X-RAY DETECTION OF BONES IN WHITEFISH FILLETS

Arnar Pór Stefánsson
Master of Science
Electrical Engineering
December 2013
School of Science and Engineering
Reykjavík University

M.Sc. RESEARCH THESIS

ISSN 1670-8539
X-ray Detection of Bones in Whitefish Fillets

by

Arnar Þór Stefánsson

Research thesis submitted to the School of Science and Engineering at Reykjavík University in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering

December 2013

Research Thesis Committee:

Baldur Þorgilsson, MSEE, Supervisor
Lecturer, Reykjavik University

Haraldur Auðunsson, Ph.D., Supervisor
Associate Professor and Director, Reykjavik University

Valdís Guðmundsdóttir, MSci, Examiner
Physicist, Landspitali - University Hospital
Copyright
Arnar Þór Stefánsson
December 2013
X-ray Detection of Bones in Whitefish Fillets

Arnar Þór Stefánsson

December 2013

Abstract

Advances in automation, robotics and machine vision are making high precision, value adding processes, such as automatic pinbone removal from fish fillets, a viable option. Marel Iceland ehf is currently developing such a system, where x-ray imaging is used to locate pinbones in cod fillets, which are then automatically removed from the fillet.

The goal of this project was to determine if stereoscopic x-ray imaging, where two images from a slightly different viewpoint are used to find the 3D position of objects being imaged, could be used to determine the 3D coordinates of the bones accurately enough to improve the yield of the bone removal process by e.g. controlling the angle of the cut, in addition to the $x$ and $y$ coordinates.

A simple image processing routine was developed that successfully isolates the bones from the flesh in the x-ray images, with up to 91% correct classification of bones when compared to a manually classified image, and minimal misclassification of the background (false positives).

2D images simulating perfectly segmented x-ray images are created using CT-scans of cod fillets. The pixel coordinates of the bones in the images are used to fit functions describing the 2D shape of the bones. The functions are used to find how far the position of the bones has shifted between the images. The 3D position of the bones is calculated and compared to the original CT data.

According to simulations using a rough cutting algorithm and the 3D data to control the cut, the wasted material is reduces by up to 30% when compared to a vertical 2 DOF cut.

A successful implementation significantly reduces the amount of wasted material and could give Marel an edge over the competition, but as the cost of the system is higher (both components and development) it may outweigh the gain, depending on the size of the market.
Röntengreining beina í hvítfiskflókum

Arnar Þór Stefánsson

Desember 2013

Útdráttur

Framfarir á sviðum sjálfvirkni, vélmenna og tölvusjónar, gera verðmætaaukandi vinnslu á fisk, svo sem sjálfvirka hreinsun smábeina úr fiskflókum, raunhæfan möguleika. Marel ehf vinnur að þróun sílskerfs þar sem röntgen myndir eru notaðar til að staðsetja beinin í flakinu og þau skorin sjálfvirkt úr.

Markmið verkefnisins er að ákvarða hvort 3D röntgen myndataka, þar sem tvær myndir tekna frá mismunandi sjónarhorni eru notaðar til að finna 3D staðsetningu hluta í myndinni, gefi nógó áreiðanlegar upplýsingar um staðsetning beinanna til að hægt sé að bæta nýtri vinnslunnar með því t.d. að stjórna halla skurðarins sem notaður er til að fjarlægja beinin, auk x og y hnit.

Einföld myndvinnsluáðferð var þróuð til að finna beinin í myndunum. Borið saman við handvirka merkingu beina á mynd, voru niðurstöður sjálfvirku áðferðarinnar allt að 91% rétt flokkuð bein, með lágmarks villum í bakgrunni.

Gögn úr CT myndatöku voru notuð til að búa til 2D myndir sem líkja eftir fullkomlega flokkuðum röntgen myndum. Hnit beinanna í myndunum voru nálguð með föllum, sem gefa mat á staðsetningu miðlínu hvers beins. Hliðrun á staðsetningu beina milli mynda er fundin, og notuð til að reikna 3D staðsetningu beinanna sem er borin saman við upphaflegu CT gögnin.

Gróft mat á skurði flaksins sýnir allt að 30% minni töp á hráefni þegar 3D skurður er borinn saman við 2D skurð.

Marktækur munur er á nýtingu með 3D skurði, og gætu not þessarar tænki gefið Marel forskot á samkeppnisaðila, en þar sem kostnaður hækkar (bæði framleiðslu og þróunarkostnaður) þarf að skoða mögulegan hagnað, medal annars með tilliti til væntanlegrar stærðar markaðs.
Contents

List of Figures vii
List of Tables ix
List of Abbreviations xi
List of Symbols xiii

1 Introduction 1

2 Methods & Materials 9
  2.1 Proposed Equipment Setup .......................................... 9
  2.1.1 Layout .......................................................... 9
  2.1.2 Stereoscopic Calculations ....................................... 12
  2.2 Images ............................................................. 15
  2.2.1 CT Data Description ............................................ 15
  2.2.2 Domus Images .................................................. 19
  2.3 Comparing images .................................................. 21

3 Image Preprocessing 23
  3.1 Image Classification ............................................... 23
  3.1.1 Pre-processing of Domus Images ................................. 25
  3.1.2 Background Removal .......................................... 26
  3.1.3 Thresholding .................................................... 28
  3.1.4 Image Clean-up ................................................ 31
  3.2 Image Segmentation ................................................ 31
  3.3 Selected Results ................................................... 32
  3.3.1 Thresholding Results ......................................... 33
  3.3.2 Background Removal Method .................................. 33
  3.3.3 Segmentation ................................................... 35
## List of Figures

1.1 X-ray images of a cod fillet .............................................. 3  
1.2 Stereoscopic imaging setup using two x-ray sources .................. 5  
1.3 Rendered scene demonstrating stereoscopic images ................. 6  
1.4 The different cutting methods proposed .............................. 7  
1.5 View through a fillet looking in the $y$ direction .................... 7  

2.1 X-ray image with axes overlay ........................................... 10  
2.2 Diagram showing the relative position of the x-ray source and the linescan sensors .................................................. 11  
2.3 Shifts in the apparent position of objects vary based on the height $h_n$ of the object .................................................. 13  
2.4 Example voxel data SINTEF T1 .......................................... 16  
2.5 An example of artefacts present in the CT images ................ 17  
2.6 Cod fillet being imaged .................................................... 20  
2.7 Imaging equipment and resulting image ................................ 20  

3.1 Digital image, human vs. computer vision ........................... 24  
3.2 Failed global thresholding ................................................ 25  
3.3 Histogram of entire image ((a) and (c)) compared to histograms for bone only ((b) and (d)) before and after filtering .......................... 29  
3.4 Simulated grayscale image demonstrating the triangle thresholding method ................................................................. 30  
3.5 Thresholding results ....................................................... 34  
3.6 Two high scoring results of frequency domain bandpass filtering .... 37  
3.7 Results of subtracting a background created using a large median filter . 38  
3.8 Results of subtracting a background created using a morphological closing ................................................................. 39  
3.9 Segmentation results ....................................................... 40  

4.1 Correspondence in images from Röntgen Domus .................... 42  
4.2 Examples of bones that can not be represented accurately by functions of $x$ ................................................................ 46  
4.3 Results of fitting different order polynomials to a segmented image . . . 49
4.4 Results of fitting different order polynomials to PCA transformed data . . 50
4.5 Calculated 3D for SINTEF cod data T1 compared to GT . . . . . . . 51
4.6 Calculated 3D position of bones in the fillet imaged at Röntgen Domus . . 51
List of Tables

A.1 Classification quality using frequency domain bandpass filtering. . . . . . 58
A.2 Classification quality using a large median filter for background removal. 59
A.3 Classification quality using a morphological closing for background re-
   moval. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 60
# List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D</td>
<td>Two Dimensional</td>
</tr>
<tr>
<td>3D</td>
<td>Three Dimensional</td>
</tr>
<tr>
<td>CT</td>
<td>Computed Tomography</td>
</tr>
<tr>
<td>DOF</td>
<td>Degrees of Freedom</td>
</tr>
<tr>
<td>DRS</td>
<td>Digital Radiography System</td>
</tr>
<tr>
<td>FOB</td>
<td>Free on Board</td>
</tr>
<tr>
<td>GT</td>
<td>Ground Truth</td>
</tr>
<tr>
<td>LRM</td>
<td>Linear Regression Model</td>
</tr>
<tr>
<td>NDT</td>
<td>Non-Destructive Testing</td>
</tr>
<tr>
<td>PC</td>
<td>Principal Component</td>
</tr>
<tr>
<td>PCA</td>
<td>Principal Components Analysis</td>
</tr>
<tr>
<td>ROI</td>
<td>Region of Interest</td>
</tr>
<tr>
<td>voxel</td>
<td>Volume Element</td>
</tr>
</tbody>
</table>
## List of Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>The angle formed by sensor A, the x-ray source and sensor B.</td>
<td>$[^{\circ}]$</td>
</tr>
<tr>
<td>$d_{A\rightarrow B}$</td>
<td>The distance between sensors used for stereo imaging.</td>
<td>$[mm]$</td>
</tr>
<tr>
<td>$d_{S\rightarrow A}$</td>
<td>The distance between the x-ray source and sensor A.</td>
<td>$[mm]$</td>
</tr>
<tr>
<td>$\theta$</td>
<td>The inclination of the cut compared to the (y,z) plane.</td>
<td>$[^{\circ}]$</td>
</tr>
<tr>
<td>$\mu(x, y, z)$</td>
<td>The mass attenuation coefficient at coordinates $(x, y, z)$.</td>
<td>$[cm^2g^{-1}]$</td>
</tr>
<tr>
<td>$\mu_{arb}(x, y, z)$</td>
<td>Arbitrarily chosen attenuation coefficient for voxel at coordinates $(x, y, z)$, used when generating 2D images from the 3D CT-scan data.</td>
<td>$[voxel^{-1}]$</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

In 2012 the cod catch of the Icelandic fleet was 204,000 metric tonnes which is approximately 20% of the total world catch (Hagstofa Islands, 2013a). The same year, marine products provided 42.4 % of the free on board (FOB) value of Icelandic goods export, with cod accounting for 13.1% of the total value of export (Hagstofa Islands, 2013b).

According to a EUMOFA case study from January 2013 (European Market Observatory for Fisheries and Aquaculture [EUMOFA], 2013) the most expensive cod products are fresh fillets. The same study cites the yield of fillets as 35% of live weight, and further losses of raw material can occur depending on additional processing.

Providing fresh fillets to the market requires a well positioned factory or the rapid transportation of goods, as well as high throughput processing plants. Increased automation, made possible with recent advances in robotics, computer vision and lower cost of computer power, is making further processing of fresh fillets, such as de-boning, trimming or cutting into loins, a viable choice in the plants.

