

# Volume Estimation of Fruit from Digital Profile Images


Keith Forbes

A dissertation submitted to the Department of Electrical Engineering,  
University of Cape Town, in fulfilment of the requirements  
for the degree of Master of Science in Engineering.

Cape Town, March 2000

# Declaration

I declare that this dissertation is my own, unaided work. It is being submitted for the degree of Master of Science in Engineering in the University of Cape Town. It has not been submitted before for any degree or examination in any other university.

Signature of Author .....  .....

Cape Town  
27 March 2000

# Abstract

This dissertation investigates the feasibility of using the same digital profile images of fruit that are used in commercial packing houses for colour sorting and blemish detection purposes to estimate the volumes of the corresponding individual pieces of fruit. Data sets of actual fruit volumes and digital images of the fruit that simulate both single and multiple camera set-ups are obtained. Shape features are extracted from the digital profile images using various digital image processing techniques. These shape features are used in conjunction with the actual volumes of the fruit as targets, to train and test both linear and artificial neural network methods of volume estimation. The results, in terms of error and of the associated confidence of the volume estimation methods, are presented.

# Acknowledgements

I am particularly grateful to my supervisor, George Tattersfield, who has been a source of limitless encouragement and enthusiasm. Throughout the course of this project, he has put tremendous effort into offering supervision of the highest possible standard. I would also like to thank David Durrett for his patient assistance during the data acquisition stage of the project, Bernard van Zyl and Jopie ten Cate of Vizier Systems for demonstrating and explaining various aspects of commercial fruit sorting equipment, Paul Stanford of Bridge Fruit for the provision of the Valencia orange weight data and Helen Divey of the University of Cape Town's Chemical Engineering Department for her assistance in loaning me laboratory equipment. The National Research Foundation provided me with funding for this dissertation for which I would also like to express my gratitude.

# Contents

<b>Declaration</b>	<b>i</b>
<b>Abstract</b>	<b>ii</b>
<b>Acknowledgements</b>	<b>iii</b>
<b>Contents</b>	<b>iv</b>
<b>List of Figures</b>	<b>vii</b>
<b>List of Tables</b>	<b>xii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Competing volume estimation methods . . . . .	3
1.3 Aims and methods of this project . . . . .	5
1.4 Preview of later chapters . . . . .	6
<b>2 Overview of Volume Estimation from Profile Images</b>	<b>9</b>
2.1 Literature review . . . . .	9
2.2 Discussion of approach . . . . .	12
2.3 Regression analysis and volumetric spread . . . . .	17
2.4 An analytically derived volume estimate for a mathematically described object . . . . .	20
2.4.1 Estimating the volume and associated error of an ellipsoid . . . . .	22
2.4.2 An example of ellipsoid volume estimation . . . . .	28
<b>3 Image Processing Methods</b>	<b>31</b>
3.1 Introduction . . . . .	31

3.2	Region labelling . . . . .	32
3.3	Binary morphology . . . . .	33
3.4	Boundary sampling . . . . .	37
3.5	Fourier descriptors . . . . .	39
3.6	The Hotelling transform . . . . .	44
3.7	Convex hull . . . . .	46
<b>4</b>	<b>Function Approximation Methods</b>	<b>48</b>
4.1	Introduction . . . . .	48
4.2	The structure of a multi-layer perceptron . . . . .	52
4.3	Preprocessing of data . . . . .	54
4.4	Training the multi-layer perceptron . . . . .	55
4.5	Comparison with some other predictor methods . . . . .	57
4.6	Testing function approximators . . . . .	58
<b>5</b>	<b>Data Acquisition</b>	<b>64</b>
5.1	Volume measurement . . . . .	65
5.2	Photographing the fruit . . . . .	68
<b>6</b>	<b>Preprocessing, Segmentation and Feature Extraction</b>	<b>75</b>
6.1	Preprocessing of the original images . . . . .	76
6.2	Size calibration . . . . .	77
6.3	Combining the colour bands . . . . .	79
6.4	Removal of the pear stalks . . . . .	85
6.5	Selection of the feature set . . . . .	87
6.5.1	Area . . . . .	88
6.5.2	Perimeter . . . . .	88
6.5.3	Aspect ratio . . . . .	89
6.5.4	Fourier ellipse sum . . . . .	90
6.5.5	Fourier circle sum . . . . .	92
6.5.6	Mean diameter . . . . .	92
6.5.7	Roundness . . . . .	93
6.5.8	Convexity . . . . .	94
6.5.9	Solidity . . . . .	94
6.5.10	Extent . . . . .	94
6.5.11	Energy of curvature . . . . .	94

6.5.12	Mean distance . . . . .	95
6.5.13	Variance of distance . . . . .	95
6.5.14	Circularity . . . . .	95
6.6	Feature computational complexity . . . . .	96
<b>7</b>	<b>Volume Estimation from Features</b>	<b>98</b>
7.1	Testing Miller's method for oranges . . . . .	98
7.2	Testing Marchant's method for potatoes . . . . .	101
7.3	Linear regression methods . . . . .	102
7.4	Redistributing volumes for purposes of error comparison . . . . .	108
7.5	Multi-layer perceptron regression methods . . . . .	113
<b>8</b>	<b>Conclusions</b>	<b>117</b>
<b>A</b>	<b>Selecting Weight Thresholds for Packing Fruit</b>	<b>120</b>
A.1	Statement of the packing problem . . . . .	120
A.2	Defining the cost function . . . . .	121
A.3	Testing the cost function . . . . .	126
A.4	Minimising the cost function . . . . .	127
A.5	Optimising thresholds for Valencia oranges . . . . .	129
A.6	Taking weight error into account . . . . .	130
<b>B</b>	<b>Volume Estimation Results Tables</b>	<b>134</b>
	<b>Bibliography</b>	<b>137</b>

# List of Figures

1.1	The basic components of an automated fruit sorting system. . . . .	1
1.2	Chessboard pattern projected onto an apple to show three-dimensional surface characteristics of the fruit by means of analysing the distortions of the grid pattern on the fruit's surface. . . . .	4
2.1	Two orientations of the same pear. . . . .	13
2.2	Profiles of two pears of equal volume, but the area of (a) is 10% less than the area of (b). . . . .	13
2.3	Flow chart to show an example of the progression from digital image to volume estimate. . . . .	16
2.4	Plot of computer-generated values of V versus S. . . . .	18
2.5	Parameterization of a fruit to arbitrary accuracy. . . . .	21
2.6	An ellipsoid with two of its semi-axes equal. . . . .	22
2.7	Probability density functions $p_A(a)$ and $p_B(b)$ . . . . .	23
2.8	Photograph of the ellipse at an arbitrary angle. . . . .	23
2.9	A point rotated anti-clockwise by $\theta$ in the $xz$ -plane. . . . .	24
2.10	Side view of the ellipsoid. . . . .	26
2.11	Example of probability density functions $p_A(a)$ and $p_B(b)$ . . . . .	28
2.12	Two examples of ellipses with volume estimates and associated uncertainties, given the PDFs in Figure 2.11. . . . .	28



2.13	(a) is the PDF of volume for the ellipse shown in Figure 2.12, and is the sum of (b) and (c) which represent the cases $r_2 = b$ and $r_1 = b$ respectively. . . . .	29
3.1	Black object pixel with (a) 4-connected and (b) 8-connected neighbours shown in grey. . . . .	32
3.2	(a) Binary image, with (b) three 4-connected objects, or (c) two 8-connected objects. . . . .	33
3.3	Example of binary erosion and dilation of an image. . . . .	33
3.4	An example of (a) a $6 \times 6$ binary image, $A$ , and (b) a $2 \times 2$ structuring element, $B$ . . . . .	34
3.5	The (a) erosion and (b) dilation of the binary image shown in Figure 3.4(a) using the structuring element shown in Figure 3.4(b). . . . .	35
3.6	The (a) opening and (b) closing of the binary image shown in Figure 3.4(a) using the structuring element shown in Figure 3.4(b). . . . .	36
3.7	Example of binary opening and closing of an image. . . . .	37
3.8	Boundary sampling. . . . .	37
3.9	Portion of $f(t)$ with the boundary shown projected on the side of the box. . . . .	39
3.10	An example of (a) $f(t)$ , and (b) magnitude of corresponding $F(\omega)$ . . . . .	41
3.11	Reconstruction of a lemon boundary from Fourier descriptor pairs. . . . .	42
3.12	Reconstruction of a pear boundary from Fourier descriptor pairs. . . . .	42
3.13	Formation of an ellipse from the first Fourier descriptor pair. . . . .	43
3.14	Rotation using the Hotelling transform . . . . .	44
3.15	Using the Hotelling transform to align a pear with the $x$ -axis. . . . .	46
3.16	An example of a set of boundary points with the convex hull shown. . . . .	46
3.17	Sorted points with convex hull points ringed. . . . .	47

4.1	Plot of orange weight versus volume with a linear regression line shown.	49
4.2	A set of data points with (a) the linear regression line; (b) the curve $y = ax^3$ ; (c) an exact interpolation of the data points. . . . .	50
4.3	A neuron or unit, the basic element of an MLP. . . . .	52
4.4	The structure of a single-output multi-layer perceptron network with a single hidden layer. . . . .	53
5.1	The pycnometer that was built for the purpose of measuring individual fruit volume (with a pear inside). . . . .	66
5.2	Using the pycnometer to measure fruit volume. . . . .	67
5.3	Mirror chamber and camera. . . . .	68
5.4	Top view and side view of the mirror chamber and camera. . . . .	69
5.5	Four cameras looking down at the fruit. . . . .	70
5.6	Computer-generated graphic of the mirror chamber with coloured ball. . . . .	70
5.7	Top view of the coloured ball in the mirror chamber. . . . .	71
5.8	Side view of the coloured ball in the mirror chamber. . . . .	71
5.9	Camera's view of the ball and reflections. . . . .	72
5.10	Bottom view of the ball with grey hidden region. . . . .	72
5.11	The effects of perspective at finite distance from a sphere. . . . .	73
6.1	Locating the four fruit images from the digital photograph. . . . .	76
6.2	Mirror chamber with size calibration sphere. . . . .	78
6.3	Manual identification of foreground and background showing (a) the original, and (b) the mask image. . . . .	80
6.4	Mask images indicating the regions of Figure 6.3(a) to be used for target values in training as (a) the background and (b) the foreground; and showing the pixels used in training an MLP on recognition of (c) the background and (d) the foreground. . . . .	81

6.5	Plot of red, green and blue values for the training pixels selected from the orange image shown in Figure 6.3. . . . .	82
6.6	(a) The normalised MLP output for Figure 6.3(a); and (b) the thresholded binary image of (a). . . . .	83
6.7	(a) The final binary image of the orange; (b) pseudo-colour image formed from original image and final binary image to show MLP segmentation accuracy. . . . .	84
6.8	Pla and Juste's thinning-based method of stalk detection. . . . .	85
6.9	Close-up of stalk meeting body with thinning-based segmentation. . .	86
6.10	Binary morphology-based method of stalk detection. . . . .	87
6.11	Close-up of stalk meeting body with morphology-based segmentation. . .	87
6.12	Measurement of object width and length. . . . .	89
6.13	Computer generated profile images of (a) upright; (b) rotated 30°; (c) rotated 60°; (d) rotated 90° pear shapes with corresponding plots of Fourier descriptor magnitudes to illustrate the Fourier ellipse sum feature. . . . .	91
6.14	(a) roughly side-on view of a lemon; (b) roughly end-on view of a lemon. . . . .	93
6.15	Mean number of floating point operations for computing shape features. . .	96
7.1	Scatter plot of volume versus area for the data set of orange images, showing the $V = KA^{\frac{3}{2}}$ curve for ideal spheres. . . . .	103
7.2	Plots of (a) aspect ratio; (b) roundness; (c) Fourier circle sum features versus $K$ for lemons. . . . .	107
7.3	Two perpendicular views of a potato. . . . .	112
A.1	An example of a category PDF of (a) fruit weights; (b) box weights. . .	123
A.2	An example of (a) a cost function, and (b) a probable cost distribution for a particular weight category. . . . .	125

A.3 Histogram of 1000 Valencia orange weights. . . . . 129

A.4 Best solution in each PBIL generation. . . . . 130

A.5 An example showing (a) an error function for a weighing mechanism,  
and (b) the actual PDF of a category with the original PDF shown  
dashed. . . . . 131

A.6 Histogram of error distribution for (a) orange volume estimation, and  
(b) orange weight estimation. . . . . 132

# List of Tables

2.1	Spread of $K$ for different fruit types. . . . .	14
3.1	Steps to calculate the convex hull in Figure 3.17. . . . .	47
5.1	Number of images acquired for each fruit. . . . .	74
5.2	Means and standard deviations of fruit volumes in the four data sets. .	74
6.1	Mean area, $\bar{A}$ , standard deviation, $STD(A)$ , and corresponding distance multiplier, Mult., for objects in calibration images. . . . .	78
7.1	Errors on volume estimates using mean volume, area and $area^{\frac{3}{2}}$ . . . .	104
7.2	Correlation coefficient of $K$ and shape features for different fruit types.	106
7.3	Errors on volume estimates using the best predictor of $K$ for each fruit.	108
7.4	Errors on volume estimates with original and redistributed data sets. .	110
7.5	Errors on volume estimates from multiple images using the average of area with $K$ -estimator volume estimation results. . . . .	111
7.6	Errors on volume estimates from single images using area with an implicit $K$ -estimator as MLP inputs. . . . .	114
7.7	Errors on volume estimates from multiple images using area with an implicit $K$ -estimator from each image as MLP inputs. . . . .	115
A.1	Box counts and currently-used Bridge Fruit weight category thresholds for Valencia oranges. . . . .	121

B.1	Errors on volume estimates using linear regression methods. . . . .	135
B.2	Errors on volume estimates using MLP function approximation. . . . .	136

# Chapter 1

## Introduction

### 1.1 Background

With the rise of consumerism since World War II, customers have become more discerning of quality and also more demanding of it. For example, supermarkets are under increasing pressure to provide fruit and vegetable produce of the highest quality. Now, more than ever before, fruit<sup>1</sup> is displayed and sold to consumers in batches of uniform quality, shape, colour, weight, volume and density. Because of these increasingly stringent requirements, and also as a result of advances in the available digital hardware and in image processing techniques, there has recently been much interest in using digital image analysis in grading systems at packing houses [21, 30, 39].

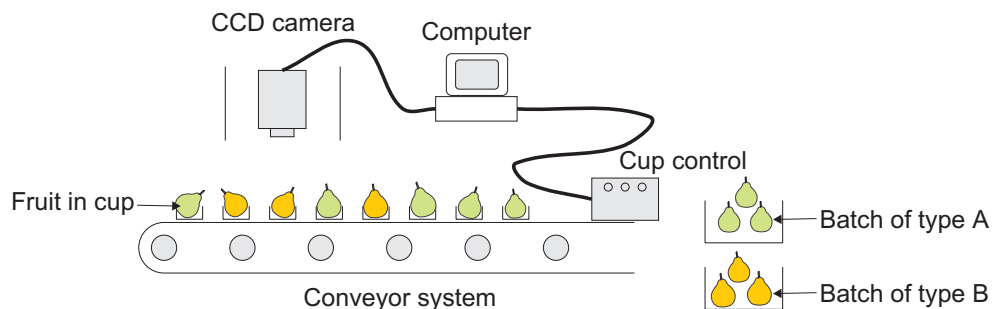


Figure 1.1: The basic components of an automated fruit sorting system.

---

<sup>1</sup>For convenience, the vegetable produce investigated in this project will always be referred to as fruit, since apart from the potato all of the produce is the biological fruit of the parent plant.

Such systems vary from one another, but usually consist of one or more charge-coupled device (CCD) cameras mounted above the packing line ‘looking’ at the fruit as it passes along on a conveyor. A diagram of the typical components of a grading system that uses digital image analysis in this way is shown in Figure 1.1. As the fruit pass under the camera they are photographed to form digital profile images. These images are analysed by computer and the results of the analysis are used to sort the fruit into batches of consistent colour, size and quality. Sorting is often implemented through the sending of signals to the cup control mechanisms on the roller on which fruit pieces sit, so as to redirect each fruit to its appropriate batch. Automation of the fruit sorting process has significant economic and labour saving benefits, especially when considered in the context of the relative inability of human graders to sort the fruit quickly and consistently [31]. Typically, a packing line is over 75m long and may pack one million pieces of fruit per day [36]. The throughput rate of such an inspection system might be 10 pieces of fruit per second [38].

This dissertation investigates the possibility of using these same digital images, which are mainly used in existing systems for colour sorting and blemish detection, to estimate the individual *volume* of each piece of fruit as it passes along the conveyor system. An estimate of the volume of each fruit would be a useful addition to a fruit sorting system for the following reasons:

- If the volume *and* the weight (determined by a mechanical weighing device on the packing line) of the fruit were known, then it would be easy to compute the fruit density, which is a useful measure for identifying the presence of *hidden* defects such as frost-damage and internal damage caused by insects. Such a system would be useful, for example, for oranges, the eating quality of which is related roughly proportionally to the density of the fruit [23].
- Alternatively, if the fruit densities were assumed to be constant, then the system could estimate fruit weight from the volume, replacing the need for a weighing device on the packing line. Such a system was developed by Marchant [21] for potato sorting—a technique which was justifiable because potatoes do not differ greatly in density from one another.
- Used in conjunction with a weighing device, such a weight estimate from volume could also be used to verify the output of the weighing device (which may jam and give an incorrect reading in an environment where fruit debris and spillage of fruit juice are unavoidable).



- A knowledge of fruit volume allows supermarkets to display fruit in batches of consistent size. Furthermore, consistently-sized fruit can be packed efficiently for transport and for display at the supermarket. Such consistency is also an ever-increasing demand of today's consumers.

Commercial fruit sorters require an estimate of fruit volume that can be calculated rapidly, that is easily incorporated into existing systems and that is cheap to implement. The root-mean-squared (RMS) error on the volume estimates should be less than 5% [38] to be of use either as a replacement or as verification of the currently-used mechanical weighing devices.

## 1.2 Competing volume estimation methods

At first glance, the solution to the problem of volumetric determination of fruit may seem to lie within the field of multiple view geometry—reconstruction of the three-dimensional surface of the fruit from multiple views provided by two or more cameras. Such an approach, however, would suffer several important time and cost disadvantages:

- Although some fruit sorting systems have multiple camera set-ups to view a larger portion of the fruit's surface than would be possible with a single camera, many single-camera systems would require the installation of a new camera at an appropriate viewing angle to reconstruct the three-dimensional structure of the fruit. Such an expense is not justified, considering the value of the volume information compared with the cost of such an installation [38].
- If multiple appropriately-positioned cameras were available, the problem of matching points in the various images would be difficult. This is because, as biological objects, the fruit lack visible features such as flat surfaces meeting at obvious corners that would be required for three-dimensional surface reconstruction of unlabelled images [28]. The fruit is often largely uniform in colour, and so without some marking system such as the projection of structured light onto the fruit, accurate matching of enough points to provide a surface model would be very daunting.

- Even with two cameras, less than half of the three-dimensional fruit surface could be reconstructed. If many cameras were used to determine the three-dimensional surface characteristics of the whole fruit, the problem of fitting the various pieces together would be highly computationally intensive and would be subject to error.

Projection of structured light onto a biological object for three-dimensional surface determination has been used before with success [5, 41]. This method is used to facilitate the point matching problem and has the added advantage that only one camera is required, because the view of the structured light pattern from the position of the projector is known and need not be photographed—in effect, the projector takes the place of one of the cameras. It still has the disadvantage that less than half the fruit surface can be modelled, since the optical plane of the camera must be at an angle to the plane of projection for the grid to be distorted. Figure 1.2 demonstrates the idea by showing the projection of a grid pattern onto an apple. Although occlusion of the projected light by protruding or indented features on the fruit's surface may yield benefits, such as allowing identification of the position of the calyx and the stem [41] (these concave regions cause discontinuity in the projected light), it remains a disadvantage to the accurate modelling of the three-dimensional surface.



Figure 1.2: Chessboard pattern projected onto an apple to show three-dimensional surface characteristics of the fruit by means of analysing the distortions of the grid pattern on the fruit's surface.

Another surface-determination technique is *shape from shading*, which recovers surface shape from changes in image intensity. This requires fixed illumination and imaging conditions, as well as a surface with known reflective properties. In the case of fruit grading, the objects to be inspected are biological, with surfaces that have unpredictable intensity variations. These difficulties, coupled with the fact that it is described as being of limited use in practice [17], indicate that this method is an unlikely candidate for the estimation of fruit volume.

Solutions to the three-dimensional shape determination problem which incorporate the use of ultrasonic devices [19] or x-rays have considerable potential for providing an accurate volume estimate, but the expense of the installation of the device would by far outweigh the benefits to the packing house of having volume measurements for their fruit.

### **1.3 Aims and methods of this project**

The above disadvantages have motivated the development of methods which estimate fruit volume from the fruit profile images that are currently used for colour and blemish detection by existing sorting systems. It is the development and testing of these methods that form the subject matter for this dissertation.

The methods make use of the *a priori* knowledge that the fruit is of a particular cultivar, and attempt to estimate the statistically-most-likely volume of the fruit by forming a function of profile shape features (such as perimeter and area). The function is trained on typical samples of fruit from the particular cultivar to determine the function parameters. The problem is thus one of function approximation or regression. This approach has the advantages of high speed (since the volume estimate will simply be a calculable combination of a few rapidly extractable shape features), of being easily adaptable to different fruit and packing house set-ups without the need for explicit camera calibration, and also of relatively low cost (since no extra hardware need be added to the sorting system).

Algorithms were developed to estimate the volume both of odd-shaped fruit such as the pear and the potato and of more regularly-shaped fruit such as oranges and lemons. The algorithms are robust in the sense that, provided that they are trained in the conditions in which they will be utilised, they are not reliant on the particular cultivar or

camera set-up used in the packing house (unlike an algorithm developed by the author [7, 9] in previous work, which relied on the mechanics of the conveyor system to align the fruit so that a side view was always seen by the camera). As long as the methods developed in this project are adequately trained on samples taken from the cultivar population, they will adapt to the specific environment of the packing house.

The purpose of this project is thus to develop statistical methods to estimate volumes from profile images of fruit and to test the performance of these methods experimentally. By quantifying the performance of these methods (calculating the RMS errors attained with use on different fruit types, namely oranges, pears, lemons and potatoes), their expected suitability for particular uses on packing lines can be established.

The practical work in the project included photographing and measuring the actual volumes of fruit samples to form data sets. In addition, the photographs were processed using digital image processing techniques to extract numerical features. These features were used, in conjunction with the actual measured volumes as target values, to form methods of volume estimation from the profile image features using artificial neural network and other statistical methods. The results of the various methods were tested on portions of the data set not used in the training process, in order to assess the performance of the different methods on unseen data.

The results of this study are for the use of manufacturers of vision-based fruit sorting equipment. To incorporate volume estimation from profile images into a fruit sorting system, the system would have to be trained on a sample of fruit from the population to be sorted. These sample fruit would have to have their volumes accurately measured to serve as target volumes before the system could be commissioned to estimate volumes from a function of shape features from digital photographs. After the training process, by analysing the corresponding digital image, a volume estimate would be given for each fruit that had passed beneath the camera.

## **1.4 Preview of later chapters**

Chapter 2 of this dissertation introduces the available published literature on the estimation of fruit volume from profile images. The general approach to estimating fruit volume statistically is then discussed in terms of how one might expect a volume estimation system to function. The important issues to be taken into account are pre-

sented. A mathematical analysis is then performed on a hypothetical ellipsoid ‘fruit’ to illustrate the inherent uncertainty in volume estimation from profile images, and to highlight some of the theoretical issues raised in approaching the volume estimation problem.

The bulk of this project is concerned with image processing and function approximation. Some of the theory and background to these two subjects, particularly as they relate to what was done in this project, is given in Chapters 3 and 4.

Chapter 3 presents the image processing methods which were used in this project to reduce the digital profile images of fruit from the camera into a format from which features could readily be extracted for regression analysis. The theory behind some of the shape features to be extracted is also discussed.

Chapter 4 discusses the theory behind the use of artificial neural networks (ANNs) as tools for regression and classification. ANNs play an important role in this project as they are used to combine the features extracted from the profile images of the fruit to form an estimate of the volume of the fruit. Attention is specifically given to the multi-layer perceptron (MLP) architecture of an ANN, as the MLP was the chosen ANN for function approximation in this project. The chapter also discusses many points which are applicable to function approximation in general. This includes explanations of the method by which the function approximators are tested, and of how an estimate of the confidence in the test results can be quantified.

Chapter 5 explains the experimental method used to measure the actual volumes (for use as target values in regression analysis and for testing the performance of the volume estimation methods) of hundreds of pieces of fruit. This chapter also details how thousands of digital images of fruit were obtained, using a mirror chamber that was designed and built by the author for the purpose of providing multiple images of each fruit piece. The mirror chamber could be used to obtain four images of a single piece of fruit simultaneously, and fruit pieces were invariably photographed in six orientations within the chamber so that 24 images of each fruit piece were available as data.

Chapter 6 describes the key operations of preprocessing, segmentation and feature extraction that were performed on the digital images for the various different fruit. This entire process essentially reduces an image to a vector of numbers which collectively describe the important characteristics of the corresponding profile image.

Chapter 7 shows how the various features from the digital images were combined using linear and artificial neural network methods, to provide volume estimates for the fruit data. The performances of the different methods are discussed and compared.

Chapter 8 concludes the investigation by summarising the results and their implications.

The details of a solution developed for a related optimisation problem concerning the selection of weight thresholds for packing fruit into boxes are described in Appendix A. It is shown how this optimisation problem is an interesting example of how a reduction in fruit weight estimate error (which could be derived from a volume estimate) can translate directly into savings for the packing house.

Tables of the error results and of the associated statistics for the various volume estimation methods developed in this project are repeated for convenience in Appendix B.

## Chapter 2

# Overview of Volume Estimation from Profile Images

This chapter discusses various aspects of the problem of volume estimation from profile images, beginning with a review of published literature on the subject. It explains how an initial volume estimate can be obtained from the area of the fruit profile, and how this estimate could be improved using an implicit estimation of the fruit's orientation with respect to the camera. This estimation of orientation is formed by taking the shape, as opposed to the size, of the fruit profile into account.

Towards the end of the chapter, a theoretical derivation of probable volume is presented for a simple mathematically-defined object. This is done to show how, in theory, an expected volume with an associated uncertainty can be calculated from a given profile of an object, provided that *a priori* knowledge about the population of such objects exists. Later in this dissertation, it is shown how, in practice, an estimated volume with an associated uncertainty can be calculated from a given profile of a fruit, using *a priori* knowledge about the population of such fruit.

### 2.1 Literature review

Although the available literature describes many machine vision fruit grading systems, the main task of these systems has been to classify fruit in terms only of colour and quality. For instance, at the University of Cape Town, Kay [18] developed colour clas-

sification functions for sorting fruit into standard colour classes from digital images of the fruit, and Henry [14] used similar methods and included some geometrical measurements to investigate the classification of dried fruit into standard quality classes.

Some implemented fruit sorting systems described in the literature include an on-line system for mango-grading [29] and a real-time video grading system for oranges [30]. The mango sorting system classifies mangoes according to colour, size (using linear, not volumetric, measurements), and taste (using infra-red sensors). The orange sorting system uses multiple orthogonal views of each orange, as it is propelled through an inspection chamber, to identify surface defects on the oranges. The diameters of the orange profiles are measured from the digital images.

Some commercial fruit sorting system manufacturers have sites on the Internet [1, 6, 15]. The fruit sorting systems which they implement are described as ‘custom designed’, and are fully integrated with the conveyor system. They promise to sort fruit into colour, size, shape and weight categories, and use recording software to keep track of each lot of produce.

In the available literature, relatively little attention is given to the sorting of fruit by size. Those systems that have been implemented to give some form of measure of the size of the fruit usually use a direct measurement from a two-dimensional image (such as fruit diameter), as opposed to using size and shape features explicitly to estimate *volume* as is done in this project. The former method is of little use with fruit that is not approximately spherical because there is usually no guarantee that the fruit will be appropriately orientated with respect to the camera; and the widest portion of the fruit, for instance, may happen to be either hidden or visible to the camera, giving inconsistent results even with the same piece of fruit (at different orientations).

Documentation of only two systems which estimate volume from shape parameters was found after an extensive world-wide library search. Both methods use relatively simple functions of one or two shape features to estimate volume. These shape features are measured or extracted from digital profile images of the fruit.

The first of these systems is the machine vision system described by Miller [23, 39] to estimate the volume of oranges for the purpose of identifying freeze damage. Freeze-damaged fruit have a lower-than-average density and can thus be identified and rejected through the use of some measurement of density. In the days prior to the determination of orange density from volume estimated by a machine vision system and



weight measured by a load cell, oranges were separated by density through the use of emulsions. The separating equipment contained an emulsion of purified mineral oil and water, whose specific gravity was adjusted to be more than that of the damaged fruit but less than that of the undamaged fruit. Sound fruit sunk onto one roller elevator, damaged fruit floated onto another; and the separated fruit were carried out of the emulsion [23].

From a digital image of an orange at arbitrary orientation, Miller used an algorithm which measured the orange diameter eight times at successive angular increments of  $22\frac{1}{2}^\circ$ . Using the arithmetic mean of these eight readings as the diameter,  $d$ , and by approximating the shape of the oranges to that of a sphere, the formula

$$V_{orange} \approx \frac{\pi d^3}{6} \quad (2.1)$$

gave readings with what the author described as “a larger variation than the estimate by liquid displacement” [23]. As Miller’s results were not quantified, his method is repeated on a new data set and is then compared with an alternative method later in this study. The alternative method estimates the volume as a function of fruit profile area. It is felt that this alternative method is simpler, and that it may even reduce volume estimation error because an area measurement is likely to be more immune to noise than the diameter measurements used by Miller.

The second system is described by Marchant [21, 37] as a process which estimates the weight of potatoes by forming a function of shape features. Although volume is not explicitly derived, Marchant’s assumption of uniform potato density essentially reduces his problem to one of volume estimation, albeit only by implication. It is far easier to obtain a training set with known target weights than with known target volumes, since the acquisition of accurate volume by measurement of liquid displacement is much more time consuming than measuring weight using a balance. It thus makes more sense, where density may be assumed constant, to use weight as the target variable of a function of shape features, since in this case nothing can be gained by first using volume and then dividing by a measured average density. The weight of potatoes was estimated by Marchant, using the average of

$$W_{potato} \approx \frac{(\text{projected area})^2}{\text{length}} \quad (2.2)$$

derived from 12 different images as the potatoes were rolled along a conveyor system. The average error on the estimated weight was 7%. Marchant’s method of weight

estimation for potatoes is repeated later in this study, using a new data set of potatoes, as an exercise in volume estimation and for comparison with other volume estimation methods.

## 2.2 Discussion of approach

This section discusses some of the issues and methods that are relevant in developing and testing methods for estimating fruit volumes. Most of these issues are discussed theoretically, with little reference to the actual data used in this project. The section thus serves to introduce the reader to some concepts that affect the approach taken in this project, and that are discussed with reference to the actual data used in the project in later chapters.

This study attempts to make a better estimate of volume than the methods described by Miller and Marchant, by using further shape features to parameterize the boundary shape of the fruit profiles and by using different methods for combining these features. Such an approach generalises the method, making it particularly applicable to odd-shaped fruit (such as the pear) which have boundary profiles of different size and shape depending on the fruit's orientation (which can in turn be implicitly predicted to an extent because most pears are approximately radially symmetric [8]). This discussion is primarily based on the pear, yet is applicable in many ways to other fruit. As will be seen later, the pear is a useful example because it clearly highlights the main issues in estimating the fruit volume from profile images.

If all pears (of differing volume) were of exactly the same shape, and were photographed in exactly the same orientation, then the pear volume would be a monotonic function of the area of the fruit profile. If the pear were sufficiently far from the camera, so that a profile boundary formed by effectively-parallel tangential rays is captured, then the volume would be related to the profile area by

$$V_{pear} = KA_{pear}^{\frac{3}{2}} \quad (2.3)$$

where  $V_{pear}$  is the volume of the pear in cubic centimetres,  $A_{pear}$  is the area of the pear profile image in pixels and  $K$  is constant for such shape-and-orientation-invariant pears. However, in the real world case where the shape and orientation of the fruit varies,  $K$  is not constant. Estimating the value of  $K$  from two-dimensional digital

profile images of the fruit can be seen as the crux of the volume estimation problem in this study.

(a) Smaller  $K$ (b) Larger  $K$ 

Figure 2.1: Two orientations of the same pear.

In a fruit sorting system, the projected profile area of the fruit as seen by the camera varies depending on the orientation of the fruit. As an example, Figure 2.1 shows the same pear photographed in two different orientations, resulting in different projected areas and different boundary profile shapes. Profile areas may also differ between pears of different intrinsic shapes. For example, a long thin pear of the same volume as a regular pear may have a larger projected area when viewed side-on than the regular pear. Figure 2.2 shows an example of such a case.

(a) Larger  $K$ (b) Smaller  $K$ 

Figure 2.2: Profiles of two pears of equal volume, but the area of (a) is 10% less than the area of (b).

By examining the standard deviations of  $K$  with a real data set of pear images, it is possible to gain an impression of the relative importance of the two factors that may cause  $K$  to differ from image to image, namely the variation in orientation and the variation in shape from pear to pear. In this project, four image data sets were formed, one for each of oranges, pears, lemons and potatoes. The acquisition of these data sets is described in Chapter 5. Table 2.1 shows, in the second column, the standard

Fruit type	$\frac{\text{STD}(K)}{\bar{K}}$ , one fruit	$\frac{\text{STD}(K)}{\bar{K}}$ , different fruit	$\bar{K}$
Orange	0.035	0.035	$5.32 \times 10^{-5}$
Pear	0.068	0.070	$6.19 \times 10^{-5}$
Lemon	0.091	0.091	$5.45 \times 10^{-5}$
Potato	0.155	0.151	$5.19 \times 10^{-5}$

Table 2.1: Spread of  $K$  for different fruit types.

deviation of  $K$ ,  $\text{STD}(K)$ , divided by the mean of  $K$ ,  $\bar{K}$ , calculated for the 24 images of each individual piece of fruit in the data sets. The value shown is an average derived from many measurements, each obtained over 24 images of each fruit piece at random orientations. The third column shows the standard deviation over the mean for  $K$  calculated for *different* fruit in random orientations. The value shown is the average of 24 such calculations for each fruit. The fourth column shows the average value of  $K$  for each fruit. The pears show a slightly larger value than the other fruit, since the data was gathered with a slightly different camera set-up, with the camera further back from the fruit than it had been with the other fruit types. This is the reason that the standard deviations were divided by the means for purposes of comparison.

Table 2.1 indicates, as would be expected, that the variation of orientation is the main contributor to variance in  $K$ , rather than the difference in shape between different fruit of the same type. This is because the change in the standard deviation of  $K$  when moving from one fruit at any orientation to any fruit at any orientation is small. Note that it would not be possible to calculate the standard deviation of different fruit at the same orientation, since the absolute orientation of a real fruit cannot usefully be defined because it has no rigorously predefined body axis. There is no need explicitly to define orientation for real fruit, but the concept of orientation is very useful in thinking about the problem.

Table 2.1 is also useful in showing how the spread of  $K$  varies in different fruit types. Oranges have a low spread of  $K$ , since they are almost spherical, and the profile area does not differ much with orientation. Pears and lemons have a greater spread than oranges, since the same fruit exposes different profile areas at different orientations. One might expect that the pear's  $K$  value could be predicted to an extent from examining the shape parameters of an image. This is because the pear's top and side views have distinctly different shapes. The prediction of  $K$  for lemons also seems viable, because lemons are roughly ellipsoidal and possess a certain degree of rotational symmetry. Later in this chapter, a theoretical examination of ellipsoids shows that there is a future in estimating volume (and hence predicting  $K$  implicitly) with a probabilistic approach. The potatoes have a large spread of  $K$ , since they are irregularly shaped. Although there is little obvious evidence that  $K$  could easily be predicted in the case of potatoes, a statistical approach may be able to give a probabilistic estimate of  $K$ .

By measuring certain shape features (for example circularity or aspect ratio [40, 32]) from the profile images of a pear, it is possible to determine the degree to which a pear is viewed from the side or the top [8], or how oblate or prolate it is. The shape features,  $x_1, x_2, \dots, x_n$ , may be of use in approximating a value for  $K$ , and hence in deriving a volume estimate. That is to say

$$K \approx f(x_1, x_2, \dots, x_n) \quad (2.4)$$

In fact it is better to use these features to predict volume directly and only *implicitly* to predict  $K$  or orientation. By directly calculating the volume of the fruit using

$$\hat{V} = f(x_1, x_2, \dots, x_n, A) \quad (2.5)$$

where  $\hat{V}$  is the estimated volume and  $A$  is the area of the fruit profile, the assumption that Equation (2.3) holds is no longer necessary. It remains, however, useful for descriptive purposes and also for a linear regression approach described later.

