Decimation and Analysis: An Internal Bleeding Detection Framework for 3D Ultrasound

by

Michael Vukovich

A thesis submitted in conformity with the requirements for the degree of Master of Applied Science
The Edward S. Rogers Sr. Department of Electrical and Computer Engineering
University of Toronto

© Copyright 2018 by Michael Vukovich
Abstract

Decimation and Analysis: An Internal Bleeding Detection Framework for 3D Ultrasound

Michael Vukovich
Master of Applied Science
The Edward S. Rogers Sr. Department of Electrical and Computer Engineering
University of Toronto
2018

This thesis proposes a decimation-analysis (D-A) framework for detecting internal bleeding in 3D ultrasound medical scans. The structure consists of two main components: decimation, in which an input volume is reduced to suspect regions, and analysis, in which feature extraction is performed on said suspect regions. The D-A framework was designed to identify intuitive, easily-identifiable features of blood pools as an aid for obtaining more complex shape information that can be assessed and matched to “library” data created from training volumes. The D-A framework is presented here for its concept and structure rather than for its actual performance, as rigorous, accurate testing was beyond the scope of the thesis; however, access to accurately segmented ground truth volumes (something missing during the experiment) would enable more in-depth development and validation of the framework.
Acknowledgements

I would like to thank my supervisor, Professor Kostas N. Plataniotis, for his incomparable levels of support, understanding, guidance and encouragement during my academic career. I would also like to thank the members of my MASc examination committee - Professors Dimitris Hatzinakos, Stergios Stergiopoulos, and Andreas Veneris - for their participation in the exam and their invaluable feedback.

Furthermore, I would like to thank my Communications Group labmates, in BA4157, for their help, guidance and interaction along the way, and for helping to provide a warm, welcoming and enjoyable environment in which to work.

I would also like to thank my father, George Vukovich, for helping and encouraging me all my life in both my academic pursuits and my personal growth.

Finally, I would like to thank the many other people in my life who has helped me come this far in terms of academic success, self-esteem, and contributing to society.
# Contents

Acknowledgements iii  
Table of Contents iv  
List of Figures vi  
List of Tables viii  

1 Introduction 1  
1.1 Motivation ................................................. 1  
1.2 Thesis Organization ........................................ 3  

2 Background 4  
2.1 The FAST Exam .............................................. 4  
2.2 Automated Free Fluid Detection ........................... 6  
2.3 Pre-processing .............................................. 7  
2.4 Prior work .................................................. 7  
2.4.1 Feature Space Clustering Method (Zagrodsky et al.) .... 7  
2.4.2 Two-Stage Level Set Method (Kim et al.) .............. 10  
2.4.3 Other methods ........................................... 12  

3 The D-A Framework: Overview and Design 14  
3.1 System Overview ............................................. 14  
3.2 Background ................................................ 15  
3.2.1 3D Data ................................................ 15  
3.2.2 Thresholding ............................................. 15  
3.2.3 Feature Extraction ....................................... 17  
3.3 Module Descriptions ....................................... 20  
3.3.1 Pre-Processing and Feature Volume Creation .......... 20  
3.3.2 Development .......................................... 20  
3.3.3 Operation ............................................. 23  

4 Testing and Results 28  
4.1 Experimental Setup ......................................... 28  
4.1.1 Dataset ................................................. 28
List of Figures

1.1 2D (left) and synthetic 3D (right) ultrasound scans for abdominal organs. ............... 2

2.1 Ultrasound scan of abdominal region with internal bleeding[30]. Note the appearance of
the fluid and its location relative to the liver and kidney. ........................................ 5

2.2 Locations of four common scan regions in the FAST exam[11]; the Morison’s Pouch view
is most important for detecting internal bleeding. .................................................... 5

2.3 Diagram of the general steps and modules for most internal bleeding detection algorithms.
The development steps (left) are used to produce feature templates/values that are used
to assess results for the operation step (right). ....................................................... 6

2.4 Example of the effects of speckle noise and denoising on ultrasound imaging. ............. 8

2.5 Feature images used to define blood pools in Zagrodsky’s method[36]. ....................... 9

2.6 Left: Feature space for Zagrodsky’s method, with a cluster represented using circles.
Right: Cluster mapped back to image space[36]. ...................................................... 10

2.7 Sample segmentation results for (a) MVRG, (b) CVLS, and (c) Kim’s method[14]. The
pre-defined “ground truth” is shown as a dotted line, while segmentation results are shown
in red. .................................................................................................................. 11

2.8 Results for each of the two parallel modules in Ito’s method[11]: (a) low-brightness filter-
ing, (b) organ area segmentation. ............................................................................ 12

2.9 Demonstration of Noll’s method. Top: Demonstrating shadow detection (right) in an
ultrasound scan (left)[26]. Bottom: Highlighting the kidney (right) in an ultrasound scan
(left). .................................................................................................................... 13

3.1 Pipeline for D-A development and operation. ........................................................... 15

3.2 Demonstrating how the data volumes can be traversed along any of 3 dimensions to obtain
different 2D frame cuts. ......................................................................................... 16

3.3 Example of intensity-based thresholding on an XY cross-section of a sample volume. ... 17

3.4 Conceptual image depicting the nature of keypoints and the descriptor vectors associated
with each. .............................................................................................................. 18

3.5 Illustrating the difference between global and local features. .................................... 19

3.6 Pipeline of preparation steps for D-A. ...................................................................... 21

3.7 Demonstrating the steps of pre-processing. .............................................................. 21

3.8 Manually segmenting a blood pool to create a ground truth mask. ......................... 22

3.9 Pipeline of pre-decimation for D-A. ....................................................................... 23

3.10 Pipeline of pre-analysis for D-A. .......................................................................... 24
3.11 Simple visualization of Algorithms 1 and 2, in which descriptors (yellow) are sorted based on the presence of their keypoints (red) in thresholded regions (white).

3.12 Visualization of automatic thresholding via decimation; note how this does not clearly isolate the blood pool as with Fig. 3.8.

3.13 Pipeline of decimation for D-A.

3.14 Pipeline of analysis for D-A.

4.1 Segmenting an AIBP (Definition 4.1.1) in 2D to create one frame cut of an ASGT.

4.2 XY frame cut of each of the initial six training volumes, with AIBP’s highlighted in red. Volumes 06 and 13 contained two AIBP’s, and were eliminated.

4.3 Pipeline for the facsimile of Zagrodsky’s algorithm.

4.4 Pipeline for the facsimile of Kim’s algorithm.

4.5 Calculating BACC over different thresholds for the Zagrodsky and Kim facsimiles. D-A, the most important algorithm here, was omitted due to the fact that \( \mu_s \) was calculated as 0 in all cases.

A.1 Processing pipeline for MaxPol lowpass filtering.
## List of Tables

4.1 List of dataset volumes sorted by size. .................................................. 31
4.2 Values of $p_{max}$ for different values of $k$ for the Zagrodsky facsimile. ........ 34
4.3 Output values and thresholds for each algorithm tested. .......................... 35
4.4 Highest BACC values obtained for each algorithm. ............................... 36

A.1 Sample results for MaxPol 3D denoising. ................................................. 42
A.2 Sample for median filter denoising. ......................................................... 42
A.3 Sample results for ideal filter denoising. .................................................. 43
A.4 Comparing the results of standard and homomorphic ideal filter denoising. .... 43
A.5 Sample results for Butterworth filter denoising. ........................................ 45
A.6 Comparing first- and second-order Butterworth filtering. ......................... 46
A.7 Comparing standard and homomorphic Butterworth filtering. .................... 46
A.8 Varying the eliminated band for wavelet filter denoising. ............................ 47
A.9 Varying the wavelet type for wavelet filter denoising. ............................... 48
A.10 Varying the wavelet order for wavelet filter denoising. .............................. 48
A.11 Comparing single and double decomposition for wavelet filter denoising. .... 49
A.12 Comparing standard and homomorphic wavelet filtering. ......................... 49
A.13 Comparing all general denoising methods for first sample frame (60). ......... 51
A.14 Comparing all general denoising methods for second sample frame (80). ....... 53
A.15 Comparing all general denoising methods for third sample frame (100). ....... 55
A.16 Comparing all general methods for fourth sample frame (120). .................. 57
Chapter 1

Introduction

This thesis focuses on automated processing of 3D medical ultrasound (US) scans, and proposes a framework known as decimation-analysis (D-A) for detecting internal bleeding within scans of trauma patients’ abdominal cavities. Because such detection must be performed by a computerized system, with little to no manual input, the problem falls within the domain of image processing. In most or all previous approaches, the ultrasound scan is treated as 3D volumetric data or a set of 2D image data, and an algorithm is used to identify regions that can be identified as internal bleeding. D-A does so in two steps: decimation, where the volume is thresholded into suspect regions, and analysis, where feature extraction is used to analyze these suspect regions.

1.1 Motivation

Abdominal trauma is a leading cause of death worldwide, often arising from causes such as vehicular accidents, sport or military injuries, gunshot wounds, or child abuse[18]. While penetrating abdominal trauma is relatively simple to diagnose, blunt abdominal trauma often occurs together with organ injuries, and necessitates analysis beyond basic physical examination[27] to identify internal bleeding. Such diagnosis is usually performed using one of three methods: diagnostic peritoneal lavage (DPL), computed tomography (CT), or ultrasound (US)[27][12]. DPL is a surgical diagnostic method in which fluid is extracted using a catheter[12], while CT and ultrasound are imaging methods that employ X-ray arrays and high-frequency sound waves, respectively[33].

Some patients suffer from hemodynamic instability, in which abnormal or unstable hemodynamic parameters (such as blood pressure) can lead to spontaneous death[24]. Prompt and accurate diagnosis is essential to avoid complications, as bleeding must be detected and stopped as soon as possible[12][18]. In cases like these, DPL is undesirable due to its invasive nature and the level of skill required, despite its high sensitivity and specificity[12]. While it benefits from similarly high measures of accuracy as well as fast scanning speed[33], CT is also ill-suited to hemodynamic instability due to its high cost, lengthy reconstruction time, and especially the immobility of the large, bulky CT scanning machines - the patient must be transported away from the resuscitation area, which puts them at risk due to transportation time[12][18]. Ultrasound is the diagnostic method of choice because it addresses the disadvantages of DPL and CT: it is non-invasive, portable, fast, produces images in real time, and can be performed without having to leave the resuscitation area[12][18][4].
Ultrasound is used to assess hemodynamically unstable patients using standardized procedures such as the Focused Assessment with Sonography in Trauma (FAST) exam[12]. While ultrasound is well-suited to such patients, accurate and/or prompt results are difficult to obtain, for the following reasons:

- Accurate analysis requires specific knowledge from trained physicians and ultrasonographers (trained experts), who might not be available on the scene because paramedics are usually dispatched instead. [18]

- In order to be assessed by a trained expert, the patient must be transported to a hospital. Transportation times cause delays, which prevent prompt analysis and put patients at risk of death during transportation[18] - thus nullifying a key advantage of US.

