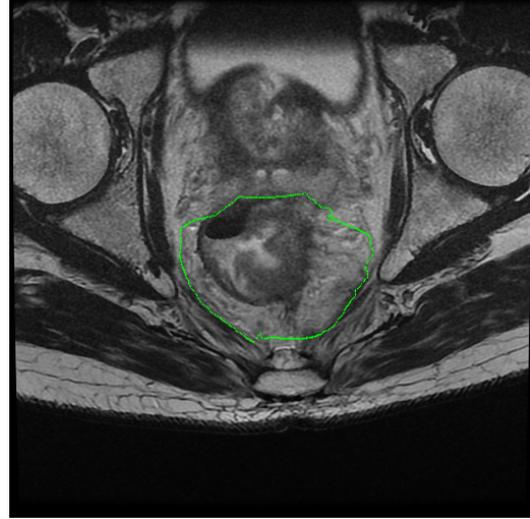


Figure 4.3: MRI slice #14, axial view. Overlaid with mesh deformed via Laplacian deformation, for $N = 6$.

deformation. This result suggests that when $N \geq 6$, there is no statistically significant difference in terms of the resultant volume, and the proposed algorithm is robust once a minimum N of 6 is used. This is also why the upper limit of N was set at 8, because the algorithm seemed to be stable by then.

To understand why this is the case, we look at MRI slices #14, #20 and #25, for $N = 5, 6, 7, 8$, where Laplacian deformation was used. These slices, with mesh overlays, are shown in Figures 4.2 to 4.13. These are the MRI slices for which initial landmarks were chosen, and we expect that the search for points along the mesorectum boundary yields a better fit as N increases.

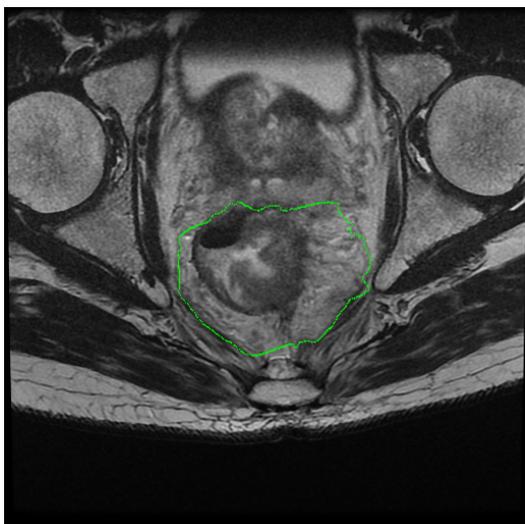


Figure 4.4: MRI slice #14, axial view. Overlaid with mesh deformed via Laplacian deformation, for $N = 7$.



Figure 4.5: MRI slice #14, axial view. Overlaid with mesh deformed via Laplacian deformation, for $N = 8$.

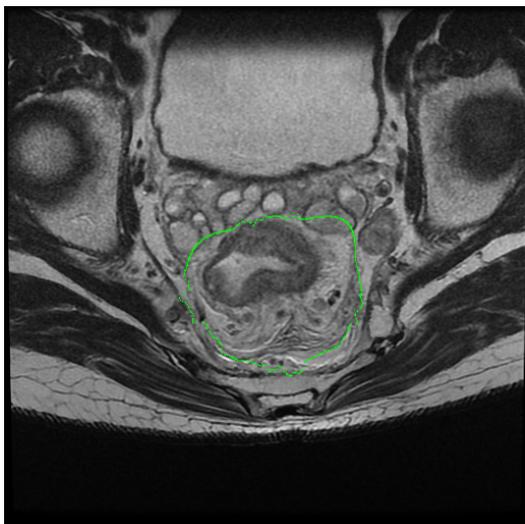


Figure 4.6: MRI slice #20, axial view. Overlaid with mesh deformed via Laplacian deformation, for $N = 5$.



Figure 4.7: MRI slice #20, axial view. Overlaid with mesh deformed via Laplacian deformation, for $N = 6$.