Replacing manual labour with automation has several advantages. While workers can in many cases outperform machines, efficient manual cutting and trimming of fillets requires training and experience, and can be time consuming. A properly tuned machine will provide good results from day one, work tirelessly and perform consistently every day of the week. Automated solutions also have the potential to significantly increase the production speed, improve yields and take less space on the factory floor. However they require fewer, but better educated, employees.

Fully or partially automated processing plants, for fish and other consumer products are becoming increasingly popular, and the competition between companies providing automated solutions for the market is growing.
Problem Description

The project is done in cooperation with Marel Iceland ehf (a provider of advanced solutions for fish, meat, and poultry processing), and in parallel with another research project (APRICOT) aiming to develop equipment capable of automatically cutting pinbones from cod fillets.

The pinbones are a series of small bones that remain in the flesh after filleting the fish. According to a study performed by SINTEF (a Scandinavian multidisciplinary research organization), the number of pinbones in each cod fillet ranges from 9 to 17 with an average number of 13. The same study reports the average thickness of pinbones in cod to be approximately 0.8 mm and their length 17 mm on average. The authors mention that due to limitations in resolution of the equipment used, the automatic estimation of the length and width is not perfect. Manual measurement of the bones reveals them to be slightly longer and narrower than the CT scans indicate (Thielemann, Bakken & Schulerud, 2012).

The bones are located in the thickest and most valuable part of the fillets. Removing these bones is desirable, and is often done by the processor or vendor prior to selling to consumers. The bone removal is generally done by hand, a rather time consuming and monotonous task - especially if high yield is desired. This can be done with tweezers, pliers or very careful cutting.

If throughput is more important than yield, e.g. in factories, the pinbones can be removed by making a v-shaped cut in the fillet as shown in Figure 1.1b. Using this method, workers can quickly remove all the bones from a fillet with some accuracy, but a considerable amount of raw material is lost.

One method to automate the pinbone removal process is using an x-ray sensor to detect the bones and then a suitable method of cutting the piece containing bones off. Because of the small size of pinbones and the speed of factory conveyor belts, a successful design requires relatively high resolution images (10 pixels/mm) to accurately locate all the bones, and close to real time processing of data to maintain production speed.

A successful design resulting from APRICOT will be able to improve the post-deboning yield compared to the v-cut, by closely tracing the outline of the bones to be removed (Nordic Innovation, 2012). The cut will be performed on a plane (two degrees of freedom (DOFs)), where the two dimensional (2D) projection of the bones visible on a x-ray image will be used to calculate the optimal path (an example is provided in Figure 1.1b). For cutting, movement along one axis could be provided by the conveyor belt, and the other
axis by the lateral movement of the cutting instrument. In the following chapters this method will be referred to as vertical cutting.

**X-Ray Imaging**

Since their discovery, x-rays have been applied in many fields. This includes research in many disciplines, medical diagnoses and various applications of non-destructive testing (NDT). NDT methods are useful in many cases, especially when valuables could be damaged if mishandled (paintings analysis), when the rapid examination of complex objects is required (baggage security check) and as in the case of food processing, where ideally every item should be scanned for contaminants or defect.

In a recent review of trends in the applications of imaging technologies to the inspection of fish and fish products (Mathiassen et al., 2011), several uses of x-ray technology are mentioned. For automation applications, x-rays have been used mainly for automated
sorting and grading. Foreign object detection is most common, and some manufacturers provide equipment specifically intended for pinbone detection (e.g. SensorX by Marel HF, Garðabær, Iceland and cutting line solutions from Valka EHF, Kópavogur, Iceland). There is ongoing research both into improving bone detection accuracy (Mery et al., 2011) using modern computer vision techniques, and into the usage of planar x-ray imaging for weight estimation which shows promising results (Veliyulin et al., 2011).

Technologies using x-rays for automated process control, e.g. pinbone removal or trimming, are not yet common, partially due to the prohibitive equipment cost. Another factor is, that even though interpreting x-ray images is easy for humans, in part due to the various methods of contrast enhancement (windows, manual adjustment of thresholds) and filters, automatically processing the images can be relatively complex.

The position of bones in fillets is predictable to some extent, however, variations in size, shape, separation, and the precise location of bones (due to the genetic variation and the age and handling of the cod in question) combined with the varying thickness of the fillet, the possible presence of other objects, noise in the x-ray image and shadows in the image caused by defects in the fish; make robust computer vision solutions an interesting challenge.

Using dual energy x-ray imaging, where a non-linear combination of images obtained using sensors for different energy x-rays simultaneously, can improve the detectability of bones in some cases (Batchelor, Davies & Graves, 2004), but adds to the complexity of image processing (computation cost), as well as requiring more expensive equipment. High dynamic range, high resolution sensors used with modern computer vision methods have also been demonstrated a successful solution to the bone detection problem (Mery et al., 2011).

**Stereoscopic Imaging**

Stereoscopic imaging is a method to obtain the three dimensional (3D) position of objects in a scene. The location of identifiable features or objects in an image pair is compared, and the displacement of points between the images (Figure 1.2) can be used to calculate depth information - provided the camera position is known. Three or more cameras can be used for increased accuracy of depth information (Szeliski, 2011, Ch. 11.6) and multiple sequential frames from a moving video camera can be used to the same effect.

In human vision, stereoscopy contributes to perception of depth and has been used to convey 3D information in print for many years. Viewing stereo images was a popular form of entertainment in the 1800s but became less common in the early 1900s. Stereo
Figure 1.2: Stereoscopic imaging setup using two x-ray sources. If the position of each source is known, the height of an object \((h_n)\) can be determined by examining the apparent displacement of objects in images. The position of an object in the image is indicated by the intersection of a ray and the image plane. Red for source A and blue for source B.

Pairs found use in medical x-ray imaging in the late 1800s and early 1900s (Davidson, 1916), later mostly replaced by computed tomography (CT) scans where a more complete and accurate 3D model can be created.

Today stereoscopy is again gaining popularity in the entertainment industry, is used in aerial stereophotogrammetry, and most recently in robot localization and navigation. Stereoscopic x-ray imaging has been developed for baggage checks at airports (Evans & Robinson, 2000), where adding depth information to the scans makes discerning individual items within the baggage much easier for inspectors.

The hard problem of recovering 3D information is the matching of points between images (the correspondence problem). The difficulty of this problem is shown in Figure 1.3, where even though the scene is relatively simple, many of the spheres are partially or completely occluded (in one or both images), creating unexpected shapes, and the displacement of others is large enough to completely change the order and neighbourhood of the spheres. The magnitude of the problem is exaggerated in these images, compared to the cod fillets, by using a lot of identically shaped objects, closely packed in a limited area with a greatly varying depth.
Figure 1.3: Stereo pair images suitable for free viewing. The scene contains 40 randomly distributed spheres, and demonstrates the difficulty of the correspondence problem. The scene is rendered in Matlab.

The simplicity of the scene being imaged in the project (cod fillets on a conveyor belt), means the correspondence problem can mostly be solved using a setup where there is a relatively small angle between the beams used to create the images. This restricts the maximum displacement of objects making the correspondence problem significantly easier.

**Research Objective**

The objective of this project is to determine if stereo x-ray images can provide an accurate enough estimate of the 3D position of pinbones to control a three DOF cutting system \((x, y)\) intended to improve the yield, when compared to the automated vertical \((x, y)\) cutting system. Where \(x\) and \(y\) are the coordinates of the cut at height \(z = 0\), and \(\theta\) is the angle between the cut and a vertical line passing through the \((x, y)\) coordinates. The angle \(\theta\) is on a plane parallel to the \(xz\)-plane (Figure 1.4).

All software was developed using MATLAB® (The MathWorks Inc., 2013).

2D images simulating perfectly segmented x-ray images are generated using a dataset containing CT scans of cod fillets taken by SINTEF. The 2D shape of the bones as seen in these images is described using functions, and correspondence in each image pair is determined. This data is used to calculate the 3D shape of the pinbones, which is compared to the original CT data.

For completeness, a simple bone detection image processing routine is demonstrated on cod fillet x-ray images, provided by Röntgen Domus (a radiology centre base in Reykjavik, Iceland).
Figure 1.4: The different cutting methods proposed. A fillet is represented by a cuboid (its bounding box), red shows a cut resulting from a 2 DOF \((x, y)\) cutting mechanism and the blue path a 3 DOF one \((x, y, \theta)\). The thicker paths indicate where the cut intersects either the top or bottom surface of the fillets bounding box. A red dotted line is used to highlight the difference between the 2 DOF and 3 DOF approach to cutting.

As discussed, the size of the industry means an improvement in yield, even a small one, can potentially provide a valuable amount of extra raw material. By tilting the cut the yield can be improved considerably, as shown in Figure 1.5.

Figure 1.5: View through fillet looking in the \(y\) direction. Image created using the CT dataset. Red line indicates the position of a vertical cut that successfully removes the leftmost bone. Angling the cut (blue line) enables a much closer cut with significant improvements to yield.

According to my literature search, this is the first reported application of stereoscopic x-ray imaging to either bone detection in fish, or more generally to foreign object detection in the food industry. The application of x-ray imaging in food processing has been focused on single view (planar) images for foreign object detection and sorting (Mery, 2013; Mathiassen et al., 2011). The focus of the project is on cod fillets, but the technology has the potential to be used for bone removal in other species of fish, and other food products.
**Thesis Outline**

The work is split into two main themes, image processing and classification, discussed in chapter 3, and the modelling of bones and 3D calculations in chapter 4. Chapter 2 provides a theoretical background and information about the images used and chapter 5 concludes the thesis.
Chapter 2

Methods & Materials

In this chapter a basic theoretical design for the layout of sensors capable of recording stereo pair images is suggested. A coordinate system for working with the images is defined, and will be used in the following chapters. The equations for calculating depth information from the stereo pairs are derived, and a description of the images used in the project is provided.

2.1 Proposed Equipment Setup

Marel is developing a system where x-ray images are used to determine the location of pinbones in fillets, which will then be used to control an automated cutting system capable of \((x, y)\) movement. The goal of this project would be an extension to the system being designed by Marel, and therefore minimizing the changes required to implement the new imaging system is advantageous. That, combined with the high cost of x-ray sources, suggests a single source system similar to the one designed for security screening by Evans and Robinson (2000).

2.1.1 Layout

The fillets travel on a conveyor belt and enter the scanner (a pair of line scan sensors) "head first". The fillet is imaged while moving on the conveyor belt, and ideally the processing of the images starts as soon as the first pixels have been acquired, to maintain production speed.
Coordinate System

For the remainder of the thesis the following two coordinate systems will be used when referring to the images or equipment.