Computer-aided diagnostic (CAD) systems address these two points by automating any steps that would normally rely on expert knowledge[18], allowing for prompt diagnosis that can be performed on-site by a paramedic. The purpose of such a system is to analyze a patient’s ultrasound scan to detect internal bleeding with little, if any, manual input. While 2D ultrasound is in common use (including for the FAST exam), it lacks a fixed coordinate system with respect to the patient’s body, making it inappropriate for the development of a CAD system[18]. It is therefore preferable to work with three-dimensional ultrasound imaging rather than 2D. 3D ultrasound’s use of such a coordinate system allows for more accurate probe placement, localization, and volume measurement for internal structures, which cannot be accomplished in 2D[18][4].

It should be noted that 3D ultrasound can be true or synthetic. The former consists of visualization in full 3D space, while the latter consists of individual 2D scans that are assembled in a 3D array(Fig. 1.1). Synthetic 3D US was used for this thesis due to its greater available and accessibility, as true 3D US is a new development.

![3D ultrasound](image)

Figure 1.1: 2D (left) and synthetic 3D (right) ultrasound scans for abdominal organs.

The field of 3D ultrasound research, in general, is underdeveloped as of the time of writing. New contributions to the field are therefore highly desired, especially to solve crucial problems such as saving lives. This thesis proposes a new 3D US internal bleeding detection framework: the decimation-analysis (D-A) framework. Rather than a standalone algorithm in its own right, D-A is designed as a modular layout that can be built upon and refined to create new detection algorithms; ideally, this would help advance the field of 3D US research and provide a structure for more accurate and meticulous algorithms than those currently in the field.
The structure and operation of D-A is largely based on [36], but uses a two-step decimation-and-analysis approach to impose an (ideally) more accurate and meticulous structure. Decimation is performed to identify suspect regions that possess basic properties of blood pools, and analysis is performed to directly assess shape information in these suspect regions. Ideally, this approach would improve on previous approaches by operating in 3D space, and by imposing two layers of “narrowing down” the volume: one based on intuitive and easily-identifiable features, and one that calculates and compares more complex shape information, whereas previous approaches tend to use only the former (such as [36]) or the latter (such as [14]).

1.2 Thesis Organization

This thesis is organized as follows:

- **Chapter 2: Background** - Presents background on ultrasound imaging, the FAST exam, and other existing methodologies. In particular, two existing methods for detecting internal bleeding in ultrasound scans are described and compared in detail, with emphasis on why a new system is required. Both alternative methods loosely follow a general framework that is described here as well.

- **Chapter 3: System Overview and Algorithm Design** - Provides a detailed description of the D-A framework proposed in this thesis. Begins with a general system overview and an explanation of feature extraction, then provides a step-by-step breakdown with details and justification for each step/procedure.

- **Chapter 4: Results and Discussion** - Explains the experimental setup used to test the algorithm, provides results from the setup, and discusses their implications and ramifications.

- **Chapter 5: Conclusion and Future Work** - Summarizes the thesis and discusses future work that can be derived from the D-A structure.
Chapter 2

Background

Ultrasound is the diagnostic method of choice for identifying internal bleeding in hemodynamically unstable patients. The primary concern of this thesis is how to detect such bleeding. The blood (a free fluid) collects in blood pools, which are typically located in gaps between organs and can be identified via distinct properties (Fig. 2.1):

- **Intensity**: Free fluid has low intensity (appears darker) due to low echogenicity.
- **Gradient**: Blood pools have strong gradients at their edges due to the surrounding organ boundaries and corresponding changes in brightness.
- **Curvature**: If the gap between surrounding organs has a known general shape, the suspect region’s local curvature can be analyzed.
- **Location**: If organ locations are already known, the gap between them (and therefore the possible location of the blood pool) can be located more easily.

A proper set of characteristics must be determined and chosen for blood pools to distinguish them from artifacts such as fluid-filled organs and acoustic shadows. Some methods, such as the FAST exam, rely on manual observation and identification of internal bleeding after scans have been taken. Such manual procedures suffer from caveats such as knowledge requirements that make them unsuitable for hemodynamically unstable patients, thereby nullifying a key advantage of ultrasound. Computer-aided diagnosis (CAD) is essential to mitigate these caveats, and facilitates early identification of bleeding symptoms[25]. CAD can be performed with 2D or 3D ultrasound, and can segment out regions of interest determined by quantitative data or by direct analysis of local shape information.

This section provides a review of previous work in the field. The D-A framework proposed in this thesis provides a means of implementing a two-stage process (“coarse” decimation followed by “fine” analysis) that combines multiple approaches to produce refined results.

2.1 The FAST Exam

The FAST exam is a widely-used routine for diagnosing trauma using ultrasound, specifically to detect internal bleeding in the form of free fluid[7] [11]. In order to detect the fluid, the ultrasonographer takes up to six internal 2D scans of the patient’s abdomen (Fig. 2.2). Out of these views, the right upper
quadrant (RUQ) view, also known as Morison’s Pouch, shows more visible differences if free fluid is present[18].

Figure 2.2: Locations of four common scan regions in the FAST exam[11]; the Morison’s Pouch view is most important for detecting internal bleeding.

FAST is preferred in emergency situations due to its portability, and trauma physicians and ultrasonographers are typically provided with proper training[35]. However, when injuries occur in remote regions such as battlefields, such personnel is often unavailable; only paramedics are sent to the scene[18]. For the FAST exam to be conducted properly, the operator must possess the knowledge required for proper placement of the ultrasound probe, proper identification of internal organs, and proper distinction between free fluids and fluid-filled organs[18] - knowledge that paramedics typically lack.
Trauma patients must therefore be transported to a hospital where proper trauma physicians and ultrasonographers are available. Transportation time causes a lengthy delay between initial treatment and proper diagnosis, which can result in death due to hemodynamic instability[18]. Computer-aided diagnosis can mitigate this by providing automatic aid for probe placement and detection of free fluid within; this thesis, and the D-A framework herein, focus on the latter.

2.2 Automated Free Fluid Detection

Prior research exists for internal bleeding detection algorithms for both 2D and 3D ultrasound, moreso the former. Fig 2.3 depicts the major steps (modules) that are followed by most such algorithms, for both development and operation. The development and operation phases can be summarized as follows:

- **Development**: In order to identify internal bleeding, the algorithm must know what characteristics are desired in the first place. Training is therefore necessary in order to define these characteristics in a quantitative form, whether as a template for comparison or as a desired set/range of values.

- **Operation**: Once the desired features have been defined, the algorithm is ready for operation. A query image or volume serves as the main input. The algorithm must analyze the query image/volume to determine the presence of desired features.

![Diagram of the general steps and modules for most internal bleeding detection algorithms.](image)

Figure 2.3: Diagram of the general steps and modules for most internal bleeding detection algorithms. The development steps (left) are used to produce feature templates/values that are used to assess results for the operation step (right).

Data must be properly prepared for the main steps of both development and operation. In many cases, the data is provided in a raw file format that must be converted to a pixel/voxel image or volume format. Afterwards, pre-processing takes place, in which images/volumes are enhanced to compensate for disruptive factors such as speckle noise and artifacts.
2.3 Pre-processing

In most cases, raw ultrasound data is not fit for direct analysis. This can be due to a number of factors:

- Medical ultrasound imaging inherently suffers from speckle noise due to random scattering off different types of tissue. This has negative impacts on both spatial resolution and image contrast, and makes structures more difficult to detect[18][17].

- Acoustic shadowing can occur as a result of absorption or scattering off various structures, including ribs[18][26].

- Some areas, mostly organs such as the kidney and liver, physically cannot contain free fluid and analyzing them is a waste of processing time.

Most detection algorithms therefore incorporate a pre-processing module to mitigate the above factors. However, excessive denoising and pre-processing in general can result in the loss of salient information such as important shapes and edges (Fig. 2.4). A balance must therefore be found between eliminating harmful interference and preserving salient features.

Pre-processing almost always incorporates some form of denoising. Different detection algorithms use different denoising procedures, usually dependent on the needs of the algorithm. These methods are often interchangeable; a breakdown and comparison of different denoising methods can be found in Appendix A.

After denoising, shadows and major organs can be identified and/or segmented in order to mitigate their effects and/or remove them from the scan. This is a field of research in its own right, as can be seen in Noll et al.[26], but can easily be used as a supplement within the D-A framework if available.

2.4 Prior work

Once volumes are pre-processed, different approaches can be taken to detect blood pools (and, prior to doing so, to extract and store training data). Some approaches are based on quantitative analysis of local features, while other approaches use region-based segmentation. More sophisticated approaches consist of two stages of analysis (a “coarse” and a “fine” stage) in order to refine results and provide other advantages.

D-A is a two-stage approach that uses quantitative analysis of image features. As such, two prior algorithms have been designated as “competing” methods. The first, by Zagrodsky et al. (2007), is a one-stage approach that maps image features to a “feature space” where specific regions correspond to blood pools[36]. The second, by Kim et al., is a two-stage approach that uses two different types of segmentation to provide more results than using either stage in isolation[14]. D-A attempts to draw on the advantages of each of these two methods while addressing their respective shortcomings.

2.4.1 Feature Space Clustering Method (Zagrodsky et al.)

Zagrodsky et al. developed an algorithm that uses a “feature space” to determine internal bleeding in 2D ultrasound scans[36]. Each pixel of the image is analyzed for three features: local intensity, local gradient, and curvature (Fig. 2.5). Curvature is determined using two of the moment invariants determined by
Figure 2.4: Example of the effects of speckle noise and denoising on ultrasound imaging.

(a) Sample image (left) and raw noisy US image (right)[21].

(b) US image denoised with a median filter of increasing window size from a) to e)[21].
Hu (1962)[9][28]. These three features are mapped to a “feature space”, a three-dimensional space where coordinates of a point are defined by the values of each of the three features (Fig. 2.6). $k$-means clustering is then performed, in which the feature space data is grouped into $k$ number of arbitrary regions. An “arbitrary threshold” is used to define feature values corresponding to internal bleeding. If a “significant part” of a cluster is located within this threshold, then internal bleeding is present. The cluster in question can then be mapped back to image space to observe the location of the blood pool.

Zagrodsky’s system is intuitive and easily understood, due to the fact that it is based on tangible properties of internal bleeding (as defined by the authors). The features that it assesses are highly reproducible, and it does not rely on the spatial location of any blood pools within the query volume.

However, the algorithm does not exactly match the needs of this project, as it is designed for 2D ultrasound rather than 3D. Due to the fact that it operates in feature space, the system does not take local geometry into account, and requires arbitrarily defined thresholds (presumably determined from sample data) as opposed to being able to discover and process local image information. This leads to additional problems due to its lack of reproducibility: the terms “arbitrary threshold” and “significant part” are undefined, thus providing no information on how to accurately analyze the feature space for internal bleeding. Ultimately, feature space clustering is left as an abstract and complex concept that is difficult to replicate usefully.