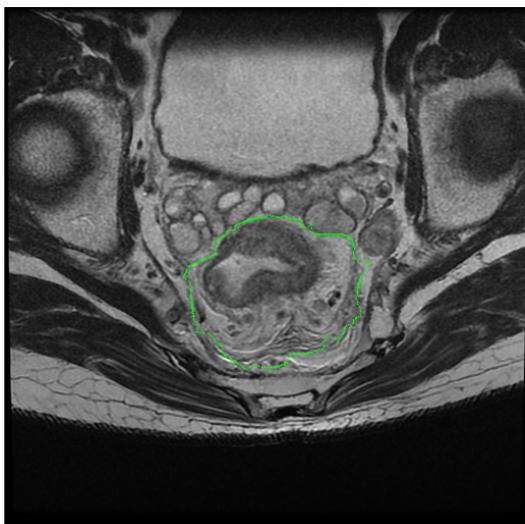


Figure 4.8: MRI slice #20, axial view. Overlaid with mesh deformed via Laplacian deformation, for $N = 7$.



Figure 4.9: MRI slice #20, axial view. Overlaid with mesh deformed via Laplacian deformation, for $N = 8$.

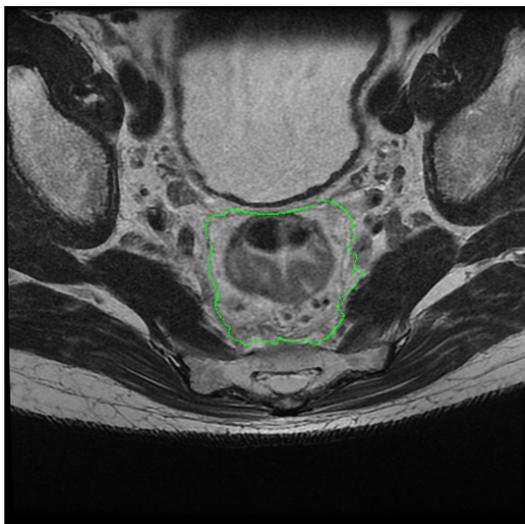


Figure 4.10: MRI slice #25, axial view. Overlaid with mesh deformed via Laplacian deformation, for $N = 5$.

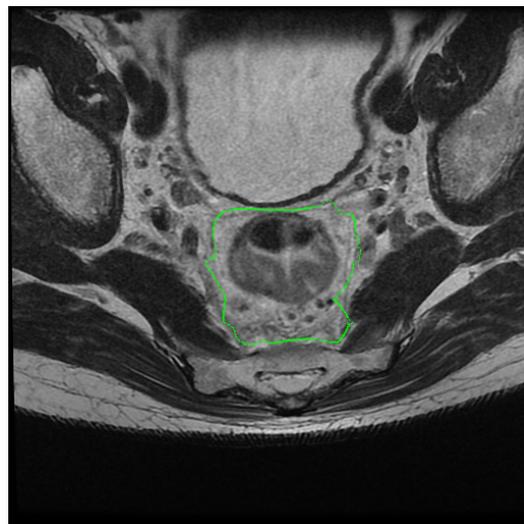


Figure 4.11: MRI slice #25, axial view. Overlaid with mesh deformed via Laplacian deformation, for $N = 6$.

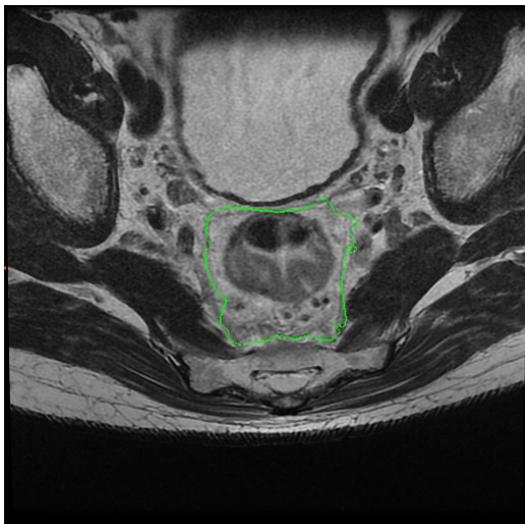


Figure 4.12: MRI slice #25, axial view. Overlaid with mesh deformed via Laplacian deformation, for $N = 7$.