![X-ray image with axes overlay](image)

Figure 2.1: X-ray image with axes overlay.

When referring to individual pixels or objects in images, a coordinate system with its origin in the upper left corner is used (Figure 2.1). The pixel at the origin has coordinates \((x, y) = (1, 1)\), \(x\) increases to the right, and \(y\) increases down.

For the equipment, the origin is defined to be at the centre of the conveyor belt, directly below the x-ray source. The belt moves in the positive \(y\) direction (Figure 2.2). This maintains some consistency between the expected orientation of the fillets and the equipment coordinate system.

The x-ray beam angle, \(\alpha\) is the angle formed between the lines connecting the x-ray source to the sensors in the \((y, z)\) plane.

Sensor Positions

The distance from the x-ray source to sensor A, \(d_{S\rightarrow A}\), is assumed to be 500mm, close to the distance used in SensorX. Two line scan sensors should be placed under the conveyor
belt, one directly beneath the x-ray source (sensor A) and the other with a slight offset in the positive y-direction (sensor B), indicated by $d_{A \rightarrow B}$. The magnitude of the offset depends both on $d_{S \rightarrow A}$ and the desired x-ray beam angle, $\alpha$. Figure 2.2 shows the suggested setup.

Figure 2.2: Diagram showing the relative position of the x-ray source and the two linescan sensors. Sensor A is placed directly below the x-ray source. Note: not to scale.

If $\alpha$ is large, the correspondence problem becomes more difficult to solve as the projected shape of a bone can be considerably different in the images from sensors A and B. A large $\alpha$ also increases the probability of bones overlapping, which makes the image segmentation process less accurate.

An angle that is too small has the opposite problem, its easy to match points between images (consider the simplest case where $\alpha$ is $0^\circ$) but the available depth information is limited.
Using high resolution sensors improves the accuracy and enables the use of smaller angles between sensors, as small variations in bone position between images are detected more accurately.

Aligning the images in a stereo pair is required before calculating 3D information. This could be done by finding features or edges in the images and comparing, but that may not be robust enough. A better solution could be to time the image acquisition and align based on the time it takes the conveyor belt to move $d_{A\rightarrow B}$, and fiducials (small markers embedded in the belt and visible on the x-ray image) could increase the accuracy of this method by acting as calibration landmarks.

### 2.1.2 Stereoscopic Calculations

The calculations required to find the position are relatively simple once the bones have been identified and the correspondence between the images has been found.

When imaging objects from different viewpoints a shift in image coordinates is observed. The magnitude of the shift is dependent on the distance of the object from the sensor (height above the conveyor belt), and the x-ray angle, $\alpha$. Figure 2.3 shows how the image coordinates change depending on the vertical position of the object, for a two dimensional scene.

The shape of the x-ray beam can affect the shape of objects visible in the image. The ideal beam shape for minimizing unwanted object shape distortion is a parallel beam. Assuming that the x-ray source can be thought of as an ideal point source, and that line scan sensors are used to record the images, the beam passing through the object is shaped like a fan (Figure 2.2). Depending on where the object is located, relative to the source/sensor pair, and the shape of the x-ray beam, the projection of the objects visible in the image is affected.

An example of the distortion can be seen in Figure 1.3, where the projected images of some of the spheres appear elliptic instead of circular. The effect is most noticeable near the upper and lower border of the image for spheres that are located high above the image plane.

Increasing $\alpha$ increases the length of the path the x-rays have to travel. This increases the divergence of rays before reaching the sensor, which slightly magnifies the distortions described. Moving the sensor to get the second image in the stereo pair adds another dimension to the distortions, now the position of the object in the image shifts in the $y$-direction as well.
Figure 2.3: Shifts in the apparent position of objects vary based on the height $h_n$ of the object. Distances $y_{nA}$ and $y_{nB}$ are measured from a fixed reference point to the position of object $n$ in the relevant image.

The position of the fillet inside the x-ray imaging equipment is close to the centre of the conveyor belt, where the rays are mostly parallel, and therefore the $x$-shift of image coordinates is negligible. As the fillets are thin, compared to the source height and the region of interest (ROI) is relatively close to the centre of the conveyor belt, the shifts in $x$-position may be ignored when matching and modelling bones.

If the fillet has rotated or shifted on the conveyor belt, moving the bones out of the ideal position for imaging, the $x$-coordinates of the bones in the image should be corrected before calculating the 3D position to maximize yield (and make sure the bones are accurately cut out). An alternative solution would be to detect images where this is the case, and re-image the fillet.

Once the images have been acquired and the correspondence found, calculating the 3D position of the detected points does not require complex mathematics, only the application of ratios of similar triangles. Prior to calculating the 3D position, the bones are modelled
and matched under the assumption that the \( x \)-coordinates do not change significantly between images.

Figure 2.3 shows how the simplified system used for depth calculations is defined.

Superimposing the image from sensor B on the image from sensor A reveals that the triangles formed by the objects actual position \((x, y, h)\) and the positions of the objects in images A and B \((x, y, 0)\) (the image plane is defined as the \( x, y \) plane at \( z = 0 \)); and the triangle formed by the x-ray source, and sensors A & B are similar.

The height of the x-ray source is known to be \( H \), the distance between sensors A & B is \( d_{A\rightarrow B} \) and the shift in position of object \( n \), \( \Delta y_n = y_{nA} - y_{nB} \). The height of object \( n \), \( h_n \) can therefore be calculated as

\[
h_n = H \frac{\Delta y_n}{d_{A\rightarrow B}}
\]  

Provided that the points have been correctly identified, the fillets maintain their shape while going through the x-ray machine, and the only movement of the fillet is a steady pace in the \( y \)-direction - the 3D position of bones can be determined with reasonable accuracy.

This approach has limitations, as bones having a major axis with a direction several degrees around the \( y \)-axis can not be positioned reliably. Many different bone shapes can produce the same shadow, and the uniformity and the lack of visible features on the bones in x-ray images make satisfactory pixel to pixel correspondence close to impossible.

Bones that are close to vertical in the fillet (aligned with the \( z \)-axis) will be difficult to detect, as one image displays the bone as a dot and the other as a line. Again correspondence between the images is unreliable at best.

A possible solution to this is to detect which bones are aligned closely with either the \( z \)-or \( y \)-axis and removing them by making a thin vertical cut. Limited yield improvement over the cutting method being developed by Marel is possible.

A more difficult class of errors are overlapping bones. Bones may be separable in one image, but not the other, or the bones might combine to form different shapes that are difficult to analyse. Some of the shapes may be resolved using advanced computer vision algorithms, but there is always the possibility of misinterpretation. The results of these errors can go both ways, either cutting too close to the bones or too far away.
If the bones are detected to have an abnormal shape, *i.e.* one that is not easily modelled by polynomials or other fitting functions, the most reliable way to remove the bones would be to use the vertical cutting algorithm.

### 2.2 Images

For the design and testing of algorithms and methods in the project, two sets of images were used. For the image processing part, x-ray images taken at Röntgen Domus were used, but as the accurate 3D position of the bones in the fillet is not known they are not suitable for developing the depth calculations (ground truth (GT) data not available). The 3D position of bones is calculated and shown, but the accuracy of the results is not known.

For the 3D part of the project, a dataset provided by Marel, containing CT scans of a number of fillets will be used. The CT scans are used to generate ‘perfectly’ segmented 2D images, the 3D position of the bones is calculated and the results compared to the original 3D data.

#### 2.2.1 CT Data Description

The images originate from SINTEF (Thielemann et al., 2012), and according to the included info file, the images were obtained using equipment at Rikshospitalet in Norway. The dataset includes tomographic images of 17 different cod fillets, each provided in both polygon file format (*.ply), and as a three dimensional Matlab matrix where each entry corresponds to a volume element (voxel).

The fish in the dataset was automatically filleted using either a Baader or a Marel filleting machine. The resolution of different images varies based on the width of the fillet, ranging from $[0.1950, 0.1950, 0.250]$ mm/step ($[z, x, y]$) to $[0.4760, 0.4760, 0.400]$ mm/step, in a non-cubic array. Each file has a separate ID number TMx and TBx where $x \in \{1 \ldots 8\}$, and one file labelled T1. **T1** (Figure 2.4) has the highest resolution and will be used to present the results in Chapter 4.

As the .ply format files seem to be corrupt and unusable for some of the fillets, the voxel array was used to generate the 2D images.

The data contained in the arrays is a processed version of the original x-ray data, the full history of post-processing is not known, but the data has at least been segmented. Each
Figure 2.4: Voxel data SINTEF T1, (a) shows both the fish and bones, and (b) shows the bones isolated.

A voxel is assigned an id number indicating the object it belongs to. A voxel can belong to one of the following: not a part of the fillet (id = 0), soft tissue (id = 10) or one of the bones (id ∈ \{101, 102, \ldots, (100+n_{bones})\}).

When viewed from the top, some unrealistically square artefacts are visible (Figure 2.5), this may be due to errors in reconstruction, an effort to remove imaging errors, or because of segmentation issues.

Although having unedited images would be preferred, the segmentation of the data is useful. It enables the generation of images containing only bones and provides the ability to exclude certain bones, e.g. skip every second bone to eliminate overlap in the generated image.

**Generating 2D Images**

The 2D images are generated using a simple algorithm, similar to volumetric ray casting. The intensity of each sensor pixel is calculated by tracing a ray from the centre of the
pixel towards the source position, effectively ignoring any non-ideal behaviour of the x-rays (e.g. scattering). The sum of the values of the elements in the fillet array, $\mu_{arb}(x, y, z)$, that a ray passes through, multiplied by the distance the ray travels in each voxel is used as the intensity of the pixel. $\mu_{arb}(x, y, z)$ is an arbitrarily chosen attenuation coefficient used to generate the 2D images from the 3D CT-scan data. Here the coefficients used are 10 for flesh (not used when simulating perfectly segmented images), and 100 for bones.

\[ p_i = \sum \mu_{arb} \cdot d_{vox} \]  

Coordinates representing the position of the x-ray source and each sensor pixel in relation to the fillet are determined such that the source is located directly above the centre of the fillet. The sensor is positioned parallel to the direction of movement, that is; every pixel in the sensor has the same $y$ coordinates, but the $x$ coordinates vary along the sensor width. Generating images that have different $\alpha$ is done by shifting the $y$ position of the sensor relative to the source (changing $d_{A \rightarrow B}$).

The sensor size (width in mm) and resolution (number of pixels in sensor) are parameters set while initializing the scene, and the width of each pixel is calculated based on those. There is an upper limit to the number of sensor pixels per mm, as the relatively low resolution (2-3 pixels/mm) of the array data limits the quality of created images.