The features  $x_1, x_2, \dots, x_n$  should be specifically chosen to identify the probable shape and orientation of the pear from the given profile image. As a function,  $f$  is highly non-linear, but it may be approximated very effectively by an artificial neural network (ANN) that has been trained with a set of shape features, areas and corresponding target volumes for many different pears in different orientations [8]. The ANN attempts to approximate  $f$  by minimising the RMS error on the volume estimates. It is preferable to have a large training set and few shape features to encourage the ANN to

generalise and not to model the specific training set [3]. The use of only a few features has the added advantage of rapidly producing a volume estimate, reducing the number of calculations that would need to be made, for example, in a commercial sorting line. A good start for identifying satisfactory shape features to feed into an ANN is to check for correlation between each proposed shape feature and  $K$ . A feature which shows correlation is a good predictor of  $K$  and can be used in conjunction with profile area to train an ANN to provide a volume estimate.

By taking multiple images of the fruit either at fixed angles or at successive random orientations, it is possible to improve the volume estimate by even more than is achieved by averaging the volume estimates from the multiple images calculated separately. For instance, a function

$$\hat{V} = f'(x_{1_{image\ A}}, \dots, x_{n_{image\ A}}, x_{1_{image\ B}}, \dots, x_{n_{image\ B}}) \quad (2.6)$$

for two images  $A$  and  $B$ , taken at right angles to one another, will outperform the mean of the two volume estimates obtained from calculating two separate  $f$  functions, one for each image. A top view of a long thin pear will result in a volume estimate biased towards that of the average shape for pears, as no length information is present and the ANN would assume the pear to be of typical volume for its profile area. If a side view image is also available, then features derived from both images in conjunction will provide a better volume estimate than can be obtained from averaging estimates derived from the two views taken separately. This is because working with two separate views biases the estimate from the side view image (presumably a reasonable volume estimate because of the fruit's near symmetry about its stem-calyx axis) towards a volume that is typical of the mean pear volume for the top-view profile area.

Figure 2.3 shows the steps that are undertaken to derive a volume estimate from a single digital image.

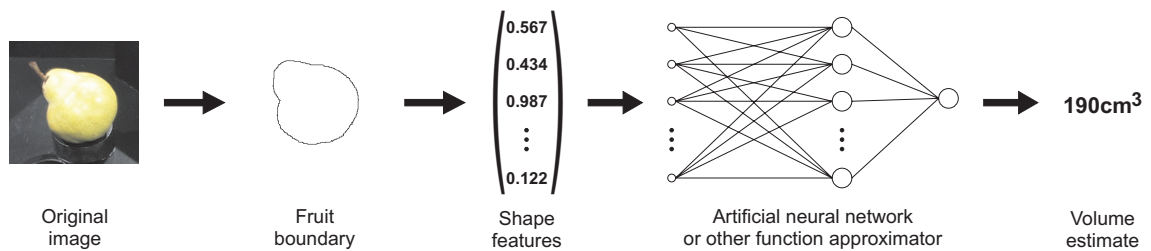


Figure 2.3: Flow chart to show an example of the progression from digital image to volume estimate.

The basic method of approach for the evaluation of the effectiveness of volume estimation algorithms has been as follows:

1. Accurately measure the actual volumes of a training set and test set of fruit.
2. Acquire multiple images of each fruit in the set.
3. Preprocess and segment the fruit images to identify the fruit profile boundaries.
4. Extract appropriate shape features from the fruit profile boundaries.
5. Using the actual fruit volumes as targets, train linear functions or ANNs to estimate the fruit volumes from the extracted features of the training set.
6. Test and compare the performances of the different volume estimation methods.

### 2.3 Regression analysis and volumetric spread

This section explains why it is necessary to consider the spread in the population of fruit volumes when interpreting the results of a statistical volume estimation. Consider a population of  $n$  fruit with volumes  $\mathbf{V} = (V_1, V_2, \dots, V_n)$ , each described by a single shape feature stored in corresponding elements of the vector  $\mathbf{S} = (S_1, S_2, \dots, S_n)$ , which is some shape feature function formed from features extracted from  $n$  profile images of the fruit (one image per fruit). For instance,  $\mathbf{S}$  might comprise

$$S_i = C \cdot \frac{width_i}{length_i} \cdot area_i^{\frac{3}{2}} \quad (2.7)$$

where  $S_i$  is the  $i^{th}$  element of  $\mathbf{S}$ ,  $C$  is a constant multiplier to convert  $S_i$  to the same units as  $V_i$  (say cubic centimetres), and  $width_i$  is the width,  $length_i$  is the length and  $area_i$  is the area (all counted in pixels) of the fruit object in the profile image of fruit number  $i$ . However, for the purposes of the discussion in this section, it is not important how  $\mathbf{S}$  is calculated. The shape feature  $S$  might be related to the corresponding volume  $V$  (dropping the subscript  $i$  and now treating  $V$  and  $S$  as variables) by the equation

$$V = S + u \quad (2.8)$$

where  $u$  is an uncertainty or error term, due to the fact that the fruit was randomly orientated when photographed and that all fruit in the population are not identically

shaped. Quantisation errors of the shape features due to the finite resolution of the camera, errors in the precise positioning of the fruit for the purposes of photographing it, any inconsistencies in lighting conditions and so forth may also contribute to  $u$ , to a limited extent. For any given shape feature  $S$ ,  $\hat{V} = S$  is the expected value for the volume of the corresponding piece of fruit. Thus  $u$  behaves as a random variable with zero mean in this context.

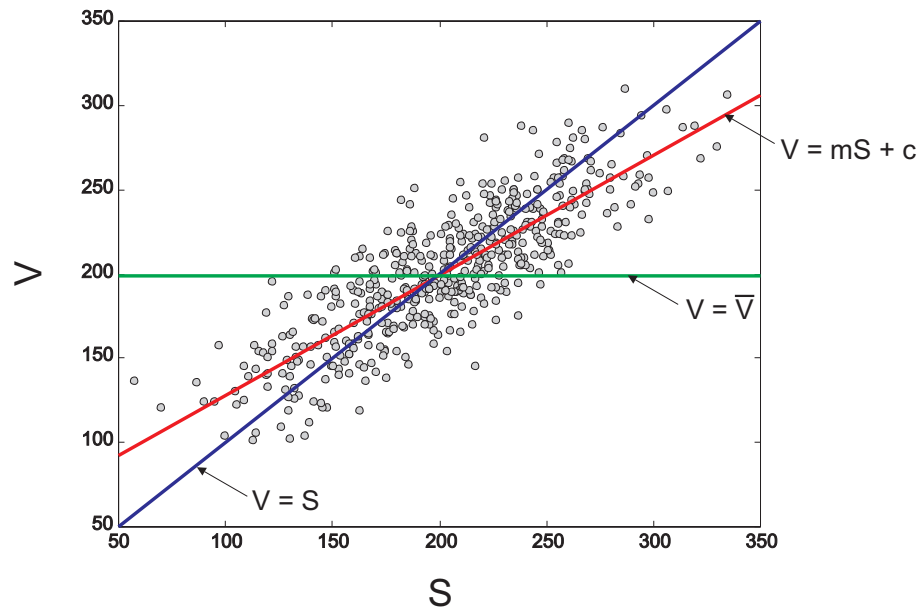


Figure 2.4: Plot of computer-generated values of  $V$  versus  $S$ .

Figure 2.4 shows an example of a plot of  $V$  versus  $S$ . The  $n = 500$  values of  $V$  in this plot were generated randomly by sampling a normally distributed probability density function (PDF) with a mean of 200 and a standard deviation of 40. The values of  $S$  were obtained by subtracting  $u$ , which was in turn obtained by sampling a normally distributed PDF of zero mean and a standard deviation of 25.

The blue line has the equation  $V = S$  and can be used to estimate  $V$  from a given  $S$ . Using  $\hat{V} = S$  to predict  $V$  from  $S$  gives an RMS error of 25.4 for this particular set of 500 points.

The green line has the equation  $V = \bar{V}$ , where  $\bar{V}$  is the mean of the 500 values of  $V$ . Volume can also be predicted using this equation. Using  $\hat{V} = \bar{V}$  amounts to predicting the volume of any new sample to be the mean volume of the population, and entirely disregarding the value of  $S$ . This method gives an RMS error of 40.2 on the 500 points.



The red line shows the linear regression line  $V = mS + c$  for the set of points. The slope,  $m$ , of this line may be calculated using a standard result from linear regression theory [22]

$$m = \frac{n \sum SV - (\sum S)(\sum V)}{n \sum S^2 - (\sum S)^2} \quad (2.9)$$

and the intercept,  $c$ , may be calculated using the equation [22]

$$c = \bar{V} - m\bar{S} \quad (2.10)$$

where  $\bar{S}$  is the mean of  $S$ . Using  $\hat{V} = mS + c$  to predict the values of  $V$  from  $S$  gives an RMS error of 21.1 over the 500 data points in this example. Note that although one would usually use a separate *test set* of data points to evaluate the RMS error on prediction, not using a test set does not in any way detract from the validity of the argument.

Although  $\hat{V} = S$  is the best estimator of volume in the absence of any information other than  $S$ , in the given example there *is* additional information, namely the distribution of volumes. One would expect new points drawn at random from the same population (considering the existing  $n = 500$  points to be a sample drawn from the population) to produce lower RMS errors on volume estimates when using the existing regression formula  $\hat{V} = mS + c$  than when using  $\hat{V} = S$ . This is because linear regression implicitly makes use of the *a priori* knowledge of the distribution of volumes when predicting volumes from new data. The  $\hat{V} = S$  method uses no knowledge of any *a priori* information (although this information could be used explicitly with this method in conjunction with Bayes' Theorem [20], but it is far easier to use regression methods which implicitly make use of the *a priori* distribution of the population).

A population with a greater spread (standard deviation) of volumes would have a regression line closer to the  $V = S$  line. In the extreme case, where the spread is infinite, the PDF of the volumes is uniform and the regression line is the  $V = S$  line. This is equivalent to having no *a priori* knowledge of the volume distribution of the population, with any volume being equally likely. This, of course, is impossible in practice for several reasons, one of which is that fruit with negative volumes do not exist.

A population with less spread would have a regression line that was closer to the  $V = \bar{V}$  line. There is no spread in the extreme case where all fruit are of the same volume. In this case the regression line is the same as the  $V = \bar{V}$  line. If the spread of the volume

is small enough, then a volume estimate using  $\hat{V} = \bar{V}$  would outperform an estimate using  $\hat{V} = S$  in terms of RMS error.

The purpose of this section is thus to highlight the fact that the RMS error associated with a given volume estimation method should be seen in the context of the distribution of volumes from which the RMS error was derived. Two volume estimation methods cannot be directly compared unless they are evaluated on data sets with the same distribution of volumes<sup>1</sup>. A volume estimation method for a particular type of fruit will not have one RMS error value associated with it, but would instead produce different RMS errors when applied to populations with different overall volume distributions. A good volume estimation method is one in which the spread of the ‘random variable’  $u$  is small. This would mean that the size and shape of the profile images are appropriately taken into account. The better the volume estimation method, the lower will be the effect of improving the volume estimate through knowledge of the volume distribution. One would thus expect a reasonably good estimation method to be relatively insensitive, in terms of a change in RMS error, to changes in the spread of the volume distribution of the population. A poor method would, by contrast, decrease in performance with increasing spread of volume distribution, as most of its information would come from the *a priori* knowledge of the volume distribution. In the extreme case, the  $\hat{V} = \bar{V}$  method makes no use of the shape or size of the profile images. This method would be expected to perform increasingly poorly, without bound, as the spread of the population distribution is increased.

## 2.4 An analytically derived volume estimate for a mathematically described object

In this section, the volume estimates, associated uncertainty and error estimates are derived for a simple mathematically described population of objects. The derivation

---

<sup>1</sup>The probable volume of a piece of fruit is a function both of the PDF of fruit volumes,  $V$ , and the PDF of  $u$ , neither of which is necessarily known (although they can be estimated from sample values). Nor is either PDF necessarily normally-distributed or necessarily independent of the other. This means that there is no simple, direct way of comparing volume estimation results for fruit data sets with different volume PDFs (such as dividing the RMS errors by the standard deviation of the volume PDFs). It is, however, enough for one to be aware that a direct comparison of RMS errors would not be meaningful.

is entirely theoretical, and makes the assumption that all measurements are perfect. This is done to demonstrate the inherent uncertainty on a volume estimate due to lack of information alone. The overall uncertainty can of course be expected to be larger in a real system with additional sources of error such as those due to imperfect measurement, quantisation, camera imperfections and so forth.

From this demonstration, it can be seen that the *a priori* knowledge of the distribution of the sizes and hence of the distribution of the volumes of the mathematical object (an ellipsoid) is not only important in estimating volume, but is fundamentally necessary in solving this problem.

The three-dimensional surface shape characteristics of a fruit can be parameterized to arbitrary accuracy by a vector  $\mathbf{x} = (x_1, x_2 \dots x_n)$ . Any piece of fruit from a population of fruit (or cultivar) could be described by a particular instance of  $\mathbf{x}$ . The parameterization could, for instance, take the form of a set of spherical harmonics, or it could simply be a set of co-ordinate triplets defining points on the fruit's surface in three-dimensional space. Figure 2.5(a) shows a computer-generated pear. It has been parameterized with a set of 64 co-ordinates in an  $8 \times 8$  mesh pattern as shown in Figure 2.5(b) and to a greater accuracy with 256 co-ordinates in a  $16 \times 16$  grid pattern in Figure 2.5(c). The value of  $n$ , the size of  $\mathbf{x}$ , for the parameterization shown in Figure 2.5(c) would thus be  $n = 16 \times 16 \times 3 = 768$ . Using this parameterization, the three-dimensional surface shape characteristics of any fruit from this cultivar may be described by a vector of 768 values.

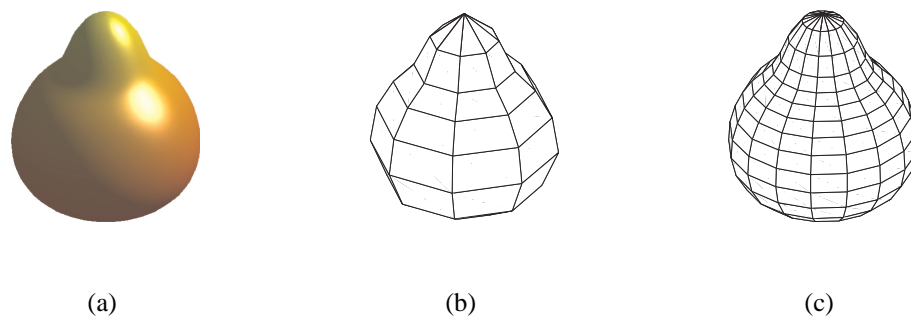


Figure 2.5: Parameterization of a fruit to arbitrary accuracy.

Each parameter of a fruit is determined probabilistically, remembering that for real fruit the probability functions associated with a population of fruit are hidden and can only be estimated. By considering a piece of fruit which is parameterized by a vector,

$\mathbf{x}$ , and which is randomly selected from a population, the inherent uncertainty associated with a volume estimate from a photograph of such a fruit can be demonstrated. To obtain the fruit profile boundary as if in a photograph, the piece of fruit is placed in a Cartesian co-ordinate system and is rotated to a random orientation. A ‘photograph’ is then taken of the fruit, with the camera looking down at the  $xy$  plane from  $z = \infty$ .

### 2.4.1 Estimating the volume and associated error of an ellipsoid

The estimation of volume and the calculation of the uncertainty on the volume estimate can become complex, even with the number of parameters,  $n$ , being low. For purposes of demonstration, consider a theoretical population of fruit each of which is an ellipsoid with two of its three semi-axes equal. Such an ellipsoid is a surface of revolution. It can be obtained by revolving the ellipse  $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1, z = 0$  around the  $x$ -axis. An example of this type of ellipsoid is shown in Figure 2.6 at arbitrary orientation. The ellipsoid is parameterized by only two values,  $a$  and  $b$ .

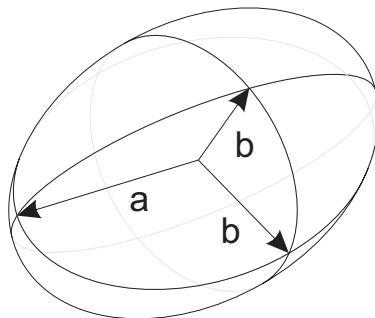


Figure 2.6: An ellipsoid with two of its semi-axes equal.

Figure 2.7 shows two PDFs which generate values  $a$  and  $b$  according to certain probabilities. In this case, for purposes of simplicity, the two probability functions are independent of one another.

Figure 2.8 shows a photograph of the ellipsoid taken at an arbitrary orientation with the camera looking down onto the  $xy$ -plane from infinity (there are no projection effects, the resulting ellipse is simply the ellipsoid projected onto the  $z = 0$  plane). Two parameters or features,  $r_1$  and  $r_2$ , can be extracted from the photograph. These two features completely describe the ellipse’s shape (yet describe nothing of its orientation).

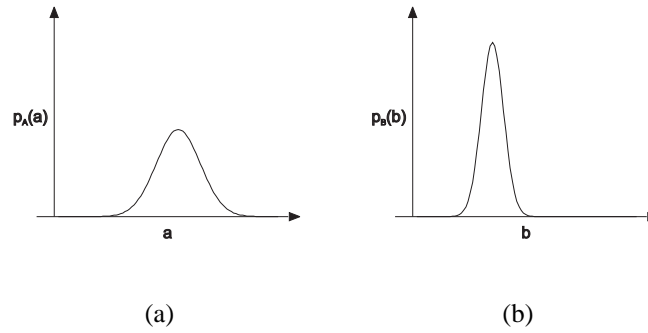


Figure 2.7: Probability density functions  $p_A(a)$  and  $p_B(b)$ .

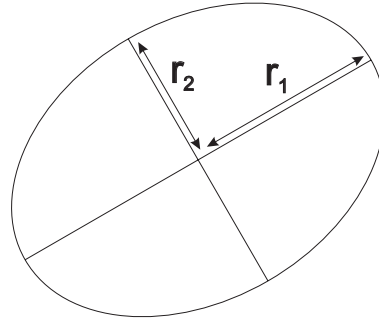


Figure 2.8: Photograph of the ellipse at an arbitrary angle.

The equation for the surface of the ellipsoid in Figure 2.6 is

$$\frac{x^2}{a^2} + \frac{y^2 + z^2}{b^2} = 1 \quad (2.11)$$

The equation of the ellipse in Figure 2.8, obtained by projecting the ellipsoid onto the  $xy$ -plane, can be found by setting  $z = 0$  in Equation (2.11):

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \quad (2.12)$$

This is only possible because the projection happens to be the same as a cross-section through the ellipse at  $z = 0$ , which is not in general the case at other orientations.

To rotate the ellipsoid to any other orientation, it can be rotated about the  $x$ -,  $y$ - and  $z$ -axes by arbitrary angles. The rotation of a single point anti-clockwise about the  $y$ -axis through an angle  $\theta$  is illustrated in Figure 2.9. Mathematically, this rotation is performed using the rotation matrix as shown below:

$$\begin{bmatrix} x_{new} \\ z_{new} \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x_{old} \\ z_{old} \end{bmatrix} \quad (2.13)$$

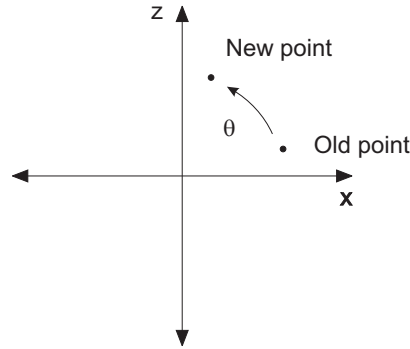


Figure 2.9: A point rotated anti-clockwise by  $\theta$  in the  $xz$ -plane.

Since the ellipsoid is radially symmetrical about the  $x$ -axis, rotation about the  $x$ -axis has no effect on the ellipsoid's apparent position or shape. The entire ellipsoid can be rotated about the  $y$ -axis by the angle,  $\theta$ , yielding a new equation for the ellipsoid

$$\frac{(x\cos(\theta) - z\sin(\theta))^2}{a^2} + \frac{y^2 + (x\sin(\theta) + z\cos(\theta))^2}{b^2} = 1 \quad (2.14)$$

Since there is no reference angle for the camera plane, which is parallel to the  $xy$ -plane, there is no need to rotate the ellipsoid about the  $z$ -axis: that is, the ellipse parameters  $r_1$  and  $r_2$  are unaffected by any rotation of the ellipsoid about the  $z$ -axis, so Equation (2.14) can be used to describe an arbitrarily-rotated ellipsoid.

The probability density function for the volume of the ellipsoid can be calculated, given  $r_1$  and  $r_2$ , which are the lengths as shown in Figure 2.8. The ellipse in the photograph could have come from a range of different ellipsoids, each with a corresponding probability. Any ellipsoid with a  $b$  value equal to one of the two  $r$  values in the photograph, and an  $a$  value such that the other  $r$  value lies between  $a$  and  $b$  could produce such an ellipse at some specific orientation. Each ellipsoid has an associated probability of occurrence, which can be calculated from the PDFs of the ellipsoid's parameters,  $a$  and  $b$ .

The volume,  $V$ , of the ellipsoid is given by the formula [34]

$$V = \frac{4\pi}{3}ab^2 \quad (2.15)$$

For a given volume, there are two cases which must be considered. Either  $r_1 = b$  or  $r_2 = b$ . These two cases give rise to two possible values of  $a$ ,  $a_1$  and  $a_2$ , and two

possible values of  $b$ ,  $b_1$  and  $b_2$ . These quantities are related to the observed lengths,  $r_1$  and  $r_2$  as follows:

$$a_1 = \frac{3V}{4\pi r_2^2} \quad (2.16)$$

$$b_1 = r_2 \quad (2.17)$$

$$a_2 = \frac{3V}{4\pi r_1^2} \quad (2.18)$$

$$b_2 = r_1 \quad (2.19)$$

Each case occurs with its associated PDFs,  $p_A(a)$  and  $p_B(b)$ , for  $a$  and  $b$ . The probability of the ellipsoid being of a specific volume given a photograph from which  $r_1$  and  $r_2$  are obtained is (after some derivation)

$$p_V(V | r_1, r_2) = \frac{\sum_{n=1}^2 p_A(a_n) p_B(b_n) U[(b_n - r_n)(b_n - a_n)] U[(r_n - a_n)(b_n - a_n)]}{\int_0^\infty \sum_{n=1}^2 p_A(a_n) p_B(b_n) U[(b_n - r_n)(b_n - a_n)] U[(r_n - a_n)(b_n - a_n)] dV} \quad (2.20)$$

where the unit step  $U(t)$  is defined by [24]

$$U(t) \equiv \begin{cases} 0 & t < 0 \\ \frac{1}{2} & t = 0 \\ 1 & t > 0 \end{cases} \quad (2.21)$$

The unit step functions in Equation (2.20) are used to remove cases which could not occur, because of the constraint that  $r$  must fall between  $a$  and  $b$ . The denominator ensures that the final volume PDF is of unit area. The summation signs are used to consider both of the two cases, where  $b = r_2$  and where  $b = r_1$ .

From this PDF the expected value of the volume and the uncertainty in terms of standard deviation can be calculated. The RMS error for the system over many volume estimations of many ellipsoids can also be derived.

The mean value of the PDF

$$E[V] = \int_0^\infty V p_V(V | r_1, r_2) dV \quad (2.22)$$

is the expected value of the volume for a given  $r_1$  and  $r_2$ .

The standard deviation of the PDF

$$\sigma_V = \sqrt{E[V^2] - E[V]^2} \quad (2.23)$$

gives a measure of the uncertainty on the volume estimate.

The root-mean-squared error (RMSE) for a theoretical system which estimates the ellipsoid volume from a photograph is

$$RMSE = \sqrt{\frac{\int_0^{2\pi} \int_0^{\infty} \int_0^{\infty} \left( \frac{4\pi}{3} ab^2 - E[V | r_1(a, b, \theta), r_2(a, b, \theta)] \right)^2 p_A(a) p_B(b) db da d\theta}{\int_0^{2\pi} \int_0^{\infty} \int_0^{\infty} p_A(a) p_B(b) db da d\theta}} \quad (2.24)$$

Equation (2.24) integrates the squared error of all possible ellipsoids in all possible orientations and multiplies in each case by the associated probability of occurrence. The denominator is again for purposes of normalisation.

To use Equation (2.24), the functions  $r_1(a, b, \theta)$  and  $r_2(a, b, \theta)$  must be derived. As can be seen in Figure 2.10, which shows the side view of an ellipsoid with the camera taking the profile photograph (as shown in Figure 2.8) looking down from on top, the value of  $r_1$  can be determined by setting  $y = 0$  and locating the turning point of the

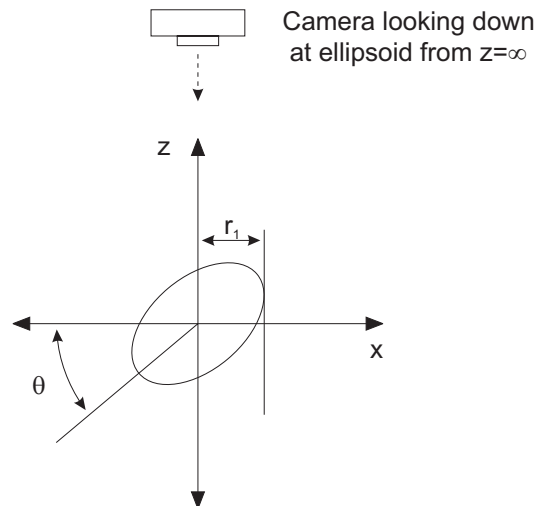


Figure 2.10: Side view of the ellipsoid.



ellipse in the  $xz$ -plane. The equation of this ellipse is

$$\frac{(x \cos(\theta) - z \sin(\theta))^2}{a^2} + \frac{(x \sin(\theta) + z \cos(\theta))^2}{b^2} = 1 \quad (2.25)$$

This can be rewritten as

$$\left( \frac{\cos^2(\theta)}{a^2} + \frac{\sin^2(\theta)}{b^2} \right) x^2 - \left( \frac{2 \cos(\theta) \sin(\theta)}{a^2} - \frac{2 \cos(\theta) \sin(\theta)}{b^2} \right) xz + \left( \frac{\sin^2(\theta)}{a^2} + \frac{\cos^2(\theta)}{b^2} \right) z^2 = 1 \quad (2.26)$$

and, by introducing the quantities

$$A = \frac{\cos^2(\theta)}{a^2} + \frac{\sin^2(\theta)}{b^2} \quad (2.27)$$

$$B = -\frac{2 \cos(\theta) \sin(\theta)}{a^2} + \frac{2 \cos(\theta) \sin(\theta)}{b^2} \quad (2.28)$$

$$C = \frac{\sin^2(\theta)}{a^2} + \frac{\cos^2(\theta)}{b^2} \quad (2.29)$$

the equation becomes

$$Ax^2 + Bxz + Cz^2 = 1 \quad (2.30)$$

Now taking the derivative of this equation with respect to  $z$ , and writing  $x' = \frac{dx}{dz}$  for convenience:

$$2Ax x' + Bx + Bx' z + 2Cz = 0 \quad (2.31)$$

$$\Rightarrow x' = -\frac{Bx + 2Cz}{2Ax + Bz} \quad (2.32)$$

At the turning-point  $x' = 0$ , so

$$-\frac{Bx + 2Cz}{2Ax + Bz} = 0 \quad (2.33)$$

$$\Rightarrow z = \frac{-Bx}{2C} \quad (2.34)$$

Substituting this into Equation (2.30) gives

$$\left( A - \frac{B^2}{2C} + \frac{B^2}{4C} \right) x^2 = 1 \quad (2.35)$$

$$\Rightarrow r_1 = x = \sqrt{\frac{4C}{4AC - B^2}} \quad (2.36)$$

The value of  $r_2$  is simply the radius of the circular cross section of the ellipsoid,  $r_2 = b$ .

### 2.4.2 An example of ellipsoid volume estimation

The use of the equations described above may now be demonstrated through a specific example.

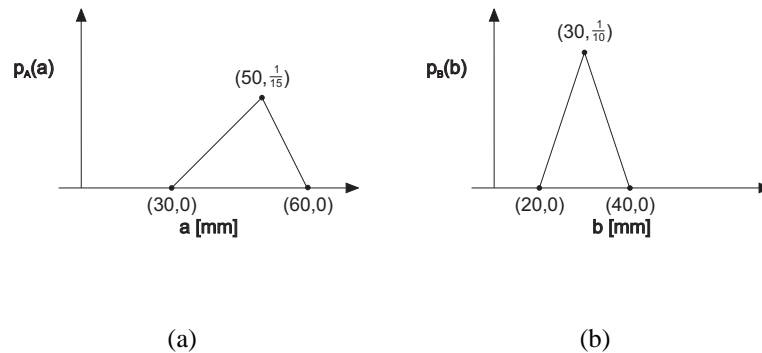


Figure 2.11: Example of probability density functions  $p_A(a)$  and  $p_B(b)$ .

Figure 2.11 shows an example of a pair of PDFs which describe the characteristics of a family of ellipsoids.

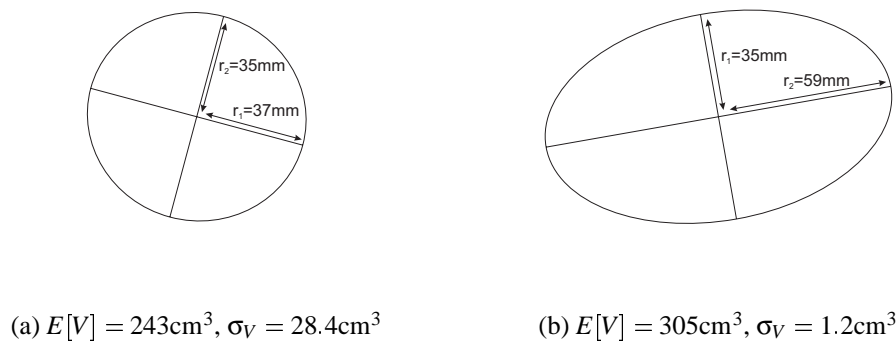


Figure 2.12: Two examples of ellipses with volume estimates and associated uncertainties, given the PDFs in Figure 2.11.

Figure 2.12(a) shows an ellipse (representing a photograph of an ellipsoid from the family described by the PDFs in Figure 2.11). Its corresponding expected volume of  $243\text{cm}^3$  was calculated using Equation (2.22) after first using Equation (2.20) to calculate the PDF shown in Figure 2.13(a). The associated standard deviation of  $28.4\text{cm}^3$  was calculated using Equation (2.23). Note how the PDF is the sum of two functions,

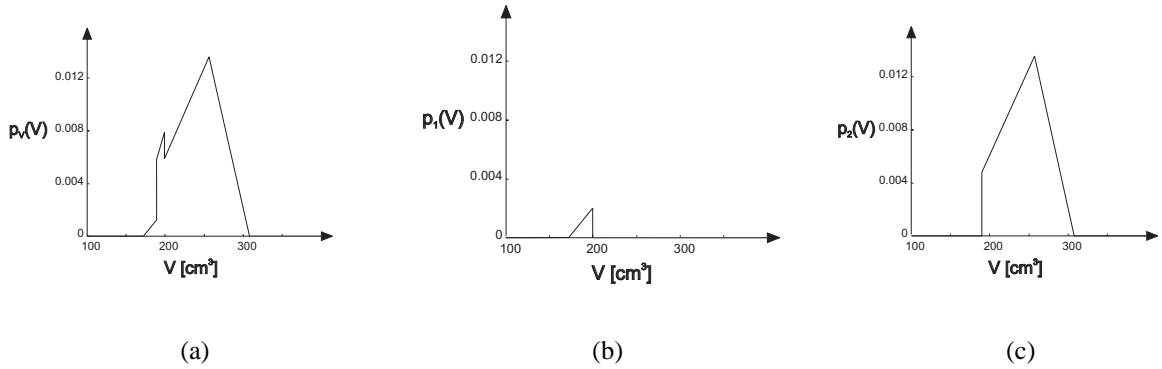


Figure 2.13: (a) is the PDF of volume for the ellipse shown in Figure 2.12, and is the sum of (b) and (c) which represent the cases  $r_2 = b$  and  $r_1 = b$  respectively.

the first (shown in Figure 2.13(b)) assuming that the ellipsoid's axis of symmetry corresponds to  $r_1$ , and the second (shown in Figure 2.13(c)) assuming that it corresponds to  $r_2$ . These two functions are summed in the numerator in Equation (2.20).

Figure 2.12(b) shows an ellipse with a corresponding expected volume of  $305\text{cm}^3$  and associated standard deviation of  $1.2\text{cm}^3$ . The uncertainty is much lower than in the first example, since there is a much smaller range of ellipsoids that could have produced the projected ellipse.

The RMS error for a system which estimates volumes of the ellipsoid fruit in this example was calculated to be  $22.8\text{cm}^3$  using Equation (2.24). In order for the error on a volume estimate to be calculated, the actual volume of the ellipsoid must be known, but to measure the uncertainty by means of the standard deviation, only the ellipse parameters  $r_1$  and  $r_2$  need be known. For instance, an ellipsoid with  $a = 56.1\text{mm}$  and  $b = 34.3\text{mm}$ , oriented at an angle of  $\theta = 322.5^\circ$ , will produce an ellipse with parameters  $r_1 = 49.2\text{mm}$  and  $r_2 = 34.3\text{mm}$ . The volume of this ellipsoid is  $276.4\text{cm}^3$  and the expected volume calculated by looking at the ellipse is  $260.1\text{cm}^3$ , resulting in an absolute error of  $16.3\text{cm}^3$ . However, the uncertainty measure on the volume estimate calculated with no knowledge of the actual ellipsoid is  $12.5\text{cm}^3$  (using Equation (2.23)).

A computer was used to evaluate the integrals in the various equations, but there is no reason to assume that they could not be solved entirely analytically. A computer simulation which generated random ellipsoids using the given PDFs confirmed the RMS error result.

This discussion shows that even if a fruit's generating parameters and PDFs are known (which is not the case in real systems), there is an intrinsic error associated with the volume estimate due to the lack of given information. This is because the volume can only be determined probabilistically. In a real system, where the parameterization and associated probabilities are hidden, further errors are accumulated, as these parameters are implicitly, yet imperfectly, derived by the volume estimation system. The ellipsoid example thus helps to show that since a fruit profile image could have been produced by fruit with a range of volumes, the statistically-most-probable volume must be chosen as the volume estimate.

It can also be seen from this ellipsoid demonstration that the knowledge of the distribution of shape features (and thus of volumes, which are functions of shape features) is of great importance, if a statistical approach is to be taken in the volume estimation problem. This information is implicitly incorporated into a real volume estimation process during the training stage, in which the system is exposed to a number of sample features which must typify the population from which they were drawn. This is usually achieved by selecting the samples randomly from the population.

# Chapter 3

## Image Processing Methods

This chapter describes the methods and terminology of some of the tools that were used for digital image processing and analysis in this project. The digital image processing is required firstly to preprocess the data set of fruit images into a format from which features can be extracted, and secondly to extract and measure these features. Eventually, the features are to be combined to form a volume estimate of the fruit from whose image they were extracted and measured.

### 3.1 Introduction

The digital fruit images used in this project, and those used in packing houses for colour sorting and blemish detection, are obtained using a charge coupled device (CCD) camera. The CCD is a two-dimensional array of light-sensing elements on a chip. Sensor elements are located spatially to correspond with the pixels in an image. These sensors form the red, green and blue (RGB) values for each pixel in the output digital image when exposed to light. An RGB image is thus comprised of three matrices of intensity bands, one for each of R, G and B. Each of the bands on its own forms a *grey-scale* image.

By convention the intensity values usually range from 0 (dark) to 255 (light), but they are also often scaled to run from zero through to one, so that they can be interpreted as the proportion of full intensity. A *binary image* is an image which contains only ‘ones’ and ‘zeros’ and may be obtained by thresholding a grey-scale image (setting all pixels

above a certain value to zero and all others to one). In some conventions, the ‘ones’ in a binary image represent foreground pixels and the ‘zeros’ represent background pixels.

## 3.2 Region labelling

Region labelling is the division of an image into *objects* or sets of connected pixels. Generally the ‘ones’ in a binary image form the objects, and the ‘zeros’ form the background. Two object pixels are directly connected to one another if they are neighbours of one another. Pixel neighbourhoods can be defined in terms of 4-connectivity, in which the two pixels share a common side, or of 8-connectivity in which the two pixels share either a common side or a common corner (see Figure 3.1).

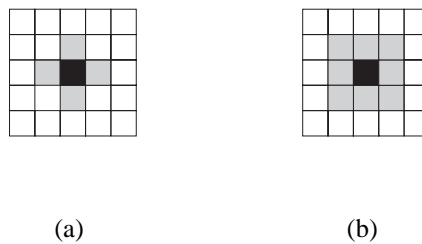


Figure 3.1: Black object pixel with (a) 4-connected and (b) 8-connected neighbours shown in grey.