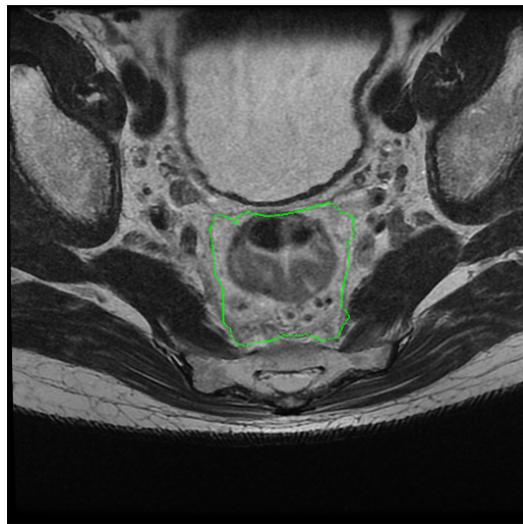


Figure 4.13: MRI slice #25, axial view. Overlaid with mesh deformed via Laplacian deformation, for $N = 8$.

Table 4.3: p -values for comparison of differences in mesorectum volume across Laplacian deformation and TPS.

N	p -value
4	.07
5	.71
6	.88
7	.68
8	.95

Intuitively, with a higher N , greater variation in the mesorectum boundary can be captured, especially if there is a sharp angle, or if the boundary is very diffuse and ill-defined. This was true, especially for MRI slices #14 and #20, where additional points helped the algorithm delineate the upper portion of the mesorectum boundary better. However, as can be seen from the figures above, for $N \geq 6$, the additional details captured do not seem to significantly affect the shape of the mesh, as the algorithm is able to find a good fit to the mesorectum boundary with lesser initial landmarks.

Next, we look at whether the choice of mesh deformation technique, Laplacian deformation or TPS, has a significant influence on the resultant mesorectum volume. Similar to above, the Welch's t -test was chosen and the significance level is set at 0.05. The resulting p -values are shown in Table 4.3.

All p -values were found to be above 0.05, thus the results suggest that there is no statistically significant difference in resultant mesorectum volume, whether Laplacian deformation or TPS was used. However, the p -value for $N = 4$ is quite small at $p = .07$, hinting that there might be a difference. Nevertheless, since the minimum N was found to be 6, when $N \geq 6$, it does not really matter whether Laplacian deformation or TPS is used.

Table 4.4: Percentage standard deviation in mesorectum volume measurements.

N	% Standard deviation (Laplacian)	% Standard deviation (TPS)
4	1.2%	0.96%
5	1.5%	3.2%
6	0.86%	1.0%
7	1.7%	1.7%
8	2.1%	1.4%

4.3 Experiment 2: Consistency

Experiment 2 is to determine the consistency of the proposed algorithm. The consistency of the proposed algorithm is measured here by the percentage standard deviation in mesorectum volume measurements over 5 replications, as shown in Table 4.4.

The percentage standard deviations are all very low, below 4%. This suggests that across multiple replications, when the operator initializes landmarks at different points along the mesorectum boundary, the proposed algorithm is still able to achieve a good fit to the mesorectum boundary, and the resultant volumes are highly consistent.