For each step of the algorithm both the sensor and the source are shifted in the $y$-direction by an amount equal to the width of a sensor pixel.

Using this method the images generated show the most relevant properties expected in the x-ray images, such as varying levels of zoom and shifts in projected position depending
on where the object being imaged is located, resulting from the fan shape of the x-ray beam.

**Algorithm 1: 2D Image Generation**

<table>
<thead>
<tr>
<th>Input</th>
<th>: 3D fillet array</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output</td>
<td>2D simulated x-ray image</td>
</tr>
</tbody>
</table>

Initialize sensor and source position relative to fillet;

while not at the end of the fillet array do

  for each pixel in sensor do
    Trace ray to source;
  end

  Shift source and sensor along the y axis, relative to fillet;
end

The simulated x-ray image could be made more realistic by adding several of the non-ideal parameters present in real systems. The rays used could originate from random points within the area of the pixel (instead of the precise centre) and end in a random point on a source modelled either as a line or a square, several rays could be averaged for each pixel giving a smoother image.

The class labels used in the array could be substituted for the correct \( \mu(x, y, z) \) for bones and soft tissue and the more accurate equation used,

\[
\log_e \left( \frac{I_0}{I_d} \right) = \sum_i d_i \mu_i 
\]

or

\[
I_d = I_0 e^{-\sum_i d_i \mu_i} 
\]

Where \( I_0 \) is the intensity detected from an unobstructed x-ray source, \( I_d \) is the detected intensity with object being imaged in place, \( d_i \) is the distance travelled through material \( i \) and \( \mu_i \) is the linear attenuation coefficient. The sums are along each ray, assume a discrete object, and a monochromatic photon beam passing through a series of homogeneous materials (Suetens, 2009, Ch. 2).
The accuracy of the simulated x-ray images could be improved further by using a larger section of the photon energy spectrum, the corresponding attenuation coefficients, and simulating noise in the image.

Images are created to show how the fan beam distortion affects the apparent shape of the bones, and to determine if the change in shape causes difficulty in finding correspondence between images. The algorithm is also used to generate images from different viewpoints (by changing $d_{A \rightarrow B}$) to compare the accuracy of calculated 3D shape for different $\alpha$ and determine a suitable angle for this dataset.

Although a more detailed simulation of x-ray images is possible, this project only requires a binary image showing the location of the bones, so the simple image generation algorithm is therefore sufficient.

### 2.2.2 Domus Images

A cod fillet was imaged at Röntgen Domus from different viewpoints ($\alpha$ equal to $[0, 15, 30, 45]^\circ$). The image was recorded using a Canon CXDI-31 digital radiography system (DRS), capable of producing 12 bit (4096 grey values) 6.5 MP images at 10 pixels/mm, with a 50 kVp, 200 mAs x-ray.

As can be seen in Figure 2.6 the fillet is larger than the DRS, but is placed such that the ROI is well within the plate.

The fillet contains 17 bones, 16 pinbones and one larger bone that breaks the pinbone pattern. Figure 2.7 shows the equipment used, and the resulting image.

The bones are clearly visible in all images, and the change in apparent bone shape is as expected, the bones stretch in the $y$ direction, but the width of the bones remains constant (no significant stretching in the $x$ direction). The image taken at a 30° angle shows some overlap of bones, which increases as the angle is increased. The maximum usable angle for this imaging set-up and this particular fillet is between 15 and 30°. This angle will most likely be smaller for the simulated images, as separating the bones in lower resolution images is more difficult.

The stored images are not identical in size, i.e. the number of pixels in each image is different. This is most likely a result of the automatic post-processing of the image, as areas of the image that contain no information are of limited use in medical imaging and are removed.
Figure 2.6: Cod fillet being imaged

Figure 2.7: (a) shows the equipment used for imaging and (b) shows the resulting image
This difference in image size makes automatic alignment of images for stereoscopic processing bit more difficult, but as this problem is not likely to occur on this scale in an industry setting (sensor width and position fixed, no unwanted processing of image data), the images are manually aligned in stead of developing a method for automatic alignment.

2.3 Comparing images

In order to compare the different methods applied to the images, a metric needs to be defined. Manually ranking the images based on quality is unreliable and biased results are extremely likely.

Classification

The goal of classification is to determine which pixels in the images show bones and which show flesh. A GT image is created, by manually marking the location of bones in the image, and used as a reference for scoring the automatic classification.

Each image to be scored is split into three separate areas based on the GT image; pixels showing bones, the area within a 15 pixel distance from the bones (roughly the average width of the bones), and the rest of the image. When referring to those areas, they are numbered (1, 2, 3) respectively. Each area is scored separately, and a weighted sum of scores is calculated and used for indicating the quality of the classification.

The division of the image into separate areas is necessary because what qualifies as a successful classification depends not only on how many pixels are correctly classified, but also on which pixels are correct.

If a score is based on the entire image (not separated into different areas) is used, a classification rate of 99.08 % can be achieved by classifying the entire image as background (Domus image, \(\alpha = 0^\circ\)). While a classification receiving a high score using this metric may be correct, the ratio of background pixels to bone pixels is too high for this score to be useful (too many incorrect classifications can receive high scores). The purpose of the different score areas is explained in the following text.

The first score shows how accurately the bones have been classified and is calculated as a ratio of correctly labelled pixels \(p_{1,cc}\) to the total number of pixels in the area \(p_1\). A perfect result, achieved by correctly classifying every pixel labelled as bone in the GT image, gives a result of 1.
The second score measures how well the area closest to the bones is classified. A good score here indicates that the bones are easily separable (or at least as separable as in the GT image). This score uses the ratio of misclassified pixels $p_{2mc}$ to the total number of pixels in the area. The lower this score the better the image.

$$s_2 = 1 - \frac{p_{2mc}}{p_2} \quad \text{Score for area 2} \quad (2.6)$$

The third score is defined identically to the second, but for another area. This score has a higher multiplier in the final score, as the area being scored is huge compared to areas 1 and 2. Score three checks the image for areas classified as bones, where there are none. Minimizing this score is essential, as incorrectly classifying soft tissue as bones would cause a loss of valuable raw material.

$$s_3 = 1 - \frac{p_{3mc}}{p_3} \quad \text{Score for area 3} \quad (2.7)$$

The weights used both indicate the relative importance of scores, and fix any difference in the scale of the three sub-scores. The final score used for comparison is

$$s_{total} = w_1 s_1 + w_2 s_2 + w_3 s_3 \quad (2.8)$$

where $w_n$ is the weight for score $n$, $p_n$ is the total number of pixels in an area, $p_{ncc}$ is the number of correctly classified pixels in area $n$, and $p_{nmc}$ is the number of misclassified pixels.
Chapter 3

Image Preprocessing

In order to control the cutting mechanism based on an x-ray scan of a fillet, a way for the control algorithm to retrieve information about where the bones are, is required. This is accomplished using image processing and machine vision techniques. This chapter outlines simple methods for image classification and segmentation, and demonstrates on the $\alpha=0^\circ$ x-ray images from Röntgen Domus.

3.1 Image Classification

The goal of image classification is to assign each pixel in an image to one of two or more classes defined for the task. For bone detection two classes will be used, either bone, or no-bone. A third class could be added for labelling empty pixels containing neither bone nor fish, but that would increase the complexity with no immediate gain.

The classes are represented by class labels, 1 for bone and 0 for no-bone, and the results of any classification are displayed as a binary image (bone - white, no-bone - black).

Digital Images

The 12-bit x-ray images are stored as 16-bit grey-scale arrays, and even though it is relatively easy for us to detect bones in the images, it is not necessarily an easy task to enable a computer to do the same. Most humans rely on vision to obtain information about their surroundings, and we have evolved with strong pattern recognition skills. This, combined with years of experience to draw from, and the various image processing tools available (e.g. contrast enhancements, filtering and intensity value remapping) mean we can
fairly easily distinguish objects in images, even when only partial information is available. However, the strong pattern recognition skills make humans prone to misinterpreting both live scenes and images, by detecting non-existent patterns. A computer can only 'see' the numbers comprising the pixels (Figure 3.1), and because the image contains no extra information about the relations of the pixels, a computer is unable to distinguish objects or understand images without the appropriate image processing algorithms.

![Images](image1.png)

**Figure 3.1:** With the available context the bone in image (a) is not difficult to distinguish from the background. Zooming in to a section of the bone (b) makes it more difficult to identify. Image (c) shows what the computer 'sees'.

The x-ray images contain only different levels of grey, and the first thought when processing the images is often that since the bones are easily visible by eye, a simple application of a global image threshold must be able to separate the bones from the fish. In general, this does not work well. The varying thickness of the fillet and the size of the bones will in most cases result in high misclassification rate (many bones classified as flesh and vice versa).

Another way to solve this would be to use basic adaptive thresholding, where the threshold level is varied based on local image characteristics. This method has a higher chance success, but there are cases where automatically choosing a threshold using this method would create unwanted and incorrect details in the classified image (e.g. emphasize x-ray shadows present because of defects in the fish).

The method described in this chapter uses a combination of basic image processing algorithms to enhance the visibility of bones while removing the flesh, with thresholding to create a binary image. Noise and objects unlikely to be bones are removed, and the
Arnar Þór Stefánsson

Figure 3.2: Failed direct thresholding of x-ray image using Otsu’s method.

resulting image is used directly to control the vertical cut. Some additional processing is required for the 3D localization of bones.

3.1.1 Pre-processing of Domus Images

For reliable 3D calculations the images used need to be perfectly aligned. The Domus x-ray images have been modified (cropped) by the software used to interface with the x-ray equipment. Inspection revealed that the spatial resolution of the images has not been changed, and the images were realigned by manually cropping.

The images themselves are of high quality, low noise and with good contrast between bones and flesh.

A $[n \times n]$ median filter is applied to the image to reduce noise. Median filters are edge preserving and suitable for noise reduction in images where edge and object location is of critical importance (Gonzalez & Woods, 2002, Ch. 3.6.2). The filter is a ranking filter, where pixels in a neighbourhood are ranked and a new value for the (usually) centre pixel is chosen based on the value of the surrounding pixels. Max, min and median filters are examples of ranking filters.

The smallest bone in the image is only around 2-4 pixels wide (0.2-0.4 mm), which limits the size of the usable filter to $3 \times 3$ pixel neighbourhood. Using a larger neighbourhood is expected to reduce the visibility of smaller bones thus risking loss of information.
3.1.2 Background Removal

The variable background (because of the changing thickness of the fillet) presents a problem for global thresholding. Isolating the bones was done using three different methods, which were then compared. The methods used were frequency domain bandpass filtering using a Fourier transform; and two versions of background subtraction where the background images are created using different low pass filters, either median filtering with a large neighbourhood; or a morphological closing of the original image (Gonzalez & Woods, 2002; Szeliski, 2011; Russ, 2011).