The region labelling process begins by scanning the binary image for the first instance of an object pixel. A search algorithm is then implemented to locate all pixels connected to the initial pixel, so that the connected pixels can be assigned to the same region, namely region one. After this, region one may be removed from the binary image (by setting each of its pixels to the value of the background) and the process of locating another object pixel and its connected pixels may be repeated. The process continues until all object pixels have been labelled. Figure 3.2 shows an example of a binary image with three labelled objects if connectivity is defined as 4-connectivity or two labelled objects if connectivity is defined as 8-connectivity. In this project connectivity, for the purposes of region labelling, will be defined as 4-connectivity.

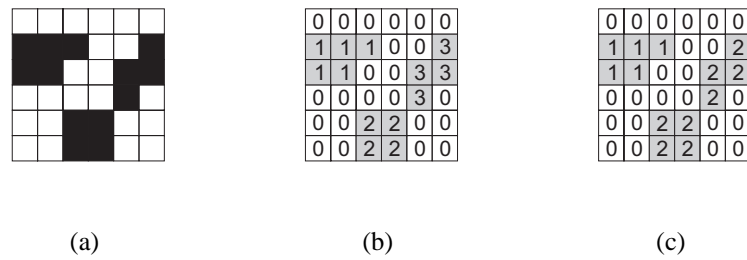


Figure 3.2: (a) Binary image, with (b) three 4-connected objects, or (c) two 8-connected objects.

### 3.3 Binary morphology

Binary morphology [40, 32, 16] is a non-linear branch of image processing that is used to alter the geometrical structure of a binary image. The two fundamental processes of morphology are *erosion* and *dilation*. These processes are used for the alteration of an image with another image called a structuring element. The structuring element is usually smaller than the image. An example of erosion and of dilation is shown in Figure 3.3.

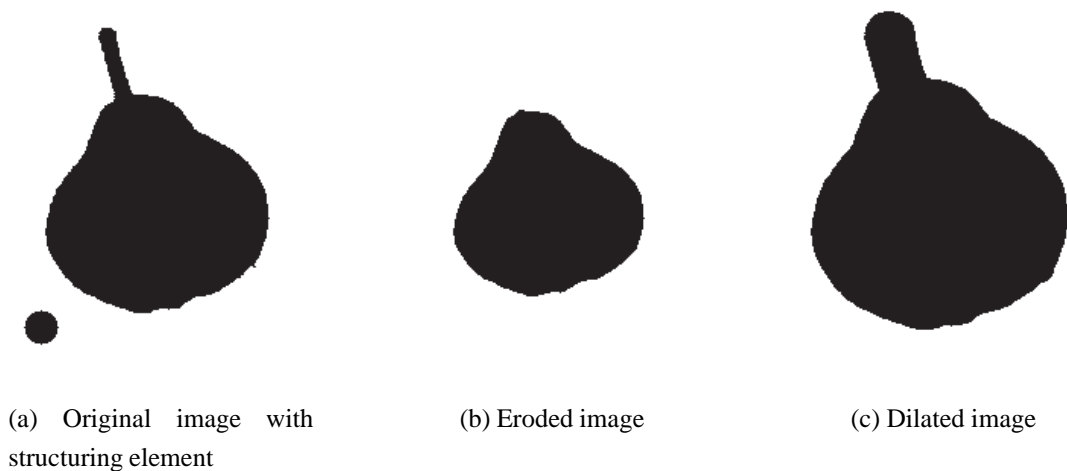


Figure 3.3: Example of binary erosion and dilation of an image.

Erosion uses a structuring element to reduce the number of ‘ones’ or non-zero pixels in an image. The origin of the structuring element is overlaid over each pixel in the original image, and only if every non-zero value of the structuring element is contained in the original image should the corresponding pixel in the resultant image be set to

‘one’. The erosion of an image,  $A$ , by a structuring element,  $B$ , is written  $A \ominus B$ , where  $A$  and  $B$  are treated as the sets of non-zero pixels in the images. Figure 3.4(a) shows an example of a binary image,  $A$ , with ‘ones’ shown in black. The co-ordinate locations of the black pixels form the set  $A$  which completely describes the binary object:

$$A = \{(1, 1), (2, 1), (2, 2), (2, 3), (3, 1), (3, 3), (3, 4), (4, 1), (4, 3), (4, 4)\} \quad (3.1)$$

Similarly, for  $B$ , in Figure 3.4(b),

$$B = \{(0, 0), (0, 1), (1, 0)\} \quad (3.2)$$

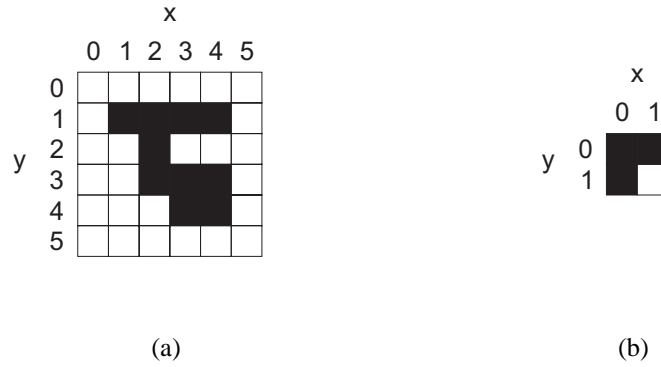


Figure 3.4: An example of (a) a  $6 \times 6$  binary image,  $A$ , and (b) a  $2 \times 2$  structuring element,  $B$ .

Erosion is mathematically defined as follows [40]:

$$A \ominus B = \bigcap_{b \in -B} \{t \in \mathbb{I}^2 : t = a + b, a \in A\} \quad (3.3)$$

where  $\mathbb{I}^2$  is the two-dimensional space of an image. Thus, erosion is the intersection of the translations of  $A$  by  $-B$ . In this case

$$-B = \{(0, 0), (0, -1), (-1, 0)\} \quad (3.4)$$

Since in this case  $-B$  has 3 elements, there are 3 translations of  $A$ :

$$\begin{aligned} A_{(0,0)} &= \{(1, 1), (2, 1), (2, 2), (2, 3), (3, 1), (3, 3), (3, 4), (4, 1), (4, 3), (4, 4)\} \\ A_{(0,-1)} &= \{(1, 0), (2, 0), (2, 1), (2, 2), (3, 0), (3, 2), (3, 3), (4, 0), (4, 2), (4, 3)\} \\ A_{(-1,0)} &= \{(0, 1), (1, 1), (1, 2), (1, 3), (2, 1), (2, 3), (2, 4), (3, 1), (3, 3), (3, 4)\} \end{aligned}$$



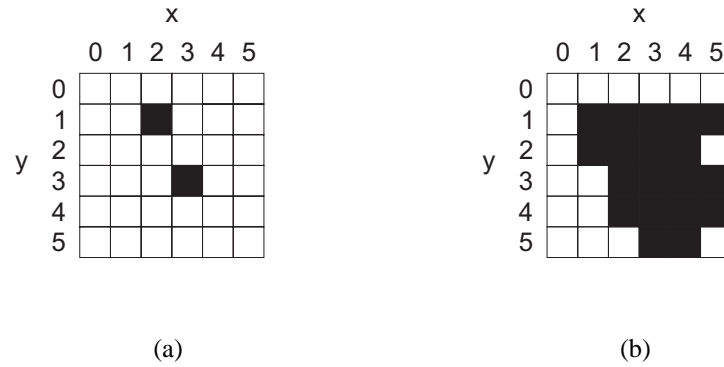


Figure 3.5: The (a) erosion and (b) dilation of the binary image shown in Figure 3.4(a) using the structuring element shown in Figure 3.4(b).

The intersection of  $A_{(0,0)}$ ,  $A_{(0,-1)}$  and  $A_{(-1,0)}$  is the set of the pixels in the eroded object as shown in Figure 3.5(a). This intersection consists of two pixel co-ordinates: (2, 1) and (3, 3).

Dilation uses the structuring element to increase the number of non-zero pixels in an image. The origin of the structuring element is overlaid over each pixel in the image in turn. If the image pixel is non-zero, then each pixel is added to the resultant image. The dilation of an image,  $A$ , by a structuring element,  $B$ , is written  $A \oplus B$  and is mathematically defined as follows [40]:

$$A \oplus B = \bigcup_{b \in B} \{t \in \mathbb{I}^2 : t = a + b, a \in A\} \quad (3.5)$$

Thus, dilation is the union of the translations of  $A$  by  $B$ . In the example case,  $B$  has 3 elements, so there are 3 translations of  $A$ :

$$A_{(0,0)} = \{(1, 1), (2, 1), (2, 2), (2, 3), (3, 1), (3, 3), (3, 4), (4, 1), (4, 3), (4, 4)\}$$

$$A_{(0,1)} = \{(1, 2), (2, 2), (2, 3), (2, 4), (3, 2), (3, 4), (3, 5), (4, 2), (4, 4), (4, 5)\}$$

$$A_{(1,0)} = \{(2, 1), (3, 1), (3, 2), (3, 3), (4, 1), (4, 3), (4, 4), (5, 1), (5, 3), (5, 4)\}$$

The union of the sets  $A_{(0,0)}$ ,  $A_{(0,1)}$  and  $A_{(1,0)}$  gives the co-ordinates of the pixels in the dilated object as shown in Figure 3.5(b).

Since the structuring element is commonly disk-shaped, it can often help to visualise dilation as the result of rolling the disk around the outer boundary of the object, and then forming the resultant object by adding to the original object all the pixels through

which the disk passes. Likewise, erosion can be visualised as the result of rolling the disk around the inner boundary of the object, and forming the resultant object by removing from the original object every pixel through which the disk passes. (The example shown in Figure 3.3 is of erosion and dilation by a disk-shaped structuring element.)

The morphological operations of opening and closing are defined [40] as combinations of erosions and dilations of a binary image. The opening of an image,  $A$ , by a structuring element,  $B$ , is written  $A \circ B$ , and is defined as

$$A \circ B = (A \ominus B) \oplus B \quad (3.6)$$

Opening a binary image smooths the contours of objects within it, eliminating small islands and sharp spikes.

The closing of an image,  $A$ , by a structuring element,  $B$ , is written  $A \bullet B$ , and is defined as

$$A \bullet B = (A \oplus B) \ominus B \quad (3.7)$$

Closing a binary image smooths the contours of the objects within it, eliminating small holes and fusing narrow breaks.

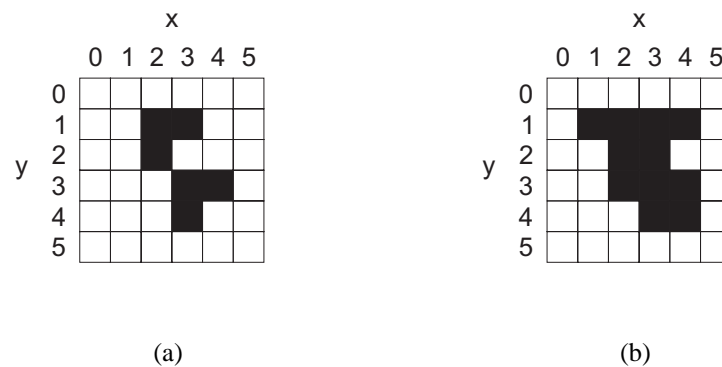


Figure 3.6: The (a) opening and (b) closing of the binary image shown in Figure 3.4(a) using the structuring element shown in Figure 3.4(b).

Following the example of Figure 3.4, the results of opening and closing  $A$  with the structuring element  $B$  are shown in Figure 3.6. Another example, which presents the opening and closing of a fruit image using a disk structuring element, is shown in Figure 3.7. Note how opening the image removes the stalk of the fruit (see Figures 3.7(a) and (b)).

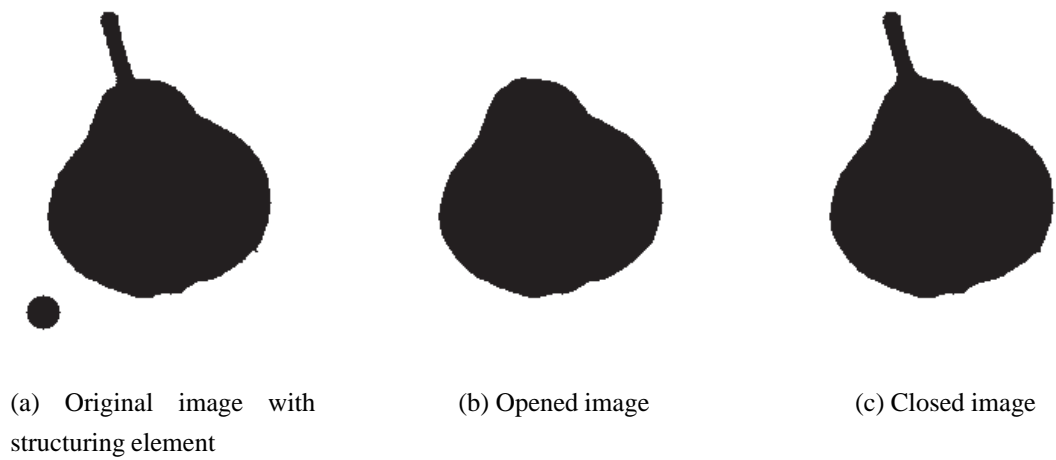


Figure 3.7: Example of binary opening and closing of an image.

Opening and closing are both idempotent operations [16]; that is, after the initial opening or closing, further opening or closing with the same structuring element has no effect:

$$A \bullet B = (A \bullet B) \bullet B \quad (3.8)$$

$$A \circ B = (A \circ B) \circ B \quad (3.9)$$

### 3.4 Boundary sampling

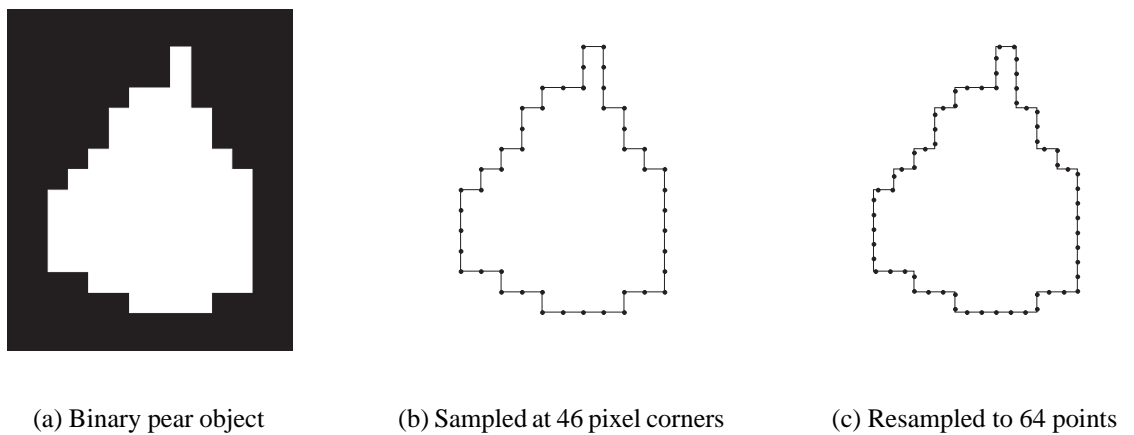


Figure 3.8: Boundary sampling.

Figure 3.8(a) shows a binary image containing a pear object. It was chosen to be low resolution to demonstrate the sampling algorithm that is to be discussed in this section. It is often far more efficient in terms of speed and storage space to extract features from a binary object by keeping a list of boundary co-ordinates than it would be by operating directly on the binary image itself. The boundary of a binary object can be described by the list of co-ordinates of the pixel corners that lie on the line separating background and foreground (see Figure 3.8(b)). Note that pixel corner co-ordinates are used, rather than the pixels themselves, as this describes the true boundary between foreground and background. Each co-ordinate is one unit distance apart from its predecessor in the list.

The list of co-ordinates is formed by first locating an arbitrary starting position known to be on the object's boundary. This can be done by selecting the upper-right corner of the object pixel with the maximum  $y$  value, for instance. If there is more than one such pixel, then, of those pixels, the one with the maximum  $x$  value may be used. After adding the first corner's co-ordinates to the list, the boundary is traversed by moving on to the next pixel corner position that lies on the boundary and adding its co-ordinates to the list, and so progressing through all the corner points until the starting point is returned to.

Often it is desirable to use a fixed number of sample points to describe an object. For instance, for a group of objects it might be useful to have each object's boundary stored in an array of, say, 64 points (see Figure 3.8(c)), instead of each object having a different number of samples to describe its boundary. Also, when implementing the fast Fourier transform (FFT) it is useful to have a number of points which is a power of two (e.g.  $2^6 = 64$ ). It is possible to resample the boundary using the existing list of boundary co-ordinates, whilst keeping the new sample points equidistant along the boundary. The length of the boundary is known from the existing list of co-ordinates, and so, by dividing this length by the desired number of new samples, the distance that must separate the new samples can be determined. A new co-ordinate list can then be compiled by traversing the boundary as marked by the old co-ordinate list, and by storing the co-ordinates of the points on the boundary that are the appropriate length apart from one another (see Figure 3.8(c)).

### 3.5 Fourier descriptors

The Fourier descriptors of an object boundary provide a means of describing the object through the use of the one-dimensional discrete Fourier transform. The discrete Fourier transform is defined for complex-valued, periodic functions, and this permits two-dimensional, closed boundaries to be treated in an elegant mathematical fashion.

The  $x$  and  $y$  co-ordinates  $(x, y)$  that lie on the boundary of an object can be represented as complex numbers,  $x + jy$ . By periodically traversing the boundary, for example over time  $t$ , the function

$$f(t) = x(t) + jy(t) \quad (3.10)$$

can be formed. This function is periodic with a period of  $T$  and is defined over the range  $-\infty < t < +\infty$ . An example of a portion of such a function is shown in Figure 3.9.

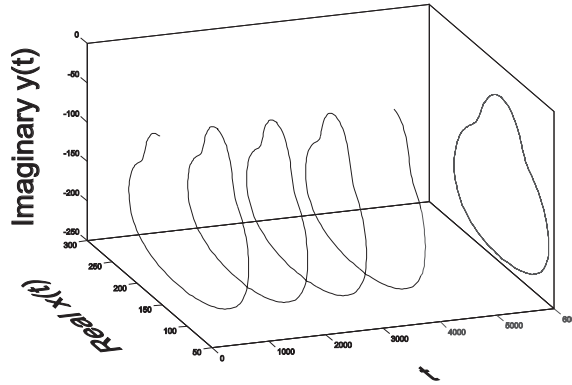


Figure 3.9: Portion of  $f(t)$  with the boundary shown projected on the side of the box.

The Fourier transform of  $f(t)$  is defined as

$$F\{f(t)\} = F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t} dt \quad (3.11)$$

Since  $f(t)$  is periodic,  $F(\omega)$  is discrete [24][25].  $F(\omega)$  is zero for all values of  $\omega$  except multiples of  $\frac{2\pi}{T}$ ; and, at multiples of  $\frac{2\pi}{T}$ ,  $F(\omega)$  is a weighted impulse that is a multiple of the Dirac delta,  $\delta(t)$ , which has the following general properties:

$$\delta(t) = 0 \quad \text{for } t \neq 0, \quad \text{and} \quad \int_{-\infty}^{\infty} \delta(t) dt = 1 \quad (3.12)$$

In particular, a Dirac delta function  $c\delta(t - k)$ , where  $c$  and  $k$  are constants, has weight, or area,  $c$  at the point  $t = k$ .

The function  $f(t)$  can be sampled  $N$  times per period with a train of equally spaced unit impulses, known as a Dirac comb

$$\delta_T(t) = \sum_{n=-\infty}^{\infty} \delta(t - n) \quad (3.13)$$

giving a sampling period of  $\frac{T}{N}$  (since there are  $N$  samples per period of length  $T$ ). Every impulse of the Dirac comb samples  $f(t)$  to form

$$f_s(t) = f(t)\delta_T(t) = f(t) \sum_{n=-\infty}^{\infty} \delta(t - n) \quad (3.14)$$

$f_s(t)$  is periodic and discrete, and so  $F_s(\omega)$  is also both periodic and discrete with  $N$  impulses repeated periodically. The weights of these  $N$  impulses are known as the *Fourier descriptors* of the boundary  $f(t)$ . They are also the values of the Discrete Fourier Transform (DFT) of the  $N$  boundary points. The DFT of an  $N$ -element, one-dimensional function  $g(r)$ , sampled where  $r = 0, 1, 2, \dots, N - 1$  is

$$G(u) = \sum_{r=0}^{N-1} g(r)e^{-\frac{j2\pi ur}{N}} \quad \text{for } u = -\frac{N}{2}, \dots, -1, 0, 1, \dots, \frac{N}{2} - 1 \quad (3.15)$$

The  $N$  values of  $G(u)$  are the Fourier descriptors obtained from the sample values of  $g(r)$ . The values of  $g$  correspond to the weights of the impulses of  $f_s$ . Likewise, the values of  $G$  correspond to the weights of the impulses of  $F$ .

If the original boundary function  $f(t)$  is strictly band-limited, then it can be recovered from  $f_s(t)$  provided that  $N$  is large enough to satisfy the Nyquist sampling criterion [25]. In practice, the boundary is unlikely to be strictly band-limited, but, since the boundary function is a finite power signal,  $F(\omega)$  decays to zero at slowest like  $\frac{1}{\omega}$  [24]. Figure 3.10(a) shows an example of a boundary described by  $f(t)$ , with the magnitude of the corresponding  $F(\omega)$  shown in Figure 3.10(b). Note how quickly  $F(\omega)$  dies away, implying that most of the information about the shape lies in the lower-order Fourier descriptors, and that the effects of aliasing at a high sampling rate (say  $N = 32$  samples or more in this case) are negligible. The effects of aliasing in recovering  $f(t)$  from  $f_s(t)$  can be made acceptable by making  $N$  sufficiently large.

The inverse Fourier transform

$$F^{-1}\{F(\omega)\} = f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega)e^{j\omega t} d\omega \quad (3.16)$$

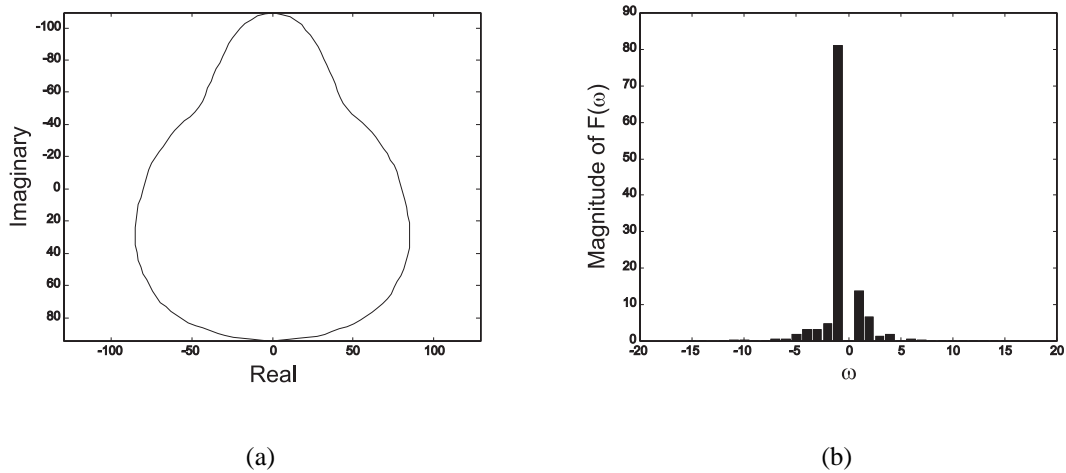


Figure 3.10: An example of (a)  $f(t)$ , and (b) magnitude of corresponding  $F(\omega)$ .

can be used to reconstruct the original object boundary. The integral becomes a summation of Dirac delta weights multiplied by complex sinusoids. The inverse Fourier transform (as opposed to the inverse DFT) is required, so that the points *in between* samples will be defined in the spatial domain.

A continuous boundary can be reconstructed from the windowed discrete frequency domain function,  $F_s(\omega)\text{Rect}\left(\frac{N\omega}{2\pi T}\right)^1$ . For the boundary to be continuous, the corresponding  $F(\omega)$  must be non-periodic. This function,

$$F(\omega) \approx F_s(\omega)\text{Rect}\left(\frac{N\omega}{2\pi T}\right) \tag{3.17}$$

is formed by retaining only the portion of  $F_s(\omega)$  centred around zero, to form  $F(\omega)$  when  $f(t)$  is band-limited and adequately sampled and to form an approximation to  $F(\omega)$  in the non-band-limited case.

Since  $f(t)$  is periodic, when adequately sampled to fulfil the Nyquist sampling criterion, the following equation holds

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega)e^{j\omega t} d\omega = \frac{1}{N} \sum_{u=-\frac{N}{2}}^{\frac{N}{2}-1} G(u)e^{\frac{j2\pi ut}{N}} \tag{3.18}$$

---

<sup>1</sup> $\text{Rect}\left(\frac{\omega}{\Omega}\right) \equiv \begin{cases} 1 & |\omega| < \frac{\Omega}{2} \\ \frac{1}{2} & |\omega| = \frac{\Omega}{2} \\ 0 & |\omega| > \frac{\Omega}{2} \end{cases}$

This is because the integration of Dirac deltas becomes a summation of their areas. It is thus possible, in the adequately sampled band-limited case, to derive  $f$  for *any* value of  $t$  by using the Fourier descriptors.

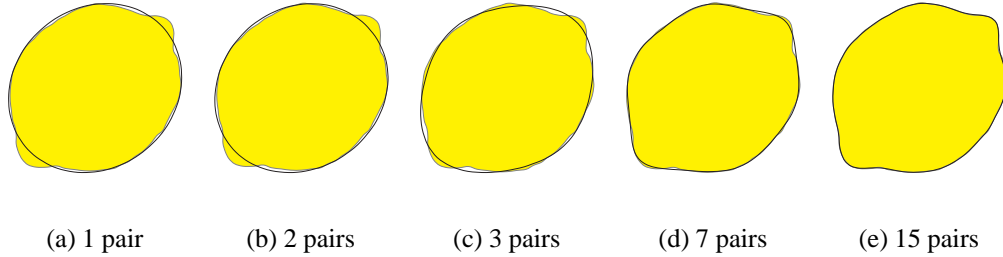


Figure 3.11: Reconstruction of a lemon boundary from Fourier descriptor pairs.

Figure 3.11 shows reconstructed lemon boundaries using various numbers of Fourier descriptor pairs with  $N = 256$  samples. The profile of the original object is shown in colour in each case for purposes of comparison. It can be seen how the low-order Fourier descriptors contain information about the gross shape of the object, and the higher-order descriptors account for the final details.

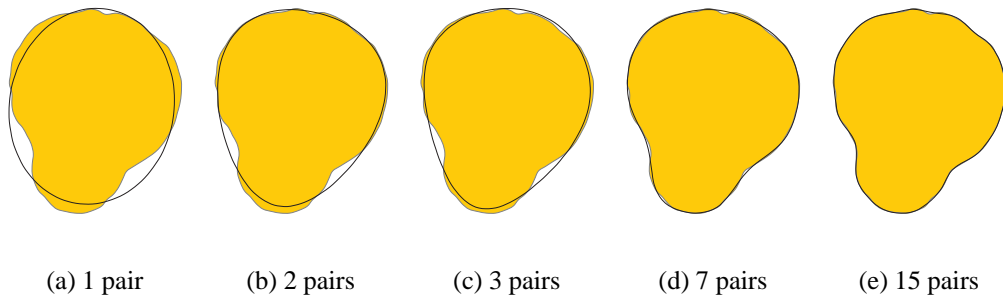


Figure 3.12: Reconstruction of a pear boundary from Fourier descriptor pairs.

Figure 3.12 shows a similar set of images formed from a reconstructed pear profile shown in colour. The first pair of Fourier descriptors describe the ellipse

$$f(r) = G_1[\cos(\frac{2\pi r}{N}) + j \sin(\frac{2\pi r}{N})] + G_{-1}[\cos(\frac{2\pi r}{N}) - j \sin(\frac{2\pi r}{N})] + G_0 \quad (3.19)$$

As  $r$  increases, the first term in Equation (3.19) can be visualised as a vector describing a circle in an anti-clockwise direction about the origin. The radius of the circle is determined by the magnitude of  $G_1$  and the starting angle is determined by the angle of  $G_1(0)$ . The second term of the equation is similarly described by  $G_{-1}$ , and would



rotate clockwise with increasing  $r$ . The sum of all three terms describes an ellipse with a centroid offset from the origin by  $G_0$ . This can be seen in the example shown in Figure 3.13. More complex shapes can be visualised in a similar manner. The vector  $G_2$  would rotate at twice the frequency of  $G_1$ .

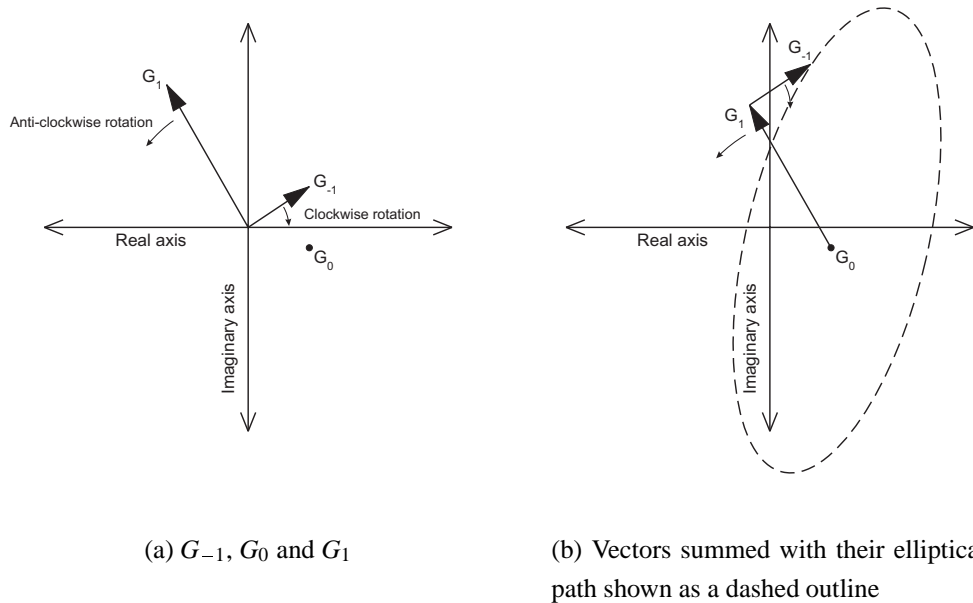


Figure 3.13: Formation of an ellipse from the first Fourier descriptor pair.

The Fourier descriptors have many useful properties. In practice they are calculated by using the FFT implementation of the DFT on an array of co-ordinates in complex number form. The DC value,  $G_0$ , is the centroid of the object. By changing  $G_0$  and then taking the inverse FFT of  $G$ , the object can quickly and easily be translated.

Since the Fourier transform is linear, multiplication by a constant in the frequency domain will increase the area of the object by the square of the constant factor.

The magnitudes of all of the Fourier descriptors except  $G_0$  are invariant to rotations and translations of the object. All rotation information is in the phase of the Fourier descriptors. Rotation of the object about the origin by an anti-clockwise angle of  $\theta$  can therefore be implemented by multiplying every Fourier descriptor by  $e^{-j\theta}$ .

The magnitudes of the lower-order Fourier descriptors are often used for describing the shape of objects. Lower-order Fourier descriptors are useful because they can parameterize the object to good accuracy within a small vector, and because they are invariant to rotations and translations of the object.

### 3.6 The Hotelling transform

The *Hotelling* or *Karhunen-Loève transform* [40] is used to calculate the angle that the major axis of an object within an image makes with a co-ordinate axis such as the  $x$ -axis. This information is often used to align the object, so that its major axis is parallel, for example, to the  $x$ -axis (see Figure 3.14).

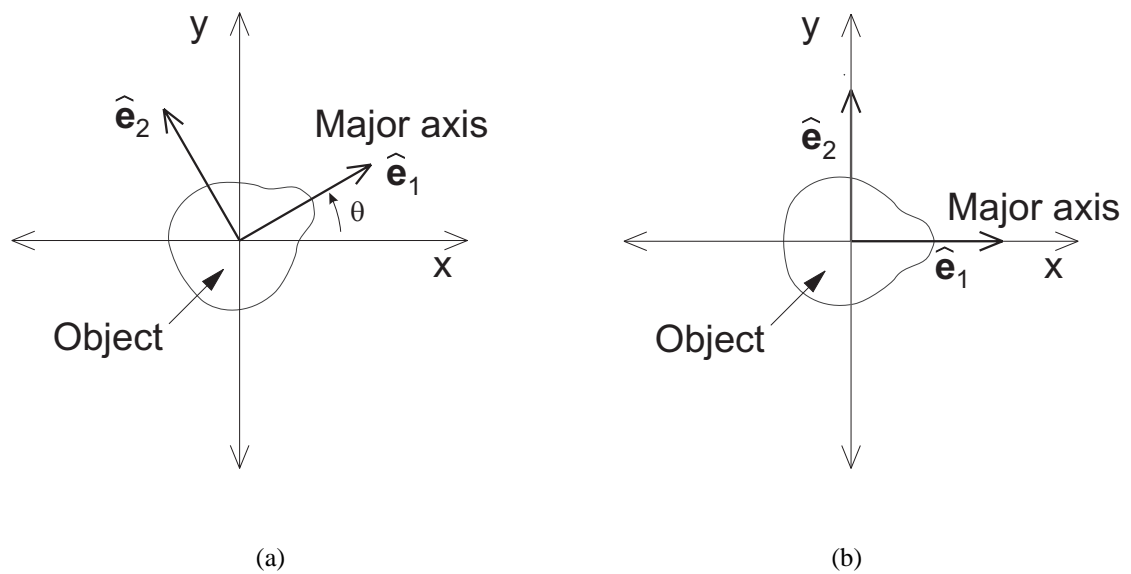


Figure 3.14: Rotation using the Hotelling transform: (a) the original object orientation; (b) after rotation through an angle of  $-\theta$ .

The Hotelling transform is based upon the eigenvalue decomposition of the covariance matrix of the pixel co-ordinates  $(x_i, y_i)$  of the object to be aligned.

This covariance matrix is written [40]

$$\mathbf{C} = \begin{bmatrix} C_1 & C_2 \\ C_3 & C_4 \end{bmatrix} \quad (3.20)$$

where

$$C_1 = \frac{1}{N} \sum_{i=1}^N x_i^2 - \bar{x}^2 \quad (3.21)$$

$$C_2 = \frac{1}{N} \sum_{i=1}^N x_i y_i - \bar{x} \bar{y} \quad (3.22)$$

$$C_3 = \frac{1}{N} \sum_{i=1}^N x_i y_i - \bar{x} \bar{y} \quad (3.23)$$

$$C_4 = \frac{1}{N} \sum_{i=1}^N y_i^2 - \bar{y}^2 \quad (3.24)$$

$C_1$  is the variance of the  $x$  co-ordinates and  $C_4$  is the variance of the  $y$  co-ordinates.  $C_2$  and  $C_3$  are the covariances, which are always equal to one another in this context. For an object that has its major axis aligned with the  $x$ -axis,  $C_2$  and  $C_3$  are zero.

The eigenvector corresponding to the larger eigenvalue of  $\mathbf{C}$  gives the orientation of the object's major axis with respect to the  $x$ -axis. The larger eigenvalue is given by

$$\lambda_1 = \frac{(C_1 + C_4) + \sqrt{(C_1 + C_4)^2 - 4(C_1 C_4 - C_2 C_3)}}{2} \quad (3.25)$$

and its corresponding eigenvector is therefore

$$\mathbf{e}_1 = \begin{bmatrix} 1 \\ \frac{\lambda_1 - C_1}{C_2} \end{bmatrix} \quad (3.26)$$

The angle that the major axis of the object makes with the  $x$ -axis is

$$\theta = \arctan \left( \frac{\lambda_1 - C_1}{C_2} \right) \quad (3.27)$$

An object can have its major axis aligned with the  $x$ -axis by rotating the object through the angle  $-\theta$ . This rotation can be carried out in a number of manners. For instance, the equation

$$\begin{bmatrix} x_{new} \\ y_{new} \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x_{old} \\ y_{old} \end{bmatrix} \quad (3.28)$$

will rotate an object, to align it with the  $x$ -axis, using the origin of the co-ordinate system as the centre of rotation.