Aside from the obvious advantage of operating in 3D space rather than 2D, D-A draws on the

\[ \text{Figure 2.5: Feature images used to define blood pools in Zagrodsky’s method[36].} \]

\[ \text{While Hu’s moment invariants have been translated to 3D[31], they have not been used for this thesis in order to preserve the original 2D concept of Zagrodsky’s algorithm.} \]
main strength of Zagrodsky’s method by using basic blood pool features in a similar, intuitive manner: intensity and gradient are used for decimation, and feature extraction is used for analysis in a similar manner to Hu’s moments. D-A in fact builds on this simplicity and intuitivity by interpreting and using these features in a more straightforward manner than feature space clustering: features are first used to separate the volume into suspect and non-suspect regions, then a different set of features is extracted directly from the remaining regions.

2.4.2 Two-Stage Level Set Method (Kim et al.)

Numerous internal bleeding detection algorithms are designed around the concept of two stages: a “coarse” step provides a general idea or rough identification of regions of interest, which is then refined into more specific observations using a “fine” step. An archetypical example is provided by Leibe et al. (2006); while not specifically designed for medical image processing, an object categorization method is presented that recognizes and then segments shapes of interest[16]. Marsousi (2013) is a more specific application designed for internal bleeding in 3D US: it first uses active contours (“snakes”) to provide an initial segmentation result, which is then followed with level set function is used to improve the accuracy of the segmented region[19].

While Marsousi’s method fulfills the exact task required by this thesis and presents a viable competitor to D-A, Kim et al. (2014) presents a simpler alternative[14] that serves the same purpose. This method attempts to segment out regions of interest (i.e. blood pools) using region growing and level set methods, which correspond to “coarse” and “fine” segmentation steps, respectively. The first step (“coarse segmentation”) uses minimum-variance region growing (MVRG)[29] to provide an approximation of the blood pool region. While MVRG is simple, fast and does not heavily depend on the location of an initial seed point, it is inaccurate around the edges of blood pools due to the local increase in intensity variance[14]. Therefore, a second step (“fine segmentation”) is necessary, which employe Chan-Vese level set (CVLS) methods[2][5]. On its own, CVLS would be less than ideal for the task due to its dependence on an initial seed point, as well as its processing complexity. However, the initial region provided by MVRG addresses both of these issues by providing a pre-segmented region. The combination of coarse and fine segmentation provides more accurate results than either MVRG or CVLS on its own (Fig. 2.7).
Kim’s method operates directly in the 3D image space, which allows it to take local geometry and image features into account. Its two-stage layout also allows for more accurate and refined results than single-stage segmentation procedures. In addition, the use of MVRG allows the initial seed point to be placed anywhere, which eliminates the cost of manual seed point selection without sacrificing accuracy.

However, Kim’s method suffers from an essential flaw: it does not consider the actual characteristics of blood pools, unlike with [36]. As such, in theory, Kim’s method could mistakenly segment fluid-filled organs or even acoustic shadows rather than actual regions of interest. Additionally, both the MVRG and CVLS steps require “ground truth” in order to operate - when changing the shape of the region, both methods compare it to a pre-defined region (a pre-segmented blood pool from a training volume) for similarity. While this is not as problematic as requiring expert knowledge during the diagnosis itself, expert knowledge is necessary when providing a training set for the algorithm; in turn, the training set must be exhaustive enough to be able to cover the different shapes and sizes of blood pools for the view in question. This also requires the blood pool to occupy similar spatial locations in the query volume and at least one of the volumes used for training.

D-A provides the same coarse-to-fine layout as Kim’s method, and is designed to provide the corresponding advantages in terms of accuracy and refinement. While D-A suffers from the same requirement of accurate “ground truths” (which should be heavily considered for future derivative work), it addresses the main shortcoming of Kim’s method in that it directly accounts for blood pool characteristics in a
similar manner to Zagrodnsky’s method[36].

2.4.3 Other methods

Two other internal bleeding detection methods were noted during production of the thesis. These methods were not used as “competing methods” against which to compare D-A, each for their own reasons.

Low-brightness analysis (Ito et al.)

Ito’s method, as explained in [11], detects blood pools by searching for low-brightness areas between the liver and kidney. It consists of two main modules: low-brightness extraction and organ segmentation, which are independent of each other. Low-brightness extraction (Fig. 2.8a) uses a specialized 8-directional filter to detect and assess local changes in brightness around each pixel. Organ segmentation (Fig. 2.8b) is performed via region growing; an operator manually marks two seed points, after which they are grown until they touch edges (identified via edge extraction). Each of the seed points spawns the liver or the kidney, which are assessed in terms of extremal values to determine which organ is which.

![Figure 2.8: Results for each of the two parallel modules in Ito’s method[11]: (a) low-brightness filtering, (b) organ area segmentation.](image)

In general, Ito’s method appears to be underdeveloped and limited compared to other methods in the literature. Its main advantages are that it operates directly in image space and accounts for organs directly. However, it shares some shortcomings with Zagrodnsky’s method in that it is designed for 2D ultrasound and does not account for shadows. In fact, Ito’s method is even more limited due to its reliance on manual input, a shortcoming directly addressed by Kim’s method. The algorithm would not also function as desired because other “edges” exist in most scans besides organ boundaries, and would be inappropriate for scans that do not contain the liver and kidney. Finally, much like Zagrodnsky’s method, the most significant shortcoming is the lack of reproducibility: it is not stated how internal bleeding is detected after delineating low-brightness areas and major organs. As such, Ito’s method was not considered for the project due to its large amount of caveats.
Artifact elimination (Noll et al.)

Noll’s method[26] mainly focuses on pre-processing, and operates on 3D volumes. It attempts a sophisticated approach by directly addressing artifacts that interfere with analysis: shadows and major organs. Noll’s method compensates for shadows and distinguishes them from shadows by approximating the intensity of the ultrasound rays used to create the scan. Afterwards, the kidney is segmented by analyzing the scan slice-by-slice and generating a “heat map”, and the liver is segmented by using region growing to delineate its blood vessel structure and then using a convex hull to extract the parenchyma. Visualizations of these steps can be found above in Figure 2.9. Noll then suggests a non-specific method for segmenting the free fluid after shadows and organs have been dealt with.

Noll’s method differs strikingly from the methods previously discussed in that it focuses on simplifying the actual task of free fluid detection, as opposed to proposing a method for the detection itself. In fact, [26] directly states that the step had not yet been actively tested due to time constraints. However, the method is very exhaustive in how it deals with unwanted artifacts, and operates directly in image space and 3D space. As such, Noll’s method is very appropriate to use in conjunction with a well-developed free fluid detection algorithm, rather than being used as a detection method in its own right.
Chapter 3

The D-A Framework: Overview and Design

This chapter describes the author’s main contribution: the *decimation-analysis* (D-A) framework, an algorithmic layout that combines simple, intuitive features of blood pools with more complex shape descriptors. The main idea here is to reduce search space by thresholding query volumes according to basic blood pool properties (such as regional intensity), then to analyze shape information *only* over that reduced search space. In doing so, the D-A layout combines concepts from prior work:

- As in [36] and [11], the D-A layout refines search space by using basic, intuitive properties of desired features.
- As in [14], the D-A layout uses a “coarse” step (thresholding) in order to aid the operation and results of a “fine” step (shape descriptors).

The D-A should be primarily noted for its concept and structure rather than the algorithm’s actual performance. As described in Chapter 4, when testing the layout, many components were rudimentary due to time, experience and scope constraints.

3.1 System Overview

Fig. 3.1 depicts the general steps that were undertaken for development and operation of the D-A framework. Green boxes denote inputs, blue bubbles denote steps, yellow boxes denote outputs from development that are passed to operation as inputs, and the red box denotes final output. Prior to operation, sample data must be prepared from a “training set” of sample volumes so that it may be compared to query data. This sample data consists of thresholding limits for image features within the blood pool (*pre-decimation*), and of a “library” of shape descriptors that encode the shapes of blood pools (*pre-analysis*). During operation, a query volume is loaded and thresholded. It is thresholded based on the sample data (*decimation*), and any shape descriptors within thresholded regions are extracted and compared to the descriptor “library” (*analysis*).
3.2 Background

To provide an understanding how 3D data is processed through D-A, an explanation of “synthetic” 3D volumetric data, and how to divide it into voxels and planes, is necessary. Additionally, the two main concepts behind the D-A framework consist of image thresholding (for decimation) and feature extraction (for analysis). Each of these two concepts shall be described here, with the latter in particular requiring explanation due to its complexity.

3.2.1 3D Data

As alluded to in Chapter 1, while D-A is nominally designed for 3D volumes, a distinction must be made between true 3D and synthesized 3D. True 3D ultrasound works in a fully three-dimensional space; however, it is a new development and is not readily available for research at this level. D-A is designed for synthesized 3D volumes, in that they consist of an array of two-dimensional planar slices. While the resulting array of voxels can be explored in all three dimensions (Fig. 3.2), the frame cuts consist of data along a specific plane, which will henceforth be labeled as the XY plane; this is an essential fact in considering how 2D denoising methods can be applied to the volumes.

3.2.2 Thresholding

Image thresholding is a procedure in which images are analyzed pixel-by-pixel (or voxel-by-voxel) for whether each element has feature values (such as for intensity or gradient) that fall within a specified range (Fig. 3.3). In this context, thresholding is used to create a binary “mask” volume $\tilde{M}$ that can be used to isolate suspect regions. Within the mask, each voxel at position $\vec{x}$ is assigned a value of 1 or 0:

$$M(\vec{x}) = \begin{cases} 
1 & \text{lowlim}_f \leq M^f(\vec{x}) \leq \text{hilim}_f \\
0 & \text{otherwise}
\end{cases} \quad \forall \vec{x} \in V$$
Figure 3.2: Demonstrating how the data volumes can be traversed along any of 3 dimensions to obtain different 2D frame cuts.
where \( V \) is the original volume, \( M^F \) is the matrix of feature values (depicted as a feature image or feature volume), and \( \text{lowlim}_f \) and \( \text{hilim}_f \) are the lower and upper limits, respectively, of the feature value.

![Sample XY volume slice.](image1.png)  
![Thresholded “mask” version of slice.](image2.png)

Figure 3.3: Example of intensity-based thresholding on an XY cross-section of a sample volume.

In the context of this thesis, thresholding serves two main purposes: to ensure that any regions analyzed possess one or more basic features of blood pools, and to reduce search space to increase efficiency. Intensity-based thresholding was performed in some form in both [36] and [11], and is commonly used for free fluid detection; blood and similar fluids show up as dark on ultrasound scans due to low echogenicity, and regions can therefore be classified as suspect if their intensity falls below a certain threshold.

### 3.2.3 Feature Extraction

Feature extraction is a process used to locate salient features in an image and to encode and extract their shape information[3]. It consists of two main steps: detection and description. Detection produces keypoints that indicate the locations of salient features[6], and description produces descriptors that correspond to each keypoint and encode local shape information in vector form. The D-A framework uses a three-dimensional variant of the Speeded Up Robust Features (SURF) algorithm[15], whose detection and description steps shall be explained here. Figure 3.4 depicts a simple conceptualization of feature keypoints (red) and descriptors (yellow). Note that this depiction is highly simplified for comprehension purposes; in most cases, hundreds or thousands of keypoints are located, and descriptors are of very high dimensionality.