4.4 Experiment 3: Efficiency

Experiment 3 aims to measure the efficiency of the proposed algorithm. The total time taken is the sum of computation time taken and operator interaction time. The computation tasks can be divided into:

1. Initialization
 - (a) load dependency libraries

- (b) load MRI data
 - (c) process MRI metadata
 - (d) display MRI slices for operator to place landmarks on
 - (e) convert coordinate systems
2. Finding landmarks on mesorectum boundary
- (a) apply anisotropic filter
 - (b) invert pixel intensities
 - (c) calculate global centroids
 - (d) record intensity profile of landmarks
 - (e) search within the conical region
 - (f) calculate loss function along search window and determine the point at which there is lowest loss
 - (g) smoothen points found by taking the average of 3 points if possible, otherwise the average of 2 points
 - (h) remove outliers
3. Mesh deformation
- (a) (only for TPS) downsample data
 - (b) convert coordinate systems
 - (c) calculate placement of initial spherical mesh
 - (d) map landmarks to points on surface of spherical mesh (to identify source points)
 - (e) perform Laplacian deformation or TPS

- (f) display mesh
- (g) calculate coordinates of mesh triangles

4. Mesh overlay

- (a) determine which points lie on the plane
- (b) determine which points surround the plane
- (c) solve for intersection between mesh and plane
- (d) segment the image for each MRI slice
- (e) calculate mesorectum volume
- (f) display overlay
- (g) export overlay as image file

The computational time taken when running the proposed algorithm across 5 replications, when $N = 4$, is shown in Table 4.5. Under the design of the proposed algorithm, N does not have a direct impact on the computational time required. Rather, it is the spacing between landmarks that determines how much processing is required, which varies depending on the operator's choice of initial landmark placement. In this study, it was assumed that the spacing is usually non-uniform, which is reasonable given the preference of placing more landmarks at the top of the mesorectum boundary. This suggests that as N increases, time taken decreases. Thus, measuring the computational time taken when $N = 4$ should give the upper bound. The average computation time for the Laplacian method is 3.7 minutes while for the TPS method, the average time taken was slightly longer at 4.9 minutes. The TPS method took longer because of the additional step of downsampling required.

As a rough estimate, the operator interaction time is taken to be proportional to the number of clicks required to perform the segmentation. For the proposed algorithm,

Table 4.5: Computational time taken for proposed algorithm (Mean \pm standard deviation).

Task	Average timing (seconds)
Initialization	7 \pm 3
Finding landmarks	40 \pm 10
Mesh deformation (Laplacian)	30 \pm 10
Mesh overlay (Laplacian)	150 \pm 20
Mesh deformation (TPS)	60 \pm 10
Mesh overlay (TPS)	180 \pm 60
Total (Laplacian)	220 \pm 20
Total (TPS)	290 \pm 60

it is $3N$, because only N landmarks for 3 MRI slices are required. If N is set to 6, which has been found to be the minimum N for which the algorithm is stable when using both Laplacian deformation and TPS, only 18 clicks are needed. This can be considered minimal effort on the part of the operator. Granted, the TPS downsampling task requires operator intervention in the form of specifying which downsampling factor is reasonable, but this can be treated as the equivalent of a single click. If some operator intervention is required during the search for points — plausible scenarios include overlapping curves due to landmarks being placed too close to each other — the initial landmarks or default parameters can be tuned for a better fit.

4.5 Summary

As shown by Experiment 1, the proposed algorithm, using either Laplacian deformation or TPS, is highly robust requiring only a small number ($N = 6$) of operator-initialized landmarks on 3 MRI slices. From Experiment 2, the algorithm has also been found to be highly consistent, with $<4\%$ standard deviation in mesorectum volume for any

N. Based on Experiment 3, the efficiency of the algorithm is decent, requiring a computational time of under 5 minutes regardless of mesh deformation technique. Comparing Laplacian deformation and TPS, Laplacian deformation requires less effort on the part of the operator because there is no need for a downsampling action, and is computationally faster by about 1 minute. There does not seem to be any statistically significant difference on the mesorectum volume found whether Laplacian deformation or TPS was used.

Conclusion & Future Work

5.1 Limitations

While the proposed algorithm has been shown to be highly consistent, this is partially due to experience with the algorithm, where initial landmarks were chosen to come from the same region instead of at random. For example, for MRI axial slice # 25, it was observed that initializing landmarks near the corners gave good fitting of the mesorectum boundary, and points elsewhere led to a poor fitting. However, in reality, a operator may not know where to initialize landmarks for any given MRI slice. To accommodate the variation in choice of initial landmarks, the ideal situation would be that the proposed algorithm is able to clearly delineate the mesorectum boundary regardless of where the operator-specified landmarks were, so long as they were accurately placed on the mesorectum boundary.