Figure 3.15 shows an example of using the Hotelling transform to align an object in an image. The binary object in Figure 3.15(b), formed from the original image in Figure 3.15(a), was used to determine that the major axis of the pear was at an angle

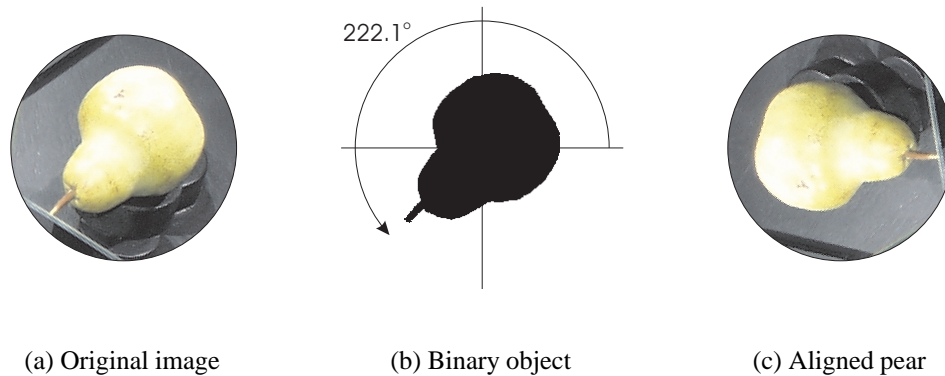


Figure 3.15: Using the Hotelling transform to align a pear with the  $x$ -axis.

of  $222.1^\circ$  with respect to the  $x$ -axis. Figure 3.15(c) shows the image with the object rotated so that it is aligned with the  $x$ -axis. With the object in such a position, its length (for instance) can easily be measured.

If the object is a boundary defined by a list of co-ordinates in complex number form, then the object can quickly be rotated by calculating the DFT of the points, multiplying the resultant Fourier descriptors by  $e^{j\theta}$  and then calculating the inverse transform.

### 3.7 Convex hull

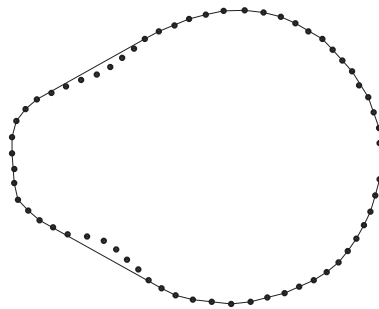


Figure 3.16: An example of a set of boundary points with the convex hull shown.

The convex hull of a set of co-ordinate points is analogous to the path of a rubber band around a grid of nails positioned at each co-ordinate point. The path is defined by the points which the rubber band touches. An example is shown in Figure 3.16.

To find the convex hull of a set of points, the first step is to locate the point with the

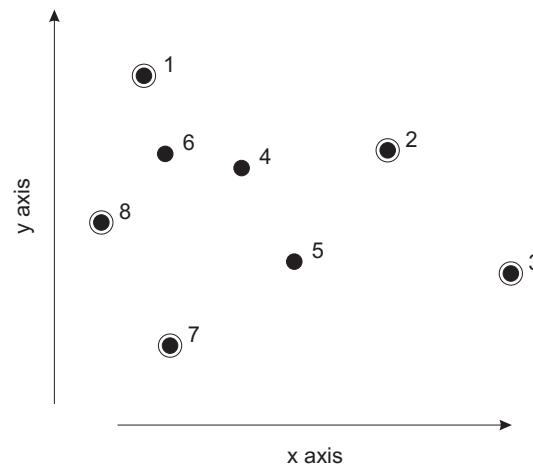


Figure 3.17: Sorted points with convex hull points ringed.

maximum y co-ordinate. This is shown as point 1 in Figure 3.17. The remaining points are then sorted according to the angle that they make at point 1 with a horizontal line passing through point 1. Points for which a left turn involves making a smaller angle to proceed to the next point in the sorted set than would a right turn are removed from the set of points which define the convex hull. This process starts off with all points as candidates. If a left turn is made, then the point is not considered any further and the algorithm backtracks to reconsider the point's predecessor. The steps to calculate the convex hull in Figure 3.17 are shown in Table 3.1.

Step	Previous Point	Current Point	Next Point	Turn ( $< 180^\circ$ )	Current hull list
1	-	1	2	Right to 2	1, 2, 3, 4, 5, 6, 7, 8
2	1	2	3	Right to 3	1, 2, 3, 4, 5, 6, 7, 8
3	2	3	4	Right to 4	1, 2, 3, 4, 5, 6, 7, 8
4	3	4	5	Left to 5	1, 2, 3, 5, 6, 7, 8
5	2	3	5	Right to 5	1, 2, 3, 5, 6, 7, 8
6	3	5	6	Right to 6	1, 2, 3, 5, 6, 7, 8
7	5	6	7	Left to 7	1, 2, 3, 5, 7, 8
8	3	5	7	Left to 7	1, 2, 3, 7, 8
9	2	3	7	Right to 7	1, 2, 3, 7, 8
10	3	7	8	Right to 8	1, 2, 3, 7, 8
11	7	8	1	Right to 1	1, 2, 3, 7, 8

Table 3.1: Steps to calculate the convex hull in Figure 3.17.

# Chapter 4

## Function Approximation Methods

Once shape features have been extracted from the fruit profile images, they must be combined in some way to form an estimate of volume. This chapter describes the workings of the artificial neural networks (ANNs) that were used in this project to combine shape features to form volume estimates for the corresponding pieces of fruit. The testing methods used on ANNs and other function approximation methods in this project are also explained. This includes the determination of errors and of the associated confidence in the results. The results of the implementation of the methods described in this chapter are presented in Chapter 7. Only a summary of the basic issues which are relevant to this project are described in this chapter; the more general field of function approximation is dealt with in far more completeness in texts such as those by Bishop [3] and Haykin [13].

### 4.1 Introduction

In the context of this project, function approximation is the determination of an adequately accurate mapping of one or more input variables to an output variable. In general, a suitable form of mapping can be found only with a set of sample input-output data points known as a *training set*. The mapping is usually modelled by a function with adjustable parameters that are determined with the aid of the training set. This process of adjusting the parameters is known as *training* or *learning* [3].

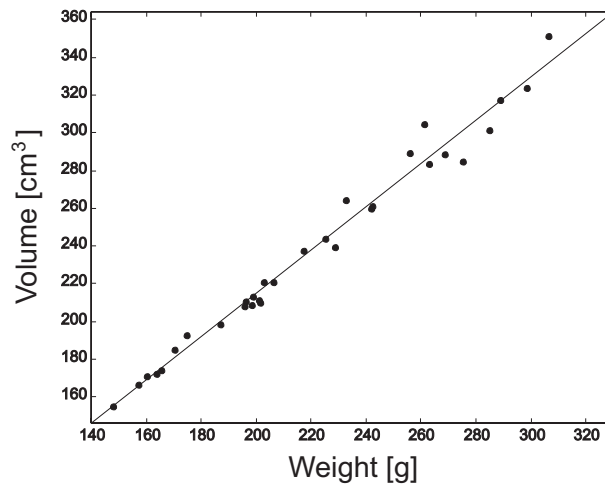


Figure 4.1: Plot of orange weight versus volume with a linear regression line shown.

In one of its simplest forms, function approximation is the determination of a linear (straight) regression line for a set of data points. For instance, Figure 4.1 shows a plot of weight versus volume for 31 oranges. The regression line has been calculated (using Equations (2.9) and (2.10) introduced in Chapter 2), and is shown in the plot. It is clear from this graph that the regression line equation could be used to predict the volumes of new oranges given their weight values. Implicit in the assumption that a linear regression line should be used for predicting future volumes is a linear relationship between weight and volume. This linear relationship is a *model* for the relationship between weight and volume and is in this case reasonable, since one would expect that the volume of a piece of fruit would be directly proportional to its weight, because fruit density is usually almost constant within a batch (with any density variations being small and unpredictable).

A model must be formed from knowledge of the problem or through an understanding of the source of the data. For example, if it was known that orange density increased with size because the peel (which is of a lower density than the rest of the fruit) formed a smaller proportion than the remainder of the fruit in larger fruit, then the model could be adjusted appropriately.

Consider the set of 10 data points in Figure 4.2. The points were computer-generated using a pseudo-random number generator and are of the form  $y = ax^3 + n$ , where  $n$  is a noise term with a PDF of normally distributed values and zero mean. (Note that the  $y = 0$  axis is not shown on the plot and does not correspond to the base of the bounding

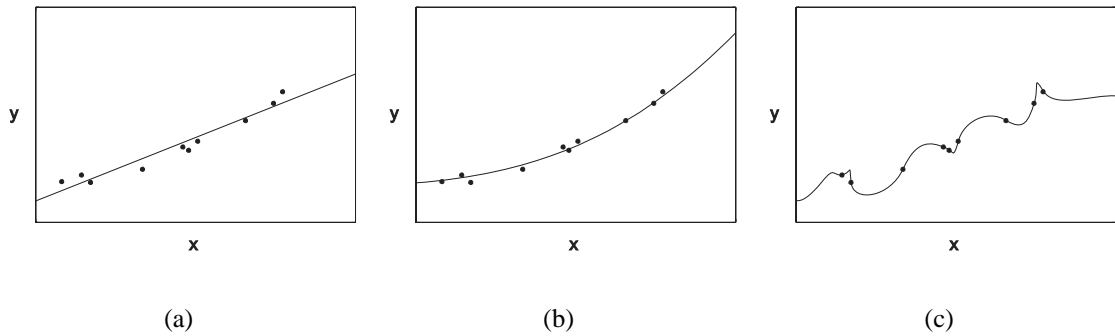


Figure 4.2: A set of data points with (a) the linear regression line; (b) the curve  $y = ax^3$ ; (c) an exact interpolation of the data points.

rectangle). The  $x$  values can be used to predict the  $y$  values because  $y$  is a function of  $x$ . Figure 4.2(a) shows the linear regression line which could be used to predict  $y$  from  $x$ . If one did not know how the  $x$  values and their corresponding  $y$  values were generated, then a linear regression line would be a reasonable method of predicting future  $y$  values from future  $x$  values. Although many functions to which linear regression could be applied are not inherently linear, linear regression may nevertheless yield reasonable results where the data comes from a small portion of a non-linear curve which is approximately linear. This approximate linearity over small ranges is often observed because of the smoothness of many real-world functions.

If extra information about the data is available then it is possible that a better model than the linear model could be formed. Perhaps the  $x$  values correspond to the diameters of oranges and the  $y$  values correspond to the weights of the oranges to be predicted from the diameters. A model can be made for this situation, for example by assuming that  $y = ax^3$  and then by finding the value of  $a$  that minimises the RMS error<sup>1</sup> on the set of data points. Such a curve is shown in Figure 4.2(b).

Often a model is not available for regression, but it is known that the relationships between the variables are not linear. In this case, a function with more flexibility than the straight line used in linear regression can be applied to the problem. For instance,

<sup>1</sup>The RMS error is a very commonly used cost function for minimisation, because it provides the lowest expected variance on the error. Minimising the RMS error is equivalent to minimising the sum-of-squares error. Other error functions are, however, sometimes more appropriate in some specific situations [3]. In Appendix A, it is shown that in the case of volume estimation of fruit to estimate fruit weights for the purposes of packing at a commercial packing house, minimising the RMS error corresponds to a direct monetary saving.



an  $n^{\text{th}}$  order polynomial  $y = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$  could be used to predict the values of  $y$  from given values of  $x$ . The constants  $a_n, a_{n-1}, \dots, a_1, a_0$  would first have to be optimised to minimise the RMS error on the data set. For any  $n$  data points, a perfect fit to the data points is possible with a polynomial of order  $n - 1$  or greater.

Figure 4.2(c) shows an example of another function (not a polynomial in this case) that is also flexible enough to interpolate each of the 10 data points exactly. Such a function, although it gives zero error on the data set (or *training set*) is quite likely to perform poorly when predicting  $y$  values from any new values of  $x$ . This is because the function has poor ability to *generalise* due to its high flexibility (which arises from a high number of parameters such as the 11 constants which must be optimised in a  $10^{\text{th}}$  order polynomial). In other words, the function is modelling the specific training set rather than the underlying input-output data relationship which created it. It is this underlying relationship which must be modelled so as to reduce the errors in predicting unseen cases of  $y$  from  $x$ . In order to do this, a regression function must be flexible enough to model the underlying population, yet not so flexible that the specific data points of the sampled training set are being modelled. The function shown in Figure 4.2(a) could be seen as an example of a function which lacks the flexibility to model the underlying cubic relationship between  $x$  and  $y$ , in contrast with the function shown in Figure 4.2(c) which is too flexible (not smooth enough) to provide good generalisation. A function which looks like the one shown in Figure 4.2(b) would appear to be a good compromise between flexibility and smoothness, and might be expected to give the best results of the three functions when used on unseen data. It is, however, difficult to predict the true performance of a function approximator in advance; such a prediction would have to make assumptions about the smoothness of the underlying generating function. The true performance of a function approximator can only be determined when it is tested on unseen cases.

ANNs can be seen as a form of regression function which can model arbitrary continuous functions where an explicit model relating the functional form of the output to the inputs is not known. There are ways of limiting the number of parameters, so that the ANN produces a smooth function which generalises and performs well on predicting unseen data. ANNs also handle multiple inputs with ease, whereas multi-variable polynomials rapidly become cumbersome as the number of variables increases.

## 4.2 The structure of a multi-layer perceptron

ANNs have their historical roots in the modelling of the operation of real brains [4, 12], but in this project they are used simply because they are a very effective form of universal function approximator [13]. By thresholding the output of the ANN, as is done in the segmentation stage of this project, the ANN can also serve as a classifier as well as being used for regression.

The multi-layer perceptron (MLP) is a type of ANN [3, 13]. It is a network of neurons or *units* which were initially developed to be roughly analogous to the neurons in an organic brain [26]. These units are very simple, in the sense that each has an output which is a relatively straightforward function of its inputs.

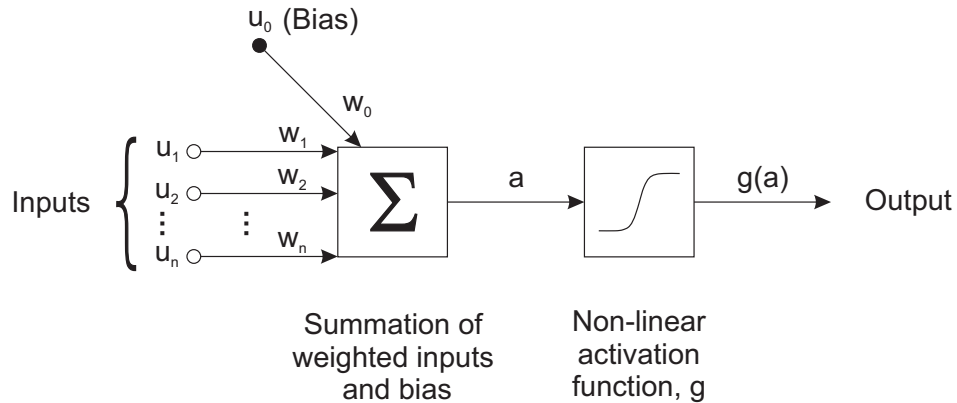


Figure 4.3: A neuron or unit, the basic element of an MLP.

Figure 4.3 shows the structure of an MLP unit. The elements of the input vector  $\mathbf{u} = (u_1, u_2, \dots, u_n)$ , together with a bias value  $u_0$  that is usually fixed at  $u_0 = 1$ , form a weighted sum, using the corresponding unit weights,  $\mathbf{w} = (w_1, w_2, \dots, w_n)$ . This weighted sum,  $a$ , serves as the input to a non-linear monotonic *activation function*,  $g$ . The activation function is needed in order to add non-linearity to the network. Almost any non-linear function works adequately as an activation function [11], but it helps, for purposes of training the network, to have a differentiable function. The output of the unit,  $g(a)$ , can be written

$$g(a) = g\left(\sum_{i=0}^n w_i u_i\right) = g(\mathbf{w}^T \mathbf{u}) \quad (4.1)$$

where  $\mathbf{w}^T$  is the transpose of the vector  $\mathbf{w}$ .

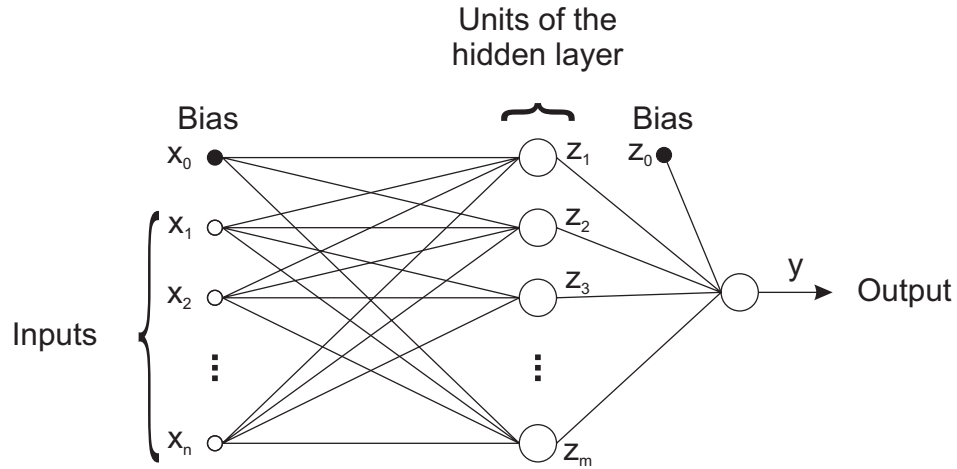


Figure 4.4: The structure of a single-output multi-layer perceptron network with a single hidden layer.

These units can be combined to form an MLP, as shown in Figure 4.4 in which each unit is represented by a circle. All of the units except for the output unit are termed *hidden units*, or units in the *hidden layer* since their output, which forms the input to the output unit, is not directly accessible. In this project, the MLPs always have a single output unit and a single hidden layer of units, and the term ‘MLP’ will refer to such MLPs; however, MLPs with multiple output units and multiple hidden layers are possible and do have applications. A single-layer MLP is a universal function approximator that can approximate any continuous function to arbitrary accuracy, given enough hidden units and if the weight values,  $\mathbf{w}$ , are selected appropriately [3].

For an MLP with  $n$  inputs,  $m$  hidden units and hyperbolic tangent activation functions

$$g(a) = \tanh(a) \equiv \frac{e^a - e^{-a}}{e^a + e^{-a}} \quad (4.2)$$

the outputs,  $z_j$ , of each of the hidden units can be written

$$z_j = \tanh \left( \sum_{i=0}^n w_{hidden,ij} x_i \right) \quad (4.3)$$

where  $w_{hidden,ij}$  is the weight value for input  $i$  in hidden unit  $j$  and  $x_i$  is the  $i^{th}$  input of an input vector  $\mathbf{x}$ . An MLP need not have the same type of activation function in the output unit as in the hidden layer units. MLPs used for regression often have linear activation functions, such as  $g(a) = a$ , in the output unit. This is equivalent to passing

the weighted sum of inputs directly to the output of the MLP. The output,  $y$ , of the MLP can therefore be written

$$y = \sum_{j=0}^m w_{out,j} \tanh \left( \sum_{i=0}^n w_{hidden,ij} x_i \right) \quad (4.4)$$

where  $w_{out,j}$  is the weight value for input  $j$  in the output unit. Linear activation functions in the output unit, together with hyperbolic tangent activation functions in the hidden layer units, were used in the MLPs for this project. Empirically, it is often found that such a combination gives rise to faster convergence of training algorithms than other choices of activation functions [3].

### 4.3 Preprocessing of data

There are several reasons why it is desirable to preprocess the input data to change its form and to reduce its dimensionality. In theory, the individual pixel values of a profile image of a piece of fruit could be used as input to a function approximator. However, due to the high dimensionality of the input space, this approach would require far too many training set samples, and far too much time, for the effective training of the function approximator. This problem is known as the *curse of dimensionality* [11]. Since one is forced to work with a limited quantity of training data, increasing the dimensionality of the input space leads to the point where the mapping of data is very sparse, providing a poor representative mapping [3]. For this reason, in the context of the volume estimation problem, it is better to discard information and to represent the profile images by an input vector of low dimensionality consisting of shape features that describe the profile image of the fruit. Such an input vector can be used to train a function approximator in a reasonable amount of time and can be expected to assist the function approximator in providing a representative mapping to the desired output (since the data is not sparsely distributed in the input vector space).

A further preprocessing stage consists of scaling the input data to zero mean and unit standard deviation. This speeds up the learning process by ensuring that the input data starts out the training process by triggering the activation functions in a relatively sensitive region. If the input to the hyperbolic tangent activation function is very high, its output becomes very close to one; likewise, if the input is very low (negative), its output becomes close to minus one.

## 4.4 Training the multi-layer perceptron

Training the MLP for regression amounts to the selection of the  $n \times m$  weight values in the hidden layer units and the  $m$  weight values in the output unit so that the error on  $y$ , the MLP output and a function of the input vector  $\mathbf{x}$ , is minimised. The cost function for this optimisation problem is evaluated using a training set of input vectors,  $\mathbf{x}_k$ , and corresponding target values,  $t_k$ . The error on an MLP output or prediction,  $y_k$ , formed from inputs  $\mathbf{x}_k$ , is  $t_k - y_k$ . The training process attempts to minimise the sum-of-squares error

$$E_N = \sum_{k=1}^N (t_k - y_k)^2 \quad (4.5)$$

where  $N$  is the number of samples in the training set. There exist numerous optimisation techniques that can be used to determine the weight values. Since the MLP output is an analytical function of its inputs (see Equation 4.4), it is possible to make use of the gradient of the error as a function of the weight values for a particular input vector  $\mathbf{x}$ . The gradient information improves the effectiveness of optimisation of the weight values and can be efficiently calculated by a process known as *back-propagation* [3].

The calculation of the gradient information begins by *forward-propagating* the input vector through the MLP to determine the outputs of the various units. This is equivalent to using the MLP in the usual manner to calculate  $y$  from  $\mathbf{x}$ . With reference to the basic unit shown in Figure 4.3, the partial derivative of the error with respect to the weights of a unit can be written

$$\frac{\partial E}{\partial w_i} = \frac{\partial E}{\partial a} \frac{\partial a}{\partial w_i} \quad (4.6)$$

This is often rewritten using the notation

$$\delta \equiv \frac{\partial E}{\partial a} \quad (4.7)$$

where values of  $\delta$  are termed *errors* [4]. Differentiating the summation

$$a = \sum_{i=0}^n w_i u_i \quad (4.8)$$

gives

$$\frac{\partial a}{\partial w_i} = u_i \quad (4.9)$$

Now, using these equations we can rewrite the partial derivative of the error with respect to the  $i^{th}$  weight of a unit as

$$\frac{\partial E}{\partial w_i} = \delta u_i \quad (4.10)$$

The values of the partial derivatives corresponding to weights in the output unit can easily be calculated. Since the activation function in the output unit for the MLPs used in this project leaves the weighted sum unaffected ( $g(a) = a$ ),

$$\delta_{out} \equiv \frac{\partial E}{\partial a_{out}} = \frac{\partial E}{\partial y} \quad (4.11)$$

and, because  $E = (t - y)^2$  for a given training example,

$$\frac{\partial E}{\partial y} = 2(y - t) \quad (4.12)$$

Using Equation (4.10) gives

$$\frac{\partial E}{\partial w_{out,j}} = 2(y - t)z_j \quad (4.13)$$

for the gradients corresponding to weights in the output unit. The value of  $z_j$  (which is the output of hidden unit  $j$  and also the  $j^{th}$  input to the output unit) is calculated during the forward-propagation stage.

In the case of the hidden units, the values for  $\delta_{hidden,j}$  are found by implementing the chain rule,

$$\delta_{hidden,j} = \frac{\partial E}{\partial y} \frac{\partial y}{\partial a_j} \quad (4.14)$$

The first factor in this product has already been calculated and the second factor can be calculated as follows:

$$\frac{\partial y}{\partial a_j} = \frac{\partial (w_{out,j} \tanh(a_j))}{\partial a_j} \quad (4.15)$$

$$= w_{out,j} \operatorname{sech}^2(a_j) \quad (4.16)$$

Again using Equation (4.10) gives

$$\frac{\partial E}{\partial w_{hidden,ij}} = 2(y - t)w_{out,j} \operatorname{sech}^2(a_j)x_i \quad (4.17)$$

for the gradients corresponding to weights in the hidden units. This stage of the calculation is known as *error back-propagation* because  $\delta_{out}$  must be *propagated back* through the network in order to calculate the values of  $\delta_{hidden}$ .

The gradient information can be used as part of a learning algorithm to update the network weights. One commonly-used learning algorithm is *steepest descent* [3]. The steepest descent algorithm starts by initialising each weight to some random value. The weight values are then iteratively updated by moving a small distance in the direction of steepest descent on the error surface. At each step, the error gradient with respect to the weights,  $\nabla E$ , is calculated by summing the derivatives for each case in the training set, to give a batch total of  $\nabla E_N$ . So the distance,  $\Delta \mathbf{w}_\tau$ , which must be added to the weights after each step  $\tau$  is

$$\Delta \mathbf{w}_\tau = -\eta \nabla E_N |_{\mathbf{w}_\tau} \quad (4.18)$$

where  $\eta$  is the *learning rate*. Provided that the learning rate is sufficiently small,  $E_N$  can be expected to be reduced after each successive step of updating the weights.

There are many other more complex optimisation techniques, the derivations of which can be found in the texts by Bishop [3] and Haykin [13]. Such techniques make use of other variables to increase the speed of convergence and to improve the probability of locating the global minimum of the error function. The optimisation technique that was ultimately chosen for training the MLPs in this project was the *quasi-Newton method* [3]. Newton-type methods use a quadratic approximation to the error surface around the point of interest. Unlike the direction of the maximum negative gradient, which is used to update the weight vector for the steepest descent method of learning, the *Newton direction*, which is calculated using Newton-type methods, points directly at the minimum of the error function. It turns out that the computation of the second derivatives of the error function, which forms a matrix known as the *Hessian* that is used to calculate the local quadratic approximation to the error surface, is highly computationally expensive. For this reason, quasi-Newton methods which build up an approximation to the Hessian matrix using first derivative information from the error function are often used.

## 4.5 Comparison with some other predictor methods

In this section a few alternatives to the MLP structure are briefly discussed to motivate the choice of the MLP architecture in this project.

Of the many ANN architectures other than the MLP, the radial-basis function network (RBFN) is one of the most widely used [3, 13]. RBFNs have pre-assigned hidden

units. The output weights can be calculated by solving a simple quadratic optimisation problem [11]. So, RBFNs have the advantage that they are quick to train. The RBFN hidden units use the distance between the inputs and a prototype vector followed by transformation with a localised function [3], whereas an MLP has a more distributed representation of a function in which many hidden units contribute to the output value for a given set of inputs. This makes MLPs more difficult and time-consuming to train than RBFNs, but, once trained, MLPs tend to generalise better, avoiding interpolation of noisy data, and performing more adequately in regions of sparse data [11].

Greene [11] suggests an MLP structure where the hidden units have randomly assigned weights, and only the weights of the output unit need be trained. Such a method reduces the training of the network to a simple quadratic optimisation problem, yet retains many of the desirable generalisation properties of the classically-trained MLP. After some investigation using the data sets from this project, this method was found to be simple to code and very fast to train, yet it failed to match the predictive accuracy of the MLP for volume estimation from shape features.

By keeping a data-base of all previous cases, it would be possible to match a new boundary to the best fit of the boundaries in the database and output that volume. This method would require far too much storage space and computational time, however, because the new case would have to be fitted against all known cases. This would be equivalent to using an ANN that has no ability to generalise. An approach where the average of the  $K$  best fits of a new case to the existing cases within a data base is known as the  $K$ -nearest neighbours approach [3].

The MLP and linear methods used in this project were thus selected after some investigation of other methods. The linear methods provide very fast and simple volume estimation based on some simple reasoning about the approximate relationships between the shape features and the fruit volume. The MLP structure was chosen for its relative simplicity and promising results after some initial experimentation.

## 4.6 Testing function approximators

Once the optimisation of the network weights is complete, the MLP function approximator can be tested to determine the RMS error which it produces on results. A set of  $N$  volume estimates,  $\mathbf{y} = (y_1, y_2, \dots, y_N)$ , with corresponding actual or target



volumes,  $\mathbf{t} = (t_1, t_2, \dots, t_N)$ , has an RMS error,  $RMSE$ , calculated using

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (t_i - y_i)^2}{N}} \quad (4.19)$$

The *RMS percentage error* is also sometimes useful for interpreting results, and is calculated using

$$RMS \text{ percentage error} = \sqrt{\frac{\sum_{i=1}^N \left(\frac{t_i - y_i}{t_i}\right)^2}{N}} \times 100\% \quad (4.20)$$

Note that although it is somewhat useful in interpreting system performance, in this project the RMS percentage error is not the quantity that is minimised during the training processes.

The MLP is usually tested on a set of data known as the *test set*, which should be distinct from the training set used to optimise the network weights. The MLP is tested on data that is separate from the training set because testing the MLP on the training set data may exaggerate the performance of the MLP, since the MLP may have learned the specific patterns of the training set rather than the underlying functions which generated them.

It is often desirable to train an MLP with as large a training set as possible, since a larger training set is a better representation of the true nature of the population of input-output combinations. However, this may leave few unseen test cases for an accurate calculation of the expected RMS error. This is especially important where data are difficult to obtain, such as is the case in this project where measuring the volumes of fruit for target values is very time consuming. In cases where data are very easy to obtain, the time taken to train the MLP is the constraining factor, and the size of the training set may have to be reduced accordingly. In such cases, a large amount of unseen data would be readily available for use as a test set.

In this project, a technique known as *jackknifing* was used to evaluate the RMS error associated with the MLP and other function approximator outputs. Jackknifing is a technique which is employed to make the best use of a relatively small data set for testing and training. The method works by iteratively removing one case from the data set for use as a test set and using the remaining data as a training set. The procedure is repeated so that each case in the data set has a turn at being tested on a function

approximator (either an MLP or some other type of function approximator) that has been trained on the remaining data. The squared error results are averaged to give the expected mean-squared error that the tested function approximator would produce on sets of unseen data. The square root of the mean-squared error gives the RMS error, which is conveniently in the same units as the output of the function approximator (cubic centimetres in the case of volume estimation).

The main disadvantage of jackknifing is that it is highly computationally intensive, because the function approximator must be retrained for each new test case. Often, jackknifing is implemented by selecting groups of data, one at a time for use as a test set, instead of only selecting a single case. This reduces the computational intensity required for the jackknifing process, yet still makes effective use of the available data provided that the groups of data are small with respect to the size of the total data set. In this project, jackknifing was implemented by removing the group of cases corresponding to one fruit, and training on the remaining data. This was done because each piece of fruit had been photographed many times, resulting in separate cases of data corresponding to the same piece fruit. The group of cases corresponding to the removed fruit was used for testing. This process was repeated so that all of the data corresponding to each fruit was removed exactly once for testing. By doing this, the function approximator was being tested on data corresponding to a fruit that it had not ‘seen’ during training. Removing one case at a time, as opposed to removing the group of cases corresponding to one fruit at a time, would have had the function approximator trained on data corresponding to the same fruit that the test case data corresponded to (data from different images of the same fruit would have been in both the training set and in the test set). This would have been undesirable, because the function approximator should be tested on data that is in no way likely to be more related to the training data than a sample picked at random from the population.

The jackknifing procedure used in this project produces  $n$  sets of mean-squared errors, where  $n$  is the number of fruit in the data set. Each of the  $n$  mean-squared errors is the mean of  $m$  squared errors, where  $m$  is the number of sets of function approximator inputs corresponding to each fruit. In this project there are cases of  $m = 24$  where features from one image of the fruit is used as input to the function approximator,  $m = 12$  where features from two images are used,  $m = 6$  where features from four images are used and  $m = 1$  in the case where features from all twenty four images of the fruit are used. The expected long-term mean-squared error of the function

approximator,  $\mu_{SE}$ , can be calculated by taking the mean of the  $n$  mean-squared errors. This is the same as the mean of all of the  $n \times m$  squared-errors (because of the linearity introduced by working with the mean-squared errors as opposed to the root-mean-squared errors), but calculating the mean in the former manner allows one to attach an associated confidence to the calculated mean-squared error as a measure of the likely performance of the function approximator over many unseen cases.

One would have higher confidence in a mean-squared error result that was used to predict the performance of a function approximator if it were formed from  $n \times m$  squared-error results from  $n \times m$  different fruit than if it was formed from just  $n$  squared-error results from  $n$  different fruit (given that  $m > 1$ ). The case in this project is slightly more complex. There are  $n \times m$  squared errors coming from only  $n$  fruit. This is because each of the  $n$  fruit was used more than once by photographing the same fruit at different orientations. Therefore we know that we can be more confident in our calculated mean-squared error than in the case where  $n$  squared errors from  $n$  fruit are used, because we have  $n$  fruit, but more than one squared error from each; and we know that we can be less confident than the case where  $n \times m$  squared errors from  $n \times m$  fruit are used, because although we have  $n \times m$  squared errors, the squared errors came from only  $n$  different fruit.

A confidence statistic for the mean of the  $n$  mean-squared errors,  $\overline{MSE}$ , can be formed by calculating the sample standard deviation of the  $n$  mean-squared errors. The sample standard deviation,  $s$ , is calculated using [22]

$$s = \sqrt{\frac{\sum_{i=1}^n (MSE_i - \overline{MSE})^2}{n - 1}} \quad (4.21)$$

where  $MSE$  is an  $n$ -element vector, each element being the mean-squared error of the  $m$  squared errors associated with images from the  $n^{th}$  fruit. A confidence interval for  $\overline{MSE}$ , which is the expected mean-squared error of the function approximator given the  $n \times m$  squared errors from  $n$  different fruit, can be formed using the  $(1 - \alpha)100\%$  confidence interval formula [22]

$$\overline{MSE} - t_{\alpha/2} \frac{s}{\sqrt{n}} < \mu_{SE} < \overline{MSE} + t_{\alpha/2} \frac{s}{\sqrt{n}} \quad (4.22)$$

where  $t_{\alpha/2}$  is obtained from a lookup table. For the 95% ( $\alpha = 0.05$ ) confidence interval, the values of  $t_{\alpha/2}$  relevant to this project are:

$$\begin{aligned}
t_{\alpha/2} &= 1.960 && \text{for } n = 31 \text{ in the case of oranges} \\
t_{\alpha/2} &= 1.960 && \text{for } n = 64 \text{ in the case of pears} \\
t_{\alpha/2} &= 2.074 && \text{for } n = 23 \text{ in the case of lemons} \\
t_{\alpha/2} &= 2.093 && \text{for } n = 20 \text{ in the case of potatoes}
\end{aligned}$$

Equation (4.22), and others which use the  $t$  statistic, are derived using the assumption of normally distributed values of  $MSE$ . However, it has been shown empirically that the statistics are often reliable even if drawn from a non-normal distribution [22]. Although one would expect a distribution of squared errors to be non-normally distributed, here one is usually dealing with sets of the means of 24 of such squared errors, requiring only that the spread of these mean-squared errors from fruit to fruit be roughly normally distributed for the confidence interval to be valid.

From Equation (4.22) we can observe that as the number of fruit,  $n$ , is increased, the confidence interval becomes narrower and we can be more certain that the expected mean-squared error,  $\overline{MSE}$ , represents the true mean of the population of all squared errors,  $\mu_{SE}$ . Thus, as  $n$  approaches infinity,  $\overline{MSE}$  approaches  $\mu_{SE}$ . We can also expect the confidence interval to become narrower as  $m$ , the number of squared errors per fruit becomes larger. This is because increasing  $m$  reduces the standard deviation of  $MSE$ ,  $s$ : for small  $m$ ,  $s$  incorporates squared errors due to differences in both fruit and orientation (as well as any experimental inconsistencies such as the extent to which the fruit does not sit snugly in the cup); but as  $m$  becomes large,  $MSE$  becomes a better estimate of the true mean-squared error for a particular fruit (the mean of all squared errors from all possible orientations of the fruit), and  $s$  begins to vary only because of differences in the spread of possible squared errors from fruit to fruit. As  $m$  approaches infinity,  $s$  approaches a limit where it describes only the variability in spread of squared error from fruit to fruit rather than any variability due to orientation. This confirms and quantifies the intuition that we may be more confident in our estimate of  $\mu_{SE}$  if we use squared errors from a data set containing multiple images of the same fruit instead of a single image per fruit, even although the same number of actual fruit is used.