#### Feature Detection

Feature detectors use a specialized function to pinpoint the locations of “distinctive” or “discriminative” features, typically locating keypoints at the local maxima of the function in question. While there is no clear-cut definition for what constitutes “distinctiveness” or “discrimination”, features that present rich visual information in well-defined spatial locations are generally preferred[23]. In the context of this thesis, this would ideally refer to distinctive shapes that characterize a blood pool, such as the cavity between the liver and kidney; however, more research and testing would be required to find shape
detectors and descriptors that identify the specific desired features, and what said features would consist of.

Images possess both global and local features (Fig. 3.5). Global features are based on an assessment of the entire image, and provide a summary of properties such as outline shape, overall colour, and texture[34][37]. However, they are not well-suited to this project; all ultrasound scans in the dataset are superficially similar, whether or not they contain internal bleeding, and the blood pools are small and diverse enough to not make a significant contribution to global features. Additionally, occlusion and clutter can have a severe impact on the detectability of global features[34]. Local features, on the other hand, are based on a direct assessment of image patches around points of interest. They are better-suited to blood pool detection not only due to the nature of blood pools, but because they are more robust to occlusion and clutter[34].

Feature Description

After keypoints are located via feature detection, feature description must be performed. Feature descriptors project image data onto vector space[10] to provide a vector representation of local image information[6][23]. These descriptors are constructed using specialized functions such as image statistics or transforms[13].

For this project and similar contexts, feature extraction requires prior training in order to construct a “library” of keypoints and descriptors from a training set. This “library” consists of desired shapes and features that are matched to query shapes (or vice-versa) via their respective descriptor vectors. One way to perform this “matching” is to calculate and compare the distances between different library-query
Figure 3.5: Illustrating the difference between global and local features.
descriptor pairs. Similarity and “best matches” are assessed via this distance measure.

The SURF Algorithm

The current incarnation of the D-A framework uses SURF, or Speeded Up Robust Features[1][32], for feature extraction. SURF uses an approximation of local Gaussian matrices for feature detection. Known as the “Fast-Hessian Detector”, the function uses the determinants of local Hessian matrices for the approximation and locates keypoints at the local maxima of the corresponding function. For feature description, SURF uses Haar wavelet responses to construct descriptors of very high dimensionality.1

A three-dimensional variant of SURF[15] was presented to the author and used for the project as its feature extraction algorithm; due to the complexity of such feature descriptors, a performance evaluation of similar algorithms was beyond the scope of the thesis. Future derivatives of this work may substitute and compare different algorithms, particularly ones that are more specifically tailored to the needs of the project.

3.3 Module Descriptions

This section describes each of the five modules of the D-A framework: preparation, pre-decimation, pre-analysis, decimation, and analysis. Each module is explained in the form of a flowchart.

3.3.1 Pre-Processing and Feature Volume Creation

Figure 3.6 depicts the sequence of preparation steps that must be performed on each volume. For both development and operation, the initial step is to load volumetric data from memory. Once loaded, speckle noise must be suppressed, due to the fact that all volumes suffer from its effects. However, if denoising is too strong, then salient features can be accidentally erased or blurred out. Additionally, the vast majority of denoising filters are designed for two-dimensional images. A balance must therefore be found between eliminating noise and preserving salient features; see Appendix A). In order to assess different image features (such as intensity, gradient, curvature), feature volumes must be constructed. These follow the same principle as the feature images from [36], in that each voxel is assigned a value based on the magnitude of the respective feature at that point. As in [36], volumes can be dilated with a round kernel after denoising in order to merge contiguous features. As with denoising, salient features must be preserved, and therefore the type and size of kernel must be chosen appropriately. Fig. 3.7 demonstrates each of these steps via a single XY frame cut.

3.3.2 Development

Developing the D-A framework requires the use of training volumes - referred herein as the set T of volumes \( V_n \) - to collect information on blood pools. Two types of information must be collected: feature thresholding limits \( \text{lowlim}_f \) and \( \text{hilim}_f \) for each feature \( f \) in the set \( F \) of features, and a library \( L \) of shape descriptor vectors (SURF in this case). This, in turn, requires blood pools within the training set to already be segmented. Ideally, segmentation would be performed by an expert for reasons explained in Chapter 1, but for this project, the author created rough approximations of “ground truths” via manual

---

1Throughout this thesis, SURF descriptors are not represented visually. This is because of their abstract nature, arising from their very high dimensionality and complexity.
Figure 3.6: Pipeline of preparation steps for D-A.

(a) Sample XY volume slice.
(b) Denoised volume slice.
(c) Dilated volume slice (intensity feature volume).
(d) Dilated volume slice (gradient feature volume).

Figure 3.7: Demonstrating the steps of pre-processing.
segmentation, as explained later in Subsection 4.1.1. Ground truth masks $GT_n$ for each volume $V_n$ are defined by the following for each voxel position $\vec{x}$ (Fig. 3.8):

$$GT_n(\vec{x}) = \begin{cases} 
1 & \text{if } \vec{x} \text{ is in blood pool} \\
0 & \text{otherwise} 
\end{cases} \quad \forall \vec{x} \in V_n$$

The resulting set of ground truth masks shall be referred to as $T_{GT}$.

**Pre-Decimation: Determining Thresholds**

Fig. 3.9 depicts the pre-decimation process. To prepare this module, the threshold limits $lowlim_f$ and $hilim_f$ must be determined for each feature $f$ during development. These two limits should represent a reasonable range of feature values that would be found in blood pools. The first step is to obtain “feature volumes” $FV_{f,n}$ for each feature $f$ and each original volume $V_n$, in a manner akin to [36] (Fig.). Afterwards, the set of all values $featvals_f$ of each feature $f$ can be collected within the regions segmented by $T_{GT}$:

$$featvals_f = \bigcup \{(FV_{f,n}(\vec{x}) : GT_n(\vec{x}) = 1) \forall \vec{x} \in GT_n \forall GT_n \in T_{GT} \quad (3.1)$$

where $\vec{x}$ denotes voxel position (feature values are collected voxel-by-voxel).

$lowlim_f$ and $hilim_f$ can then be determined by operating on $featvals_f$. One simple example (used for the project) is to use mean plus/minus standard deviation:

$$lowlim_f = \mu_{featvals_f} - \sigma_{featvals_f}$$
$$hilim_f = \mu_{featvals_f} + \sigma_{featvals_f} \quad (3.2)$$

Such simplistic limits would not be recommended for a more thorough application of the D-A framework. Ideally, $lowlim_f$ and $hilim_f$ should be based more directly on the distribution of feature values within the blood pools.
Pre-Analysis: Generating the Descriptor Library

To prepare the analysis module, a “library” $L$ of shape descriptors is necessary. Fig. 3.10 depicts the pre-analysis process that goes into creating this library. Ideally, these descriptors would cover and encompass shape information that would be found within blood pools. The first step is to generate all keypoints $k$ and descriptors $d$ for each training volume $V_n \in T$, resulting in the corresponding sets $K_n$ and $D_n$ which are in turn stored in the banks $T_K$ and $T_D$ over all volumes. Afterwards, the descriptors are stored in the library depending on whether or not their corresponding descriptors could be found within the segmentations provided by $T_{GT}$ (Algorithm 1, Fig. 3.11):

**Algorithm 1:** Generating SURF descriptor library for the analysis step

<table>
<thead>
<tr>
<th>Data: segmentation masks $T_{GT}$, SURF keypoint banks $T_K$, SURF descriptor banks $T_D$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Result: SURF descriptor library $L$</td>
</tr>
<tr>
<td>initialization;</td>
</tr>
<tr>
<td>for each $K_n \in T_K$ do</td>
</tr>
<tr>
<td>for each keypoint $\hat{k}$ in $K_n$ do</td>
</tr>
<tr>
<td>if $GT_n(\hat{k}) = 1$ then</td>
</tr>
<tr>
<td>fetch corresponding descriptor $\hat{d}_k$ from $D_n$;</td>
</tr>
<tr>
<td>add $\hat{d}_k$ to library $L$;</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>end</td>
</tr>
</tbody>
</table>

3.3.3 Operation

Once sample data has been collected and stored, the D-A framework is ready to operate. A single query volume $V_n$ is loaded and assessed. In decimation, the query volume is reduced to suspect regions whose feature values fall within thresholds that suggest internal bleeding. In analysis, the SURF descriptors of those regions are extracted and compared to the library $L$ via a distance measure $s_{mea}$. If $s_{mea}$ meets one or more conditions (such as falling below a threshold), then a match is present, and internal bleeding
is detected.

**Decimation**

The decimation step (as laid out in Fig. 3.13) consists of automatic thresholding, using the threshold limits found in Equation 3.2. Assuming that volume $V_q$ is the query volume, the decimation step creates a new segmentation mask $M_n$ based on whether each voxel at position $\vec{x}$ falls within the thresholding range for each feature (Fig. 3.12):

$$M_n(\vec{x}) = \begin{cases} 
1 & \text{lowlim}_f \leq F_{f,n}(\vec{x}) \leq \text{hilim}_f \forall f, \forall \vec{x} \in V_q \\
0 & \text{otherwise}
\end{cases}$$

(3.3)

**Analysis**

The analysis step, as visualized in Fig. 3.14, involves obtaining all SURF keypoints $K_q$ and descriptors $D_q$ over the volume $V_q$, then applying results from the decimation step to reduce the set. In this way,
Figure 3.12: Visualization of automatic thresholding via decimation; note how this does not clearly isolate the blood pool as with Fig. 3.8.

Figure 3.13: Pipeline of decimation for D-A.
only descriptors corresponding to keypoints that “survive” the decimation would remain (Algorithm 2, Fig. 3.11 above):

![Pipeline of analysis for D-A.](image)

**Algorithm 2:** Reducing set of SURF descriptors to those within $M_q$ thresholded region

<table>
<thead>
<tr>
<th>Data:</th>
<th>segmentation mask $M_q$, SURF keypoint bank $K_q$, SURF descriptor bank $D_q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Result:</td>
<td>Thresholded SURF descriptor bank $D_{tq}$</td>
</tr>
</tbody>
</table>

Initialization;
load $K_q$;

for each keypoint $k$ in $K_q$ do
  if $M_q(k) = 1$ then
    fetch corresponding descriptor $d_k$ from $D_q$;
    add $d_k$ to bank $D_{tq}$;
  end
end

Afterwards, each descriptor in $L$ is matched pairwise to its nearest neighbour in $D_{tq}$, and a function $s_{mea}$ of the distance between nearest-neighbour pairs is used to determine matches. Nearest neighbours and the distance between them are common performance evaluators for feature matching algorithms[22]. For this project, the “minimum average distance” $\mu_s$ was chosen for $s_{mea}$ as calculated below, but can easily be substituted with a different, possibly more appropriate measure.
**Algorithm 3:** Matching vectors to determine internal bleeding

<table>
<thead>
<tr>
<th>Data: SURF descriptor library $L$, SURF descriptor bank $Dt_n$, distance threshold $s_{thres}$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Result:</strong> is internal bleeding present?</td>
</tr>
<tr>
<td><strong>for each descriptor $\vec{l}$ in $L$ do</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>end</strong></td>
</tr>
<tr>
<td>calculate mean value $\mu_s$ over all $s$;</td>
</tr>
<tr>
<td><strong>if</strong> $\mu_s \leq s_{thres}$ <strong>then</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>else</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>end</strong></td>
</tr>
</tbody>
</table>
Chapter 4

Testing and Results

The D-A framework was tested and compared to author-programmed facsimiles of Zagrodky’s[36] and Kim’s[14] internal bleeding detection algorithms. Results were obtained by testing each algorithm with all 49 data volumes that were provided; this query set includes the same volumes used in the training set $T$, as they were not excluded from the query set due to the relatively sparse number of volumes that contained internal bleeding.