Frequency Domain Bandpass Filtering

Images, as well as other signals, can be expressed as a sum of a series of sinusoids. The 2D Fourier transform can be used to decompose an image to its frequency components. The results are represented by an image, the same size as the original, where each pixel indicates the contribution of a given frequency component to the original image.

The lowest wavelength signal that can occur in an image is one composed of alternating black and white pixels. The frequency of that signal, the maximum spatial frequency in the image, is

$$f_{\text{max}} = \frac{1}{2 \text{ pixels}}$$  \hspace{1cm} (3.1)

In the frequency domain image, the zero frequency component or DC-value (the average brightness) of the original image is displayed at the centre and the maximum frequency at the edges of the image.

Filtering the original image is done by manipulating the frequency domain representation. Certain frequencies, depending on application (e.g. high frequencies for edge detection), can be isolated or suppressed in order to emphasize or remove features in the image.

Because the bones are thin compared to most other features of the fillet and the bone/background contrast is relatively good, they can be isolated using a high pass filter or bandpass filter. The bandpass filter has the advantage of reducing or removing a significant part of image noise as well as isolating the bones.

For this implementation a bandpass filter is created by subtracting a second order Butterworth lowpass filter with a low cutoff frequency from one with a higher cutoff. The value of a Butterworth filter at coordinates (u,v) is given by
Where $D(u, v)$ is the distance of point $(u, v)$, $D_0$ is the cutoff value and $n$ is the order of the filter.

Prior to filtering the image it is padded by zeroes to increase it’s size to $[N \times N]$, where $N$ is the next power of two larger than the longer edge of the image, to make sure the frequency resolution of the Fourier domain image is equal in both directions.

**Median Background Subtraction**

In addition to the frequency filtering, the bones can be isolated by subtracting a background image from the original. The background image is created by ‘smoothing’ the original image, using a low pass filter or another suitable function.

In this case, median filters using a large window are used to prepare the background image. Median filters have minimal effect on edge positions in the filtered image, and a correctly sized filter will remove the bones and other small details from the background. A median filter with a neighbourhood that has a radius slightly larger than the average width of the thickest bone (the width of the bone in the image is 18 pixels) is expected to give good results, provided the distance from bone to bone is large enough to accommodate the window. The main drawback of median filtering is the time it takes to process the image. Real time implementations are limited to small window sizes.

**Morphological Closing Background Subtraction**

Here the background image is created using morphological closing. The closing operation is the erosion of the dilation of the original image.

The dilation operation closes any ‘holes’ that are in the image (the bones) by expanding the lighter areas of the objects in the image by a distance equal to the radius of the structuring element used. This operation is then followed by the erosion of the image using the same structure element. This reduces the size of the objects in the image to equal the original (e.g. fillet borders in same place in original and closed image) without reopening closed holes (the bones are removed).
The process of subtracting a closed image from the original is known as black top-hat filtering or bottom hat filtering and is useful for feature extraction and image enhancement.

The thickest bones in the x-ray image are between 16 and 18 pixels wide. Therefore the radius of the structure element expected to give the best performance is 9 or 11 pixels.

### 3.1.3 Thresholding

When two distinct classes of objects (bone and background in this case) are present in a grey-scale image they can be separated by choosing an appropriate threshold, $T$. The intensity value of each pixel, $f(x, y)$, is compared to the chosen threshold and assigned to a class depending on which side of the threshold the value falls (Gonzalez & Woods, 2002, Ch. 10.3)

$$
g(x, y) = \begin{cases} 
1 & \text{if } f(x, y) > T \\
0 & \text{if } f(x, y) \leq T 
\end{cases} \tag{3.3}
$$

where $g(x, y)$ is the resulting binary image. The easiest way to choose the value of $T$ is to manually set the threshold, which for single images potentially gives the best results, but has limited use in automated processes.

Two automatic threshold selection methods are used, Matlab’s graythresh: an image processing toolbox function implementing Otsu’s method, which selects the threshold value to minimize the intraclass variation, and the triangle method (Zack, Rogers & Latt, 1977).

Both methods use an image’s histogram, a representation of the distribution of grey values in the image, to calculate the threshold. Otsu’s method assumes a bimodal distribution, and the triangle method works well with tailed Gaussian distributions.

Comparing the histogram of the entire image (Figure 3.3, (a) and (c)) and the histogram limited to bones (Figure 3.3, (b) and (d)) shows that prior to processing, the bones are spread out behind a multi-modal distribution.

Processing the images reveals the bones by setting the majority of background pixels to zero.

The images being classified fall somewhere between those two distributions, and can either be viewed as bimodal, where one peak is considerably smaller than the other, or
Figure 3.3: (a) shows the histogram for the entire image prior to processing, (c) shows the histogram after processing, (b) shows the distribution of pixel values for bone pixels prior to processing and (d) shows the distribution bone pixels after processing the image.
unimodal with a small tail. Figure 3.4 shows a typical histogram distribution but the scale has been changed significantly for clarity (the highest peak scaled down, and the tail magnified).

**Triangle**

To find the threshold value using this method, both the highest peak in the histogram (most common grey value) and the brightest grey-value present in the image is found. The perpendicular distance from the line connecting those points (points A and B in Figure 3.4) to the histogram is maximized, and the value of the threshold determined from the coordinates of intersection between the maximized line and the histogram.

![Figure 3.4: A histogram of a simulated greyscale image. Perpendicular distance from the line (A-B) to the histogram is maximized, and the coordinates used for the threshold. The histogram used shows a typical distribution of values after filtering the image, but the scale has been changed for clarity.](image-url)
3.1.4 Image Clean-up

After thresholding a significant amount of noise is present in the binary image, it is removed by applying a series of morphological operations in the following order:

**Thin** Removes pixels from the objects, while maintaining the connections between regions. For each pass the outermost pixels of an object are removed. If an object contains holes, pixels are removed from both borders. This operation affects random noise (irregular structure, holes in objects) more than the bones (relatively large, no holes), making the noise more likely to be removed by area open.

**Area open** Removes all objects that have an area smaller than a chosen value.

**Thicken** Restores the bones to their original size by adding pixels to the border.

**Dilate** Fills in gaps that can occur on the edges of bones.

**Clear border** Assuming that all relevant bones appear close to the centre of the image, this removes false positives that touch the edges of the image. This command is mainly used to remove the effects of markings placed on the images from Röntgen Domus, and may not be needed for the final design.

These operations remove most of the false positives from the binary image. The final step assumes that any bones visible in the image have an approximately linear shape. Matlab’s [regionprops](https://www.mathworks.com/help/images/ref/regionprops.html) function calculates the eccentricity of an ellipse having the same second moments as the object detected in the image. The eccentricity of an ellipse is the ratio of the distance between its foci and major axis length. For circles the eccentricity is zero, but for long and thin objects (bones) the eccentricity approaches one.

Any object that has an eccentricity lower than a certain threshold is removed from the image.

3.2 Image Segmentation

The segmentation takes the classification a step further and separates and assigns labels to each object detected by the segmentation algorithm. Each labelled object is taken to be a separate bone and should be matched with a corresponding object in the second image of the stereo pair. The simplest way to label the bones is to use the thresholded images directly and assigning a label to each region visible without further processing. This is sufficient if the classification is accurate, and there is no overlap of bones in the image. If overlapping bones are present in the image more advanced image processing methods...
that are able to separate the bones that are touching, such as the watershed algorithm, are needed.

Because of the limitations of the 3D calculations, bones that have their major axis orientation aligned to the $y$-axis give unreliable results. Those bones will be cut using a vertical cutting profile and therefore their position in the image is marked separately. The orientation of the bones is automatically determined using the ‘Orientation’ measurement in the `regionprops` function.

### 3.3 Selected Results

To determine the ideal parameters of the preprocessing functions for use with these images, each parameter is varied over a sizeable region and every combination of parameters within these limits is tested. The scoring method detailed on page 21 is used to estimate the segmentation quality of each combination.

The reasoning for the choice of parameters is provided in the relevant subsections. A summary of the parameters and the range over which they are varied for these images is provided below.

**Median filter**

A $n \times n$ neighbourhood around each pixel is used to determine the median value.  

$$n \in \{1, 3, 5, 7, 9\}$$

**Morph. Closing**

The radius in pixels, $r$, of the structure element is changed.  

$$r \in \{3, 5, \ldots, 15\}$$

**Median Background**

Here $n \in \{3, 5, \ldots, 25\}$

**Fourier Transform**

Both the lower limit, $f_{\text{low}}$, and higher limit, $f_{\text{high}}$, of the bandpass filter are changed.  

$$f_{\text{low}} \in \{1, 2, 3, 4\} \text{ and } f_{\text{high}} \in \{8, 10, \ldots, 24\}$$

**Triangle Threshold**

The threshold can be shifted by a ratio, $T_{\text{shift}}$, of the highest grey value to either direction.  

$$T_{\text{shift}} \in \{-0.10, -0.08, \ldots, 0.10\}$$

**Area Open**

All objects with an area smaller than a chosen threshold, $A_T$, are removed from the image.  

$$A_T \in \{0, 10, 15, 20, \ldots, 80\}, \text{ where a threshold of 0 keeps every object.}$$
The parameters and scores are grouped based on which background removal method was used to generate the image. The 20 highest ranking combinations are shown in the appendix while the main results are summarised below.

The weights used when combining the sub-scores \( s_1, s_2 \) and \( s_3 \) are \( w_1 = 1 \), \( w_2 = -1 \) and \( w_3 = -100 \). A high value for the total score, \( s_{\text{total}} \), is a good indication of a successful classification. The bones and the area around them are given equal weights, but have opposite signs because of how the sub-scores are defined. The third score has a higher weight. That is because the scores are ratios of correctly, or incorrectly, classified pixels to the total number of pixels in the area. The majority of the image contains background pixels, which means that unless the scaling is applied, bone-sized or larger regions can be misclassified without a noticeable change of \( s_3 \).

### 3.3.1 Thresholding Results

Figure 3.5 shows the result of applying both the triangle threshold algorithm and Otsu’s method (Matlab: `graythresh`) to a preprocessed image. The image has been filtered using a 3-by-3 median filter, and the background removed by subtracting a morph. closing made with a disk shaped neighbourhood of radius 9.

Despite assuming a bimodal distribution, thresholding using the Matlab implementation of Otsu’s method (Figure 3.5a) gives surprisingly good, but rather aggressive results. Even without further processing the background in the image is almost perfectly clear, but a significant portion of the bones is removed as well.

The results of using the triangle threshold algorithm are shown in Figure 3.5b. Here all of the bones are clearly visible, but a significant amount of noise remains. Applying the clean-up procedure described in chapter 3.1.4 removes nearly every misclassified pixel from the background, leaving only a small region in this case (Figure 3.5c).