The mean and standard deviation of the mean-squared errors from each jackknifing test set group also provide a useful way of testing the significance of the improvement of one function approximation method over another. For instance, we may have a function  $f_1(\mathbf{x}_1)$  which approximates fruit volumes from a vector of profile image shape features,  $\mathbf{x}_1$ , and another function  $f_2(\mathbf{x}_2)$  which provides a different method of volume estimation of the same fruit volumes from a different set of shape features,  $\mathbf{x}_2$ . Suppose

we suspect that  $f_2(\mathbf{x}_2)$  gives a lower mean MSE from the jackknifing results than  $f_1(\mathbf{x}_1)$ , because it is inherently a better method, rather than the lower mean being accredited to chance. This may be assumed because the input vector  $\mathbf{x}_2$  contains more information than  $\mathbf{x}_1$  or perhaps  $f_2$  is a more effective functional combination of the input vector elements. The probability that such a hypothesis is correct can be estimated by assuming the MSEs obtained from the results of the jackknifing are normally distributed. The sample standard deviations,  $s_1$  and  $s_2$ , of the MSEs are estimated from the  $n_1$  samples of the mean-squared errors,  $MSE_1$ , generated by  $f_1$  and from the  $n_2$  samples of the mean-squared errors,  $MSE_2$ , generated by  $f_2$  respectively. The hypothesis to be tested is that  $\mu_1 < \mu_2$  where  $\mu_1$  is the true mean of the MSEs generated by  $f_1$  and  $\mu_2$  is the true mean of the MSEs generated by  $f_2$ . A  $t$ -test statistic can be written [22]:

$$t = \frac{\overline{MSE_1} - \overline{MSE_2}}{\sqrt{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}} \sqrt{\frac{n_1 n_2 (n_1 + n_2 - 2)}{n_1 + n_2}} \quad (4.23)$$

from which the probability,  $p$ , that  $\mu_1 < \mu_2$  can be found for the values of  $n_1$  and  $n_2$  which occur in this project (the number of degrees of freedom  $n_1 + n_2 - 2$  should be greater than 30 for a good approximation [22]) using

$$p = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^t e^{-\frac{u^2}{2}} du \quad (4.24)$$

which is evaluated using a lookup table, since no closed-form analytical solution for the integral exists [22]. We can now say with probability  $p$  that  $f_2$  is a superior function (in terms of lower RMS error) to  $f_1$ .

# Chapter 5

## Data Acquisition

The data acquisition stage of this project consisted of two parts: measuring the actual volumes of the pieces of fruit and then photographing them. The actual volumes of the fruit were measured to serve as target volumes for training function approximators to estimate fruit volumes from shape features extracted from profile images of the fruit. The fruit were photographed to acquire these digital profile images, which are similar to the images used by packing houses for colour sorting and blemish detection, and from which the shape features could later be extracted.

Four fruit types were chosen to investigate the performance that one could expect from different volume estimation methods on a range of different fruit. The orange and the potato were chosen since they represent a highly symmetrical fruit and a highly irregularly-shaped fruit respectively. Volume estimation of these fruit has been performed before and is recorded in published literature [23, 21]. The results of volume estimation on these fruit can thus be evaluated with reference to the existing methods described in the literature. In addition to oranges and potatoes, pears and lemons were also used in this project. Pears and lemons are both roughly cylindrically symmetrical fruit, the pear having a more distinctive side-view and the lemon being closer to an ellipsoid.

## 5.1 Volume measurement

The obvious answer to the problem of measuring the volume of a piece of fruit would be to submerge the fruit in a beaker of water. Two difficulties are encountered when attempting to measure the volume of a piece of fruit by submerging the fruit in a beaker of water and noting the displacement of the water. The first of these is that the fruit floats in the water and must be forcibly submerged by sinking it with a thin rigid piece of wire. Secondly, even if the fruit is successfully submerged with the wire, or if a liquid of lower density than the fruit is used (so that the fruit sinks in the liquid), the resolution to which one can measure the volume of displaced liquid is severely limited by the diameter of the beaker (which must be large enough so that the fruit can fit into the beaker). In the general case where the fruit can fit into the top of the beaker, even quite a large change in volume will correspond to a small change in the height of the liquid, so the volume change can be read off the side of the beaker only with rather limited accuracy.

A *eureka can* is an apparatus used to measure the volume of objects by Archimedes' Principle (displacement of liquid). The eureka can, which is essentially a beaker with a spillway spout, is filled with water until it overflows. The object whose volume is to be measured is then submerged in the eureka can, and the water which is displaced by the object so that it flows out of the spout is collected in a measuring cylinder. From this, the volume of the object can be determined accurately, since it is equal to the volume of displaced water that is collected in the measuring cylinder. A eureka can was built for this project, and methylated spirits (which is less dense than fruit) was used in place of water so that the fruit sank. Unfortunately, it was found that the liquid continued to drip out of the eureka can spout for many minutes after the fruit had sunk to the bottom. This severely hampered the taking of accurate measurements (at least within any reasonable period of time), and for this reason the eureka can method of volume measurement was abandoned.

Most of the other proposed methods posed other difficulties. For instance, it was suggested that the fruit should be weighed first underwater (with the weighing mechanism underwater too) and then weighed again out of the water. The difference between these two readings would yield the buoyancy force due to the displaced water, which could then be used to calculate the volume of displaced water. This method is theoretically sound, but it is far from practical, especially when one considers that a weighing de-

vice which functions both under water and in air must be used and that the water would have to be replaced with a liquid of lower density in order to have the fruit sink. It was also suggested that the fruit first be photographed and then chopped into small pieces which could be used to displace water in a long, thin measuring cylinder. This method was rejected because the fruit pieces would absorb some of the water osmotically and end up displacing less than the volume of the original fruit.

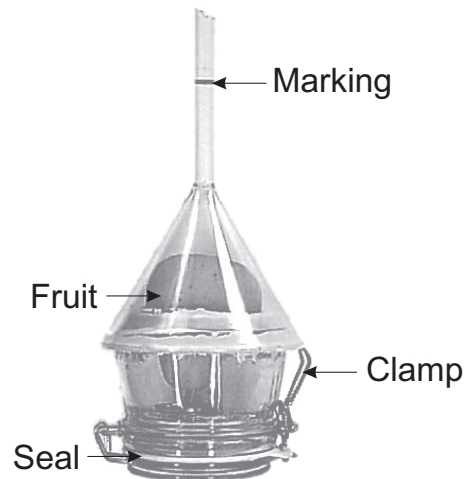


Figure 5.1: The pycnometer that was built for the purpose of measuring individual fruit volume (with a pear inside).

To avoid the problems mentioned above, the volumetric determination apparatus which is shown in Figure 5.1 was built and used to measure the individual volumes of fruit to an accuracy of better than 0.5%. It was later discovered that such a device is known as a *pycnometer* and that it is most commonly used to the measure specific gravity of soil in the study of soil mechanics [35]. Such pycnometers are, however, usually only about one tenth of the size of the pycnometer built for use in this project. The pycnometer used in this project has a wide opening at the base into which the fruit can be placed. The opening can then be clamped and sealed closed. The pycnometer was constructed from a glass jar with a clamp-type sealable lid and a glass funnel. A small marking was made on the top tube section (see Figure 5.1) so that the pycnometer could be filled accurately and repeatedly to the same volume. A 1cm change in the height of the water in the tube section corresponds to only a  $0.8\text{cm}^3$  or 0.8ml change in the volume of water in the pycnometer. Apart from yielding this accuracy, the thin tube section also solves the problem of the fruit floating in the water, as the tube is far too narrow for the fruit to float to the water's surface.



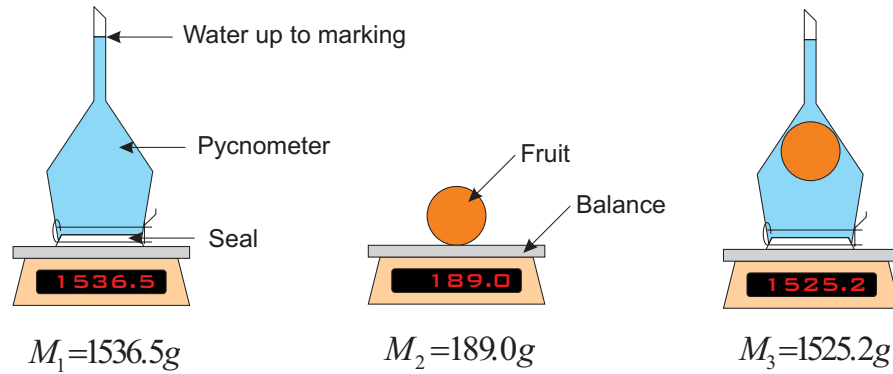


Figure 5.2: Using the pycnometer to measure fruit volume.

To use the pycnometer for volume measurement, a balance was first used to find the mass,  $M_1$ , of the apparatus filled with water to the marking on the tube section. In this project, an electronic balance accurate to one tenth of a gram was used. Next, the mass,  $M_2$ , of the fruit was measured using the balance. The fruit was then placed inside the pycnometer and the lid with a rubber seal was clamped shut. The apparatus was filled with water up to the marking, and the total mass,  $M_3$ , was measured. Fruit volume,  $V$ , is then given by,

$$V_{fruit} = \frac{M_1 + M_2 - M_3}{\rho_{water}} \quad (5.1)$$

where  $\rho_{water}$  is the density of the water used. Whilst it was realised that water density varies with both purity and temperature, the water used in this project at laboratory temperatures was pure, and had unit density to four significant figures. The numerator of the equation,  $M_1 + M_2 - M_3$ , is the mass of the water displaced by the fruit. The volume of this water is obtained by dividing by the density of the water.

This procedure is made clearer by the example shown in Figure 5.2. By using the numbers shown in the figure and Equation (5.1), the fruit's volume is calculated to be  $200.3\text{cm}^3$  :

$$V_{fruit} = \frac{1536.5 + 189.0 - 1525.2}{1.000} = 200.3\text{cm}^3 \quad (5.2)$$

The primary source of error on the volume measurement is any lack in consistency in clamping the base of the pycnometer closed due to the slight lack of repeatability in the compression of the rubber seal in the pycnometer's base. The consistency of

the pycnometer readings was tested by repeatedly measuring the volume of the same piece of fruit. Ten readings over the space of several hours gave a standard deviation of  $0.3\text{cm}^3$ . It is difficult to obtain fruit-sized objects of precisely-known volume for testing the pycnometer's measurement accuracy. However, the pycnometer volume measurement of a  $141.8\text{cm}^3$  ball, used in this project for image size calibration, agreed with the volume of the ball, determined using micrometer callipers, to  $0.2\text{cm}^3$ .

The fruit used in this project was purchased from a trader who sells fruit and vegetables in bulk quantities. When purchased from this particular trader, the fruit had only been sorted to a limited extent, so one could obtain a reasonably diverse data set, which might typify the fruit to be sorted in a packing house, for each of the fruit types. The volumes of 31 oranges, 64 pears, 23 lemons and 20 potatoes were measured using the pycnometer. This measurement process averaged around five minutes per fruit, and took place over several days.

## 5.2 Photographing the fruit



Figure 5.3: Mirror chamber and camera.

After its volume had been accurately measured, each piece of fruit was photographed in the mirror chamber as shown in Figure 5.3. The mirror chamber was built to simulate a multiple camera set-up by obtaining multiple views of each fruit from one

photograph. Six photographs were taken of each fruit, with the fruit placed by hand in a random orientation within the cup for each photograph. The cup had an inner diameter of 45mm, which was large enough to allow the fruit to sit snugly in position at different orientations, yet small enough not to obscure the fruit object boundary in any of the images. It was important that the fruit sat snugly in the cup, since any variation in the projected area of the fruit due to the fruit sitting either too far forward or too far back in the cup would translate into variation in the volume estimates.

An Olympus Camedia C1400L digital camera was used, with flash mode selected so as to enhance the contrast between the fruit and the background. The chamber was painted with matt black poster paint to limit reflection from the flash. The camera was mounted on a tripod which was rigidly clamped to the table to prevent any movement from occurring during or between the acquisition of the photographs. The mirror chamber was also rigidly fixed to the table. The resolution of the images ranged from about 2.2 pixels per millimetre for the actual fruit to about 1.8 pixels per millimetre for the reflected image in the rear mirror.

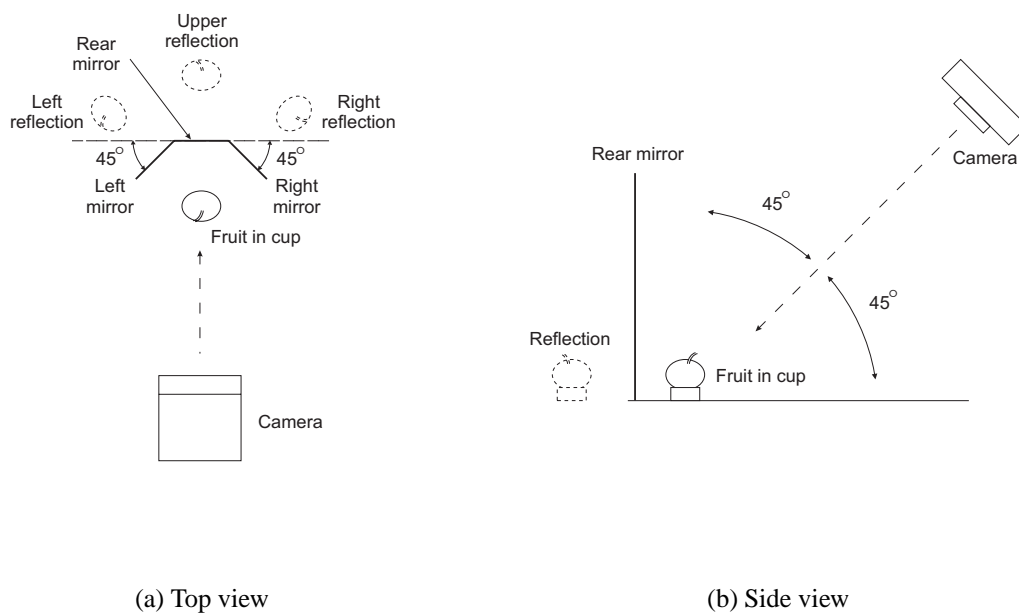


Figure 5.4: Top view and side view of the mirror chamber and camera.

Each of the three mirrors was mounted perpendicularly to the floor, with the left and right mirrors each at  $45^\circ$  to the rear mirror. The camera was positioned so that its optical axis was at  $45^\circ$  to both the rear mirror and the floor, in a plane perpendicular to both the rear mirror and the floor (see Figure 5.4). Such an arrangement maximises the angles between the fruit image boundaries without having any of the boundaries cutting through the cup. Thus, at sufficient distance from the fruit, the single camera simulates a multiple camera set-up of two pairs of cameras each with its optical axis at  $90^\circ$  to the other, with the planes formed by the optical axes of both cameras of each pair also at  $90^\circ$  to one another, all looking down on the fruit at an angle of  $45^\circ$  to the floor. This set-up is shown in Figure 5.5. Note that the images formed from reflections in the mirror chamber have to be flipped and scaled in order for the mirror chamber to be a simulation of the set-up in Figure 5.5.

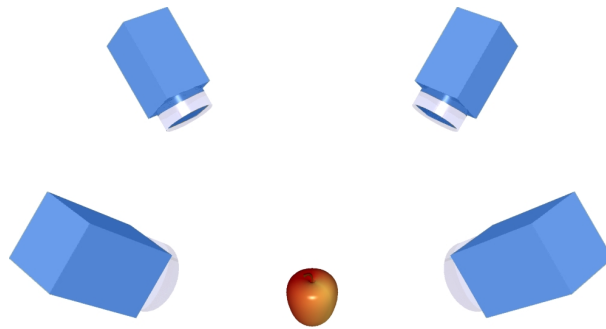


Figure 5.5: Four cameras looking down at the fruit.

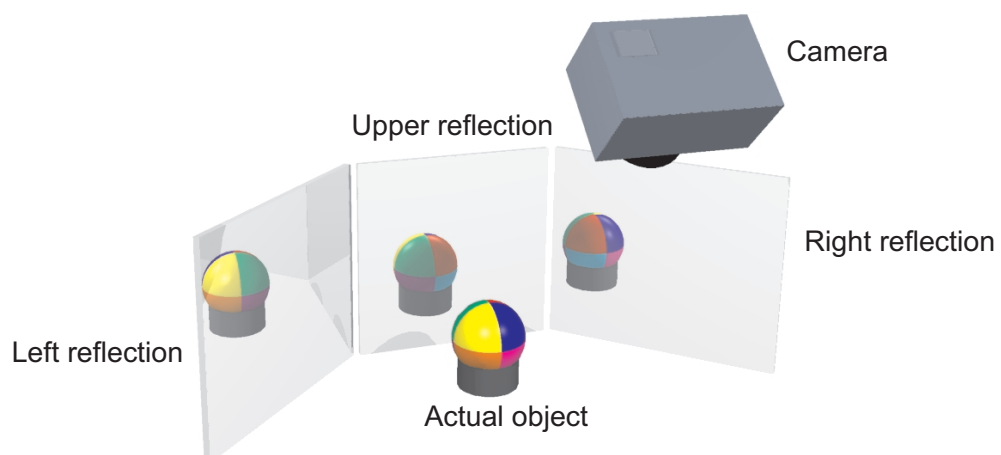


Figure 5.6: Computer-generated graphic of the mirror chamber with coloured ball.

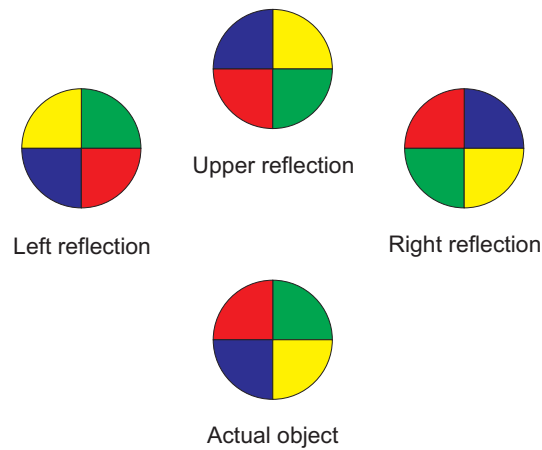


Figure 5.7: Top view of the coloured ball in the mirror chamber.

The diagrams in Figures 5.6 and 5.7 help to visualise the layout of the mirror chamber and the orientations of the images of fruit placed inside the chamber. Figure 5.6 is a computer-generated graphic of an eight-coloured ball that is placed in a cup and reflected in three mirrors, with the camera looking down on the set-up at an angle of  $45^\circ$  to the floor. Figure 5.7 shows the top view (see Figure 5.8) of the ball placed in the cup. The ball is divided into eight differently-coloured octants to help to demonstrate the appearance of the reflections from the camera's point of view. The three reflected images are also illustrated in Figure 5.7, although it should be noted that they would not be visible in this way from an actual top view because the reflective surfaces of the mirrors cannot be seen from the top.

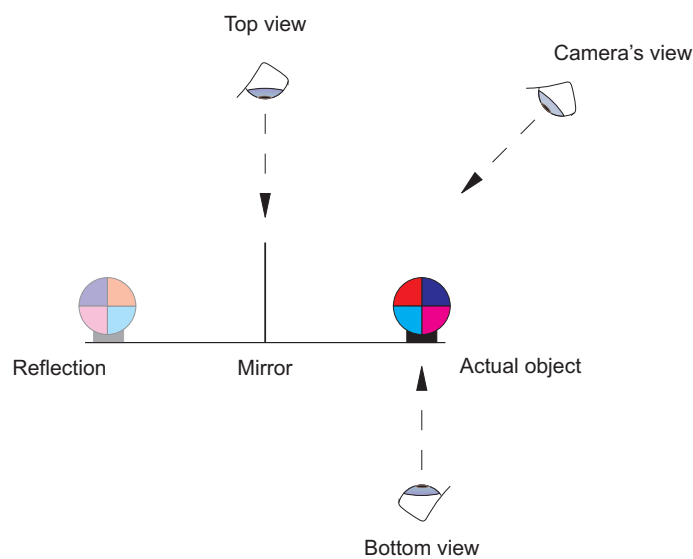


Figure 5.8: Side view of the coloured ball in the mirror chamber.

Figure 5.9 shows the four sub-images as seen by the camera (see Figure 5.8). It is useful to view this figure together with Figure 5.7: the camera is looking down at the mirror chamber at an angle of  $45^\circ$  whereas Figure 5.7 is directly from above. The boundaries of the pair of sub-images formed from the actual object and the upper reflection are in planes which are perpendicular to one another. Similarly, the boundaries of the pair of sub-images formed from the left reflection and the right reflection are in planes which are perpendicular to one another. These two pairs of planes are at an angle of  $45^\circ$  to one another.

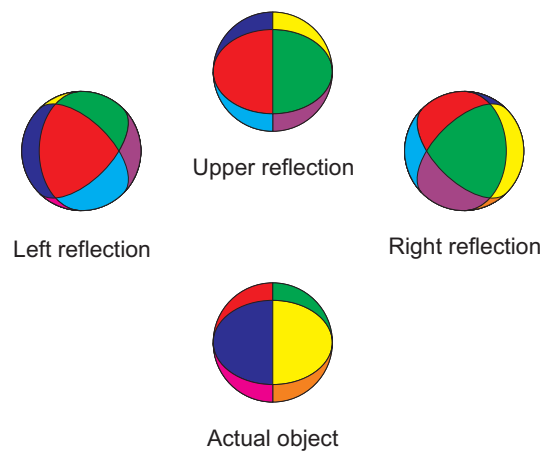


Figure 5.9: Camera's view of the ball and reflections.

Figure 5.10 shows the bottom view of the ball in the mirror chamber. The grey section represents the part of the ball which is not seen by the camera in any of the four views. Note that none of the four ball boundaries seen by the camera run through this grey section. The part of the ball that is obscured by the cup holder must lie within this section so that the ball boundaries as seen by the camera are never obscured by part of the cup. Thus, the cup must have a small enough diameter to satisfy this constraint, yet a large enough diameter so that the fruit sits snugly within it.

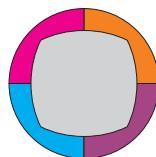


Figure 5.10: Bottom view of the ball with grey hidden region.

The assumption that the camera is sufficiently far from the fruit is essentially an assumption that the effects of perspective can be ignored. The distance from the camera to the fruit in the set-up for acquiring fruit images was about 800mm. The implications, in terms of perspective, can be understood by considering the diagram shown in Figure 5.11. The eye is at a distance  $d$  from the centre of a sphere of radius  $r$ . The

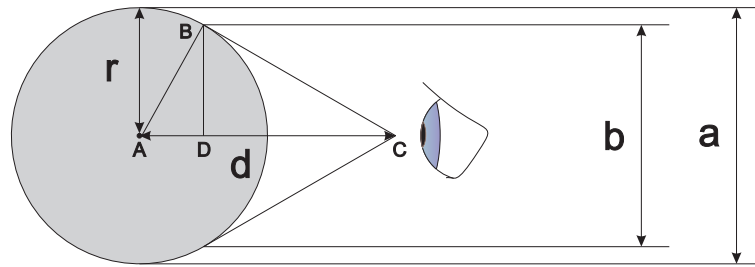


Figure 5.11: The effects of perspective at finite distance from a sphere.

length  $a$  is the diameter of the profile circle seen at infinity; it is equal to the actual diameter of the sphere. The length  $b$  is the diameter of the profile circle, and is the apparent diameter of the sphere as seen by the eye. The ratio of  $b$  to  $a$  is given by the equation

$$\frac{b}{a} = \frac{\sqrt{d^2 - r^2}}{d} \quad (5.3)$$

which can easily be derived by noting that triangle  $ABC$  and triangle  $ADB$  are similar. In the case of the camera photographing a spherical piece of fruit in the mirror chamber, the value  $r$  might typically be about 75mm (corresponding to a volume of around 221cm<sup>3</sup>) and the distance from the camera to the centre of the fruit is about 800mm. This results in the apparent diameter of the fruit,  $b$ , being 99.6% of the true diameter,  $a$ . Admittedly, different shape features would have different resultant errors in assuming that the profile image as seen by the camera was the same as the profile image that would be seen from infinity, but the given example does help to illustrate that ignoring perspective effects is unlikely to result in significant detrimental effects on the results of any calculations based on this assumption. This is especially true because the volume estimation methods developed in this project do not explicitly assume the absence of the effect of projection; the assumption was only made to simplify the thinking involved in developing some of the methods. Since the methods are based on statistical estimation rather than measurement, any small yet consistent change in a shape feature throughout the data set, as a consequence of projective effects, will be of no significance, since the change will be the same for all of the profile images.

Fruit type	Count	Images per fruit	Total images
Orange	31	24	744
Pear	64	24	1536
Lemon	23	24	552
Potato	20	24	480

Table 5.1: Number of images acquired for each fruit.

The oranges, pears, lemons and potatoes used in this project were each photographed six times at random orientations in the mirror chamber, as described above. Table 5.1 shows the resultant number of pictures acquired for each of the fruit types.

Fruit type	$\bar{V}$ [cm <sup>3</sup> ]	STD( $V$ ) [cm <sup>3</sup> ]
Orange	237.8	52.6
Pear	193.6	7.8
Lemon	224.6	60.0
Potato	239.8	48.5

Table 5.2: Means and standard deviations of fruit volumes in the four data sets.

Table 5.2 shows the means,  $\bar{V}$ , and standard deviations,  $STD(V)$ , of the volumes derived from pycnometer measurements of all of the fruit samples of each of the four fruit types. As can be seen from this table, the pear data set has a much lower spread of volumes than the other three data sets.



## Chapter 6

# Preprocessing, Segmentation and Feature Extraction

This chapter describes the processes implemented on all of the raw images of fruit that were acquired with the digital camera. These processes convert the digital images into a set of numbers, or features, which can then be used to estimate the volumes of the corresponding pieces of fruit.

Firstly, several preprocessing stages are necessary to locate the four pieces of fruit within each of the original images and then to form new images containing each individual piece of fruit separately. The new images are very similar in scale and resolution to the images used commercially for colour and blemish sorting.

A scaling factor is next determined for each of the four mirror chamber views. The scaling factors are used to bring the fruit profile boundaries to the same scale.

The methods used for combining the three colour bands to form grey-scale images are then explained. The grey-scale images are necessary because they are used as an intermediate stage in forming binary images, in which the fruit may easily be separated from their backgrounds.

Before features can be measured from the pear images, the stalk section must be removed from each image. The reasons for doing this and the methods used are explained later in this chapter.

The segmentation to identify the fruit object within the profile image in this project is not necessarily the same as the processes used in existing commercial systems. The specifics of the segmentation process used differ from system to system [38]. The purpose of the segmentation stage of this project is to isolate the experimental images so that function approximators based on features obtained from well-segmented images can be designed and later meaningfully tested.

The fruit boundaries identified from the resulting binary fruit images are then used to extract shape features for each image. This process of selecting and measuring features is described in the final section of this chapter.

## 6.1 Preprocessing of the original images

The first stage in the computer processing of the original  $1280 \times 1024$  pixel images from the digital camera was to form separate image files for the four pieces of fruit within each of the original images from the camera. This was necessary since locating the fruit within the large image would be very computationally expensive. For this reason, each of the large images was cropped to form four smaller images of the relevant portions for future processing.

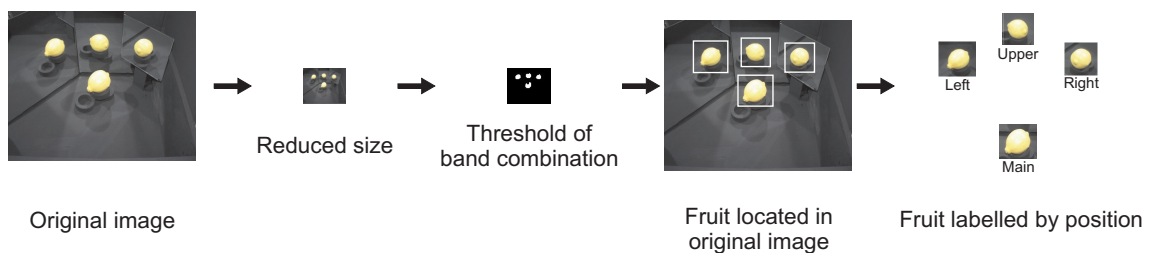


Figure 6.1: Locating the four fruit images from the digital photograph.

The process of forming separate image files for each fruit is summarised in Figure 6.1. Firstly a small  $256 \times 205$  pixel reduced resolution image was formed as a resized version of the original image. Rather than using the original images, smaller resized images were processed to locate the fruit objects because this greatly reduced the computational time required to process the entire set of images. From these resized images, grey-scale images were formed from the sum of the red and green bands less twice the blue band. The new images,  $P$ , were thus each formed using the linear

combination

$$P(x, y) = R(x, y) + G(x, y) - 2B(x, y) \quad (6.1)$$

where  $R$ ,  $G$  and  $B$  are the red, green and blue bands of the original image respectively. This method of combining the bands is relatively fast to compute and tends to highlight the fruit (which, being yellow/brown/orange in colour, have high red and green values and low blue values for each of the fruit types used in this project). It was important that the method should be relatively fast on account of the size of the task: 834 images of oranges, pears, lemons and potatoes, each  $1280 \times 1024$  pixels, had to be cropped into 3336 smaller images.

Next, the grey-scale images were thresholded to form binary images, as shown in the third picture in Figure 6.1. For each fruit type, the threshold value was simply found by inspection (with reference to several image histograms), as it was felt that more elaborate threshold locating methods [14] would be unnecessarily time consuming. All that was required at this stage was the rough location of the fruit objects within the large image. The objects (connected regions) in these binary images formed the set of initial candidates for the locations of the fruit objects. The four largest binary objects were identified as the approximate locations of the four fruit for each image. Several much smaller binary objects were sometimes present on the binary images, but these had usually formed as a result of specks of dust in the mirror chamber.

A corresponding rectangular region for each fruit within the original image was then determined. This was done by multiplying the boundary co-ordinates of the four binary objects by a scaling factor. An additional 50 pixels was added to the lengths and widths of each rectangle to ensure that the entire fruit was contained. The images within these rectangular regions were then copied from the original image. These regions were saved to disk to form the four image files for each of the original images. The  $x$  and  $y$  co-ordinates of the centroids of the binary fruit objects were used to label the four fruit image files, to allow later determination of whether the image came from the left, right or upper mirror, or if it were the main view of the fruit.

## 6.2 Size calibration

The fruit images were calibrated for size by using images of an orange sphere with a consistent diameter of 64.7mm (measured with micrometer callipers), the volume of

which was confirmed with the pycnometer. During the data capture stages for all of the fruit types, a photograph of the sphere was taken after every ten fruit photographs. This was done for each fruit type to ensure that the relative dimensions of the four fruit images per photograph were well-known, even if the data capture was taking place over a period of several days. In this way variations in the geometry between the data capture runs of the different fruit types could be taken into account (especially since the apparatus was reassembled for each data capture run for a particular fruit type).

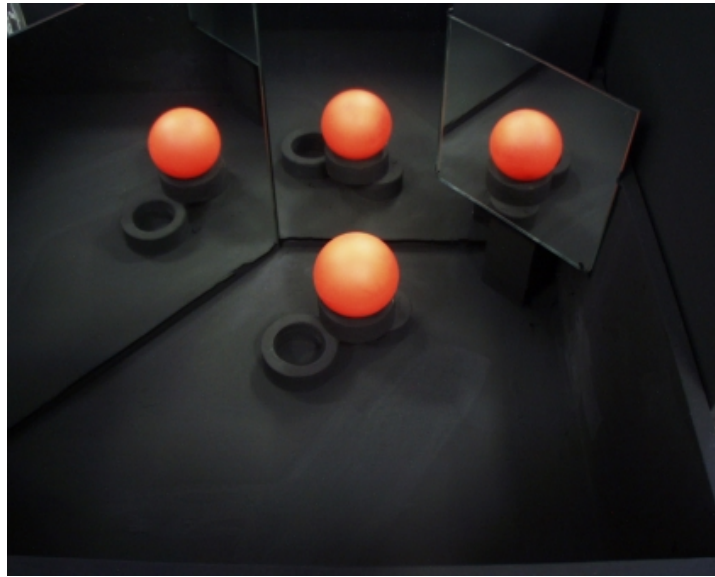


Figure 6.2: Mirror chamber with size calibration sphere.

An example of one of the calibration images is shown in Figure 6.2. The number of pixels (area,  $A$ ) in the resultant circular objects were counted for each of the four views. The circular objects were identified by thresholding the red bands, and the threshold value was easily found by inspection of the red band histograms. The results for all of the calibration photographs were averaged.

Run	Left mirror view			Right mirror view			Upper mirror view			Main view		
	$\bar{A}$	$STD(A)$	Mult.	$\bar{A}$	$STD(A)$	Mult.	$\bar{A}$	$STD(A)$	Mult.	$\bar{A}$	$STD(A)$	Mult.
Orange	12773.3	77.2	1.193	13126.3	55.3	1.177	12634.5	79.1	1.200	18193.0	34.4	1.000
Pear	11729.3	18.1	1.165	11245.7	26.7	1.190	10994.0	21.8	1.204	15931.3	30.4	1.000
Lemon	12747.3	28.7	1.194	12981.3	14.2	1.183	12538.7	55.2	1.204	18175.0	34.0	1.000
Potato	12612.5	139.3	1.199	13061.0	99.0	1.179	12476.5	115.3	1.206	18139.0	60.8	1.000

Table 6.1: Mean area,  $\bar{A}$ , standard deviation,  $STD(A)$ , and corresponding distance multiplier, Mult., for objects in calibration images.

This information is shown for each data capture run, identified by the name of the fruit, in Table 6.1. From these average areas, a distance multiplier was calculated to scale any distance in the mirror sub-images to an equivalent distance in the main view sub-image. The formula for doing this calibration is

$$Multiplier_{view} = \sqrt{\frac{\bar{A}_{view}}{\bar{A}_{main\ view}}} \quad (6.2)$$

### 6.3 Combining the colour bands

The purpose of the segmentation stage is to identify and isolate the fruit objects within the images. The required result is a binary image with ‘ones’ labelling the pixels corresponding to fruit within the image and with ‘zeros’ corresponding to background sections. The binary images were formed from grey-scale images which were in turn formed from the original RGB fruit images. A binary image was formed for each of the fruit images in the four data sets (orange, pear, lemon and potato).

In combining the three colour bands to form the grey-scale image, it was very desirable for the fruit sections of the images to consist of higher intensity values than the background. This was necessary to allow the conversion of the grey-scale image into a binary image which would accurately distinguish foreground from background. The rudimentary method described above in Equation (6.1) is quite effective for quickly combining the RGB bands so that the fruit object can be located roughly within the image. For a more accurate identification of the fruit object, however, a more general (non-linear) combination of the bands must be made.

Multi-layer perceptron (MLP) neural networks were used to form non-linear combinations of the three colour bands. This was done because MLPs with a small number of nodes in the hidden layer are able to produce smooth yet not necessarily linear (and thus more flexible) functions of multiple inputs. With a small number of nodes in the hidden layer, and a large training set, there is little danger of the MLPs being over-trained, as they simply would not have the flexibility to become so. Given the already quite satisfactory results using a simple linear combination of the colour bands, it is unlikely that an MLP would need much more flexibility than a linear combination of bands for accurate labelling of fruit image pixels as background or as foreground.

Four MLPs were developed, one for each of the four fruit types. The MLPs were designed to take three inputs, the red, green and blue values for a pixel, and to output one value, the grey-level value, as a function of the three input values. These grey-level output values were then to be thresholded, resulting in a binary image.

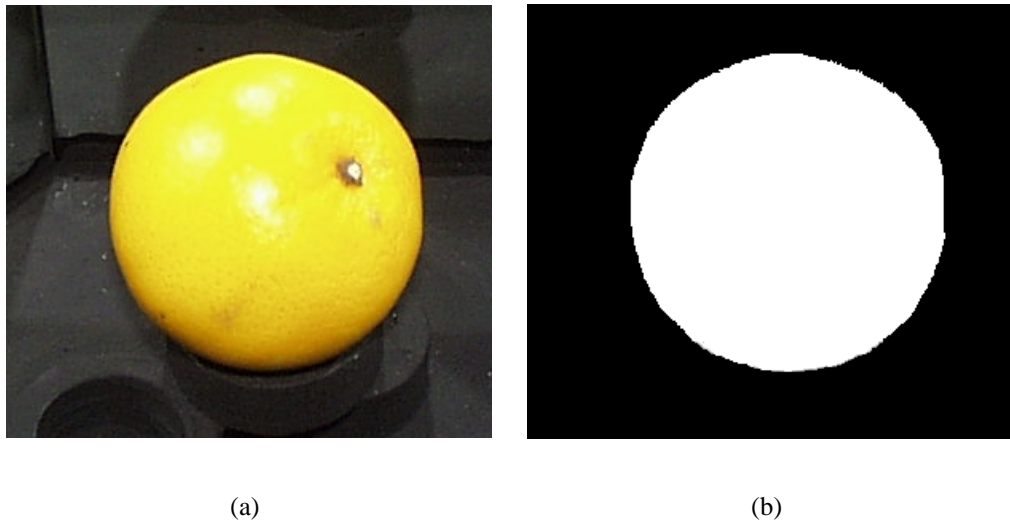


Figure 6.3: Manual identification of foreground and background showing (a) the original, and (b) the mask image.

To train the MLPs, eight binary image masks were formed for each fruit type. This was done by manually distinguishing the foreground from the background pixels for each of the selected fruit images, judging by eye whether each pixel considered was part of the fruit or part of the black background of the mirror chamber. These training images were selected by picking two fruit images at random for each of the four mirror chamber views (main, left, right and upper views) for each fruit type. Figure 6.3(a) shows a picture of an orange image selected for training an MLP and its corresponding manually-segmented mask in Figure 6.3(b). This orange is used as an illustrative example for the rest of this section.