The original intention was to judge the success of the D-A algorithm by comparing its accuracy and speed to its competitors. However, results were deemed secondary to the structure of the algorithm itself; the components that were used to program D-A and its competing algorithms were largely rudimentary due to time and resource constraints. In particular, the pre-segmented blood pools used for the training sets were rough approximations, and some algorithms were pre-packaged and were too complex to warrant extensive modification and code investigation. Therefore, the performance analysis did not prove or disprove the integrity of the D-A setup; in order to do so, future research should build on it with more sophisticated and specific components.

4.1 Experimental Setup

The D-A algorithm and its competitors were tested in MATLAB using a 3D volumetric dataset provided by Defence Research and Development Canada[18]. An appropriate set of training volumes needed to be chosen, as well as certain specific parameters such as image threshold limits. The output for each algorithm was a single value used for binary threshold-based decision making; the threshold itself was left as an adjustable parameter. Performance was to be judged by calculating an arbitrary measure of accuracy, which was to be plotted against said threshold in order to observe when accuracy would be at its highest.

4.1.1 Dataset

The dataset consisted of 49 ultrasound volumes obtained using the GE Voluson e machine[18]. Each volume has an index number from 1 to 50.\footnote{A facsimile of Ito’s[11] algorithm was originally planned, but it was abandoned due to too many caveats, most notably an overreliance on human operator input.} The volumes represent scans of the abdominal region,\footnote{Volume 14 was absent upon receipt by the author.}
collected from eight healthy volunteers and eight “abnormal” ones with in-body fluids simulating internal bleeding.

The volumes were obtained using the GE Voluson e machine, and were initially in KRETZFILE format. Once loaded, the volumes were converted to 3D voxel arrays. Each voxel has an intensity that can be mapped to an integer value $y \in [0, 1, \ldots, 255]$, ranging from 0 (black) to 255 (white)[18].

An auxiliary document[20] describes several important details about each volume: size, presence of organs (liver/kidney), and presence of bleeding. 16 volumes are known to contain bleeding, and 29 volumes are known to be of the Morison’s Pouch’ view. The Morison’s Pouch view is of note because the liver and kidney are clearly identifiable, which in turn makes free fluid easier to identify using the assumption from [36] that it can be located adjacent to organs.

Training Set

As in Chapter 3, sample data needed to be collected for both D-A and its competing algorithms using a training set $T$ of volumes. Because the sample data would be extracted from blood pools, each volume in the dataset would have to contain a segmentable blood pool for ground truth data; the resulting segmented region shall be referred to herein as the “author-segmented ground truth” (ASGT). However, segmentation was limited for two reasons:

- Originally, full 3D manual segmentation was planned. The main option available was the TurtleSeg algorithm. The license to the software was lost, and 3D alternatives were not viable.
- No prior segmentation existed for the blood pools, and no information was given on what they looked like. No expert knowledge was available - a problem highlighted in Chapter 1.

Therefore, compromises were made:

- Blood pools were segmented and approximated via a limited number of 2D slices, segmented using MATLAB’s Image Segmenter app (Figure 4.1).
- The author (with limited medical imaging experience) had to identify blood pools manually, via inference.

Each ASGT was to be constructed from four XY-plane frame cuts, thereby severely limiting its representation of the actual ground truth. Nonetheless, by investigating all 49 volumes, “author-identified blood pools” (AIBPs) were defined as the following (Figure 4.2):

**Definition 4.1.1.** Author-identified blood pool (AIBP): An author-identified blood pool is a prominent, vaguely round dark blob with bright edges, typically located at organ boundaries.

Note that this definition, by the very nature of being author-defined, was highly subjective and highlights the necessity of more solid ground truth data. The above compromises in general are a major reason why less emphasis was placed on performance evaluation: segmentation data was based on limited resources and medical imaging knowledge, whereas the information in the companion document was likely based on a wider range of resources and expert imaging knowledge.

Two conditions and one goal were therefore imposed for $T$:

**Condition 1:** Each volume had to contain internal bleeding according to the companion document[20].
Figure 4.1: Segmenting an AIBP (Definition 4.1.1) in 2D to create one frame cut of an ASGT.

**Condition 2:** Each volume must contain exactly one AIBP as per Definition 4.1.1 and the author’s discretion.

**Goal 1:** The volumes should each be of different sizes, under the presumption that different sizes represent different internal views.

For initial selection, volumes were sorted by size in order to meet the “different sizes” goal (Table 4.1)\(^3\). Six volumes - indexed 01, 04, 06, 11, 13 and 18 - were initially selected. Volumes 01, 04, 11 and 18 contained a single prominent AIBP, but volumes 06 and 13 contained two (Fig. 4.2). Volumes 06 and 13 were therefore eliminated due to the risk of incorrectly identifying a non-bleeding artifact as an AIBP. Additionally, a large amount of prior statistics and segmentations existed from volumes 01, 11 and 18, but not 04. Volume 04 was therefore also eliminated from the training set due to time constraints; it was deemed appropriate to preserve the work as is rather than reiterate everything to add one additional volume - especially given the rudimentary nature of the project. For future derivative work, a more exhaustive training set would be preferable, especially with pre-segmented “ground truths” for the blood pools determined by an expert.

**Denoising**

Each of the dataset volumes had to be denoised prior to use, for development and operation of all algorithms. Two choices had to be made in this regard: “denoising dimensionality”, and denoising method. The objective was to preserve salient features while mitigating the effects of speckle noise as much as possible.

All denoising filters available were designed for use with two-dimensional images. Due to the synthetic 3D nature of the volumes (refer back to Fig. 3.2), the option to denoise each volume along one, two or three planes (“denoising dimensionality”) was present. By default, the XY plane was chosen as the

\(^{3}\)Some volumes (bold or italic) contain internal bleeding; bolded volumes were selected for the training set.
Figure 4.2: XY frame cut of each of the initial six training volumes, with AIBP’s highlighted in red. Volumes 06 and 13 contained two AIBP’s, and were eliminated.

<table>
<thead>
<tr>
<th>Size</th>
<th>Volume numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>197x183x129</td>
<td>01, 03</td>
</tr>
<tr>
<td>197x181x125</td>
<td>02</td>
</tr>
<tr>
<td>197x175x119</td>
<td>04, 05, 06, 07, 08, 09, 10, 13, 17</td>
</tr>
<tr>
<td>178x240x178</td>
<td>11, 23-32, 33, 34-50</td>
</tr>
<tr>
<td>197x177x117</td>
<td>12</td>
</tr>
<tr>
<td>197x181x119</td>
<td>15</td>
</tr>
<tr>
<td>197x211x141</td>
<td>16</td>
</tr>
<tr>
<td>195x247x133</td>
<td>18, 21, 22</td>
</tr>
<tr>
<td>189x247x135</td>
<td>19</td>
</tr>
<tr>
<td>197x221x121</td>
<td>20</td>
</tr>
</tbody>
</table>

Table 4.1: List of dataset volumes sorted by size.
“first” plane, due to the fact that the volumes were constructed from slices along the XY plane. The XZ and YZ planes designated as the “second” and “third” planes, respectively. Each level of denoising dimensionality was tested, and results were observed along all three planes. In the end, single-plane (XY) denoising was chosen; denoising along additional planes did not help with the preservation/mitigation objectives and even caused loss of detail in some cases.

Multiple denoising methods were tested: median filtering, ideal filtering, Butterworth filtering, wavelet filtering, and MaxPol filtering[8]. A detailed comparison and log of methods, as well as visual results, can be found in Appendix A; median filtering was selected as the method of choice.

4.1.2 Setup of Each Algorithm

All algorithms - D-A, Zagrodsky, Kim and Ito (abandoned) - needed to be programmed and set up module by module. Each module could be created and adjusted in a number of ways, and this section details the processes and choices behind each major module.

D-A (Vukovich)

As described in Subsection 3.3.1, volume dilation is recommended for the D-A framework in order to facilitate coincidence of features, as in [36]. Kernel size had to be chosen in order to merge contiguous features while preserving larger features. A spherical kernel of size 5 was chosen; refer to Fig. 3.7.

For pre-decimation, a set of feature volumes had to be generated. Two of the three features from [36] - regional intensity and regional gradient - were chosen, due to the fact that they can easily be associated with the low echogenicity and bright edges of a blood pool. The third feature (Hu moments) was not used, due to the fact that the analysis module served a similar purpose (shape information). The thresholding limits were calculated according to Equation 3.2.

During the decimation step, thresholding was initially performed with statistics from both feature volumes (intensity and gradient). However, when performed in conjunction with the analysis step, the descriptor library \( L \) would be empty. As a result, it was determined that using two feature constraints was too restrictive for such a rudimentary algorithm, and only intensity thresholds were used. For future derivative work, gradient and/or other feature constraints can be easily re-added, once the algorithm has been refined to allow for proper functioning when doing so.

The analysis step, as well as pre-analysis, required a feature algorithm to be selected. As stated in 3.2.3, SURF was the method of choice. The experiment used a pre-packaged 3D SURF MATLAB algorithm based on [1] and [15]. Numerous adjustments had to be made to the code to avoid runtime errors. Due to the relative complexity of the code, it is unclear how these fixes affected the integrity of the algorithm. The code takes a very long time to generate filters in order to find SURF descriptors, which violates the principle in Chap. 1 of prompt diagnosis; it should be replaced with an alternative for future derivative work. k-nearest neighbour matching was performed using MATLAB’s built-in knnsearch algorithm.

Zagrodsky’s method

Fig. 4.3 describes the functioning of the facsimile of Zagrodsky’s algorithm[36]. Orange and red boxes denote inputs,\(^4\) blue boxes denote steps, and yellow boxes denote outputs. Pre-processing consisted

\(^4\)Red inputs were obtained from development
of denoising and dilation in the same manner as for D-A, using similar parameters. Likewise, feature volumes for intensity and gradient were obtained in a similar manner to D-A. Hu moment volumes were created using prepackaged MATLAB code.