Due to the exploratory nature of this study, there was only one set of axial MRI scans was available, which is admittedly a significant limitation. More data from different colorectal cancer patients, both male and female, will be required to better assess the robustness and consistency of the proposed algorithm.

Currently, the prototype is implemented in *Mathematica*, which was used as *Mathematica* is a very powerful high-level programming language, enabling quick prototyping for research, at the cost of computational speed. When the algorithm is to be implemented

in production, the algorithm can be made more efficient by porting it into *C++*, where computations could be sped up by an order of magnitude or two.

Finally, it should be noted that the research thus far is highly exploratory, and has yet to be implemented in a real world setting and tested by doctors. In order to ascertain the usefulness of the proposed algorithm, further work has to be conducted to refine the performance of the algorithm, and tested by the intended users.

5.2 Conclusion

Colorectal cancer affects a significant proportion of people throughout the world, and particularly so in Singapore. The mesorectum volume has been hypothesized to be an independent predictor of a positive surgical margin, as well as of surgery type (i.e. laparoscopic or open surgery) that has the best medical outcome for the patient. Measurements of mesorectum volume from MRI scans via manual segmentation are extremely time-consuming, tedious, and prone to intra-operator and inter-operator variability. In standard hospital practice, mesorectum volume is not measured at all. On the other hand, there is also no robust fully automated algorithm for this purpose, as medical MRI images are prone to significant variation and thus difficult to segment automatically.

Hence, a new semi-automated algorithm has been proposed to segment and measure the mesorectum volume from 3D MRI images. The algorithm has 4 main steps, namely initialization, finding landmarks that delineate the mesorectum boundary, deforming a mesh via Laplacian deformation and Thin Plate Spline, and overlaying the deformed mesh on MRI scans. Based on the results of Experiment 1, the algorithm has been found to be robust when $N \geq 6$, where additional landmarks are not required. As shown by the results of Experiment 2, the algorithm is also highly consistent, and able to consistently produce resultant mesorectum volumes with a standard deviation of

less than 4%. In terms of efficiency, the results of Experiment 3 suggest that relatively minimal input is required from the operator, and the computational time taken is only about 4-5 minutes. The 3 objectives of robustness, consistency and efficiency have been shown to be achieved by the experiments conducted. Laplacian deformation was also found to be faster than TPS by about 30 seconds, and both methods of mesh deformation result in highly similar volume measurements with no statistically significant difference.

This new algorithm has the potential to augment the work of medical staff in analyzing MRI scans, saving time and effort and leading to better patient outcomes. With the right information at the doctors' fingertips, the correct surgical decision can be made more frequently, leading to a lesser number of cases with surgical complications. Additionally, with this algorithm allowing consistent and quick measurements of the mesorectum volume, further research on the effect of mesorectum volume on other medical variables can be easily conducted. Going forward, more data will be required to improve the proposed algorithm's robustness, consistency and efficiency.

With the proposed algorithm being able to segment the mesorectum robustly and reliably, one natural extension is to segment the rectum as well. The rectum lies completely within the mesorectum, and it would be interesting to use the volume of the mesorectum in the two cases of including and excluding the rectum volume, as factors for surgical assessment. Beyond just using the mesorectum and rectum volumes, the densities of the mesorectum and rectum, or equivalently, their intensities in the MRI scan, can also be measured and used as factors for surgical assessment.