In order to have the MLPs concentrate on the important regions of segmentation, namely the regions close to the boundary between the fruit and the background (since it is this boundary which will ultimately describe the fruit shape), these regions in particular were used to train the MLPs. Misclassified pixels that were not connected to the boundary could later easily be corrected by filling-in or deleting regions. To select the image regions for training the MLPs, binary dilation and erosion of the original manually-segmented mask was used.

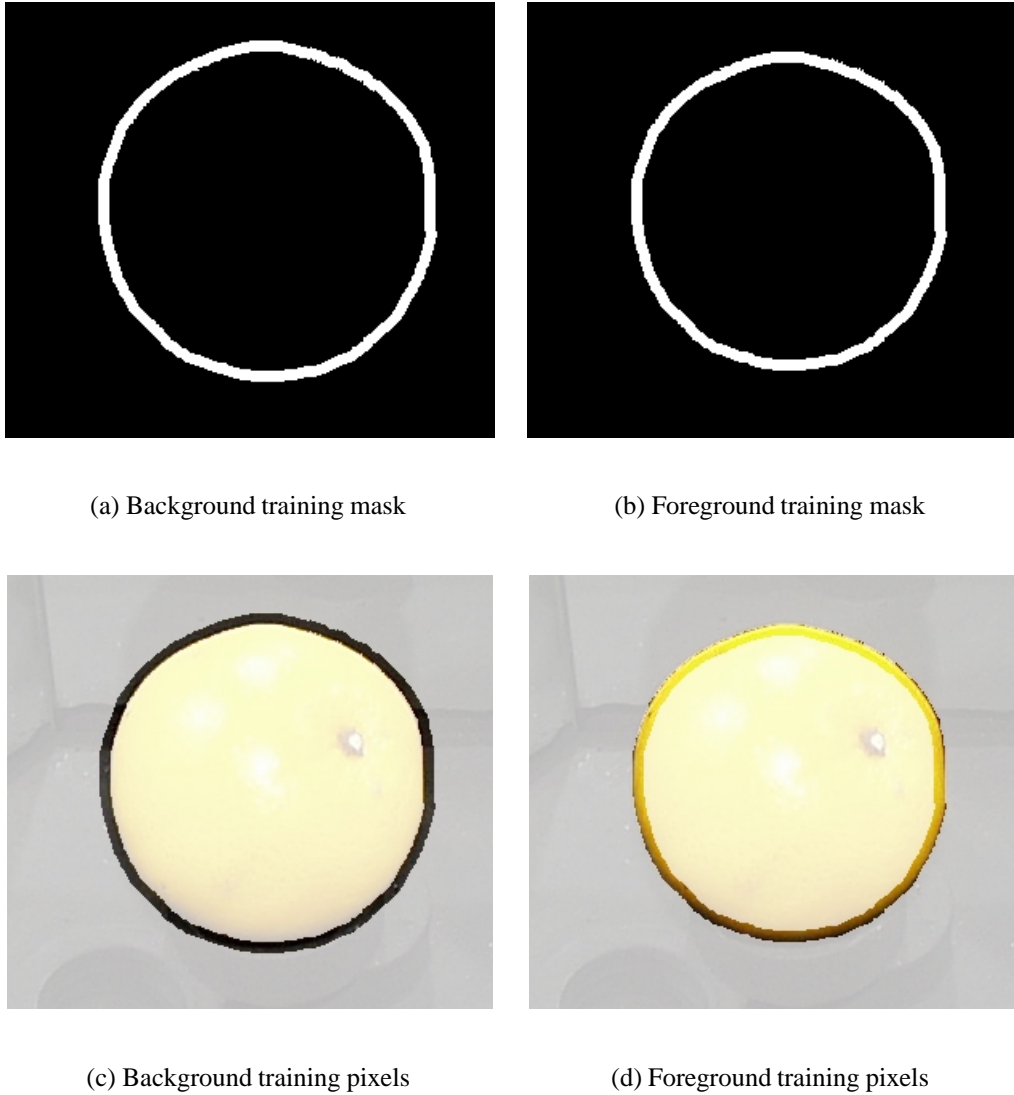


Figure 6.4: Mask images indicating the regions of Figure 6.3(a) to be used for target values in training as (a) the background and (b) the foreground; and showing the pixels used in training an MLP on recognition of (c) the background and (d) the foreground.

The mask region indicating the background region to be used was formed by dilating the original mask with a disk-shaped structuring element with a diameter of 15 pixels, and then subtracting the original mask from this binary image. This can be written mathematically as

$$B(x, y) = M(x, y) - M(x, y) \oplus S(x, y) \quad (6.3)$$

where  $B$  is the masked region identifying the portion of background to be used in training,  $M$  is the manually-segmented mask image and  $S$  is the disk-shaped structur-

ing element with a diameter of 15 pixels. Figure 6.4(a) shows the mask indicating the region of Figure 6.3(a) to be used for forming examples of background pixels for training the MLP. Figure 6.4(c) shows those background pixels explicitly against a diffuse version of the image for reference.

Similarly, the mask region indicating the foreground region to be used was formed by eroding the original mask using the disk-shaped structuring element of 15-pixel diameter, and then subtracting this binary image from the original mask, so that

$$F(x, y) = M(x, y) \ominus S(x, y) - M(x, y) \quad (6.4)$$

where  $F$  is the binary image which identifies the foreground region to be used for training. Figure 6.4(b) shows the mask that specifies the region of Figure 6.3(a) to be used for forming examples of foreground pixels for training the MLP. Figure 6.4(d) shows those foreground pixels explicitly, again against a diffuse version of the image. The structuring element used in the dilation and erosion of the mask was chosen to have a diameter of 15 pixels since it was felt that a smaller structuring element would unduly bias the MLP towards any manually misclassified pixels very close to the fruit object boundary (it is, of course, only close to the fruit object boundary that one would expect manually determined pixels to be misclassified).

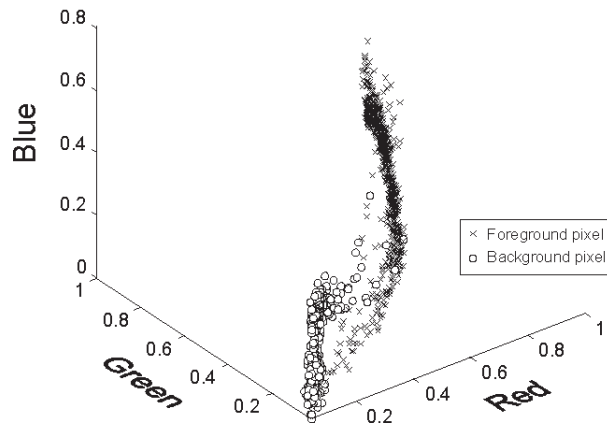


Figure 6.5: Plot of red, green and blue values for the training pixels selected from the orange image shown in Figure 6.3.

From the regions selected for background and foreground examples, 10 000 sample sets of RGB values (pixels) were randomly selected to be used as input data for train-



ing. The full masked region was not used because this would have greatly increased the time required to train the MLPs. The set of foreground RGB input vectors were each given a corresponding target value of +1; target values of -1 were used for the background. This provided a threshold value of zero, so that pixels which gave a positively-valued MLP output could be classified as foreground, and those giving a negatively-valued output could be classified as background.

Figure 6.5 shows a plot of the training set of RGB values extracted from the orange image shown in Figure 6.3(a). The plot gives an idea of the three-dimensional surface in RGB space that the MLP needs to use to divide the foreground and background values. Plots for pears, lemons and potatoes showed similar types of shapes, indicating that the same MLP structures (yet not of course weights, because the fruits are of different colours) could be used for all four fruit types.

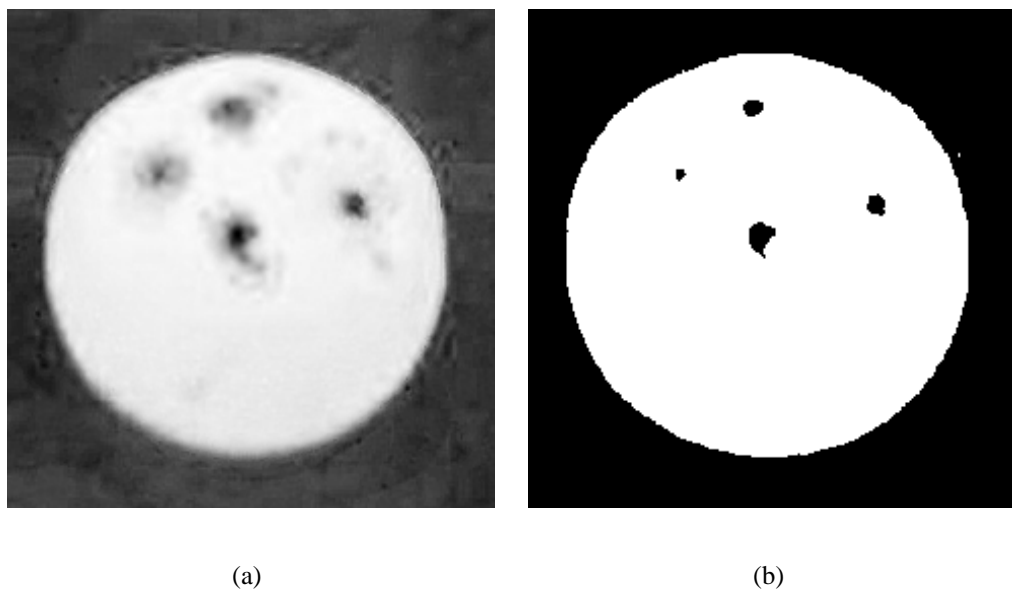


Figure 6.6: (a) The normalised MLP output for Figure 6.3(a); and (b) the thresholded binary image of (a).

The MLPs used had four hidden units with hyperbolic tangent activation functions and were each trained for 100 cycles using the quasi-Newton method to adjust the weight values. Figure 6.6(a) shows the result of using the trained MLP to calculate the grey-scale values of the combined band image from an input RGB image. The pixel values have been normalised to fall within the (0,255) range for display purposes. The grey-scale image (without normalisation) can be thresholded at zero to yield the binary image shown in Figure 6.6(b).

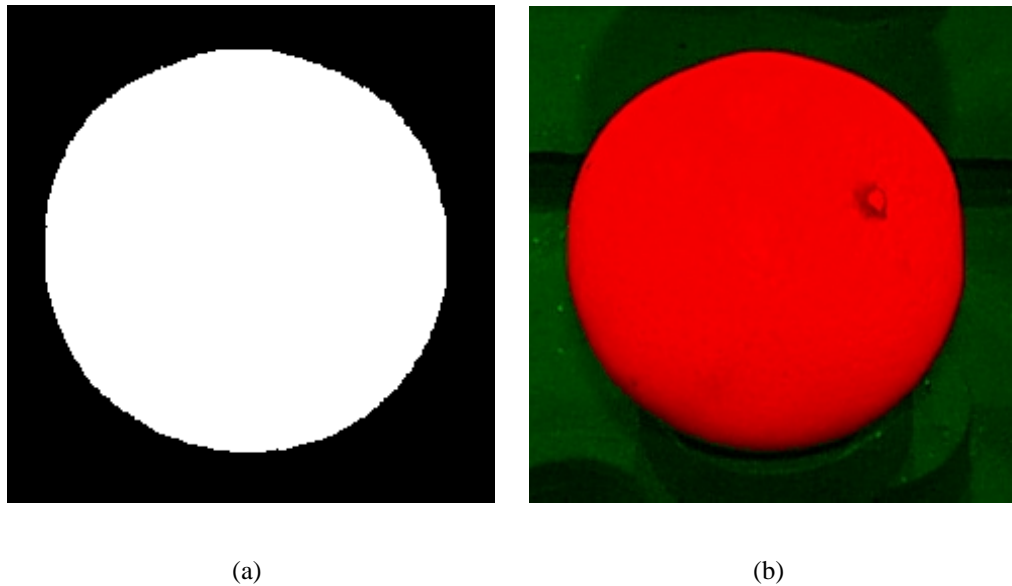


Figure 6.7: (a) The final binary image of the orange; (b) pseudo-colour image formed from original image and final binary image to show MLP segmentation accuracy.

The thresholded image may contain misclassified regions away from the boundary. In some cases this could be due to specks of dust within the mirror chamber; in others it is because some bright sections near the middle of the fruit are misclassified since pixels of such RGB values were not included in the training set. The bright sections are caused by the camera flash. The misclassified regions were quickly removed by firstly selecting the largest background region as the only background region (by setting the pixel values in all smaller-sized background regions to ‘ones’), and then from this new binary image selecting the largest foreground region as the only foreground region (by setting the pixel values in all smaller-sized foreground regions to ‘zeros’). The result of the removal of such unwanted regions is shown in Figure 6.7(a). These binary images form the final stage in the identification of fruit pixels within the fruit images.

In order to quantify the error on the MLP-based pixel classifier, a measure of error, such as the proportion of misclassified pixels, must be formed. However, determining which pixels had been misclassified would require the use of a manually-segmented image as the absolute arbiter of whether a particular pixel lies in the background or foreground. Since it was felt that the proportion of misclassified pixels in the manually-segmented image was comparable to the proportion of misclassified pixels in the image formed by the MLP, a quantitative evaluation of the error was not

attempted. Instead, the results were checked visually by means of pseudo-colour images such as the one shown in Figure 6.7(b). The pseudo-colour images were created by showing the red-band of the original image in the region indicated by the binary image to be foreground and showing the green-band in the background region. With such pseudo-colour images, it is easy to check whether or not the binary image indicates the fruit object pixels to be classified in a way that would be deemed correct by a human. It is also important to note that any errors in locating the absolute boundary of the fruit, to the extent that they are consistent from image to image, are relatively inconsequential in the statistical estimation of fruit volume (as opposed to calculating volume from exact real-world measurements). For instance, if dark brown pixels at the base of the fruit are misclassified as background pixels, it will not have an important effect on the final volume estimate, because the dark brown pixels will have been misclassified for all of the images, effectively changing the size and shape of the identified fruit objects in the same manner. Consistent treatment of images can be expected from MLPs with small numbers of hidden units as the functions represented by the MLPs are slow-varying and insensitive to most small changes in the input vectors.

## 6.4 Removal of the pear stalks

A segmentation stage was necessary to remove the stalk from the profile images of the pears. The stalk has a negligible contribution to the fruit volume (typically constituting less than 0.2% of the volume of pears) and would complicate the feature extraction process as it significantly alters the shape of the profile images.

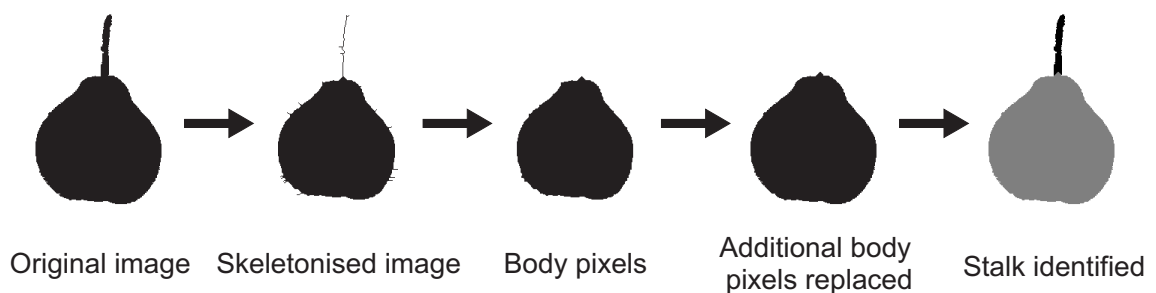


Figure 6.8: Pla and Juste's thinning-based method of stalk detection.



Figure 6.9: Close-up of stalk meeting body with thinning-based segmentation.

Pla and Juste [27] give a thinning-based approach to fruit stem identification which they describe as a modified thinning method used to characterise fruit stems from fruit profile images. The process is summarised in Figure 6.8. The first step is to form the skeletonised image [32] of the fruit profile. Each iteration of the skeletonisation algorithm removes a layer of pixels on the object boundary, so long as the removal does not cause the object to break apart.  $N$  iterations of the skeletonisation algorithm are applied to the fruit image, with  $N$  being a number dependent on the expected thickness of the stalks in the images. In the example shown in Figure 6.8,  $N = 6$ . From the skeletonised object, body pixels are identified. Body pixels are defined as pixels that would leave the object connected if removed. After the body pixels have been identified, the pixels removed in each iteration of the skeletonisation process are added back to the body, provided that they touch the body. The stalk is then defined as that portion of the original object that is not body.

The skeletonisation-based algorithm was found to be slow and to produce an undesirable small protrusion of the fruit body into the stalk object. An illustration of this can be seen in Figure 6.9. An algorithm based on binary morphological opening (described in Section 3.3) was developed as an alternative to the thinning-based method in an attempt to eliminate the undesirable protrusion of the fruit body into the stalk.

Figure 6.10 summarises the steps in this method. The difference image was formed by subtracting the original binary image from the same binary image opened with a circular disk structuring element that had 5% of the area of the original object. A structuring element of this size was found to have a diameter larger than the width of the stalks yet small enough to have little effect on the body of the fruit. The difference

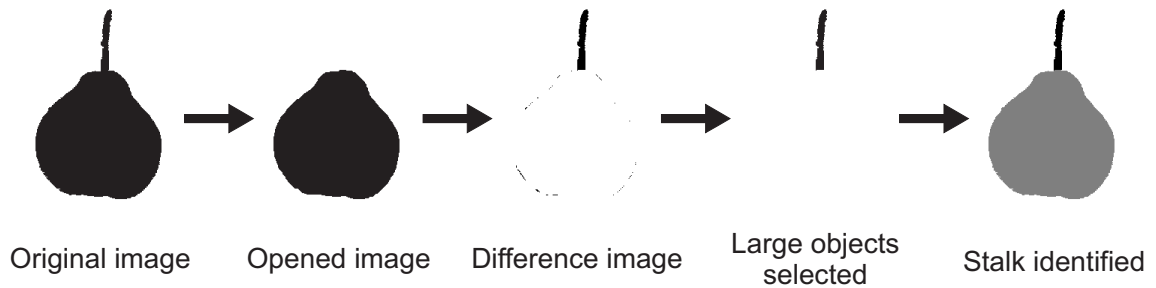


Figure 6.10: Binary morphology-based method of stalk detection.



Figure 6.11: Close-up of stalk meeting body with morphology-based segmentation.

image was region-labelled, and regions of area larger than 10 pixels were identified as stalk and removed from the original image, leaving only the body of the fruit. The results of this method were superior to those of the thinning-based algorithm, as is illustrated in Figure 6.11.

## 6.5 Selection of the feature set

A feature set was developed so as to provide a pool of features from which to draw for all of the volume estimation methods to be evaluated. The features consist partly of standard image analysis features and partly of features developed specifically for the solution to the volume estimation problem. Some of the features are redundant in the sense that they are functions of the other features, but these redundant features do have the potential to outperform the features that they are derived from when linear regression methods are used.

All of the features are relatively simple and quick to extract from the digital images. The motivation for the features to be rapidly extractable is that some fruit sorting systems require as many as ten images to be processed per second [38].

Unless otherwise specified, the features were selected because they were believed to show potential for giving an indication of the orientation of the fruit or of  $K = VA^{-\frac{3}{2}}$ . Such qualities are necessary in order to work together with a measure of profile image size to produce a volume estimate (see Chapter 2).

The fourteen features which were measured from each of the fruit profile boundaries of each of the four fruit types are listed below.

### 6.5.1 Area

The *area* of a binary object is found by counting the number of pixels in the binary object. The area of an object is a relatively noise-immune measure of object size, because every pixel in the object contributes towards the measurement.

After calculating the area of the profile object, the remaining features were calculated from a profile boundary (as described in Section 3.4) which was normalised to unit area. This was achieved by dividing the boundary co-ordinate points by the square root of the area. Using profile boundaries that were normalised to standard size, the features other than area were guaranteed to measure information about shape rather than information about size.

### 6.5.2 Perimeter

The *perimeter* of a boundary defined by a list of co-ordinates is the sum of the distances from each co-ordinate to the next. The perimeter measurement can become distorted by the fractal nature of some boundaries (fine boundary detail may exaggerate the size of the measurement). However, assuming relatively smooth boundaries, the perimeter is a possibly desirable feature on account of its simplicity and because of the speed with which it may be extracted from a profile boundary.

### 6.5.3 Aspect ratio

The *aspect ratio* of an object is the ratio of its calliper width to its calliper length. Because the object's major axis will in general be at a random orientation,  $\theta$ , with respect to the co-ordinate axes,  $(x, y)$ , its calliper dimensions are calculated by first establishing the value of  $\theta$  by means of the Hotelling transform. Once  $\theta$  is known, each point  $(x, y)$  on the object's boundary may be converted into a rotated frame  $(x', y')$  through the transformation

$$x'(\theta) = +x\cos(\theta) + y\sin(\theta) \quad (6.5)$$

$$y'(\theta) = -x\sin(\theta) + y\cos(\theta) \quad (6.6)$$

The difference between the maximum and minimum co-ordinate values of  $x'$  forms the object's *length*, while the difference between the maximum and minimum co-ordinate values of  $y'$  constitutes its *width*. Then

$$\text{aspect ratio} = \frac{\text{width}}{\text{length}} \quad (6.7)$$

Figure 6.12 illustrates measuring the width and length of an object in this way.

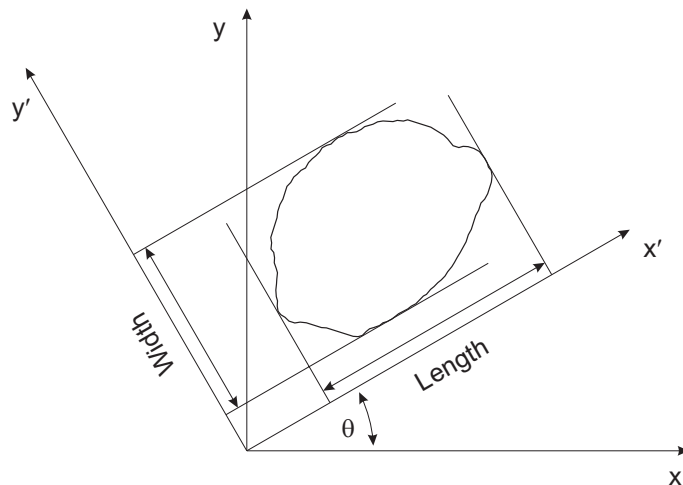


Figure 6.12: Measurement of object width and length.

### 6.5.4 Fourier ellipse sum

The *Fourier ellipse sum* was a feature developed especially for (implicitly) identifying pear orientation in this project. This feature gives an indication of the extent to which a profile view is elliptical and in so doing gives an indication of the extent to which a fruit such as the pear is being viewed side on. Since pears, when viewed from either the top or from the bottom, have essentially elliptical profiles, the Fourier descriptors of the boundary, other than  $G_{-1}$ ,  $G_0$  and  $G_1$ , are low-valued for pears seen end-on. A side view of a pear is a more complex shape than a circle or an ellipse, and therefore has at least some Fourier descriptors other than  $G_{-1}$ ,  $G_0$  and  $G_1$  that are of significant magnitude to account for this increase in complexity. The Fourier ellipse sum is a fraction formed from  $G_{-1}$  and  $G_1$  in the numerator and the sum of all of the Fourier descriptors other than  $G_{-1}$ ,  $G_0$  and  $G_1$  in the denominator. Thus,

$$\text{Fourier ellipse sum} = \frac{|G_1| + |G_{-1}|}{\sum_{i=2}^{N'} (|G_i| + |G_{-i}|)} \quad (6.8)$$

where  $N'$  is the highest order of Fourier descriptor that is ever expected to contribute significantly. In this project the value used was  $N' = 16$ .  $G_0$  is not included in the equation because it indicates nothing more than the location of the object's centroid relative to the origin of some co-ordinate system, and this is not relevant to the shape of the boundary.

To demonstrate the Fourier ellipse sum, Figure 6.13 shows four profiles of a cylindrically symmetrical pear-like object, viewed at angles of  $0^\circ$ ,  $30^\circ$ ,  $60^\circ$  and  $90^\circ$  to the side view. The three-dimensional object was computer-generated for the purpose of this illustration, and code was written to extract and sample its profile boundary at differing orientations. A plot of the Fourier descriptor magnitudes of each boundary is shown below each computer-generated profile. The vertical axis of all of the plots is to the same scale. Note how the magnitudes of the higher-order Fourier descriptors drop as the pear-like object is rotated from a side view towards the bottom view. The first negative Fourier descriptor is comparatively large in every case because it describes the main circle of the boundary (see Section 3.5). Although in the case of a simple closed boundary, the square of the first Fourier descriptor (the larger of  $G_{-1}$  and  $G_1$ ) is *roughly* proportional to the area of the object (as can be seen in Figure 6.13 where the objects are each scaled to have unit area and  $G_{-1}$ , the largest Fourier descriptor, appears to be roughly equal in each case), it is not necessarily *exactly* proportional



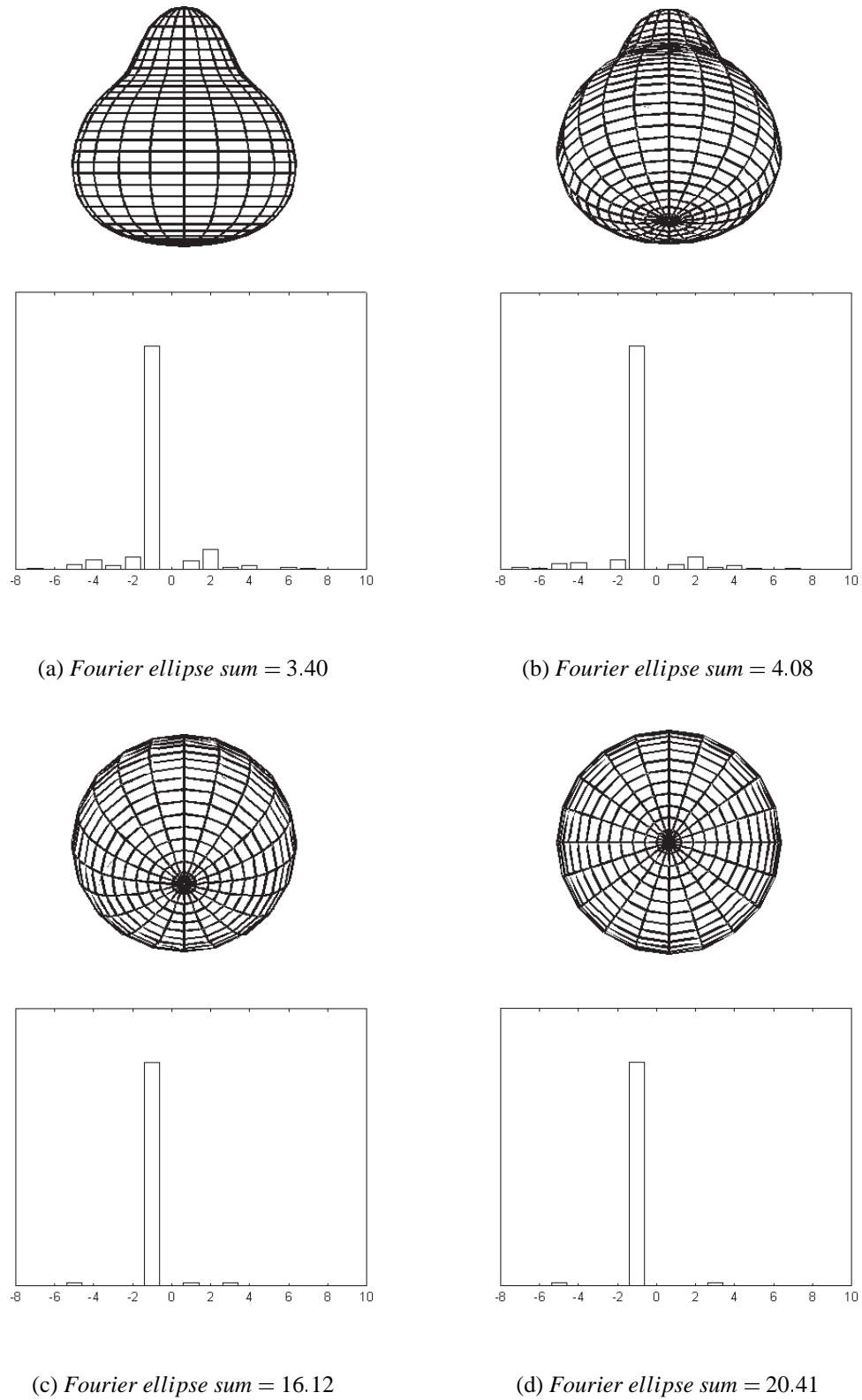


Figure 6.13: Computer generated profile images of (a) upright; (b) rotated 30°; (c) rotated 60°; (d) rotated 90° pear shapes with corresponding plots of Fourier descriptor magnitudes to illustrate the Fourier ellipse sum feature.

to the area of the object<sup>1</sup> (which is normalised to unity for all of the boundaries for measurement of all of the features other than area). The value of the Fourier ellipse sum, which can be used to give an indication of orientation for fruit with elliptical top and bottom views and more complex side views, is shown together with each of the profiles in Figure 6.13.

### 6.5.5 Fourier circle sum

The *Fourier circle sum* is an adaptation of the Fourier ellipse sum that was designed mainly to give an indication of lemon orientation. Since lemons appear roughly elliptical when viewed from the side and circular when viewed from the top, only the larger of  $G_{-1}$  and  $G_1$  was excluded from the denominator and used to form the numerator. The equation for the Fourier circle sum is

$$\text{Fourier circle sum} = \frac{\max(|G_1|, |G_{-1}|)}{\sum_{i=1}^{N'} (|G_i| + |G_{-i}|) - \max(|G_1|, |G_{-1}|)} \quad (6.9)$$

It may be noted that as a profile boundary becomes more circular, the denominator of this fraction tends to zero and the Fourier circle sum tends to infinity.

A lemon side view such as the one shown in Figure 6.14(a) has a low-valued Fourier circle sum both because of its elliptical-like shape and because of the bumps on the stem and calyx ends of the fruit, whereas the roughly circular profile view of the top or bottom of a lemon, such as is shown in Figure 6.14(b), has no Fourier descriptors of significant magnitude other than the larger of  $G_{-1}$  and  $G_1$ , and thus has a high-valued Fourier circle sum.

### 6.5.6 Mean diameter

The *mean diameter* was measured by Miller [23] to estimate the volume of oranges. Since Miller's method was to be evaluated on the data set of oranges, it was decided

---

<sup>1</sup>This fact was easily shown empirically by creating two objects of equal area and different shape. The first Fourier descriptor magnitudes of the two objects were observed to be not necessarily equal. However, the first Fourier descriptor is sometimes used as a convenient divisor for normalising shapes to standard size (in effect requiring the first Fourier descriptor of the shapes to have a magnitude of unity) [10]

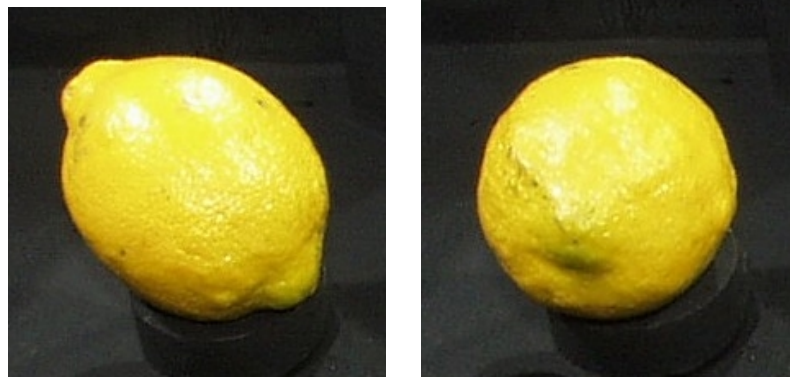
(a) *Fourier circle sum* = 3.02(b) *Fourier circle sum* = 8.84

Figure 6.14: (a) roughly side-on view of a lemon; (b) roughly end-on view of a lemon.

to include the feature in the pool of features from which to draw for each of the four fruit types. As explained in Chapter 2, the mean diameter is the average of eight diameter measurements (calliper style measurements) at successive incremental angles of  $22.5^\circ$ .

### 6.5.7 Roundness

An object's *roundness* [40] is the ratio of its area to its maximum diameter:

$$\text{roundness} = \frac{4 \cdot \text{area}}{\pi \cdot \text{maximum diameter}} \quad (6.10)$$

Since the objects were scaled to unit area, this measurement is simply inversely proportional to the maximum diameter of the object. The maximum diameter was estimated by using the maximum of the eight diameters measured to obtain the mean diameter feature described above.

The circular-like top and bottom view profile images of lemons and of pears may be expected to have a relatively low maximum diameter. This results in a high roundness value for such profiles.

### 6.5.8 Convexity

The *convexity* of an object is the ratio of its actual perimeter to its convex perimeter [32]. The convex perimeter is calculated from the convex hull of the object by deriving the convex hull as described in Section 3.7, and then by summing the distances between the points along the convex hull boundary.

$$\text{convexity} = \frac{\text{perimeter}}{\text{convex perimeter}} \quad (6.11)$$

An object with high convexity has one or more ‘bays’ around its perimeter, whereas an object whose convexity equals unity has a convex hull which conforms exactly to its outer perimeter.

### 6.5.9 Solidity

Similar to the convexity of an object, but in the higher dimension of area, is the *solidity* of an object [32]. The solidity of an object is the ratio of its area to its convex area (the area of its convex hull).

$$\text{solidity} = \frac{\text{area}}{\text{convex area}} \quad (6.12)$$

### 6.5.10 Extent

The *extent* of an object is the ratio of the object’s area to the area of its bounding rectangle.

$$\text{extent} = \frac{\text{area}}{\text{area of bounding rectangle}} \quad (6.13)$$

The length and width used to calculate the area of the bounding rectangle are the same calliper length and width used to form the aspect ratio (see Figure 6.12).

### 6.5.11 Energy of curvature

The *energy of curvature* [40] of an object boundary is an indicator of the average amount of curvature in the object boundary. The curvature of a boundary describes changes in direction and is defined in terms of the second derivative of the co-ordinates.

For points  $x_i$  and  $y_i$  on the object boundary, the curvature at that position is defined numerically as [40]

$$C_i = \sqrt{(2x_i - x_{i-1} - x_{i+1})^2 + (2y_i - y_{i-1} - y_{i+1})^2} \quad (6.14)$$

The average value of the square of this curvature, normalised to the boundary length  $P$ , gives the energy of curvature:

$$\text{energy of curvature} = \frac{1}{P} \sum_{i=1}^N C_i^2 \quad (6.15)$$

where  $N$  is the number of boundary points and the  $x$ - and  $y$ -indices should be understood to wrap around the  $N^{\text{th}}$  boundary point in the summation.

### 6.5.12 Mean distance

The *mean distance* feature is the mean of the distances from each boundary point to the centroid of the object:

$$\text{mean distance} = \frac{1}{N} \sum_{i=1}^N \sqrt{(x_i - x_{\text{centroid}})^2 + (y_i - y_{\text{centroid}})^2} \quad (6.16)$$

### 6.5.13 Variance of distance

Similarly, the *variance of distance* is the variance of the distances from each boundary point to the centroid of the object:

$$\text{variance of distance} = \frac{1}{N} \sum_{i=1}^N \left( \sqrt{(x_i - x_{\text{centroid}})^2 + (y_i - y_{\text{centroid}})^2} - \text{mean distance} \right)^2 \quad (6.17)$$

A circle would have zero variance of distance, whereas a very elongated ellipse would return a high value for this feature.

### 6.5.14 Circularity

The *circularity* is the dimensionless ratio of the square of perimeter to area:

$$\text{circularity} = \frac{\text{perimeter}^2}{\text{area}} \quad (6.18)$$

Although it is a function of perimeter, which is already one of the pool of features, and of area, which is normalised to unity, circularity is a commonly-used measure in image processing [40]; and a non-linear transform of an existing feature, although it adds no further information, does have the potential to out-perform the original feature when using linear methods.

## 6.6 Feature computational complexity

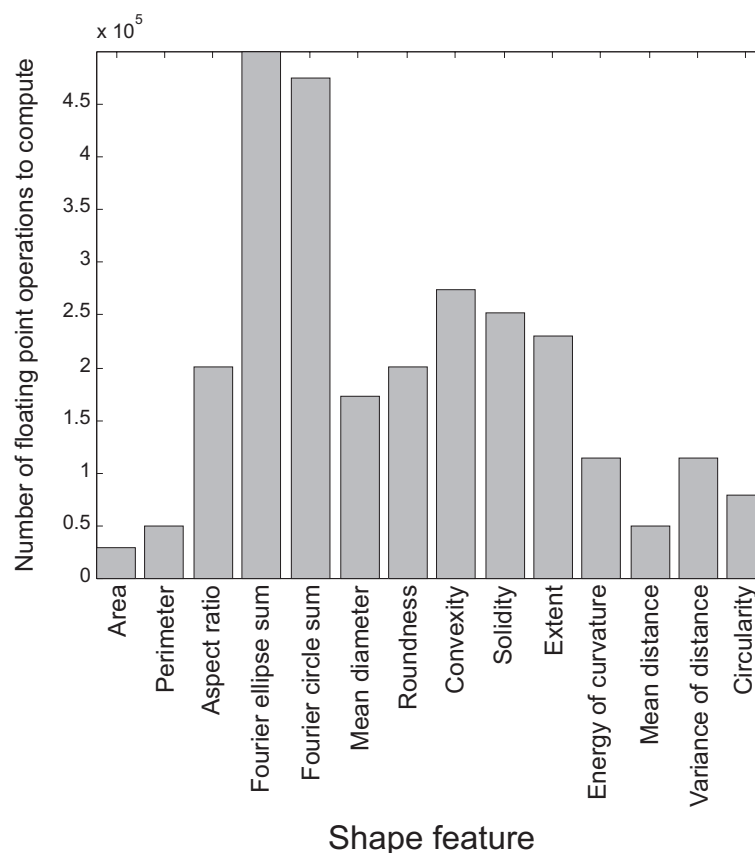


Figure 6.15: Mean number of floating point operations for computing shape features.