As explained in Subsection 2.4.1, the two most subjective parameters to replicate were those of the “arbitrary threshold” for bleeding and “significant part” for a cluster. The “arbitrary threshold” was defined for all three features; those for intensity and gradient were identical to D-A, and those for Hu’s Moments were calculated according to the same equations (Equation 3.2). As with D-A, an automatic segmentation mask $M_n$ was created over all three of these features, as per Chap. 3. The “significant part” would denote the largest possible proportion of “bleeding voxels” to total voxels within a cluster:

![Figure 4.3: Pipeline for the facsimile of Zagrodsky’s algorithm.](image)

**Algorithm 4:** Calculating $p_{\text{max}}$ for the Zagrodsky facsimile.

<table>
<thead>
<tr>
<th>Data: set of all voxels $V$, automatic segmentation mask $M_n$, set of clusters $C$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Result: largest proportion $p_{\text{max}}$ of “bleeding voxels” to total voxels within a cluster</td>
</tr>
<tr>
<td>initialization;</td>
</tr>
<tr>
<td>create placeholder for $p_{\text{max}}$;</td>
</tr>
<tr>
<td>for each cluster $c$ in $C$ do</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>end</td>
</tr>
</tbody>
</table>

As the algorithm features $k$-means clustering, a small test was performed on one volume to observe the effects of the number of clusters $k$ on $p_{\text{max}}$. There was a slight upwards trend (Table 4.2); in the end, 8-means and 16-means clustering were chosen and compared.
<table>
<thead>
<tr>
<th>Number of clusters ($k$)</th>
<th>Maximum bleeding proportion ($p_{max}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0.6304</td>
</tr>
<tr>
<td>8</td>
<td>0.7370</td>
</tr>
<tr>
<td>12</td>
<td>0.7270</td>
</tr>
<tr>
<td>16</td>
<td>0.7510</td>
</tr>
</tbody>
</table>

Table 4.2: Values of $p_{max}$ for different values of $k$ for the Zagrodsky facsimile.

**Kim’s method**

Fig. 4.4 describes the functioning of the facsimile of Kim’s method[14], using the same colour-coding as 4.3. The algorithm required the use of minimum-variance region growing (MVRG) and Chan-Vese level set (CVLS) algorithms. MVRG could not be implemented due to time constraints as well as falling well outside the scope of the project, and a more general region-growing algorithm was used instead.

In technical terms, the initial seed point was an adjustable parameter; however, based on the premise from [14] that the seed point could be placed far from the region of interest, it was arbitrarily selected to be the centre voxel of each volume in order to minimize human input.

The output of the algorithm would simply be calculated as the Dice similarity coefficient $DSC$ between the resulting segmented region $R_{seg}$ and any of the author-segmented ground truths:

$$DSC_{best} = \max_{GT_n \in I_{GT}} DSC(R_{seg}, GT_n)$$

**Ito’s method**

While Ito’s method[11] was fully programmed, it was neglected from testing for numerous reasons:

- The algorithm depended on the existence of the liver and kidney within the query volume.
- The algorithm required manual placement of seed points.
- No information was provided in [11] on how to automatically assess internal bleeding; the assumption was that it would be identified visually as being a highlighted dark region between the highlighted organs.

Due to an overreliance on manual input and external conditions, Ito’s method was omitted from testing.
4.1.3 Performance Evaluation

The output of each algorithm was in the form of a value to be compared to an adjustable threshold (Table 4.3):

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Output Value</th>
<th>Meaning</th>
<th>Adjustable Threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td>D-A (Vukovich)</td>
<td>$\mu_s$</td>
<td>Min. mean distance</td>
<td>$s_{thresh}$</td>
</tr>
<tr>
<td>Zagrodsky</td>
<td>$p_{max}$</td>
<td>Max. bleeding proportion</td>
<td>$p_{thresh}$</td>
</tr>
<tr>
<td>Kim</td>
<td>$DSC_{best}$</td>
<td>&quot;Best&quot; Dice coefficient</td>
<td>$DSC_{thresh}$</td>
</tr>
</tbody>
</table>

Table 4.3: Output values and thresholds for each algorithm tested.

Afterwards, the binary bleeding indicator variable $B$ could be calculated using one of the corresponding relations from Equation 4.1:

$$
D-A: B = \begin{cases} 
1 & \mu_s \leq s_{thresh} \\
0 & \text{otherwise}
\end{cases}
$$

$$
\text{Zagrodsky: } B = \begin{cases} 
1 & p_{max} \geq p_{thresh} \\
0 & \text{otherwise}
\end{cases}
$$

$$
\text{Kim: } B = \begin{cases} 
1 & DSC_{best} \geq DSC_{thresh} \\
0 & \text{otherwise}
\end{cases}
$$

To determine true and false positives and negatives, $B$ was compared to the “true” bleeding status of the volume in question, as denoted by [20] and Table 4.1.

A measure of accuracy needed to be chosen and calculated over all 49 volumes. In order to properly balance the value of true positives (being able to correctly detect internal bleeding) and true negatives (avoiding unnecessary treatment of non-fatal wounds), the balanced accuracy $BACC$ was chosen:

$$
BACC = \frac{(TP/P + TN/N)}{2}
$$

where $TP$ is the number of true positives, $TN$ is the number of true negatives, $P$ is the number of total positives, and $N$ is the number of total negatives.

4.2 Results

Results were taken by calculating $BACC$ for each algorithm while varying each adjustable threshold from Table 4.4. The Zagrodsky and Kim facsimiles would produce highly varied results (Figure 4.5). However, D-A did not function as desired: all values of $\mu_s$ would come out as 0, thus undermining the significance of any results. This is likely due to the tampering done to the pre-packaged SURF code to prevent runtime errors when operating with the dataset.

In terms of speed, all three algorithms took several hours to complete over all 49 volumes. In the case of D-A, the SURF algorithm took several hours to run. Because SURF was developed to provide a faster alternative to other feature extraction methods[1], this ran counter to one of its purposes. The reason for this was, again, due to runtime error prevention: the prepackaged SURF code calculates Hessian filters of different sizes for each volume, and stores them in the computer’s internal memory.

---

5For the Zagrodsky facsimile, the highest BACC was obtained when $k = 8$. 
<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Highest BACC</th>
<th>Adjustable Threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td>D-A (Vukovich)</td>
<td>0.5</td>
<td>$st_{thresh} = 0$</td>
</tr>
<tr>
<td>Zagrodsky</td>
<td>0.6212</td>
<td>$pt_{thresh} = 0.76$</td>
</tr>
<tr>
<td>Kim</td>
<td>0.8617</td>
<td>$DSC_{thresh} = 1e-5$</td>
</tr>
</tbody>
</table>

Table 4.4: Highest BACC values obtained for each algorithm.

Figure 4.5: Calculating BACC over different thresholds for the Zagrodsky and Kim facsimiles. D-A, the most important algorithm here, was omitted due to the fact that $\mu_s$ was calculated as 0 in all cases.
When operating on a new volume, the code checks if a required filter already exists in memory in order to save processing time. However, this would result in runtime errors due to the fact that some filters would be incompatible with new volumes due to size, and the code had to be modified by the author to recalculate the filters every time, resulting in a very time-consuming algorithm.

4.3 Discussion

The D-A structure is recommended more as a layout or skeleton for internal bleeding detection algorithms, rather than a complete algorithm in its own right. In order to validate D-A, a proper performance evaluation is required. Numerous improvements can therefore be made to enable a meaningful evaluation.

By far the largest shortcoming of the experiment was the inability to access or create accurately segmented ground truths. This manifested itself as two major limitations:

- **Lack of expert knowledge**: No expert knowledge was available on how to identify or locate blood pools within the dataset.

- **Limited segmentation**: Ground truth segmentation was limited to a small number of 2D slices.

Both of these limitations resulted in an incomplete benchmark to gauge segmentation success. Without proper expert knowledge, no consistency was guaranteed between the empirical data provided in[20] and “similarity” to AIBPs, which was worsened by the limitations on segmentation.

The first limitation can be addressed by providing pre-segmented ground truths, or at least the proper information/context so that researchers may segment them. This would provide for a more justifiable measure of success and a more thorough and justified assessment of blood pool features, and would also prevent confusion of blood pools with similar artifacts. The second limitation can be addressed by providing access to proper 3D segmentation software, such as TurtleSeg. This would allow for blood pools and all their salient features to be fully included and accounted for in 3D.

Other details can also be improved on for future work. The choices for $f_{lim,f,low}$ and $f_{lim,f,high}$ as in Equation 3.2 were not based on a meaningful analysis of the data, and were merely “placeholders” used to test the structure of the algorithm. They should instead be determined by a more thorough statistical analysis of the feature data, which in turn would only be meaningful if obtained from accurate ground truths.

SURF was a rather arbitrary choice for feature extraction, and does not detect a specific class of features. Once again, accurate ground truths would enable an analysis of what shape features are truly desired in blood pools. A different feature extraction algorithm could therefore be selected based on these features which, once again, would be justified via accurate ground truths.
Chapter 5

Conclusion and Future Work

The D-A framework is built around three design points: operation in 3D space, simple and intuitive use of blood pool features, and “coarse” decimation followed by “fine” analysis. These three concepts would potentially allow for a simple, accurate bleeding detection method; on paper, it can be argued that D-A is a very sound framework for internal bleeding detection. It is therefore recommended as a foundation on which to build future work, but not without addressing the two key limitations described in Section.

It should also be noted that, with the additional information provided by accurate ground truths, an auxiliary document such as [20] does not necessarily need to be the basis for performance evaluation. Success can be measured by modifying the D-A algorithm to identify and highlight blood pools outright rather than simply using a numerical measure, and by comparing such segmentations to the original ground truth - a procedure performed for kidney segmentation in [18].

Once proper ground truths are attainable, more satisfactory results can be obtained with the D-A framework. Researchers do not strictly need to adhere to the modules and methods from this thesis; D-A is based on the concept of decimation followed by analysis, rather than any specific means thereof. [19], as discussed in Chapter 2, is a two-step approach functionally similar to Kim’s two-stage level set; it can therefore be used as the “analysis” step in a similar fashion, with “decimation” used to locate ideal locations for initial seed points.

Additional steps are also possible. Noll’s algorithm[26] provides a framework for segmenting organs and shadows, therefore improving the accuracy and decreasing the probability of error for potential feature extraction algorithms. Implementing this framework in conjunction with D-A would be very helpful in further reducing the search space, as well as eliminating artifacts that can be confused with blood pools by the system or by an untrained researcher.

The initial hope was for D-A to serve as a self-contained system that would improve on its competitors. However, it seems that the most valuable contribution of this thesis is the very concept of the D-A framework, as well as the organized two-step layout that it provides for future work. An ideal situation would be for researchers to not only build D-A into a more integral and better-performing algorithm, but to use its structure to introduce new ideas to the field of trauma diagnosis.
Appendices
Appendix A

Experimentation on Denoising Methods

This appendix demonstrates and compares the results of the denoising methods tested for this thesis. Each method was tested on dataset volume 33 (as indexed by [20]), on four XY-plane frame cuts located at z-axis indices 60, 80, 100 and 120 (as per the dimensions from [20]). As explained in the thesis, the main goals of denoising were twofold:

- to eliminate the effects of speckle noise, and
- to preserve salient features.

A.1 MaxPol homomorphic filtering in cepstrum domain

This is a denoising method described in [8]. It is classified as “homomorphic”, but differs from other homomorphic Fourier filtering methods (explained later in this Appendix) in that it applies the log transform after taking the FFT of the scan rather than before. In addition, it applies a 3D lowpass filter to the 3D volume as a whole rather than filtering slice-by-slice (Fig. A.1). The filter is constructed from three lowpass filter kernels, one each for the x, y and z directions. Each of these kernels is generated from centralized derivative coefficients.