References

- [AGB07] SD Allen, V Gada, and DM Blunt. Variation of mesorectal volume with abdominal fat volume in patients with rectal carcinoma: assessment with mri. *The British journal of radiology*, 80(952):242–247, 2007.
- [BCA98] Eric Bardinet, Laurent D Cohen, and Nicholas Ayache. A parametric deformable model to fit unstructured 3d data. *Computer vision and image understanding*, 71(1):39–54, 1998.
- [BJPB07] Sarah Bond, Niranjana Joshi, Styliani Petroudi, and Mike Brady. Estimating the mesorectal fascia in mri. In *Biennial International Conference on Information Processing in Medical Imaging*, pages 650–661. Springer, 2007.
- [Boo89] Fred L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on pattern analysis and machine intelligence*, 11(6):567–585, 1989.
- [DB96] Christos Davatzikos and N Bryan. Using a deformable surface model to obtain a shape representation of the cortex. *IEEE transactions on medical imaging*, 15(6):785–795, 1996.
- [FC05] Araon Fenster and Bernard Chiu. Evaluation of segmentation algorithms for medical imaging. In *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*, pages 7186–7189. IEEE, 2005.

- [Fen10] Ding Feng. *3D Segmentation of Soft Tissues by Flipping-free Mesh Deformation*. PhD thesis, National University of Singapore, 2010.
- [HHR82] RJ Heald, EM Husband, and RDH Ryall. The mesorectum in rectal cancer surgery: the clue to pelvic recurrence? *British Journal of Surgery*, 69(10):613–616, 1982.
- [IOH⁺13] T Ishida, K Okabayashi, H Hasegawa, Y Ishii, H Kikuchi, R Seishima, and Y Kitagawa. Pelvic dimensions and mesorectal volume as predictors of surgical difficulty in laparoscopic surgery for rectal cancer. *Colorectal Disease*, 15:40, 2013.
- [KWT88] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331, 1988.
- [MBL⁺91] James V Miller, David E Breen, William E Lorensen, Robert M O’Bara, and Michael J Wozny. Geometrically deformed models: a method for extracting closed geometric models form volume data. In *ACM SIGGRAPH Computer Graphics*, volume 25, pages 217–226. ACM, 1991.
- [MT96] Tim McInerney and Demetri Terzopoulos. Deformable models in medical image analysis: a survey. *Medical image analysis*, 1(2):91–108, 1996.
- [NL98] Anke Neumann and Cristian Lorenz. Statistical shape model based segmentation of medical images. *Computerized Medical Imaging and Graphics*, 22(2):133–143, 1998.
- [oDO17] National Registry of Diseases Office. Singapore cancer registry annual registry report 2015. Singapore cancer registry annual registry report, National Registry of Diseases Office, 2017.
- [PXP00] Dzung L Pham, Chenyang Xu, and Jerry L Prince. Current methods in

- medical image segmentation 1. *Annual review of biomedical engineering*, 2(1):315–337, 2000.
- [Ric14] David Richfield. Medical gallery of david richfield, 2014.
- [SCOL⁺04] Olga Sorkine, Daniel Cohen-Or, Yaron Lipman, Marc Alexa, Christian Rössl, and H-P Seidel. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 175–184. ACM, 2004.
- [SSW88] Prasanna K Sahoo, SAKC Soltani, and Andrew KC Wong. A survey of thresholding techniques. *Computer vision, graphics, and image processing*, 41(2):233–260, 1988.
- [TB05] Michael R Torkzad and Lennart Blomqvist. The mesorectum: morphometric assessment with magnetic resonance imaging. *European radiology*, 15(6):1184–1191, 2005.
- [THL⁺07] Michael R Torkzad, Karl A Hansson, Johan Lindholm, Anna Martling, and Lennart Blomqvist. Significance of mesorectal volume in staging of rectal cancer with magnetic resonance imaging and the assessment of involvement of the mesorectal fascia. *European radiology*, 17(7):1694, 2007.
- [TRS⁺15] Muhammad Tayyab, Abdul Razack, Abhiram Sharma, James Gunn, and John E Hartley. Correlation of rectal tumor volumes with oncological outcomes for low rectal cancers: does tumor size matter? *Surgery today*, 45(7):826–833, 2015.
- [UH99] Jayaram K Udupa and Gabor T Herman. *3D imaging in medicine*. CRC press, 1999.