As the goal of the project is to be able to detect all bones present in the fillet, and thresholding using `graythresh` removes some of the smaller bones; the triangle thresholding method is used for the remainder of the chapter.

### 3.3.2 Background Removal Method

Out of the three methods tried, creating a background estimation using the morphological closing of the image and subtracting proves to be the most successful, correctly classifying 91% of the bones and a background misclassification rate of around 0.1%.
The median background subtraction is moderately successful, has a slightly lower rate of correctly classified bones and increased misclassification of the background. The method using the Fourier transform is the least successful when comparing $s_1$ and $s_3$, but $s_2$ is better than for the other methods. A low score both for $s_1$ and $s_2$ indicates that a good part of the bones and area around them is classified as background.

**Fourier Transform**

The combination of parameters giving the best results with the Fourier transform vary more than for the other methods. The results are not sensitive to prior application of a noise filter, as the high frequency noise is automatically removed by the bandpass filter.

The images in Figure 3.6 were made using the parameters from rows 1 (left image pair) and 11 (right image pair) in table A.1. The images demonstrate the theme in high scoring classifications using this method, where the background misclassification rate can be reduced significantly but at a cost to the correct classification of bones.
**Median Background**

The twenty highest scoring results, provided in the table, have some common parameters. Like the Fourier method, the application of a noise filter prior to creating the background image is not required. The background is created using a 19 to 25 pixel neighbourhood \((n \times n)\), which is close to the estimation based on the bone size. When using the larger neighbourhoods some loss of information occurs, as can be seen in Figure 3.7d.

**Morphological Closing**

The parameter combination giving the highest scoring result is almost constant, the top 20 images all use a \(3 \times 3\) median noise filter and in most cases a disk shaped structure element with a radius of 9 or 11. Figure 3.8c shows an almost perfect segmentation of the image where every bone is visible (correctly classified) and the misclassification rate of the background is minimal.

### 3.3.3 Segmentation

The highest scoring classification is used for segmentation.

The spacing between most of the bones is sufficient to accurately separate them, without using more advanced algorithms. A cluster of bones near the bottom of Figure 3.9a, is not separable but in this case the orientation of the bones would cause them to be labelled as unsuitable for 3D calculation.

The cluster of bones is detected and marked automatically, as well as the bones that are oriented along the y-axis.

### 3.3.4 Possible Improvements

Although the methods tried here give excellent results when segmenting these images, some questions about stability remain to be answered.

Advanced computer vision techniques have been shown (Mery et al., 2011) to successfully filter out false positives during classification without removing the smaller bones. Implementing a similar method as well as the watershed algorithm for segmentation has the potential to significantly increase the success rate of the algorithm. It might also be
worth exploring different combinations of thresholding or background subtraction methods, as the location of the misclassified areas changes depending on which method is applied, while the detection of bone pixels is more robust.

Limiting the bone detection to the region where the bones are most likely to be would improve the classification by filtering out many unlikely false positives. Adding this type of processing to the images produced in this chapter would result in almost perfect classification of the bones and no misclassification of the background. The position of the ROI might be determined by analysing a great number of cod fillets, and creating a model showing the probable locations of bones relative to other easily recognizable points on the fillet.

Optimization of the parameters using a large quantity of fillets (including fillets containing few or no bones), based on images acquired using the final equipment, are required.
Figure 3.6: Images showing two high scoring results of frequency domain bandpass filtering, made using parameters from row 1 ((a) and (c)), and row 11 ((b) and (d)) in table A.1. Images (a) and (b) show the resulting image directly after applying the filter, images (c) and (d) show the classification after clean-up.
Figure 3.7: Images showing the results of subtracting a background created using a large median filter, made using parameters from row 1 ((a) and (c)), and row 3 ((b) and (d)) in table A.2. Images (a) and (b) show the resulting image directly after applying the filter, images (c) and (d) show the classification after clean-up.
Figure 3.8: Images showing the results of subtracting a background created using a closing, made using parameters from row 1 ((a) and (c)), and row 5 ((b) and (d)) in table A.3. Images (a) and (b) show the resulting image directly after applying the filter, images (c) and (d) show the classification after clean-up.
Figure 3.9: Results of segmentation, (a) shows the bones as they are separated and (b) shows the bones marked as unsuitable for 3D calculations.
Chapter 4

3D Bone Localization

One of the main requirements for finding the 3D position of bones is the ability to accurately and precisely determine the position and shape of the bones detected in 2D images. Having properly segmented the images and identified the bones, the next step is finding the correspondence between images, that is: which bones in image A correspond to which bones in image B; and if any bones are occluded or otherwise missing from one of the images.

4.1 Image Correspondence

Steroscopic 3D calculations rely on being able to identify interest points or features in two images from slightly different viewpoints and creating a set of correspondences between images. The features can be points or patches and are selected such that they are easily detected and identified in both images. Identifying a correspondence pair is mainly done by tracking, where image B is searched for a feature found in image A, or by matching, where features are independently found in images and matched based on the similarity of their neighbourhoods.

The bones in an x-ray image have a limited number of usable features as they lack texture, and the shape, size and neighbourhood can change abruptly between images.

Two types of correspondence are required, first the bone in image B corresponding to a given bone in image A must be determined. Assuming that the classification and segmentation are accurate, the x-coordinates of the extreme points (leftmost and rightmost points) should not change between images. Those points combined with the maximum allowed translation, based on the fillet thickness and $\alpha$, the angle between the x-ray beams,
is used to determine bone correspondence. This can be seen in Figure 4.1, where the extreme points have been labelled (small ’x’), and correspondence drawn (green lines) on an image created by combining image A & B. During segmentation of the x-ray image, a single bone may be labelled as multiple bones due to unfavourable thresholding. The extreme points are used to partially fix this by comparing the leftmost point on the right bone segment to the rightmost point on the left bone segment. If the distance between those points is below a set threshold, the two parts are treated as one bone for the purpose of determining its shape.

This is not used in the generated images, where the labelling of objects is consistent enough to use directly for correspondence (that is, object labelled ’1’ in image A is also labelled ’1’ in image B).

Secondly because of the changing length and shape of bones, corresponding pixels within each bone pair must be found to successfully calculate the 3D shape of the bone. As the
grey-scale value and neighbourhood can change between images, and the bones have no distinctive textures, pixel matching can be rather difficult.

By modelling the shape of the bones (the centreline) and assuming again that if a bone segment has coordinates \((x, y_1)\) in image A, the coordinates of that same segment in image B should be \((x, y_2)\).

### 4.2 Describing Bones

Simple 2D images, created from the segmented CT voxel data were used to develop the method of describing the bones and finding 3D position. In the simplified images the soft tissue is ignored and several bones present in the CT-scan are left out. This minimizes the chances of bone overlap and eliminates the need for any processing or thresholding of the image.

Based on the assumption that the difference in \(x\)-coordinates of an object between images in a stereo pair is negligible, modelling the bones in each image as functions of \(x\) can significantly reduce the complexity of the task.

Bone \(i\) in image A is modelled as

\[
y_{Ai} = f_{Ai}(x)
\]

(4.1)

and its corresponding bone in image B can be described using

\[
y_{Bi} = f_{Bi}(x)
\]

(4.2)

where \(y_{Xi}\) is a vector containing the \(y\) coordinates of the estimated centreline of bone \(i\) in image \(X\), \(f_{Xi}\) is the function describing the bone and \(x\) is a vector containing values for \(x\) coordinates distributed evenly along the bone.

The input range for the functions, \(x\), is the same for both images, therefore, assuming that the bones have been successfully segmented,

\[
\Delta y = y_{Bi} - y_{Ai}
\]

(4.3)

for equally spaced points along the bone can be obtained and used to calculate the 3D position of the bone.
Different methods of representing the bones were tried, linear regression models with different basis functions, and skeletons, which will be discussed in the following sections.

4.2.1 Linear Regression Models

Linear regression models (LRMs) are used for determining the relationship between variables, often to determine how the value of a target variable (in this case \( y \)) changes for different input data sets, \( x \). Often the goal of the analysis is the construction of a predictive function \( f(x, w) \), capable of estimating the value of \( y \) for new sets of input data. A simple example function of this form is an equation for a straight line in two dimensions.

\[
y = f(x, w) = w_0 + w_1 x
\]  

(4.4)

Note that \( x \) represents a column vector of input variable for multidimensional models, while \( x \) indicates a single input variable. In this general introduction to linear regression models \( x \) is used for any input which may or may not include the \( x \)-coordinate.

It is possible to increase the expressive power of the model by replacing the linear function of input variables with a linear combination of non-linear functions of the input variables, or basis functions \( \phi_j(x) \), such that

\[
y = f(x, w) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(x)
\]  

(4.5)

where \( M \) is the total number of parameters in the model. By adding a 'zeroth' basis function, \( \phi_0(x) = 1 \), the equation can be simplified to

\[
f(x, w) = \sum_{j=0}^{M-1} w_j \phi_j(x)
\]  

(4.6)

The parameters of a model, \( w \), are found by minimizing the sum of squares error function

\[
E_D(w) = \frac{1}{2} \sum_{n=1}^{N} (y_n - w^T \phi(x_n))^2
\]  

(4.7)
where \( N \) is the number of data points available. Setting the gradient to zero and solving for \( \mathbf{w} \) gives (Bishop, 2005, Ch. 3.1) (Bishop, 2007)

\[
\mathbf{w}_{ML} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{y}
\]  

(4.8)

where \( \mathbf{w}_{ML} \) are the parameters for the maximum likelihood fit of the model to the data.

**Polynomial Basis Functions**

The simplest way to represent bones is by assuming that the bones are thin, long (compared to width) and relatively straight. Those bones could be described using a linear approximation, where

\[
f(x, \mathbf{w}) = w_0 + w_1 x
\]

(4.9)

As most bones are curved to some extent, the quality of the fit can be improved significantly by extending the model to include higher order terms. The linear fit is a special case of a \( n \)-th order polynomial basis, with \( n = 1 \). In general the basis functions take the form,

\[
\phi_j(x) = x^j
\]

(4.10)

and the complete set of function for a \( n \)-th order basis is

\[
f(x, \mathbf{w}) = \sum_{j=0}^{n} w_j \phi_j(x) = w_0 + w_1 x + w_2 x^2 + \ldots + w_n x^n
\]

(4.11)

Polynomial fits of up to fifth order were applied to the data, but no significant improvement was seen above third order. In many cases higher order fits oscillate wildly and are unsuitable to model the bones.