Table A.1 demonstrates the effect of varying $P_{cutoff}$ from 2 to 18. In keeping with previous experimentation, performance was judged via the effects on the volume slices due to the fact that the volume was synthesized from scans along the XY plane. Setting $P_{cutoff}$ to very low values ($P_{cutoff} = 2$) introduced excessive blurring. Increasing its value ($P_{cutoff} = 4$ to 8) weakened the blurring effect and started to introduce details. Higher values ($P_{cutoff} = 10$ to 19) reduced noise significantly while preserving most details; however, lower values of $P_{cutoff}$ within the range (10 to 14) had the greatest denoising effect and did not obscure details much more than higher values within the range. Visually, setting $P_{cutoff}$ to 12 seems to produce the results that best meet the experimental goals.
**Figure A.1:** Processing pipeline for MaxPol lowpass filtering.

<table>
<thead>
<tr>
<th>$P_{cutoff}$</th>
<th>Frame 60</th>
<th>Frame 80</th>
<th>Frame 100</th>
<th>Frame 120</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td><img src="image" alt="Frame 60" /></td>
<td><img src="image" alt="Frame 80" /></td>
<td><img src="image" alt="Frame 100" /></td>
<td><img src="image" alt="Frame 120" /></td>
</tr>
<tr>
<td>2</td>
<td><img src="image" alt="Frame 60" /></td>
<td><img src="image" alt="Frame 80" /></td>
<td><img src="image" alt="Frame 100" /></td>
<td><img src="image" alt="Frame 120" /></td>
</tr>
<tr>
<td>4</td>
<td><img src="image" alt="Frame 60" /></td>
<td><img src="image" alt="Frame 80" /></td>
<td><img src="image" alt="Frame 100" /></td>
<td><img src="image" alt="Frame 120" /></td>
</tr>
<tr>
<td>6</td>
<td><img src="image" alt="Frame 60" /></td>
<td><img src="image" alt="Frame 80" /></td>
<td><img src="image" alt="Frame 100" /></td>
<td><img src="image" alt="Frame 120" /></td>
</tr>
<tr>
<td>8</td>
<td><img src="image" alt="Frame 60" /></td>
<td><img src="image" alt="Frame 80" /></td>
<td><img src="image" alt="Frame 100" /></td>
<td><img src="image" alt="Frame 120" /></td>
</tr>
<tr>
<td>10</td>
<td><img src="image" alt="Frame 60" /></td>
<td><img src="image" alt="Frame 80" /></td>
<td><img src="image" alt="Frame 100" /></td>
<td><img src="image" alt="Frame 120" /></td>
</tr>
<tr>
<td>12</td>
<td><img src="image" alt="Frame 60" /></td>
<td><img src="image" alt="Frame 80" /></td>
<td><img src="image" alt="Frame 100" /></td>
<td><img src="image" alt="Frame 120" /></td>
</tr>
</tbody>
</table>
A.2 Median filtering

This method uses a moving window that is moved from left to right and from top to bottom along each 2D slice, replacing the pixel in its centre with the median intensity value within at each position. Window size was varied from 3 to 10 because this range demonstrated an adequate range of different effects from different window sizes; Table A.2 demonstrates a cross-section of results.

Setting the window size to small values provided adequate denoising and detail preservation, and met the experimental objectives. Increasing the window size introduced unnecessary blurring and the loss of smaller details, as seen with the small black spots in frames 100 and 120.

<table>
<thead>
<tr>
<th>Window size</th>
<th>Frame 60</th>
<th>Frame 80</th>
<th>Frame 100</th>
<th>Frame 120</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
</tr>
<tr>
<td>3</td>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
<td><img src="image7.png" alt="Image" /></td>
<td><img src="image8.png" alt="Image" /></td>
</tr>
<tr>
<td>6</td>
<td><img src="image9.png" alt="Image" /></td>
<td><img src="image10.png" alt="Image" /></td>
<td><img src="image11.png" alt="Image" /></td>
<td><img src="image12.png" alt="Image" /></td>
</tr>
<tr>
<td>9</td>
<td><img src="image13.png" alt="Image" /></td>
<td><img src="image14.png" alt="Image" /></td>
<td><img src="image15.png" alt="Image" /></td>
<td><img src="image16.png" alt="Image" /></td>
</tr>
</tbody>
</table>

Table A.2: Sample for median filter denoising.
A.3 Ideal filtering

Ideal filtering is a form of Fourier filtering that uses a lowpass filter with a sharp cutoff. Because speckle noise is typically of high frequency, this filter is applied to each image slice in the Fourier domain to remove noise. Cutoff frequency was varied from 30 to 90, as the two extremes of this range demonstrate the two extremes of strong blurring (for low frequencies) and unneeded pixelation (for high frequencies); results in general either suffered from blurring or pixelation. As a result, ideal filtering was deemed unsatisfactory for the experiment (Table A.3). Homomorphic ideal filtering did not fare any better, as it suffered from the additional problem of a “ray” or “waffle” pattern (Table A.4).

<table>
<thead>
<tr>
<th>Cutoff freq.</th>
<th>Frame 60</th>
<th>Frame 80</th>
<th>Frame 100</th>
<th>Frame 120</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
</tr>
<tr>
<td>30</td>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
<td><img src="image7.png" alt="Image" /></td>
<td><img src="image8.png" alt="Image" /></td>
</tr>
<tr>
<td>60</td>
<td><img src="image9.png" alt="Image" /></td>
<td><img src="image10.png" alt="Image" /></td>
<td><img src="image11.png" alt="Image" /></td>
<td><img src="image12.png" alt="Image" /></td>
</tr>
<tr>
<td>90</td>
<td><img src="image13.png" alt="Image" /></td>
<td><img src="image14.png" alt="Image" /></td>
<td><img src="image15.png" alt="Image" /></td>
<td><img src="image16.png" alt="Image" /></td>
</tr>
</tbody>
</table>

Table A.3: Sample results for ideal filter denoising.

<table>
<thead>
<tr>
<th>Type</th>
<th>Frame 60</th>
<th>Frame 80</th>
<th>Frame 100</th>
<th>Frame 120</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td><img src="image17.png" alt="Image" /></td>
<td><img src="image18.png" alt="Image" /></td>
<td><img src="image19.png" alt="Image" /></td>
<td><img src="image20.png" alt="Image" /></td>
</tr>
<tr>
<td>Homomorphic</td>
<td><img src="image21.png" alt="Image" /></td>
<td><img src="image22.png" alt="Image" /></td>
<td><img src="image23.png" alt="Image" /></td>
<td><img src="image24.png" alt="Image" /></td>
</tr>
</tbody>
</table>

Table A.4: Comparing the results of standard and homomorphic ideal filter denoising.
A.4 Butterworth filtering

Butterworth filtering is a version of Fourier filtering similar to ideal filtering, but the cutoff of the lowpass filter is more gradual. As such, cutoff frequency was varied from 30 to 90 once again; additionally, results were observed for both first- and second-order Butterworth filtering. Overall, Butterworth filtering fared much better than ideal filtering. While low frequencies once again produced severely blurred results, higher frequencies resulted in adequate noise removal and preservation of details, and therefore met the experimental objectives (Table A.5). Changing the filter order (Table A.6), or varying between standard and homomorphic filtering (Table A.7), did not have a notable effect on the results.
<table>
<thead>
<tr>
<th>Cutoff freq.</th>
<th>Frame 60</th>
<th>Frame 80</th>
<th>Frame 100</th>
<th>Frame 120</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td><img src="image1" alt="Image" /></td>
<td><img src="image2" alt="Image" /></td>
<td><img src="image3" alt="Image" /></td>
<td><img src="image4" alt="Image" /></td>
</tr>
<tr>
<td>30</td>
<td><img src="image5" alt="Image" /></td>
<td><img src="image6" alt="Image" /></td>
<td><img src="image7" alt="Image" /></td>
<td><img src="image8" alt="Image" /></td>
</tr>
<tr>
<td>60</td>
<td><img src="image9" alt="Image" /></td>
<td><img src="image10" alt="Image" /></td>
<td><img src="image11" alt="Image" /></td>
<td><img src="image12" alt="Image" /></td>
</tr>
<tr>
<td>90</td>
<td><img src="image13" alt="Image" /></td>
<td><img src="image14" alt="Image" /></td>
<td><img src="image15" alt="Image" /></td>
<td><img src="image16" alt="Image" /></td>
</tr>
</tbody>
</table>

Table A.5: Sample results for Butterworth filter denoising.
### A.5 Wavelet filtering

Wavelet filtering uses the discrete wavelet transform (DWT) to divide each image slice into “sub-images” based on different frequencies in different directions. Initially, there are four sub-image bands: LL (largest features), LH, HL, and HH (finest features). The idea is to eliminate the band that contains the noise; however, if a band contains noise and important details, then it may be necessary to decompose one of the bands further (resulting in double decomposition or higher).

Four parameters were adjusted: the band to be eliminated, the wavelet type, the wavelet order, and the decomposition level, in addition to testing both standard and homomorphic filtering.

- **Band:** LL was not offered, and would be inappropriate, as an option for elimination. Eliminating HH and HL had barely any effect on the image slices, whereas eliminating LH had a mild denoising effect (best seen magnified) that was deemed inadequate for noise removal (Table A.8).

- **Wavelet type:** All wavelets tested (biorthogonal, coiflet, Daubechies, reverse biorthogonal, symlet) produced similar results (Table A.9).

- **Wavelet order:** All wavelet orders tested produced similar results. A varying range of orders was tested for each wavelet type. Some very high orders (especially for symlets) would take several minutes or hours to produce results, and were therefore deemed inappropriate for testing due to the fact that they violated the project objective of producing results promptly (Table A.10).

#### Table A.6: Comparing first- and second-order Butterworth filtering.

<table>
<thead>
<tr>
<th>Order</th>
<th>Frame 60</th>
<th>Frame 80</th>
<th>Frame 100</th>
<th>Frame 120</th>
</tr>
</thead>
<tbody>
<tr>
<td>First</td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
</tr>
<tr>
<td>Second</td>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
<td><img src="image7.png" alt="Image" /></td>
<td><img src="image8.png" alt="Image" /></td>
</tr>
</tbody>
</table>

#### Table A.7: Comparing standard and homomorphic Butterworth filtering.

<table>
<thead>
<tr>
<th>Type</th>
<th>Frame 60</th>
<th>Frame 80</th>
<th>Frame 100</th>
<th>Frame 120</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td><img src="image9.png" alt="Image" /></td>
<td><img src="image10.png" alt="Image" /></td>
<td><img src="image11.png" alt="Image" /></td>
<td><img src="image12.png" alt="Image" /></td>
</tr>
<tr>
<td>Homomorphic</td>
<td><img src="image13.png" alt="Image" /></td>
<td><img src="image14.png" alt="Image" /></td>
<td><img src="image15.png" alt="Image" /></td>
<td><img src="image16.png" alt="Image" /></td>
</tr>
</tbody>
</table>
• **Decomposition level:** While single decomposition was deemed too weak, double decomposition removed a large amount of image detail and caused unneeded brightening (Table A.11).