Figure 6.15 shows the average number of floating point operations that were required for the calculation of each of the shape features described above. Elementary functions such as addition, subtraction, multiplication or division of a pair of real numbers correspond to a single floating point operation (flop). The number of flops was counted using the MATLAB software in which all of the routines in this project were written.

Running on a 200MHz Pentium II processor on a Windows 95 platform, a flop typically corresponds to about  $1.4\mu\text{s}$  which means that many of the routines take longer than 100ms. However, in an industrial context with faster processors and with dedicated compiled software (as opposed to the interpreted code which was written for the investigations of this project), it is apparent from these numbers that implementations of many of the feature extraction routines could offer feasibly rapid volume estimation methods. It may be noted that the number of flops required for the computation of the fourteen shape features considered varies from a minimum of about 30 000 for the calculation of area to a maximum of about 500 000 for the determination of the Fourier ellipse sum of a single fruit profile.

Although it is not the goal of this project to optimise the speed of the feature extraction process, this information is of use for rough comparisons of the likely times of extraction of the shape features. It is the extraction of these features which will add the most significant overhead to existing fruit sorting systems that already implement segmentation. The speed of extraction of a feature can be taken into account when selecting the subset of features to be used for a particular function approximation method.

# Chapter 7

## Volume Estimation from Features

This chapter presents and discusses the results of the volume estimation methods developed in this project. Firstly, the methods of Miller and Marchant are revisited, and their performance results are determined in terms of RMS error by applying them to data sets of profile images. These performances are then compared with those of some alternative methods that were developed in this project. Next, the performances and associated confidences in the linear methods developed are presented and discussed. It is then shown how the interpretation of the results can be improved by using a technique of volume redistribution on the existing data sets prior to the implementation of the volume estimation methods. The results of volume estimation using the average of several estimations after volume redistribution are also presented here. Finally, multi-layer perceptron methods are introduced and these are compared with the linear methods for volume estimations based on features from both single and multiple images. The results obtained indicate that volume estimation with RMS errors of less than  $10\text{cm}^3$  (and often considerably less) is certainly possible if employed on fruit types that have some degree of symmetry. The tables of all of the results presented in this chapter are repeated for convenience of comparison in Appendix B.

### 7.1 Testing Miller's method for oranges

Miller's method [23] of estimating orange volume from profile images, as described in Chapter 2, was tested on the data set of oranges. The method consists of averaging eight diameters taken at incremental angles of  $22.5^\circ$  from the profile image of the



orange. This average diameter was used to estimate the volume of the orange by approximating the shape of the orange to that of a sphere. A training set of oranges was used to determine the appropriate scaling factor.

After applying the jackknifing process on the data set of  $24 \times 31 = 744$  images of 31 oranges, 31 mean-squared errors (MSEs), each calculated from the volume estimations of 24 images of each fruit, were obtained. These MSEs were obtained by removing from the data the features associated with the profile images of one orange, and then using the remaining features and corresponding target values as a training set. The volume estimation function established by the training stage was used to produce volume estimates from the profile images associated with the removed fruit. This process was repeated 31 times with the profile images corresponding to each fruit being removed on one occasion each.

The results of using Miller's method for volume estimation on the data set of oranges, calculated using the methods introduced in Chapter 4, were as follows:

RMS percentage error:	5.3%
RMS error (RMSE):	12.5cm <sup>3</sup>
95% confidence interval:	9.7cm <sup>3</sup> < RMSE < 14.7cm <sup>3</sup>
Mean of MSEs:	155.1cm <sup>6</sup>
Standard deviation of MSEs:	171.9cm <sup>6</sup>

The RMS percentage error is a useful statistic for a quick interpretation of the results. Nevertheless, it is important to note that this statistic cannot directly be derived from the RMS error (RMSE). The RMSE tends to be larger in fruit of larger volume, while the RMS percentage error does not. However, it is the MSE (and thus the RMSE) which is minimised by the function approximators used in this project, and not the RMS percentage error. So, although the RMS percentage error is closely linked to the RMSE, it should be seen as the quicker, rather than the better interpretation of the performance of the function approximator.

The 95% confidence interval indicates that we can be 95% sure that Miller's method would produce an RMSE in the 9.7cm<sup>3</sup> to 14.7cm<sup>3</sup> range in the long run (after estimating the volumes of many oranges from the same population using Miller's method).

Another method of calculating the orange volume, also using the approximation that oranges are spherical, was tested. Instead of using an estimate of the diameter of

the orange in the profile image, the area,  $A$ , of the orange in the profile image was used. Assuming sphericity, and less importantly that the effects of projection can be neglected, the volume,  $V$ , of the orange can be calculated from the circular area,  $A$ , of its profile image using

$$V = \frac{4}{3\sqrt{\pi}}A^{\frac{3}{2}} \quad (7.1)$$

which can be rewritten

$$V \propto A^{\frac{3}{2}} \quad (7.2)$$

and the appropriate scaling factor can be found using a training set of profile images and corresponding target volumes.

This method produced the following results on the data set of oranges:

RMS percentage error:	3.7%
RMS error (RMSE):	8.7cm <sup>3</sup>
95% confidence interval:	7.6cm <sup>3</sup> < RMSE < 9.7cm <sup>3</sup>
Mean of MSEs:	75.3cm <sup>6</sup>
Standard deviation of MSEs:	51.1cm <sup>6</sup>

As can be seen from these results, this method provides better estimates than Miller's method. The certainty that this method is better than Miller's method can be estimated using the  $t$  statistic introduced in Chapter 4:

$$t = \frac{\overline{MSE}_1 - \overline{MSE}_2}{\sqrt{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}} \sqrt{\frac{n_1 n_2 (n_1 + n_2 - 2)}{n_1 + n_2}} \quad (7.3)$$

$$= \frac{155.1 - 75.3}{\sqrt{(31 - 1)171.9^2 + (31 - 1)51.1^2}} \sqrt{\frac{31 \cdot 31(31 + 31 - 2)}{31 + 31}} \quad (7.4)$$

$$= 2.478 \quad (7.5)$$

where the  $n_1 = 31$  elements of  $MSE_1$  are the mean-squared errors from jackknife testing Miller's method and the  $n_2 = 31$  elements of  $MSE_2$  are the mean-squared errors from jackknife testing the alternative method. The sample standard deviations,  $s_1$  and  $s_2$ , are the sample standard deviations of  $MSE_1$  and  $MSE_2$  respectively. Using a lookup table, the  $t = 2.478$  statistic with  $n_1 + n_2 - 2 = 60$  degrees of freedom corresponds to a 99.3% chance that the alternative method produces better results than Miller's method. This means that we can be 99.3% certain that the fact that the alternative

method produced a lower mean RMS error than Miller's method on the data set is not due to chance, and that it would continue to produce lower mean RMS errors in the long run if implemented or tested on new data.

The better results from the alternative method can be attributed to the fact that the method is more immune to noise than Miller's method. By averaging eight diameters, Miller moves in the direction of removing the effects of noise. However, using the area of the orange profile image incorporates information about the entire object and is less sensitive to any local variations in the boundary of the fruit profile than Miller's method.

The alternative method has an added advantage over Miller's method in that the extraction of the area feature is quicker and less complex than the extraction of the mean diameter feature required for Miller's method.

## 7.2 Testing Marchant's method for potatoes

As described in Chapter 2, Marchant [21] implicitly estimated potato volume in a machine vision sorting system by measuring the lengths and areas of profile images. The method used by Marchant was to divide the square of the area by the length of the profile image, and then to use an appropriate scaling factor determined by a training set of potatoes. Marchant's method gave the following results upon jackknife testing, using the data set of  $24 \times 20 = 480$  images of 20 potatoes created for this project:

RMS percentage error:	12.3%
RMS error (RMSE):	$30.7\text{cm}^3$
95% confidence interval:	$21.1\text{cm}^3 < \text{RMSE} < 37.9\text{cm}^3$
Mean of MSEs:	$941.2\text{cm}^6$
Standard deviation of MSEs:	$1061.9\text{cm}^6$

Marchant's method can be compared with a simpler method that uses only an area feature. This simplification is motivated by the fact that potatoes are so irregularly-shaped that there is unlikely to be any identifiable general relationship between the potato volumes and any of the profile features identified for volume estimation (apart, possibly, from the area). Making the assumption that area is approximately linearly

related to volume on the relevant portion of the volume-area curve resulted in a simple method which assumed potato volume to be proportional to potato profile image area. The scaling factor was determined using a training set of potato images and corresponding target volumes, giving the following results:

RMS percentage error:	12.3%
RMS error (RMSE):	31.3cm <sup>3</sup>
95% confidence interval:	20.1cm <sup>3</sup> < RMSE < 38.9cm <sup>3</sup>
Mean of MSEs:	976.3cm <sup>6</sup>
Standard deviation of MSEs:	1152.3cm <sup>6</sup>

Although the resultant RMS errors were very slightly greater than those obtained using Marchant's more complex method, the  $t$  statistic shows that there is only a 54.0% probability that the difference between the means is not due to chance. The very small improvement in RMSE therefore suggests that the extraction of the length feature in addition to the area feature could be seen as an overkill. Marchant relied on the mechanics of the conveyor system to align the potatoes for measurement of length, and in this project the potato images were aligned for measurement of length using the Hotelling transform. In both cases, the extra computational time and complexity provides minimal returns in terms of lower RMSE.

### 7.3 Linear regression methods

As seen in the previous section, a very simple volume estimate can be made using the area of the fruit profile image to estimate volume. However, this effectively neglects the fact that the shape and area of the fruit profile images may vary depending on the orientation of the fruit and on differences in the shape (as opposed to just size) of one fruit from another. This is particularly true in roughly cylindrically symmetrical fruit such as pears and lemons. One would assume that, for a largely spherically symmetrical fruit such as the orange, using the area feature alone would give a reasonable volume estimate, since orange profiles differ little in shape either from fruit to fruit or between different orientations; and that for more complex fruit such as pears this would be a good *first approximation*, which could be improved by considering additional shape features.

Although the relationship between profile area and volume is not linear, the range of orange profile images and corresponding volumes is small enough to lie on what could be considered an almost linear portion of a nonlinear function. This can be

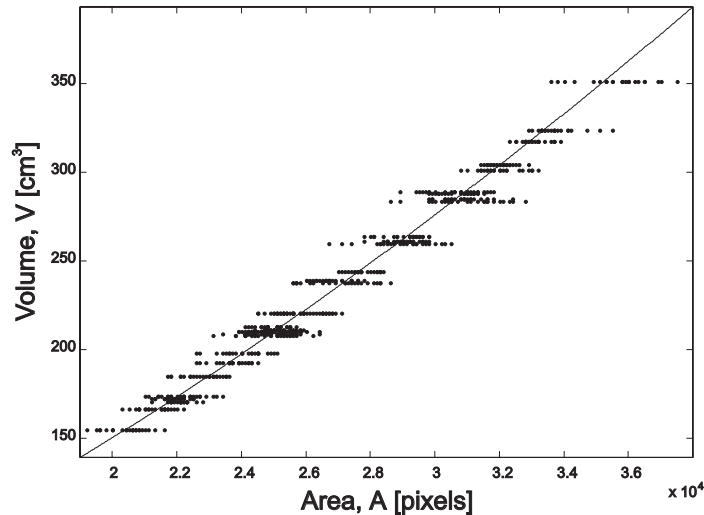


Figure 7.1: Scatter plot of volume versus area for the data set of orange images, showing the  $V = KA^{\frac{3}{2}}$  curve for ideal spheres.

seen from observing a plot of profile image area versus volume for oranges as shown in Figure 7.1. In this scatter plot, the areas (in pixels) of the 744 orange images are shown plotted against their corresponding volumes. The 31 horizontal sets of 24 dots correspond to the 31 oranges whose volumes were measured. The spread of these horizontal lines is an indication of the extent to which the oranges are not spherical, and of how the profile area varies according to the orientation of the fruit with respect to the camera. Figure 7.1 also shows an almost straight section of the  $V = KA^{\frac{3}{2}}$  curve, upon which all of the data points would be expected to lie if the oranges were perfectly spherical and if the camera were at infinity.

Standard linear regression was used to predict volume from area measurements. The results are shown in Table 7.1 where  $n$  is the number of fruit and corresponding number of MSEs, RMS%E is the square root of the mean of the squared percentage errors, RMSE is the square root of the mean of the MSEs, LO95%CI and UP95%CI are the lower and the upper bounds of the 95% confidence interval for RMSE respectively,  $\overline{\text{MSE}}$  is the mean of the  $n$  MSEs and  $\text{STD}(\text{MSE})$  is the sample standard deviation of the  $n$  MSEs, all from the results of the jackknife testing. The use of the mean volume

Estimation Method	Statistic	Orange ( $n = 31$ )	Pear ( $n = 64$ )	Lemon ( $n = 23$ )	Potato ( $n = 20$ )
Mean volume	RMS%E [%]	23.2	4.1	27.6	18.9
	RMSE [ $\text{cm}^3$ ]	<b>51.7</b>	<b>7.7</b>	<b>58.6</b>	<b>47.3</b>
	LO95%CI [ $\text{cm}^3$ ]	40.7	5.6	47.4	19.0
	UP95%CI [ $\text{cm}^3$ ]	60.7	9.4	68.0	64.1
	$\overline{\text{MSE}}$ [ $\text{cm}^6$ ]	2674.1	59.6	3438.0	2236.6
	STD(MSE) [ $\text{cm}^6$ ]	2881.4	114.7	2749.1	4005.5
Area	RMS%E [%]	3.7	3.7	8.4	12.3
	RMSE [ $\text{cm}^3$ ]	<b>8.6</b>	<b>7.0</b>	<b>18.5</b>	<b>31.3</b>
	LO95%CI [ $\text{cm}^3$ ]	7.6	5.5	16.5	20.9
	UP95%CI [ $\text{cm}^3$ ]	9.5	8.2	20.3	38.9
	$\overline{\text{MSE}}$ [ $\text{cm}^6$ ]	73.6	49.0	342.6	976.3
	STD(MSE) [ $\text{cm}^6$ ]	46.5	76.1	161.1	1152.3
Area $^{\frac{3}{2}}$	RMS%E [%]	3.7	3.7	8.2	12.1
	RMSE [ $\text{cm}^3$ ]	<b>8.7</b>	<b>7.0</b>	<b>18.4</b>	<b>30.7</b>
	LO95%CI [ $\text{cm}^3$ ]	7.6	5.5	16.2	21.1
	UP95%CI [ $\text{cm}^3$ ]	9.7	8.2	20.3	37.9
	$\overline{\text{MSE}}$ [ $\text{cm}^6$ ]	75.3	49.1	338.0	939.6
	STD(MSE) [ $\text{cm}^6$ ]	51.1	76.8	173.3	1057.4

Table 7.1: Errors on volume estimates using mean volume, area and area $^{\frac{3}{2}}$ .

of the training set as a predictor was also tested, and the results are shown in the first rows of Table 7.1. This predictor makes no use of the profile images, and bases its prediction entirely on the *a priori* knowledge of the distribution of volumes from the training set. A high error on the volume prediction using the mean volume simply indicates that there is a large spread of fruit volumes; likewise, a small error indicates a small spread. This highlights the fact that direct comparisons cannot be made between fruit types, since the data sets of the fruit types consist of different volume distributions. From this table, it is only meaningful to look for improvements in the columns (due to more elaborate volume estimation methods) for the particular fruit type with its particular distribution. Later, a normalisation process is presented which allows one to compare the suitability of volume estimation processes between the columns, for fruit of different intrinsic shapes and volume distributions.

In the case of the pear, the volume spread is so narrow that there is hardly any evidence of improvement from the mean volume predictor when an area predictor is used. However, for the other fruit sets, which have wider distributions of volumes, Table 7.1 shows a definite improvement in the volume estimates. The use of  $\text{area}^{\frac{3}{2}}$  in place of area as a predictor shows little improvement, because (as mentioned before) the relevant portion of the area-volume curve is approximately linear.

A simple linear regression technique can be used to improve the volume estimates by taking the shape of the profile image into account. To do this the method uses the equation

$$V = KA^{\frac{3}{2}} \quad (7.6)$$

$$\Rightarrow K = \frac{V}{A^{\frac{3}{2}}} \quad (7.7)$$

A shape feature is used to predict  $K$  and thus implicitly to take the orientation of the fruit into account. This is because the same fruit expose different profile areas at different orientations. Different fruit types have inherently different shapes, so different shape features are better predictors of  $K$  for different fruit types.

Table 7.2 shows the absolute value of the correlation coefficient of  $K$  and the various measured shape features calculated over all of the available fruit images for each fruit type. These correlation coefficients were used to select the best predictors of  $K$  for each fruit type. The best linear predictors of  $K$  for the different fruit types were thus:

Oranges: Mean diameter  
 Pears: Mean diameter  
 Lemons: Aspect ratio  
 Potatoes: Convexity

As might be expected, the correlation coefficients corresponding to the potatoes and to the oranges are low, indicating that prediction of  $K$  is unlikely to improve a volume estimate for the fruit. This is because oranges are approximately spherical and do not differ in profile between different orientations. Even if the orientation of the oranges could be predicted, it would be of little use, since the exposed profile area of an orange does not systematically differ with orientation. Although the potatoes expose very different profile areas at different orientations, this is largely unpredictable because potatoes are very irregularly-shaped fruit.

Shape Feature	Correlation coefficient			
	Orange	Pear	Lemon	Potato
Perimeter	0.348	0.559	0.394	0.212
Aspect Ratio	0.119	0.572	<b>0.827</b>	0.114
Fourier ellipse sum	0.299	0.604	0.586	0.127
Fourier circle sum	0.318	0.662	0.775	0.012
Mean diameter	<b>0.378</b>	<b>0.680</b>	0.753	0.083
Roundness	0.291	0.639	0.826	0.113
Convexity	0.337	0.119	0.071	<b>0.309</b>
Solidity	0.371	0.585	0.540	0.063
Extent	0.220	0.579	0.594	0.022
Energy of curvature	0.367	0.383	0.347	0.181
Mean distance	0.002	0.525	0.667	0.178
Variance of distance	0.156	0.651	0.718	0.075
Circularity	0.345	0.557	0.392	0.207

Table 7.2: Correlation coefficient of  $K$  and shape features for different fruit types. The best correlations for each fruit type are shown in bold.

The pears and lemons, on the other hand, are both regularly-shaped approximately cylindrically symmetrical fruit. This makes prediction of  $K$  from shape features more feasible than with oranges and potatoes. Figure 7.2 shows plots of three shape features versus  $K$  for lemons. The correlation and usefulness of these features as predictors of  $K$  can be seen from these plots. The plots show that although a linear regression line could be fitted to the data points to predict  $K$ , there is some scope in making use of non-linear methods (MLPs) to make better predictions of  $K$  by taking into account the underlying non-linearity of these plots.

Although pears have a more distinctive variation in shape with orientation than lemons, the differences in projected profile areas for different orientations are less in the case of pears than they are for lemons. For this reason, a better prediction of  $K$  is possible for lemons than is possible for pears. Note that the values in Table 7.2 *can* be compared meaningfully across the different fruit types. This is because  $K$  is invariant to the distribution of volumes, because it is a measure of shape.



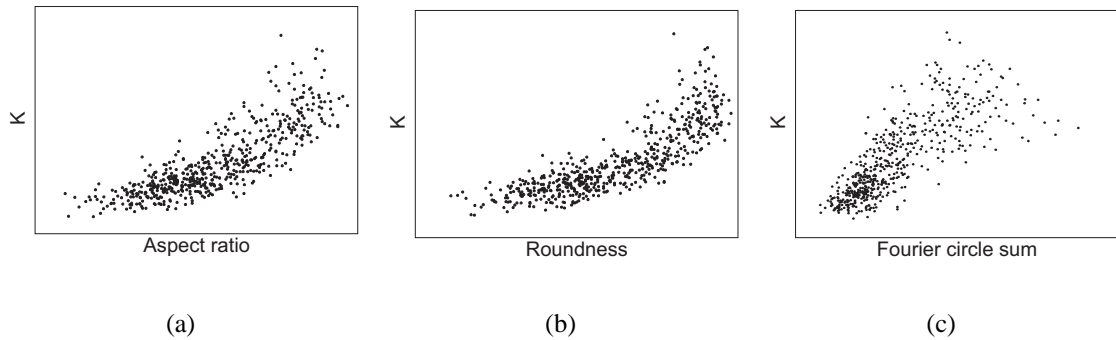


Figure 7.2: Plots of (a) aspect ratio; (b) roundness; (c) Fourier circle sum features versus  $K$  for lemons.

From Table 7.2, we can also note that the Fourier ellipse sum and Fourier circle sum features, which are shape features developed specifically for this task, performed reasonably, yet were outperformed by simpler standard image processing features. The Fourier ellipse sum was outperformed by the Fourier circle sum in the case of lemons, yet still showed reasonable correlation. This suggests that although the change from a circular profile to an increasingly less circular elliptical profile is an important indicator of the amount of area that a fruit is projecting, the change in boundary bumpiness (as indicated by the Fourier ellipse sum) due to the bumps at the stem and calyx end of the lemon also indicate the amount of projected area to an extent.

Ironically, the mean diameter feature, which was somewhat opportunistically included in the set of measured features as a byproduct of Miller's method, turned out to be a good predictor of  $K$  for oranges, pears and lemons. This does make sense if one considers the fact that an ellipse will have a greater mean diameter than a circle of the same area. So, a fruit which is manifesting a large  $K$  value in an elliptical side view would have a corresponding high mean diameter feature, compared with the low  $K$  value manifested in a circular top view which would have a corresponding low mean diameter feature.

The errors obtained for each of the fruit data sets when using area and the best shape feature to predict  $K$  are shown in Table 7.3. The volume estimates were formed using Equation (7.6). These results can be compared with the results shown in Table 7.1, which used even simpler area-based methods to estimate volume. The lemon data set shows the most dramatic improvement, the RMSE dropping from  $18.5\text{cm}^3$  in the case of the area-based predictor to  $11.9\text{cm}^3$  with the inclusion of the  $K$ -estimator. Using

Estimation Method	Statistic	Orange ( $n = 31$ )	Pear ( $n = 64$ )	Lemon ( $n = 23$ )	Potato ( $n = 20$ )
Area with $K$ -estimator	RMS%E [%]	3.4	5.4	5.0	16.2
	RMSE [ $\text{cm}^3$ ]	<b>8.4</b>	<b>10.3</b>	<b>11.9</b>	<b>40.5</b>
	LO95%CI [ $\text{cm}^3$ ]	7.2	9.5	9.1	24.0
	UP95%CI [ $\text{cm}^3$ ]	9.3	11.1	14.2	52.0
	$\overline{\text{MSE}}$ [ $\text{cm}^6$ ]	69.7	106.8	141.3	1638.4
	STD(MSE) [ $\text{cm}^6$ ]	49.5	67.1	137.3	2273.0

Table 7.3: Errors on volume estimates using the best predictor of  $K$  for each fruit.

the  $\overline{\text{MSE}}$  and STD(MSE) to form a  $t$ -statistic ( $t = 4.56$ ) shows that we can be well over 99.9% certain that the  $K$ -estimator method produces a lower RMSE.

It is interesting to note that, in the case of pears, using a  $K$  predictor increases the error even though from Table 7.2 it can be seen that there is a reasonable correlation between  $K$  and mean diameter. This increase of error is because of the small spread in volumes of the pears in the data set, rather than having anything to do with the intrinsic shape of pears. In the next section, it is explained how the volumes can be redistributed to obtain more meaningful interpretations of volume estimation methods and to achieve a direct comparison of errors between different fruit types.

## 7.4 Redistributing volumes for purposes of error comparison

The issues in understanding the effect that the volume distribution of the fruit has on the performance of a volume predictor were discussed in Section 2.3. In this section, a method of altering the data, so that more meaningful comparisons can be made, is now developed.

All of the shape features apart from area are invariant to size, since they were measured from profile images normalised to unit area. Since none of the shape features shows any significant correlation with volume (as opposed to  $K$ ), and since the effects of projection are small, any set of features from a profile image of a corresponding piece of fruit can be easily mapped to a new set of features corresponding to the same piece

of fruit at the same orientation, but with a different volume. Under this transformation, all of the features apart from area,  $A$ , remain unchanged. To change a known fruit volume,  $V_{old}$ , to a required new volume,  $V_{new}$ , the known area,  $A_{old}$ , is updated to a new area feature,  $A_{new}$ , using

$$A_{new} = A_{old} \left( \frac{V_{new}}{V_{old}} \right)^{\frac{2}{3}} \quad (7.8)$$

By specifying a new set of volumes, it is possible to obtain a very good approximation to what a data set of features from a fruit type with a specific volume distribution would be.

Using Equation (7.8) to calculate new area values, and leaving the remaining original features unchanged, a simulated data set for each of the four fruit types was formed. The new volume values were obtained by randomly sampling a normally distributed PDF of mean  $200\text{cm}^3$  and standard deviation  $50\text{cm}^3$ . Such a distribution is typical of one that might exist in a batch of oranges or pears sorted in a packing house [33]. The actual volumes of the redistributed volume data sets differed between fruit types because the numbers of fruit for each data set differed. The effects of chance differences in actual volume distribution amongst the four redistributed data sets was reduced by selecting a sample set of the PDF having an actual mean and sample standard deviation matching those of the original PDF to less than 0.1%.

The volume estimation results of using linear methods to estimate fruit volume from a single image on the original data (Orig.) are summarised in Table 7.4 together with the results on the redistributed data (Redist.). Note that for the area with  $K$ -estimator method, the RMS percentage errors of the redistributed volume data sets are often close to those of the original data sets. This is because the *a priori* knowledge of the volume distribution gained from the training set becomes of less importance as more information from the profile images is utilised.

The inclusion of a  $K$ -estimator provides noticeable improvements in the RMS error in the case of the lemons and the pears. In the case of the pears, the volume redistribution is necessary to illustrate this. This is because, as stated in the previous section, the original data set with its tight volume distribution actually shows a decrease in performance when a  $K$ -estimator is included in addition to the area feature. In the case of the original data set, the area feature is a better predictor of  $K$  than any other shape feature: the correlation coefficient of area as a predictor of  $K$  is 0.872. As the volumes of the population of fruit become spread out, area becomes uncorrelated with  $K$ , while the

Estimation Method	Statistic	Orange ( $n = 31$ )		Pear ( $n = 64$ )		Lemon ( $n = 23$ )		Potato ( $n = 20$ )	
		Orig.	Redist.	Orig.	Redist.	Orig.	Redist.	Orig.	Redist.
Mean volume	RMS%E [%]	23.2	30.4	4.1	33.5	27.6	34.3	18.9	30.7
	RMSE [ $\text{cm}^3$ ]	51.7	<b>49.4</b>	7.7	<b>49.7</b>	58.6	<b>48.9</b>	47.3	<b>48.7</b>
	LO95%CI [ $\text{cm}^3$ ]	40.7	32.7	5.6	40.3	47.4	29.2	19.0	35.6
	UP95%CI [ $\text{cm}^3$ ]	60.7	61.6	9.4	57.6	68.0	62.6	64.1	58.9
	MSE [ $\text{cm}^6$ ]	2674.1	2435.5	59.6	2469.3	3438.0	2386.6	2236.6	2370.6
	STD(MSE) [ $\text{cm}^6$ ]	2881.4	3876.1	114.7	3456.9	2749.1	3754.1	4005.5	2353.9
Area	RMS%E [%]	3.7	4.0	3.7	7.2	8.4	8.5	12.3	14.3
	RMSE [ $\text{cm}^3$ ]	8.6	<b>8.0</b>	7.0	<b>14.8</b>	18.5	<b>16.3</b>	31.3	<b>27.0</b>
	LO95%CI [ $\text{cm}^3$ ]	7.6	6.8	5.5	13.0	16.5	14.5	20.9	24.0
	UP95%CI [ $\text{cm}^3$ ]	9.5	9.1	8.2	16.5	20.3	17.9	38.9	29.7
	MSE [ $\text{cm}^6$ ]	73.6	64.5	49.0	220.3	342.6	265.2	976.3	727.9
	STD(MSE) [ $\text{cm}^6$ ]	46.5	52.6	76.1	207.0	161.1	129.5	1152.3	328.2
Area <sup>2/3</sup>	RMS%E [%]	3.7	3.8	3.7	7.0	8.2	9.6	12.1	14.3
	RMSE [ $\text{cm}^3$ ]	8.7	<b>7.7</b>	7.0	<b>14.2</b>	18.4	<b>16.7</b>	30.7	<b>27.3</b>
	LO95%CI [ $\text{cm}^3$ ]	7.6	6.6	5.5	12.8	16.2	15.2	21.1	23.8
	UP95%CI [ $\text{cm}^3$ ]	9.7	8.6	8.2	15.5	20.3	18.2	37.9	30.4
	MSE [ $\text{cm}^6$ ]	75.3	59.1	49.1	202.7	338.0	280.0	939.6	744.1
	STD(MSE) [ $\text{cm}^6$ ]	51.1	44.1	76.8	155.7	173.3	116.9	1057.4	378.7
Area with $K$ -estimator	RMS%E [%]	3.4	3.4	5.4	5.4	5.0	5.0	16.2	16.2
	RMSE [ $\text{cm}^3$ ]	8.4	<b>7.1</b>	10.3	<b>10.9</b>	11.9	<b>10.2</b>	40.5	<b>35.3</b>
	LO95%CI [ $\text{cm}^3$ ]	7.2	6.1	9.5	9.9	9.1	8.5	24.0	12.4
	UP95%CI [ $\text{cm}^3$ ]	9.3	7.9	11.1	11.8	14.2	11.6	52.0	48.3
	MSE [ $\text{cm}^6$ ]	69.7	50.0	106.8	117.8	141.3	103.0	1638.4	1243.7
	STD(MSE) [ $\text{cm}^6$ ]	49.5	36.0	67.1	83.0	137.3	72.6	2273.0	2329.1

Table 7.4: Errors on volume estimates with original and redistributed data sets.

remaining shape features, which are invariant to volume, have unchanged correlation coefficients. This means that if one is estimating the volumes of a set of pears known not to vary much in volume, then each pear's volume will be very close to the mean volume of all of the pears, and there is no need for a machine vision volume estimator. However, for more widely distributed volumes, such as the redistributed volume distribution, the  $K$ -estimator is of use, reducing the RMSE from  $14.8\text{cm}^3$  (area predictor) to  $10.9\text{cm}^3$  (area with  $K$ -estimator) in the case of the redistributed volumes.

From Table 7.4, comparisons can be made between the four fruit types. Oranges produce the lowest RMSE of  $7.1\text{cm}^3$  (area with  $K$ -estimator), due to their simple shape. Potatoes produce the highest RMSE of  $35.3\text{cm}^3$  (area with  $K$ -estimator) due to their irregular shape. Lemons and pears, however, show the biggest improvement from the area method to the area with  $K$ -estimator method, since a good prediction of  $K$  can be made for these fruit from shape features.

Methods which use increasingly more of the relevant information from the profile images (such as the inclusion of a  $K$ -estimator in addition to area) become increasingly robust towards differing volume distributions: in other words, the volume distributions do not affect the RMSE as much as when this information is not utilised. This means that the area with  $K$ -estimator method is expected to perform similarly regardless of the volume distribution.

Estimation Method	Statistic	Orange ( $n = 31$ )		Pear ( $n = 64$ )		Lemon ( $n = 23$ )		Potato ( $n = 20$ )	
		Orig.	Redist.	Orig.	Redist.	Orig.	Redist.	Orig.	Redist.
2 images from mirrors at 90°	RMS%E [%]	2.2	2.2	3.4	3.4	3.9	3.9	9.8	9.8
	RMSE [cm <sup>3</sup> ]	5.6	<b>4.5</b>	6.6	<b>7.0</b>	9.1	<b>7.8</b>	24.9	<b>22.1</b>
	LO95%CI [cm <sup>3</sup> ]	4.6	3.9	5.8	6.1	6.4	6.1	12.7	3.9
	UP95%CI [cm <sup>3</sup> ]	6.4	5.0	7.3	7.8	11.1	9.2	32.9	31.0
	$\overline{MSE}$ [cm <sup>6</sup> ]	31.0	20.3	43.3	48.9	82.8	60.6	620.4	488.3
	STD(MSE) [cm <sup>6</sup> ]	28.4	13.6	37.9	45.9	95.6	54.7	980.8	1011.4
All 4 mirror chamber images	RMS%E [%]	1.5	1.5	2.4	2.4	3.6	3.6	6.3	6.3
	RMSE [cm <sup>3</sup> ]	4.0	<b>3.2</b>	4.7	<b>5.1</b>	8.4	<b>7.2</b>	16.1	<b>14.5</b>
	LO95%CI [cm <sup>3</sup> ]	3.1	2.6	3.7	4.1	5.7	5.3	3.0	0.0
	UP95%CI [cm <sup>3</sup> ]	4.8	3.6	5.4	5.9	10.5	8.6	22.6	21.6
	$\overline{MSE}$ [cm <sup>6</sup> ]	16.1	10.0	21.6	25.5	70.7	51.1	259.7	208.9
	STD(MSE) [cm <sup>6</sup> ]	18.9	9.0	31.2	36.2	89.7	52.4	535.0	548.6
All 24 images of the fruit	RMS%E [%]	0.9	1.1	1.8	1.9	3.3	3.2	4.0	4.5
	RMSE [cm <sup>3</sup> ]	2.2	<b>2.3</b>	3.4	<b>3.9</b>	7.3	<b>6.3</b>	10.4	<b>10.1</b>
	LO95%CI [cm <sup>3</sup> ]	1.4	1.5	2.3	2.8	5.2	6.0	4.6	2.4
	UP95%CI [cm <sup>3</sup> ]	2.8	2.8	4.3	4.8	9.0	7.8	14.0	14.0
	$\overline{MSE}$ [cm <sup>6</sup> ]	4.9	5.1	11.6	15.4	53.5	39.2	108.5	101.4
	STD(MSE) [cm <sup>6</sup> ]	8.3	8.0	26.7	31.4	61.6	51.0	187.0	204.2

Table 7.5: Errors on volume estimates from multiple images using the average of area with  $K$ -estimator volume estimation results.

Table 7.5 shows the results of averaging the volume estimates from multiple images of the same fruit. The first rows in the table shows the results of averaging the area with  $K$ -estimator volume predictions from two perpendicular views of the fruit. Improved estimates are obtained for all of the fruit types. The improvements can largely be attributed to the fact that volume over-estimates due to viewing a large profile image of the fruit (such as the one shown in Figure 7.3(a)) are often compensated by a volume-underestimate in the small profile image of the fruit in the perpendicular view (such as the one shown in Figure 7.3(b)).

Although averaging the volume estimates is a convenient way of combining them, it is not necessarily the optimal method of doing so. Consider, for example, a fruit that is roughly cylindrically symmetrical, and which has a larger side view profile than top view profile. A pair of perpendicular views of this fruit could consist of two side views

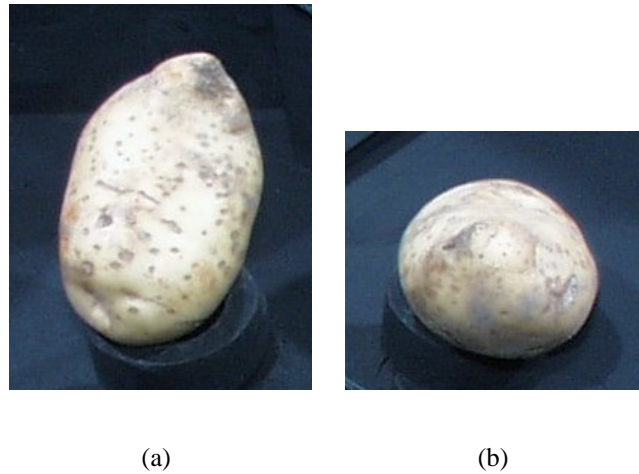


Figure 7.3: Two perpendicular views of a potato.

(two large profile objects) or a top view and a side view (one small and one large profile object), but not two top views (two small profile objects). This observation indicates that simple averaging is not necessarily the best method of combining the estimates and that it is likely that a method which takes the statistical relationships between the perpendicular views into account would give better results. Such a method is implemented implicitly by using an MLP, and is discussed in the next section.