**Gaussian Basis Functions**

The bones can also be modelled using other basis functions, such as a combination of Gaussian functions.
where $\mu_j$ controls the position of the curve and $s$ controls its width, here all of the basis functions have the same width ($s$). As before the functions are combined using weights $w_j$, controlling the magnitude of each Gaussian in the basis.

### 4.2.2 Improved Regression using PCA

Because of the curvature of bones it can be impractical to use functions of $x$ to fit the data. Figure 4.2 shows examples of bones that can not be modelled accurately using functions of $x$, as some values of $x$ would require two output values ($y$).

![Figure 4.2](image)

Figure 4.2: Examples of bones that can not be represented accurately by functions of $x$. (a) is from fillet imaged at Röntgen Domus, images (b) and (c) are taken from different fillets imaged using CT.

In the cases shown the shape of the bone could easily be described as a function of $y$, but simply rotating the image or redefining the functions is not sufficient, due to the varying orientation and shape of bones, which make conflicts difficult to avoid.

But by transforming each bone detected individually to a new coordinate system, where the major and minor axis of image intensity replace $x$ and $y$ in the LRM equations respectively this can be overcome.

This can be accomplished by applying principal components analysis (PCA) to the original data. PCA can be used to transform the original set of variables ($x$ and $y$) to their principal components (PCs). PCs are linear combinations of the original variables, ortho-
gonal to each other, such that the greatest variation of the data lies on the first PC, the second greatest on the second PC and so on (Bishop, 2005, Ch. 12.1). The number of PCs is never greater than the number of original variables.

The pixel coordinates of the bones are transformed to the new coordinate system, where the distribution of values along the first PC can be interpreted in some sense as the approximate length of the bone, and the distribution over the second PC as its width.

The data is normalized before application of PCA

$$x_{std} = \frac{x - \mu_x}{\sigma_x}$$

(4.13)

where $\mu_x$ is the mean of the $x$ coordinates and $\sigma_x$ its standard deviation. The $y$ coordinates are treated the same way. Subtracting the mean centres the data and is recommended practise, prior to applying PCA (default setting of Matlabs $\texttt{pca}$ function). Dividing by the standard deviation rescales the values to have unit variance, which ensures that all dimensions are treated with equal importance, on the same scale.

The different fitting methods are applied to the data in the new coordinate system, and the resulting fit transformed back to image coordinates and used directly to calculate the depth of different parts of the bone.

4.2.3 Skeleton

In image processing a skeleton of an object is a thin (single pixel wide) representation of the shape of the object of equal distance to two (or more) of the boundaries of the object (Sonka, Hlavac & Boyle, 2008, Ch. 13.5). The skeleton is created by removing pixels on the boundary of objects, while keeping the shape intact. The pixels remaining when no more can be removed without changing the structure are the objects skeleton.

The 3D shape of the bones could either be calculated directly from the pixels of the skeleton, or the shape of the bones determined by fitting a LRM to the skeleton.

Both applications of skeletons to the task were tried, but neither worked as hoped. The somewhat irregular boundary of the bones creates a lot of branches which affect both the shape of the LRM and make direct calculations inaccurate.

Removing or reducing the branches was not successful, as in many cases the correct branch (the one more accurately following the shape of the bone) is removed while an incorrect branch remains.
While further work may enable the use of skeletons to accurately calculate the 3D shape of bones, no evident improvement over LRMs is seen and thus their application will not pursued.

### 4.3 Selected Results

The different methods of modelling bones are applied to the available images and the resulting fits and 3D calculations using different methods are compared. In this section the results for the x-ray image from Röntgen Domus, the one used in the previous chapter, and the highest resolution SINTEF image (T1) are presented.

#### 4.3.1 Bone Modelling

**Choice of Model**

The model used to describe the bones is required to accurately follow the approximate centre of the bone, while being relatively unaffected by small shape defects present in the outline of bones after the processing of the image. As mentioned earlier, skeletons are too sensitive to irregularities on the border of bones, and are therefore not suitable to this task.

Linear regression models are less sensitive to those irregularities. Figure 4.3 compares the results of using LRMs with first, second and third order polynomial basis.

Each image shows the bones and the model that has been fitted to the bone for two overlapping images, the image from sensor A and from sensor B. In general, the bones that are oriented along the y-axis in the image can not be modelled well using this method. None of the methods manage to accurately model the small bone present close to the centre of the image.

The images show a clear improvement in fit quality when the order of the polynomial used is increased, however fits using higher order polynomials were not useful. Based on those images first and second order polynomials are not sufficient to model the bones while the third order model is better able to follow the curvature of the bones.

In addition to using polynomial basis functions, regression using a Gaussian basis was applied to several isolated bones. The resulting fits were not accurate enough to base 3D calculations on, and a full implementation was abandoned.
Figure 4.3: (a) shows the results of fitting first order polynomials to a segmented image, generated using SINTEF cod data $T_1$, (b) uses second order polynomials, and (c) third order.

**Regression on Data Transformed by PCA**

In order to improve the results the bones are rotated to a new coordinate system found by PCA. This significantly improves the quality of the fits, as can be seen in Figure 4.4.

Here the model fits the bones that are oriented along the $y$-axis much better, but because of the limitations of the imaging method the accuracy of the 3D position calculated for bones or parts of bones that lie along the $y$-axis is expected to be poor. There is still difficulty fitting the smallest bone, overfitting in the case of the third order model and the incorrect shape for the first and second order.
Figure 4.4: (a) shows the results of fitting first order polynomials to PCA transformed data (original: segmented image generated from SINTEF T1), (b) uses second order polynomials, and (c) third order.

4.3.2 3D Calculations

The 3D calculations are made using the results of fitting a third order polynomial to the PCA transformed data. The resulting 3D data can be seen in Figure 4.5, where the grey blobs show the position of bones in the original CT image and the red lines show the calculated 3D position (note that the image is best viewed in 3D).

The calculated 3D gives an accurate estimate of the 3D position and shape of most of the bones, but as expected the calculated 3D for bone segments aligned with the y-axis (the lowest parts of the three bones closest to the viewer in Figure 4.5).

The results are somewhat dependent on the angle between the different x-ray paths. If the angle is too great the correspondence problem will be more difficult to solve and bone overlap increases, but too small an angle reduces the depth resolution of the 3D calculations. For these images an angle of 10° was a successful compromise between the
two, determined by varying the location of the sensors relative to the x-ray source when generating images using the SINTEF data. The number of overlapping bones in images generated using different angles for 5 of the 17 available arrays (selected at random) was determined by manual inspection, and based on the results a 10° angle was selected. The optimal angle is dependent on the properties of the imaging system and the object being imaged, in particular the resolution of the sensor, and the size and spacing of bones. The angle should be re-evaluated and optimized when a sensor has been chosen.

Despite the relatively small angle used in this case, the low resolution (2 pixels/mm) of several of the CT files, required some bones to be removed from the voxel arrays before generating the segmented image to prevent overlap.
3D is also calculated for the fillet imaged at Röntgen Domus (Figure 4.6) but the accuracy can not be confirmed as the true position of the bones is unknown. The x-ray images are processed using the highest scoring method suggested in the previous chapter, and the bones in the resulting segmented images (Figure 3.9) are automatically matched as demonstrated in Section 4.1 (Figure 4.1). The 3D shapes calculated for the bones are believable as the height of the bones does not exceed the approximate thickness of the fillets and there are no obviously improbable or incorrect shapes.

### 4.3.3 Possible Improvements

While the results of modelling the bones using polynomials are accurate, adapting and using active contour models (snakes) for the task may be able to increase the accuracy (Szeliski, 2011, Ch. 5).

A more thorough matching of the bones could be possible, e.g. by identifying if parts of bones are missing from images and finding partial matches in cases where bones are overlapping. If identified partial 3D may be found for parts of bones. A similar method could be used for bones that have sections aligned to the y-axis where the 3D information is unreliable. By detecting those sections and marking, the chance of accidentally cutting through a bone because of incorrect information is reduced.

It may be possible to minimize the number of bones where the 3D information is calculated by controlling the orientation of the fillet on the conveyor belt. This would require imaging a large number of fillets and determining the average or most likely orientation of the bones. Changing the orientation of the fillets has a few drawbacks, a wider conveyor belt is required and it would be more difficult to position the ROI correctly below the x-ray source, increasing the possibility of projection distortions of the bones.
Chapter 5

Concluding Remarks

5.1 Increased Yield from Cut Improvement

To verify that the 3D information obtained can be used to improve the yield by employing a more advanced cutting algorithm, two cutting methods were implemented as a 'proof of concept'.

**Vertical Cut**

The simpler one is a vertical cutting algorithm, where the output of the image classification is used as input and a path for the cutting instrument is calculated. The path is always at least 12 pixels (the maximum bone radius measured in the input image) away from the nearest bone and the cutting instrument is assumed to be capable of side to side (along the x-axis) movement speed that’s twice as fast as the conveyor belt ($\frac{dx}{dt} = 2$). This cutting method is a significant improvement over the v-cut as shown in Figure 1.1b.

**3D Cut**

Since the results of the vertical cut are used as a starting point to calculate the 3D cut, the worst case scenario gives results that are identical to the vertical cut.

The fillet is divided into slices and the intersection of each slice and the calculated bone position is found. The starting point of the cut (the position of the vertical cut) is compared to the area occupied by bones for each slice, and if an improvement is found by either tilting or shifting the position of the cut, it is stored and used for the 3D cut. The area
occupied by a bone in a slice through the fillet is taken to be a disk of radius 24 centred at the intersection point, 12 pixels to account for the thickness of the bone and 12 pixels as a margin of safety.

The different cuts are compared by estimating the volume that is removed from the fillet using each cut. By using this implementation of the 3D cutting algorithm, estimations show up to 30% reduced waste of material depending on the orientation of the bones in the fillet.

This implementation of the 3D cutting algorithm is extremely inefficient and not suitable for real time calculations of the cutting path.

5.2 Conclusions & Future Work

The results obtained using the CT-scans of cod fillets show that in theory using stereo-scopic x-ray imaging is a viable way of obtaining information about the 3D position of the bones. The 3D data is accurate enough to significantly reduce the waste when used to automatically cut a fillet, but other factors such as the high cost of system components, increased complexity of the system and potential difficulty of implementing a real time process have to be taken into account.

The image processing method presented in Chapter 3 performed quite well on the available images. Further tests are required to determine the robustness of the algorithm, where the method is applied to images of different fillets in varying condition.

Using principal components analysis to transform the bone coordinates before fitting a third order polynomial to the data to model the bone position, gives results accurate enough to retrieve the 3D position of the bones with reasonable accuracy.