• **Homomorphic filtering:** Standard and homomorphic wavelet filtering produced similar results, although homomorphic filtering caused some white specks to appear along the edge of the scan area (Table A.12).

Overall, wavelet filtering was deemed inappropriate for the project, but remains a possible option due to the large variety in input parameters and setups.

<table>
<thead>
<tr>
<th>Band</th>
<th>Frame 60</th>
<th>Frame 80</th>
<th>Frame 100</th>
<th>Frame 120</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td><img src="image1" alt="Image" /></td>
<td><img src="image2" alt="Image" /></td>
<td><img src="image3" alt="Image" /></td>
<td><img src="image4" alt="Image" /></td>
</tr>
<tr>
<td>HH</td>
<td><img src="image5" alt="Image" /></td>
<td><img src="image6" alt="Image" /></td>
<td><img src="image7" alt="Image" /></td>
<td><img src="image8" alt="Image" /></td>
</tr>
<tr>
<td>HL</td>
<td><img src="image9" alt="Image" /></td>
<td><img src="image10" alt="Image" /></td>
<td><img src="image11" alt="Image" /></td>
<td><img src="image12" alt="Image" /></td>
</tr>
<tr>
<td>LH</td>
<td><img src="image13" alt="Image" /></td>
<td><img src="image14" alt="Image" /></td>
<td><img src="image15" alt="Image" /></td>
<td><img src="image16" alt="Image" /></td>
</tr>
</tbody>
</table>

Table A.8: Varying the eliminated band for wavelet filter denoising.

<table>
<thead>
<tr>
<th>Wavelet type</th>
<th>Frame 60</th>
<th>Frame 80</th>
<th>Frame 100</th>
<th>Frame 120</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td><img src="image17" alt="Image" /></td>
<td><img src="image18" alt="Image" /></td>
<td><img src="image19" alt="Image" /></td>
<td><img src="image20" alt="Image" /></td>
</tr>
<tr>
<td>Symlet order 9</td>
<td><img src="image21" alt="Image" /></td>
<td><img src="image22" alt="Image" /></td>
<td><img src="image23" alt="Image" /></td>
<td><img src="image24" alt="Image" /></td>
</tr>
<tr>
<td>Biorthogonal order 4.4</td>
<td><img src="image25" alt="Image" /></td>
<td><img src="image26" alt="Image" /></td>
<td><img src="image27" alt="Image" /></td>
<td><img src="image28" alt="Image" /></td>
</tr>
</tbody>
</table>
Table A.9: Varying the wavelet type for wavelet filter denoising.

<table>
<thead>
<tr>
<th>Wavelet Type</th>
<th>Frame 60</th>
<th>Frame 80</th>
<th>Frame 100</th>
<th>Frame 120</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coiflet order 5</td>
<td><img src="image1" alt="Image" /></td>
<td><img src="image2" alt="Image" /></td>
<td><img src="image3" alt="Image" /></td>
<td><img src="image4" alt="Image" /></td>
</tr>
<tr>
<td>Daubechies order 9</td>
<td><img src="image5" alt="Image" /></td>
<td><img src="image6" alt="Image" /></td>
<td><img src="image7" alt="Image" /></td>
<td><img src="image8" alt="Image" /></td>
</tr>
<tr>
<td>Reverse biorthogonal order 4.4</td>
<td><img src="image9" alt="Image" /></td>
<td><img src="image10" alt="Image" /></td>
<td><img src="image11" alt="Image" /></td>
<td><img src="image12" alt="Image" /></td>
</tr>
</tbody>
</table>

Table A.10: Varying the wavelet order for wavelet filter denoising.

<table>
<thead>
<tr>
<th>Order</th>
<th>Frame 60</th>
<th>Frame 80</th>
<th>Frame 100</th>
<th>Frame 120</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td><img src="image13" alt="Image" /></td>
<td><img src="image14" alt="Image" /></td>
<td><img src="image15" alt="Image" /></td>
<td><img src="image16" alt="Image" /></td>
</tr>
<tr>
<td>3</td>
<td><img src="image17" alt="Image" /></td>
<td><img src="image18" alt="Image" /></td>
<td><img src="image19" alt="Image" /></td>
<td><img src="image20" alt="Image" /></td>
</tr>
<tr>
<td>6</td>
<td><img src="image21" alt="Image" /></td>
<td><img src="image22" alt="Image" /></td>
<td><img src="image23" alt="Image" /></td>
<td><img src="image24" alt="Image" /></td>
</tr>
<tr>
<td>9</td>
<td><img src="image25" alt="Image" /></td>
<td><img src="image26" alt="Image" /></td>
<td><img src="image27" alt="Image" /></td>
<td><img src="image28" alt="Image" /></td>
</tr>
<tr>
<td>12</td>
<td><img src="image29" alt="Image" /></td>
<td><img src="image30" alt="Image" /></td>
<td><img src="image31" alt="Image" /></td>
<td><img src="image32" alt="Image" /></td>
</tr>
<tr>
<td>15</td>
<td><img src="image33" alt="Image" /></td>
<td><img src="image34" alt="Image" /></td>
<td><img src="image35" alt="Image" /></td>
<td><img src="image36" alt="Image" /></td>
</tr>
</tbody>
</table>
### Table A.11: Comparing single and double decomposition for wavelet filter denoising.

<table>
<thead>
<tr>
<th>Decomp. level</th>
<th>Frame 60</th>
<th>Frame 80</th>
<th>Frame 100</th>
<th>Frame 120</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single</td>
<td><img src="image1.jpg" alt="Image" /></td>
<td><img src="image2.jpg" alt="Image" /></td>
<td><img src="image3.jpg" alt="Image" /></td>
<td><img src="image4.jpg" alt="Image" /></td>
</tr>
<tr>
<td>Double</td>
<td><img src="image5.jpg" alt="Image" /></td>
<td><img src="image6.jpg" alt="Image" /></td>
<td><img src="image7.jpg" alt="Image" /></td>
<td><img src="image8.jpg" alt="Image" /></td>
</tr>
</tbody>
</table>

### Table A.12: Comparing standard and homomorphic wavelet filtering.

<table>
<thead>
<tr>
<th>Filtering type</th>
<th>Frame 60</th>
<th>Frame 80</th>
<th>Frame 100</th>
<th>Frame 120</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td><img src="image9.jpg" alt="Image" /></td>
<td><img src="image10.jpg" alt="Image" /></td>
<td><img src="image11.jpg" alt="Image" /></td>
<td><img src="image12.jpg" alt="Image" /></td>
</tr>
<tr>
<td>Homomorphic</td>
<td><img src="image13.jpg" alt="Image" /></td>
<td><img src="image14.jpg" alt="Image" /></td>
<td><img src="image15.jpg" alt="Image" /></td>
<td><img src="image16.jpg" alt="Image" /></td>
</tr>
</tbody>
</table>

### A.6 Comparison of all methods

Tables A.13-A.16 compare the results for all general denoising methods and for all four of the image slices used in this appendix. Parameters that offered little or negative variation (such as homomorphic variants and eliminating wavelet bands other than LH) are not depicted here, for efficiency reasons and ease of comparison. In general, frame 60 is too heavily shadowed to compare results in great detail, although some variation is visible. Frames 80, 100 and 120 show notable differences for each denoising setup. Due to the larger images, the effects of wavelet decomposition are more apparent without magnification.

Previously, median filtering and Butterworth filtering were deemed the most appropriate denoising methods for the project. Using the optimal parameters for each as determined by the author, MaxPol produces results of similar detail and noise level to the previously chosen methods (median and Butterworth), but had some issues with changing the sizes of volumes. In the end, median filtering was selected for D-A and its competitors.
<table>
<thead>
<tr>
<th>Filter Type</th>
<th>Cutoff/Freq.</th>
</tr>
</thead>
<tbody>
<tr>
<td>MaxPol, $P_{cutoff} = 12$</td>
<td></td>
</tr>
<tr>
<td>MaxPol, $P_{cutoff} = 14$</td>
<td></td>
</tr>
<tr>
<td>Median, window size 3</td>
<td></td>
</tr>
<tr>
<td>Median, window size 6</td>
<td></td>
</tr>
<tr>
<td>Ideal, cutoff freq. 60</td>
<td></td>
</tr>
<tr>
<td>Ideal, cutoff freq. 90</td>
<td></td>
</tr>
<tr>
<td>Butterworth, cutoff freq. 60</td>
<td></td>
</tr>
<tr>
<td>Butterworth, cutoff freq. 90</td>
<td></td>
</tr>
<tr>
<td>Wavelet, single decomposition</td>
<td></td>
</tr>
<tr>
<td>Wavelet, double decomposition</td>
<td></td>
</tr>
</tbody>
</table>
Table A.13: Comparing all general denoising methods for first sample frame (60).
MaxPol, $P_{cutoff} = 12$

MaxPol, $P_{cutoff} = 14$

Median, window size 3

Median, window size 6

Ideal, cutoff freq. 60

Ideal, cutoff freq. 90

Butterworth, cutoff freq. 60

Butterworth, cutoff freq. 90

Wavelet, single decomposition

Wavelet, double decomposition
Table A.14: Comparing all general denoising methods for second sample frame (80).
MaxPol, $P_{cutoff} = 12$  
MaxPol, $P_{cutoff} = 14$

Median, window size 3  
Median, window size 6

Ideal, cutoff freq. 60  
Ideal, cutoff freq. 90

Butterworth, cutoff freq. 60  
Butterworth, cutoff freq. 90

Wavelet, single decomposition  
Wavelet, double decomposition
Table A.15: Comparing all general denoising methods for third sample frame (100).
<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MaxPol, $P_{cutoff} = 12$</td>
<td>MaxPol, $P_{cutoff} = 14$</td>
<td></td>
</tr>
<tr>
<td><img src="image1" alt="Image" /></td>
<td><img src="image2" alt="Image" /></td>
<td></td>
</tr>
<tr>
<td>Median, window size 3</td>
<td>Median, window size 6</td>
<td></td>
</tr>
<tr>
<td><img src="image3" alt="Image" /></td>
<td><img src="image4" alt="Image" /></td>
<td></td>
</tr>
<tr>
<td>Ideal, cutoff freq. 60</td>
<td>Ideal, cutoff freq. 90</td>
<td></td>
</tr>
<tr>
<td><img src="image5" alt="Image" /></td>
<td><img src="image6" alt="Image" /></td>
<td></td>
</tr>
<tr>
<td>Butterworth, cutoff freq. 60</td>
<td>Butterworth, cutoff freq. 90</td>
<td></td>
</tr>
<tr>
<td><img src="image7" alt="Image" /></td>
<td><img src="image8" alt="Image" /></td>
<td></td>
</tr>
<tr>
<td>Wavelet, single decomposition</td>
<td>Wavelet, double decomposition</td>
<td></td>
</tr>
</tbody>
</table>
Table A.16: Comparing all general methods for fourth sample frame (120).
Bibliography