By using this information to control the cut position it is possible to reduce the material waste significantly, but the amount depends on the orientation of the bones in the fillet. A new optimized cutting algorithm able to perform in real time needs to be created before the methods developed here can be applied in a commercial product.
Bibliography


Mery, D. (2013). X-ray testing by computer vision. In *In proceedings of 9th ieee cvpr workshop on perception beyond the visible spectrum*


Appendix A

Scores

A.1 Image Processing Results

The following pages contain tables showing the top ranked combinations for each background removal method. In all cases the triangle method is used for thresholding and the weights used when combining scores $s_1$, $s_2$ and $s_3$ are 1, -1, and -100 respectively.
<table>
<thead>
<tr>
<th>filtType</th>
<th>filtSize</th>
<th>fLow</th>
<th>fHigh</th>
<th>threshShift</th>
<th>arOpen</th>
<th>s1</th>
<th>s2</th>
<th>s3</th>
<th>Stotal</th>
</tr>
</thead>
<tbody>
<tr>
<td>med</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td>-0.80</td>
<td>80</td>
<td>0.8381</td>
<td>0.0920</td>
<td>0.0029</td>
<td>0.4562</td>
</tr>
<tr>
<td>med</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td>-0.80</td>
<td>70</td>
<td>0.8381</td>
<td>0.0932</td>
<td>0.0030</td>
<td>0.4409</td>
</tr>
<tr>
<td>med</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td>-0.80</td>
<td>75</td>
<td>0.8381</td>
<td>0.0932</td>
<td>0.0030</td>
<td>0.4409</td>
</tr>
<tr>
<td>med</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td>-0.80</td>
<td>65</td>
<td>0.8381</td>
<td>0.0932</td>
<td>0.0031</td>
<td>0.4314</td>
</tr>
<tr>
<td>med</td>
<td>1</td>
<td>1</td>
<td>24</td>
<td>0.000</td>
<td>65</td>
<td>0.8024</td>
<td>0.0665</td>
<td>0.0031</td>
<td>0.4305</td>
</tr>
<tr>
<td>med</td>
<td>1</td>
<td>1</td>
<td>24</td>
<td>0.000</td>
<td>70</td>
<td>0.8024</td>
<td>0.0665</td>
<td>0.0031</td>
<td>0.4305</td>
</tr>
<tr>
<td>med</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td>-0.80</td>
<td>60</td>
<td>0.8446</td>
<td>0.0979</td>
<td>0.0032</td>
<td>0.4279</td>
</tr>
<tr>
<td>med</td>
<td>1</td>
<td>1</td>
<td>20</td>
<td>-0.020</td>
<td>80</td>
<td>0.8399</td>
<td>0.0754</td>
<td>0.0034</td>
<td>0.4261</td>
</tr>
<tr>
<td>med</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td>-0.80</td>
<td>55</td>
<td>0.8446</td>
<td>0.0979</td>
<td>0.0033</td>
<td>0.4202</td>
</tr>
<tr>
<td>med</td>
<td>1</td>
<td>1</td>
<td>24</td>
<td>0.000</td>
<td>55</td>
<td>0.8024</td>
<td>0.0665</td>
<td>0.0033</td>
<td>0.4089</td>
</tr>
<tr>
<td>med</td>
<td>1</td>
<td>1</td>
<td>24</td>
<td>0.000</td>
<td>60</td>
<td>0.8024</td>
<td>0.0665</td>
<td>0.0033</td>
<td>0.4089</td>
</tr>
<tr>
<td>med</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td>-0.80</td>
<td>50</td>
<td>0.8464</td>
<td>0.0996</td>
<td>0.0034</td>
<td>0.4085</td>
</tr>
<tr>
<td>med</td>
<td>1</td>
<td>1</td>
<td>24</td>
<td>0.000</td>
<td>75</td>
<td>0.7851</td>
<td>0.0656</td>
<td>0.0031</td>
<td>0.4067</td>
</tr>
<tr>
<td>med</td>
<td>1</td>
<td>1</td>
<td>24</td>
<td>0.000</td>
<td>80</td>
<td>0.7795</td>
<td>0.0644</td>
<td>0.0032</td>
<td>0.3969</td>
</tr>
<tr>
<td>med</td>
<td>3</td>
<td>2</td>
<td>24</td>
<td>0.020</td>
<td>70</td>
<td>0.8212</td>
<td>0.0667</td>
<td>0.0036</td>
<td>0.3956</td>
</tr>
<tr>
<td>med</td>
<td>1</td>
<td>1</td>
<td>24</td>
<td>0.000</td>
<td>45</td>
<td>0.8069</td>
<td>0.0669</td>
<td>0.0035</td>
<td>0.3942</td>
</tr>
<tr>
<td>med</td>
<td>1</td>
<td>1</td>
<td>20</td>
<td>-0.020</td>
<td>75</td>
<td>0.8399</td>
<td>0.0758</td>
<td>0.0037</td>
<td>0.3937</td>
</tr>
<tr>
<td>med</td>
<td>1</td>
<td>1</td>
<td>24</td>
<td>0.000</td>
<td>50</td>
<td>0.8024</td>
<td>0.0668</td>
<td>0.0035</td>
<td>0.3905</td>
</tr>
<tr>
<td>med</td>
<td>1</td>
<td>1</td>
<td>24</td>
<td>0.000</td>
<td>40</td>
<td>0.8069</td>
<td>0.0669</td>
<td>0.0035</td>
<td>0.3886</td>
</tr>
<tr>
<td>med</td>
<td>1</td>
<td>2</td>
<td>18</td>
<td>0.000</td>
<td>75</td>
<td>0.7690</td>
<td>0.0661</td>
<td>0.0032</td>
<td>0.3843</td>
</tr>
</tbody>
</table>

Table A.1: Classification quality using frequency domain bandpass filtering.
<table>
<thead>
<tr>
<th>filtType</th>
<th>filtSize</th>
<th>medSize</th>
<th>threshShift</th>
<th>arOpen</th>
<th>$s_1$</th>
<th>$s_2$</th>
<th>$s_3$</th>
<th>$s_{total}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>med</td>
<td>1</td>
<td>23</td>
<td>0.000</td>
<td>10</td>
<td>0.8570</td>
<td>0.0976</td>
<td>0.0014</td>
<td>0.6217</td>
</tr>
<tr>
<td>med</td>
<td>1</td>
<td>19</td>
<td>-0.040</td>
<td>50</td>
<td>0.8932</td>
<td>0.1282</td>
<td>0.0015</td>
<td>0.6196</td>
</tr>
<tr>
<td>med</td>
<td>3</td>
<td>25</td>
<td>0.000</td>
<td>35</td>
<td>0.9181</td>
<td>0.1344</td>
<td>0.0017</td>
<td>0.6179</td>
</tr>
<tr>
<td>med</td>
<td>3</td>
<td>25</td>
<td>0.000</td>
<td>40</td>
<td>0.9181</td>
<td>0.1344</td>
<td>0.0017</td>
<td>0.6179</td>
</tr>
<tr>
<td>med</td>
<td>3</td>
<td>25</td>
<td>0.000</td>
<td>45</td>
<td>0.9181</td>
<td>0.1344</td>
<td>0.0017</td>
<td>0.6179</td>
</tr>
<tr>
<td>med</td>
<td>1</td>
<td>25</td>
<td>-0.020</td>
<td>75</td>
<td>0.8857</td>
<td>0.1186</td>
<td>0.0015</td>
<td>0.6173</td>
</tr>
<tr>
<td>med</td>
<td>1</td>
<td>25</td>
<td>-0.020</td>
<td>80</td>
<td>0.8857</td>
<td>0.1186</td>
<td>0.0015</td>
<td>0.6173</td>
</tr>
<tr>
<td>med</td>
<td>1</td>
<td>25</td>
<td>-0.020</td>
<td>25</td>
<td>0.8998</td>
<td>0.1254</td>
<td>0.0016</td>
<td>0.6163</td>
</tr>
<tr>
<td>med</td>
<td>1</td>
<td>25</td>
<td>0.000</td>
<td>10</td>
<td>0.8723</td>
<td>0.1035</td>
<td>0.0015</td>
<td>0.6160</td>
</tr>
<tr>
<td>med</td>
<td>1</td>
<td>25</td>
<td>-0.020</td>
<td>30</td>
<td>0.8990</td>
<td>0.1244</td>
<td>0.0016</td>
<td>0.6157</td>
</tr>
<tr>
<td>med</td>
<td>1</td>
<td>23</td>
<td>0.000</td>
<td>15</td>
<td>0.8525</td>
<td>0.0965</td>
<td>0.0014</td>
<td>0.6154</td>
</tr>
<tr>
<td>med</td>
<td>1</td>
<td>19</td>
<td>-0.040</td>
<td>55</td>
<td>0.8882</td>
<td>0.1279</td>
<td>0.0015</td>
<td>0.6153</td>
</tr>
<tr>
<td>med</td>
<td>1</td>
<td>25</td>
<td>-0.020</td>
<td>50</td>
<td>0.8894</td>
<td>0.1202</td>
<td>0.0015</td>
<td>0.6153</td>
</tr>
<tr>
<td>med</td>
<td>1</td>
<td>25</td>
<td>-0.020</td>
<td>55</td>
<td>0.8894</td>
<td>0.1202</td>
<td>0.0015</td>
<td>0.6153</td>
</tr>
<tr>
<td>med</td>
<td>1</td>
<td>25</td>
<td>-0.020</td>
<td>60</td>
<td>0.8894</td>
<td>0.1202</td>
<td>0.0015</td>
<td>0.6153</td>
</tr>
<tr>
<td>med</td>
<td>1</td>
<td>23</td>
<td>-0.020</td>
<td>35</td>
<td>0.8807</td>
<td>0.1150</td>
<td>0.0015</td>
<td>0.6147</td>
</tr>
<tr>
<td>med</td>
<td>1</td>
<td>23</td>
<td>-0.020</td>
<td>40</td>
<td>0.8807</td>
<td>0.1150</td>
<td>0.0015</td>
<td>0.6147</td>
</tr>
<tr>
<td>med</td>
<td>1</td>
<td>19</td>
<td>-0.020</td>
<td>10</td>
<td>0.8602</td>
<td>0.1045</td>
<td>0.0014</td>
<td>0.6138</td>
</tr>
<tr>
<td>med</td>
<td>1</td>
<td>25</td>
<td>-0.020</td>
<td>40</td>
<td>0.8939</td>
<td>0.1205</td>
<td>0.0016</td>
<td>0.6134</td>
</tr>
<tr>
<td>med</td>
<td>1</td>
<td>19</td>
<td>-0.040</td>
<td>40</td>
<td>0.8965</td>
<td>0.1308</td>
<td>0.0015</td>
<td>0.6125</td>
</tr>
</tbody>
</table>

Table A.2: Classification quality using a large median filter for background removal.
Table A.3: Classification quality using a morphological closing for background removal.