As Table 7.5 shows, the averaging of the four mirror chamber image estimates (main, upper, left and right) yields an improvement on both the estimates from single and from perpendicular-pair images. Again, averaging the estimates is a convenient, linear method of combining the estimates, yet not necessarily optimal, and an MLP method that uses the same feature input information as the method used to produce the results in Table 7.5 is presented in the next section.

The results of averaging the estimates for all of the 24 images of each fruit are also shown in Table 7.5. The 24 images consist of the six sets of four images from the mirror chamber for each fruit. The results are presented here mainly for interest, as it is unlikely that a fruit sorting system would produce so many images of each fruit. Here it is interesting to note that the mean of 24 volume estimates of a potato is necessary to reduce the RMS error ( $10.1\text{cm}^3$ ) to roughly that which is obtained from a single image of a lemon ( $10.2\text{cm}^3$ ). An MLP which uses the same feature input information was not developed because the ratio of inputs to available data samples was far too high to produce meaningful results.

## 7.5 Multi-layer perceptron regression methods

Multi-layer perceptrons (MLPs) were used to investigate a better combination of a single  $K$ -estimator and area to predict volume for each of the fruit types. Although the best  $K$ -estimator for each fruit type was selected from the pool of shape features using a *linear* method (the linear correlation coefficient), it was decided to use the same shape features for each fruit type as input to the MLPs. This was done mainly for purposes of comparison, to investigate whether or not the non-linear MLP combinations of the same inputs would result in significantly better volume estimates than the linear methods. Figure 7.2 shows that it is likely that some non-linear relationships do exist between at least some variables. Unlike the linear methods, MLPs can produce non-linear mappings and can thus reduce the volume estimation error to the extent that a non-linear relationship exists between the selected features and the target volume in the fruit type. Another advantage of using both the  $K$ -predictor feature and area as inputs to the MLP to predict volume, instead of just using the  $K$ -predictor as an input and then using  $V = KA^{\frac{3}{2}}$  to predict volume, is that the MLPs do not have to adhere to the assumptions made in the  $V = KA^{\frac{3}{2}}$  equation (such as the assumption that the camera is at infinity).

Apart from exhaustively testing the combination of every single shape feature with area as an input to the MLP, there is no way of guaranteeing that the feature which is the best linear predictor of  $K$  is also, together with area as an input, the best predictor of volume in an MLP. The exhaustive testing was not carried out because it would have been far too time-consuming, and so only the shape features used as  $K$ -predictors for the linear methods were used at this stage with the MLPs. Nevertheless, some investigations of likely feature combinations that were carried out yielded no remarkable reduction in error.

The MLPs designed to estimate fruit volume from one shape feature with area using single images had 15 hidden units and were trained for 100 cycles using the quasi-Newton method. These values were determined empirically, and it was found that the MLP was insensitive to the exact values of the number of hidden layers and number of training cycles.

Table 7.6 shows the results of using an MLP with area and one shape feature as inputs. This table can be compared with the last row of Table 7.4, where the same features were linearly combined to produce volume estimates. The MLP volume estimates

Estimation Method	Statistic	Orange ( $n = 31$ )		Pear ( $n = 64$ )		Lemon ( $n = 23$ )		Potato ( $n = 20$ )	
		Orig.	Redist.	Orig.	Redist.	Orig.	Redist.	Orig.	Redist.
Area and shape feature	RMS%E [%]	3.3	3.5	3.6	6.1	4.5	4.9	11.8	10.4
	RMSE [ $\text{cm}^3$ ]	7.9	<b>6.8</b>	6.7	<b>10.7</b>	9.9	<b>9.4</b>	29.8	<b>26.4</b>
	LO95%CI [ $\text{cm}^3$ ]	6.8	6.1	5.2	9.7	8.1	8.0	20.8	17.8
	UP95%CI [ $\text{cm}^3$ ]	8.9	7.5	7.9	11.7	11.3	10.6	36.7	32.8
	$\overline{\text{MSE}}$ [ $\text{cm}^6$ ]	62.8	46.8	44.6	115.3	97.1	88.3	888.8	697.0
	STD(MSE) [ $\text{cm}^6$ ]	48.6	26.7	71.4	90.7	71.8	57.8	972.0	810.2

Table 7.6: Errors on volume estimates from single images using area with an implicit  $K$ -estimator as MLP inputs.

show some small improvements on the linear method using the same inputs. For instance, the RMSE on lemons dropped from  $10.2\text{cm}^3$  to  $9.4\text{cm}^3$ . Using the  $t$ -statistic on the pairs of  $\overline{\text{MSE}}$  and  $\text{STD}(\text{MSE})$  indicates that one can be 77.6% sure of an improvement in performance if the MLP is used in place of the linear method for these lemons. This indicates that more data would be required before we could be more confident in this result (say 95% certain). However, because the difference in error between the two methods is so small, it also indicates that the model used in the linear method ( $V = KA^{\frac{3}{2}}$ ) is a reasonable model that compares favourably with a more flexible method (the MLP).

Various other input combinations were experimented with for the MLP using inputs from a single image. Several methods of determining which features to add to the function approximator as extra inputs were investigated. Observing plots of the individual volume errors against the various shape features showed little correlation. Observing the profile images corresponding to the particularly high and low errors also revealed no particular pattern. Furthermore, no increase in the number of shape feature inputs was found to show any significant improvement on the RMS error. This is because the shape features are often highly correlated with one another, or contain noise. Principle component analysis [3] was experimented with to form linear combinations of the features, but the best linear combinations did not outperform the best features taken singly, and required the calculation of all of the features, which would be far too time-consuming in an implemented system. The ‘law of diminishing returns’ might be said to apply to volume estimation from shape features: the more features that are added, the lower the improvement on the RMS error. The improvements that might occur when three or more features are used as inputs to the MLP would require much larger training sets to be shown to exist than those that were available in this project.



Even if small improvements in the RMSE could be shown to exist, they would be of little practical use, since they would require the extraction of further shape features, which would be time consuming and would add complexity to the system.

The performance of MLPs on input from more than one image was tested. As explained in Chapter 2, if this data is treated together it seems likely that a better volume estimate would be produced than by averaging the volume estimates from multiple pictures, each treated separately.

Estimation Method	Statistic	Orange ( $n = 31$ )		Pear ( $n = 64$ )		Lemon ( $n = 23$ )		Potato ( $n = 20$ )	
		Orig.	Redist.	Orig.	Redist.	Orig.	Redist.	Orig.	Redist.
2 randomly selected images	RMS%E [%]	2.3	2.6	3.1	3.8	3.7	3.5	11.2	14.6
	RMSE [ $\text{cm}^3$ ]	5.5	<b>5.1</b>	5.9	<b>7.6</b>	7.6	<b>6.6</b>	29.4	<b>23.8</b>
	LO95%CI [ $\text{cm}^3$ ]	4.8	4.6	4.8	6.8	6.7	5.8	17.8	15.9
	UP95%CI [ $\text{cm}^3$ ]	6.2	5.6	6.9	8.4	8.5	7.4	37.6	29.6
	$\overline{\text{MSE}}$ [ $\text{cm}^6$ ]	30.3	25.9	35.0	58.4	57.9	43.6	865.8	565.9
	STD(MSE) [ $\text{cm}^6$ ]	21.1	14.6	50.9	52.0	31.3	24.3	1172.4	668.3
2 images from mirrors at 90°	RMS%E [%]	2.1	2.3	2.5	3.2	2.4	2.4	8.7	9.2
	RMSE [ $\text{cm}^3$ ]	4.8	<b>4.5</b>	4.8	<b>6.4</b>	5.4	<b>4.8</b>	21.6	<b>20.5</b>
	LO95%CI [ $\text{cm}^3$ ]	4.2	3.9	4.2	5.7	4.3	3.9	16.3	13.4
	UP95%CI [ $\text{cm}^3$ ]	5.3	4.9	5.4	7.1	6.2	5.6	25.8	25.8
	$\overline{\text{MSE}}$ [ $\text{cm}^6$ ]	23.0	19.8	23.3	41.3	28.7	23.5	465.0	421.5
	STD(MSE) [ $\text{cm}^6$ ]	15.4	12.4	24.7	37.9	23.7	19.1	429.2	518.3
4 randomly selected images	RMS%E [%]	1.8	2.2	2.6	2.9	3.3	3.3	9.1	9.5
	RMSE [ $\text{cm}^3$ ]	4.2	<b>4.1</b>	5.0	<b>5.8</b>	7.1	<b>6.8</b>	23.9	<b>21.2</b>
	LO95%CI [ $\text{cm}^3$ ]	3.6	3.7	4.1	5.0	5.9	4.0	14.9	14.6
	UP95%CI [ $\text{cm}^3$ ]	4.7	4.5	5.7	6.5	8.1	8.8	30.3	26.3
	$\overline{\text{MSE}}$ [ $\text{cm}^6$ ]	17.5	17.0	24.6	33.4	49.9	46.8	569.4	451.3
	STD(MSE) [ $\text{cm}^6$ ]	13.1	8.7	32.5	36.2	34.5	70.6	742.2	507.9
All 4 mirror chamber images	RMS%E [%]	1.3	1.7	1.9	2.0	1.6	1.7	5.3	5.6
	RMSE [ $\text{cm}^3$ ]	2.9	<b>3.2</b>	3.7	<b>4.1</b>	3.2	<b>3.6</b>	14.1	<b>12.6</b>
	LO95%CI [ $\text{cm}^3$ ]	2.5	2.7	2.9	3.2	2.7	2.0	8.2	2.2
	UP95%CI [ $\text{cm}^3$ ]	3.3	3.5	4.3	4.8	3.7	4.8	18.2	17.6
	$\overline{\text{MSE}}$ [ $\text{cm}^6$ ]	8.5	10.0	13.7	16.5	10.5	13.2	198.9	158.2
	STD(MSE) [ $\text{cm}^6$ ]	6.2	7.4	21.0	26.3	7.9	21.8	280.1	327.2

Table 7.7: Errors on volume estimates from multiple images using area with an implicit  $K$ -estimator from each image as MLP inputs.

Table 7.7 shows the results of two 4-input MLPs, each taking area and the best  $K$ -estimator feature from two images. This was done firstly with pairs of images of the fruit chosen at random from the 24 available images of each fruit and secondly with pairs of perpendicular images of the fruit. The results of two 8-input MLPs, each taking area and the best  $K$ -estimator feature from four images, are presented in the third and fourth set of rows in Table 7.7. Again, this was implemented firstly with the four images of the fruit chosen at random, and secondly by using the four mirror

chamber images of the fruit so as to simulate a four camera system with a fixed camera geometry.

The MLPs used in this section had 50 sets of hidden units and were trained for 150 cycles. As before, these values were determined empirically and the MLPs were insensitive to the exact values of these parameters. The increase in the required number of hidden units is attributed to the increased complexity of the problem.

The sets of random images were used to show that the structured orientation of the images would produce better results than the random sets of images. As can be seen from Table 7.7, and as would be expected, the sets of images at consistent angles outperformed the sets of randomly-chosen images. For example, in the case of the lemons, a reduction in RMSE from  $6.8\text{cm}^3$ , in the case of four randomly chosen images, to  $3.6\text{cm}^3$ , in the case of using the four images from the mirror chamber, is achieved.

Comparing these results with the average of the volume estimates used in the case of the linear methods (see Table 7.5), suggests that it is advantageous to combine the shape features from multiple images in parallel (using an MLP). For example, in the case of volumetric estimation of lemons using four images from the mirror chamber, there is an improvement in the RMSE from  $7.2\text{cm}^3$  using the linear methods to  $3.6\text{cm}^3$  using the MLP-based methods. The t-statistic ( $t = 3.203$ ) shows that in this case one can be more than 99.9% confident that there is a real improvement in using the MLP-based estimate.

# Chapter 8

## Conclusions

This project has involved the development and testing of algorithms that produce volume estimates of fruit from digital profile images of the fruit which are similar to those used in many commercial fruit sorting environments for colour sorting and blemish detection. The methods developed in the project are of a statistical nature, and are general in the sense that they do not rely on the mechanics of the conveyor system to align the fruit in any particular orientation. The fruit are assumed to be randomly oriented with respect to the camera. Since the methods used in this project rely on training for the setting of parameters, they are adaptable for use with many existing fruit sorting systems.

An experimental set-up was built to simulate the images that would be obtained in single- and multiple-camera fruit sorting systems. This took the form of a mirror chamber and a camera mounted upon a tripod. Data sets of orange, pear, lemon and potato images were captured photographically. These fruit were chosen so as to cover a range of different fruit shapes and to compare methods with those in published literature. Multiple images were obtained for each fruit, both to investigate multiple image volume estimations and to improve the confidence in the expected RMS errors produced by the methods. Target volumes were measured for all of the fruit that were photographed. This was a time consuming process, yet could be achieved accurately through the use of a purpose-built pycnometer.

Multi-layer-perceptron-based segmentation methods were developed and implemented on the fruit images, to mimic the segmentation processes which occur for colour sorting in a commercial fruit sorting environment, and to determine the fruit profile bound-

aries from the digital images. A set of shape features was then measured for each of the digital images. These shape features functioned as a pool from which to select features to be combined to form a volume estimate of the corresponding piece of fruit.

Both linear techniques and artificial neural network methods, using both single and multiple images for volume estimation were tested on the four fruit types. The methods each incorporated a function of a small number of rapidly-extractable shape features, so that volume estimation could be implemented in conjunction with the existing image processing procedures used for colour and blemish sorting, and with little increase in computational time. It was confirmed that the shape as well as the size of the fruit profile is often of importance in estimating the volume of the corresponding piece of fruit.

A jackknifing procedure was used to ensure the best realistic estimation of expected RMS error and of the associated confidence in this result for each volume estimation method. A volume redistribution technique was implemented so that meaningful comparisons could be made across the fruit types, on a volume distribution typical of that found in a batch of fruit at a commercial packing house.

Using linear methods and features from a *single* image, an RMS error of  $7.1\text{cm}^3$  was calculated to be expected of a system using such a method on a batch of oranges with a typical distribution of volumes (a mean of  $200\text{cm}^3$  and a standard deviation of  $50\text{cm}^3$ ). This means that, assuming a normal distribution of errors, about 68% of the oranges (corresponding to one standard deviation) can be expected to have their volumes estimated with an error of less than  $7.1\text{cm}^3$  (which is about the same volume as a heaped teaspoon of sugar). The low RMS errors achieved with oranges, even with a single feature from a single image, can be attributed to their simple, almost spherical shape. This volume estimate can be improved upon by using an artificial neural network to combine shape features from multiple images. Using this method with a single set of four images from four well-separated cameras looking down on the fruit is expected, for example, to give an RMS error of  $3.2\text{cm}^3$ .

In the case of the pears, after redistributing the volumes it was calculated that an RMS error of  $10.9\text{cm}^3$  could be expected if a linear method is used to combine area and mean diameter measurements from a single image (even although a large portion of the fruit is hidden due to the random orientation of the fruit). This is a noticeable improvement on the volume estimation using only area from a single image, which can be expected to produce an error of  $14.8\text{cm}^3$ . This indicates that the shape of the

profile image is useful in predicting pear volume. The pear data set, at 64 fruit and 1536 images, was the largest for any of the four fruit types investigated in this project and correspondingly produces the highest confidence in the expected RMS errors. An RMS error of  $4.1\text{cm}^3$  can be expected from an 8-input multi-layer perceptron (MLP) using two inputs from each of four images obtained from four cameras looking down on the fruit, as suggested by the mirror chamber geometry.

Similarly to the pears, the lemons produced an expected RMS error of  $10.2\text{cm}^3$  when using linear methods to combine area and aspect ratio features from a single profile image, compared with an expected  $16.3\text{cm}^3$  RMS error when using area alone. The shape of the lemon profile is thus useful in estimating the volume of the corresponding lemon. The expected RMS error can be brought down to  $3.6\text{cm}^3$  using an 8-input MLP with features from all four images of a well-spaced four-camera set-up.

Potato volume estimation gave the worst performance of the four fruit types investigated, due to the potatoes' irregular and inconsistent shapes. A single image with an area measurement resulted in an RMS error of  $27.0\text{cm}^3$ . However, this error could be reduced by using multiple images. The error was reduced to  $12.6\text{cm}^3$  using a 4-image MLP system and to  $10.1\text{cm}^3$  when using the mean of the linear estimates for all 24 images of each of the potatoes. Due both to the irregularity in shape of the potato and to the relatively small data set, the potato results have the lowest associated confidence.

The choice of which volume estimation method to implement is largely dependent on the camera system and on the images which are already available at a packing line. If a simple one-camera, one-image system is in use, then simple linear methods can be employed to estimate volume almost as effectively as the MLP methods. If multiple camera systems are being used, or if the fruit is systematically rolled between photographs by a conveyor system, then the more complex MLP methods can be brought into play to combine the extra information so as to produce a system with a lower RMS error.

Once trained on a sample of the fruit, the volume estimation methods will give rapid volume estimates which can be used either in conjunction with a weighing mechanism to identify candidate fruit pieces for rejection due to low density and probable damage, or to monitor the performance of the weighing mechanism (to check that it does not become stuck). If the volume estimation is used to estimate fruit weight, then this information can be used to pack the fruit cost-effectively into boxes (as described in Appendix A) or to sort the fruit into consistent batches.

# Appendix A

## Selecting Weight Thresholds for Packing Fruit

This appendix describes an optimisation algorithm that was developed for reducing monetary loss in the packing of fruit into boxes which are sold at a fixed labelled weight. The test results of the algorithm not only show how commercial packing houses can reduce their losses through the use of mathematical and numerical methods, but also show how the RMS error on a weight measurement, which may be determined from a volume estimate, translates directly into a monetary cost in the context of packing the fruit into boxes.

### A.1 Statement of the packing problem

At the end of a packing line, fruit is packed into boxes which are to be sold by weight. When a company called Bridge Fruit packs Valencia oranges, which is the example cited throughout this appendix [33], each box is labelled and sold as a 14.5kg box. Since each box is sold at the same weight, boxes with larger fruit require a lower count of fruit than boxes with smaller fruit. There are six different categories of 14.5kg box, as shown in Table A.1. The different categories of box contain different numbers or counts of fruit. At the end of the packing line, the fruit is packed into a box of a certain count according to the weight category into which it falls. For instance, according to Table A.1, a 140g orange would be placed in a 97 count box, since its weight is in the 138g – 164g range. Using this system of packing, a 97 count box could weigh

Count	Weight category thresholds currently in use
116	< 138g
97	138g – 164g
84	164g – 200g
69	200g – 248g
56	248g – 287g
48	> 287g

Table A.1: Box counts and currently-used Bridge Fruit weight category thresholds for Valencia oranges.

anything from  $97 \times 138\text{g} = 13.386\text{kg}$  to  $97 \times 164\text{g} = 15.908\text{kg}$ , but is, in fact, most likely to weigh just over 14.5kg. If a box weighs less than 14.5kg, then it cannot be sold and it must be repacked at a cost that has been estimated to be R4.35 per box [33]. If a box weighs more than 14.5kg, then fruit that could have been sold at a rate of R2.07 per kilogram is being given away for nothing. The threshold weights that divide the weight categories should be chosen carefully so as to minimise the expected cost per box due to the penalties incurred by over- and under-packing boxes. The optimisation problem is thus the selection of these threshold weights in order to minimise the expected average cost per box.

## A.2 Defining the cost function

In general, the expected cost per box,  $c_{\text{expected}}$ , is a function of the  $n - 1$  weight category thresholds,  $t_1, t_2, \dots, t_{n-1}$ , which separate the  $n$  categories: it is the average cost per box due to over-packing and under-packing that is expected to be incurred in the long run. For purposes of calculation, the lower threshold bound,  $t_0$ , is fixed at zero, and the upper threshold bound,  $t_n$ , is fixed at infinity. The weight range for count category  $i$  is thus specified by the range from  $t_{i-1}$  to  $t_i$ . The probability density function (PDF) of the fruit weights in every category can be estimated from  $m$  samples that have been randomly selected from the population of fruit weights. Each of the  $n$  weight categories has a corresponding number of fruit per box, and these numbers are denoted as the *counts*,  $N_1, N_2, \dots, N_n$ . The other constants for the problem are the cost per box of under-packing,  $C$ , the loss per kilogram of fruit for over-packing,  $L$ , and the labelled weight at which the boxes are to be sold,  $W$ .

The expected cost per box is the sum of the expected average costs for each category or box type,  $c_1, c_2, \dots, c_n$ , weighted by the probabilities  $P_1, P_2, \dots, P_n$ , of occurrence of the corresponding box type:

$$c_{expected}(t_1, t_2, \dots, t_n) = \sum_{i=1}^n P_i c_i \quad (\text{A.1})$$

The probabilities,  $P_1, P_2, \dots, P_n$ , are found by considering a representative sampling of  $m$  fruit pieces taken from the population of available fruit. Each probability,  $P_i$ , expresses the chance that a given box of this fruit (when all the fruit is packed) contains fruit pieces of weight category  $i$  (as defined by the threshold weights  $t_1, t_2, \dots, t_n$ ). If the sample of  $m$  fruit pieces comprises  $M_1, M_2, \dots, M_n$  fruit pieces for the  $n$  categories, the number of boxes,  $B$ , required to pack each category is given by

$$B_i = \frac{M_i}{N_i} \quad (\text{A.2})$$

and, from this, the category box probability can be written as

$$P_i = \frac{B_i}{\sum_{j=1}^n B_j} \quad (\text{A.3})$$

In order to calculate the expected cost,  $c_i$ , of packing a box of a particular category, it is necessary to calculate the PDF of the weight of the packed box,  $p_{box,i}$ . The PDF of weight for a box of category  $i$  is the convolution of  $N_i$  PDFs of weight for fruit in category  $i$ ,  $p_{fruit,i}$ . The central limit theorem [20] states that, as  $N_i$  becomes larger,  $p_{box,i}$  will tend towards being normally distributed, regardless of the shape of the distribution,  $p_{fruit,i}$ . Since the convolved fruit weight PDFs (corresponding to each fruit in the box) are independent of one another, the box weight PDF can be determined to a very good approximation by assuming a normal distribution of mean  $\mu_i$  and variance  $\sigma_i^2$ , where

$$\mu_i = N_i \times \text{Mean of weight samples in } t_{i-1} \text{ to } t_i \text{ range} \quad (\text{A.4})$$

and

$$\sigma_i^2 = N_i \times \text{Variance of weight samples in } t_{i-1} \text{ to } t_i \text{ range} \quad (\text{A.5})$$

It is therefore not necessary to know the actual PDF,  $p_{fruit,i}$ , in order to calculate a very good approximation to  $p_{box,i}$ ; in fact only the mean ( $\mu_i$ ) and variance ( $\sigma_i^2$ ) of  $p_{box,i}$  need be known [20]. The mean and variance are both best estimated by the mean



and variance of the weight samples which fall within category  $i$ . The PDF,  $p_{box,i}$ , for a particular category  $i$  defined by a set of thresholds,  $t_1, t_2, \dots, t_n$ , is then given virtually exactly by

$$p_{box,i} = \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{(x-\mu_i)^2}{2\sigma_i^2}} \quad (\text{A.6})$$

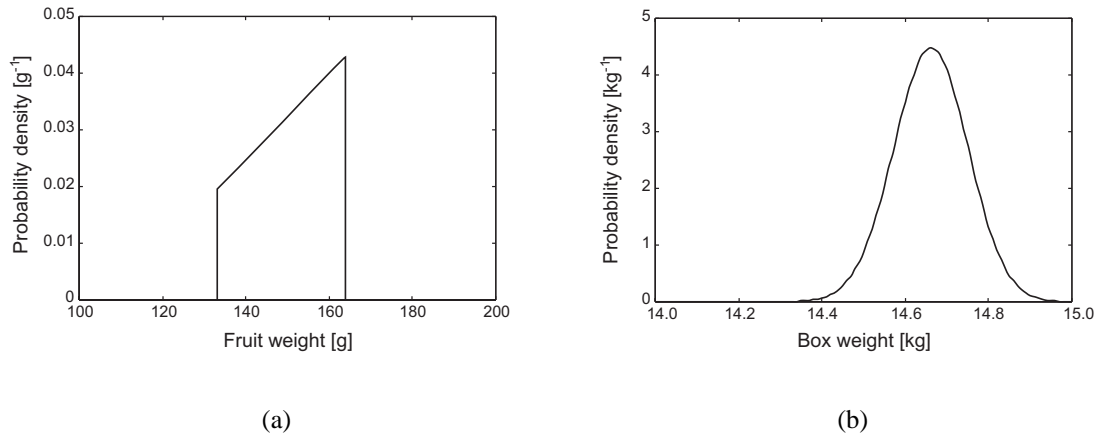


Figure A.1: An example of a category PDF of (a) fruit weights; (b) box weights.

Figure A.1(a) shows an example of  $p_{fruit,i}$  for a 97-count category that happens to have weights in a 133g to 164g range. Figure A.1(b) shows an example in which  $p_{box,i}$  is formed by convolving the PDF in Figure A.1(a) with itself 97 times. Note that even though  $p_{fruit,i}$  has steep sides,  $p_{box,i}$  is normally distributed to a very good approximation.

The box weight PDF,  $p_{box,i}$ , is used to calculate the expected average over-packing costs,  $c_{over,i}$ , and under-packing costs,  $c_{under,i}$ , for boxes of each weight category,  $i$ . Then, the expected average cost per box within category  $i$  is

$$C_i = c_{under,i} + c_{over,i} \quad (\text{A.7})$$

The probable under-packing costs are calculated by integrating the product of the cost of under-packing and the box weight PDF in category  $i$ .

$$c_{under,i} = \int_{-\infty}^W C p_{box,i} dx \quad (\text{A.8})$$

$$= C \times Q\left(\frac{W - \mu_i}{\sigma_i}\right) \quad (\text{A.9})$$

where

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{u^2}{2}} du \quad (\text{A.10})$$

$Q(x)$  cannot be integrated in closed form [22], and is instead evaluated using a look-up table. In general, for a normal distribution of mean,  $m$ , and standard deviation,  $s$ ,

$$\frac{1}{\sqrt{2\pi}s} \int_{-\infty}^x e^{-\frac{(u-m)^2}{2s^2}} du = Q\left(\frac{x-m}{s}\right) \quad (\text{A.11})$$

and so, for the right-hand tail of a distribution with zero mean

$$\frac{1}{\sqrt{2\pi}s} \int_x^{\infty} e^{-\frac{u^2}{2s^2}} du = 1 - Q\left(\frac{x}{s}\right) \quad (\text{A.12})$$

The over-packing cost,  $y$ , for a box of weight  $x$  is given by the equation

$$y = \begin{cases} 0 & : x \leq W \\ (x - W)L & : x > W \end{cases} \quad (\text{A.13})$$

The probable over-packing costs are calculated by integrating the product of the over-packing cost and box weight PDF:

$$c_{over,i} = \int_W^{\infty} y p_{box,i} dx \quad (\text{A.14})$$

This integral is simpler to evaluate if  $y'$  and  $p'_{box,i}$  are formed by shifting the functions  $y$  and  $p_{box,i}$  a distance  $\mu_i$  along the  $x$ -axis so that  $p'_{box,i}$  has zero mean:

$$y' = \begin{cases} 0 & : x \leq W - \mu_i \\ (x - (W - \mu_i))L & : x > W - \mu_i \end{cases} \quad (\text{A.15})$$

and

$$p'_{box,i} = \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{x^2}{2\sigma_i^2}} \quad (\text{A.16})$$

so that

$$c_{over,i} = \int_{W-\mu_i}^{\infty} y' p'_{box,i} dx \quad (\text{A.17})$$

The integral can now be evaluated by inspection and with the use of a  $Q(x)$  term (see Equation (A.12)):

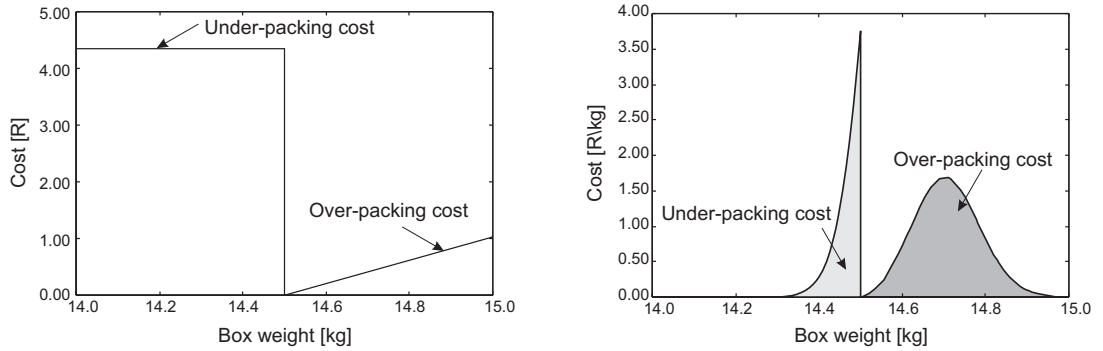
$$c_{over,i} = \int_{W-\mu_i}^{\infty} (x - (W - \mu_i))L \frac{e^{-\frac{x^2}{2\sigma_i^2}}}{\sqrt{2\pi}\sigma_i} dx \quad (\text{A.18})$$

$$= \int_{W-\mu_i}^{\infty} \left( \frac{L}{\sqrt{2\pi}\sigma_i} x e^{-\frac{x^2}{2\sigma_i^2}} - \frac{(W - \mu_i)L}{\sqrt{2\pi}\sigma_i} e^{-\frac{x^2}{2\sigma_i^2}} \right) dx \quad (\text{A.19})$$

$$= \frac{L}{\sqrt{2\pi}\sigma_i} \int_{W-\mu_i}^{\infty} x e^{-\frac{x^2}{2\sigma_i^2}} dx - (W - \mu_i)L \int_{W-\mu_i}^{\infty} \frac{e^{-\frac{x^2}{2\sigma_i^2}}}{\sqrt{2\pi}\sigma_i} dx \quad (\text{A.20})$$

$$= \left[ \frac{-L\sigma_i}{\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma_i^2}} \right]_{W-\mu_i}^{\infty} - (W - \mu_i)L \left[ 1 - Q\left(\frac{W - \mu_i}{\sigma_i}\right) \right] \quad (\text{A.21})$$

$$= \frac{L\sigma_i}{\sqrt{2\pi}} e^{-\frac{(W-\mu_i)^2}{2\sigma_i^2}} - (W - \mu_i)L \left[ 1 - Q\left(\frac{W - \mu_i}{\sigma_i}\right) \right] \quad (\text{A.22})$$



(a)

(b)

Figure A.2: An example of (a) a cost function, and (b) a probable cost distribution for a particular weight category.

Figure A.2(a) shows an example of cost as a function of box weight based upon the same typical figures mentioned in Section A.1. The probable average cost per box can be found by integrating the area under the curve shown in Figure A.2(b). Equation (A.8) is the under-packing cost portion of this integral and Equation (A.17) is the over-packing cost portion. This example was derived by using the product of the cost curve in Figure A.2(a) and the box weight PDF shown in Figure A.1(b).

### A.3 Testing the cost function

Given a set of  $m$  samples of fruit weight values, the cost function can also be estimated by *simulating* the box packing using these values. To do this, the sample weights are each assigned to one of the  $i$  categories based upon the threshold weight values. Groups of fruit weights corresponding to the counts in each category are then summed to give box weights. The excess fruit weights are discarded and the box weight costs are then determined by using a cost function such as the one shown in Figure A.2(a). Finally, the average cost per box is calculated.

Compared with the largely analytical method described in Section A.2, the simulation method has a poor ability to generalise for realistic values of  $m$ , resulting in greater error in the expected average cost. This is because it only makes use of the specific sample weight values, whereas the analytical method makes use of the variance to estimate the box weight PDFs. As  $m$  becomes large, the two methods converge on the same cost.

To test the validity of the cost function, an artificial set of 100 000 normally distributed fruit weights with a mean of 180g and standard deviation of 40g was generated using a pseudo-random number generator.

The simulated average box weight costs were compared with the costs calculated using Equation (A.1) for various sets of thresholds, using the parameters for Valencia oranges as described in Section A.1. The mean absolute difference between the two sets of results was less than R0.01.

Assuming the threshold solutions derived from these 100 000 samples to be the actual solutions, sampling again with  $m = 1000$  samples now gave mean errors of R0.07 and R0.12 for the analytical method and the simulation method respectively. Using  $m = 10\ 000$  samples gave mean errors of R0.02 for the analytical method and R0.03 for the simulation method.

The extent to which the two costs agree for large values of  $m$  reflects the extent to which the assumption of normally distributed box weights (by the central limit theorem) is valid.

## A.4 Minimising the cost function

Once the cost function,  $c_{expected}$ , can be evaluated for any set of threshold weight values,  $t_1, t_2, \dots, t_n$ , a set of threshold weight values which provide a low cost may be found.

It was decided to use an optimisation technique known as Population Based Incremental Learning (PBIL) [2] to select such a set of threshold weights. After setting up a random number generator with an initially-uniform PDF over the space of the function to be optimised, PBIL repeatedly spawns generations of populations of candidate solutions, adjusting the PDF slightly in each generation so as to favour the best candidate of the population. This is explained in greater detail below. PBIL has been shown to be a simple yet widely effective function optimisation strategy [2]. Furthermore, the PBIL algorithm can restrict the solution vector to consist of integers, which is useful since the threshold weights on the packing line can in practice only be set as integer weight values in grams [33].

As the cost function is a summation of integrals, it may be expected to be intrinsically smooth and slow-varying. Such functions do not usually pose any difficulty for optimisation algorithms, because they do not present the problem of dealing with steep ‘cliffs’ or ‘spikes’. In addition to this, the search space contains only  $2^{5 \times 6} = 2^{30}$  candidate solutions (this is explained below), and therefore could almost be searched exhaustively for the best solution. These two points suggest that it would be best to use a simple-to-code algorithm, such as the basic PBIL, rather than any more elaborate PBIL variant or other optimisation technique, which would in either case be unlikely to show any further improvement on the basic PBIL results.

PBIL encodes candidate solutions (weight thresholds in this case) as binary strings. The threshold weights were encoded by storing the differences between thresholds as binary strings. The first threshold was offset from a value of 111g (which was the smallest of the thousand sample weights for Valencia oranges). Each of the five differences was then encoded with six bits. For example, the bit string [011011 011010 100100 110000 100111] corresponds to the set of thresholds [138g 164g 200g 248g 287g], as illustrated below:

Binary number	Decimal number	Threshold weight
011011	27	$111 + 27 = 138$
011010	26	$138 + 26 = 164$
100100	36	$164 + 36 = 200$
110000	48	$200 + 48 = 248$
100111	39	$248 + 39 = 287$

In PBIL, a population of candidate solutions is created by sampling a probability vector. Each element in the probability vector corresponds to the probability that a bit in a candidate solution will be ‘one’. Each element in the probability vector is sampled by creating a pseudo-random number chosen from a uniform distribution on the interval (0,1). If the probability vector element is greater than the random number, then the corresponding bit in the candidate solution is set to ‘one’; otherwise it is set to ‘zero’. The probability vector has each of its elements initially set to 0.5.

A *generation* of, say, 100 candidate solutions is produced from the probability vector. Each candidate solution is evaluated with the cost function and the best candidate solution is selected to be used in the updating of the probability vector.

If the probability of generating a ‘one’ in bit position  $i$  is  $probability_i$ , and if the  $i^{th}$  position in the solution vector towards which the probability vector is moved is  $vector[i]$ , then the probability vector may be updated after each generation using [2]

$$probability_i = (probability_i \times (1 - LR)) + (LR \times vector[i]) \quad (A.23)$$

where  $LR$  is a parameter known as the *learning rate*.

In order to maintain diversity in the population, and to reduce the likelihood of premature convergence, a small proportion (the *mutation rate*,  $MR$ ) of the probability vector elements are randomly updated towards ‘one’ or ‘zero’ with equal probability. This is done at the end of each generation, using the same method as described in Equation (A.23). It is debatable whether the mutation rate parameter (which is fundamental to creating diversity in the genetic algorithms from which PBIL is derived) is strictly necessary to the operation of the PBIL algorithm. However, it was used in this project for the sake of working with a standard and pre-existing optimisation method [2].

PBIL repeatedly generates populations of candidates and updates the probability vector until the same best solution is generated several (say five) times in a row. The best

solution produced by PBIL has been found to yield very satisfactory minimisations of the cost function.

## A.5 Optimising thresholds for Valencia oranges

Valencia orange data was obtained for testing the threshold optimisation from Mr Stanford [33] of Bridge Fruit, which is a commercial packing house in the Western Cape. At this packing line, the oranges are packed into six different weight categories, the counts and currently-used thresholds of which are shown in Table A.1. The constants  $L$ ,  $C$  and  $W$  were given as:

Under-packing cost:  $L = \text{R}4.35$  per box  
 Over-packing cost:  $C = \text{R}2.07$  per kg  
 Labelled box weight:  $W = 14.5\text{kg}$

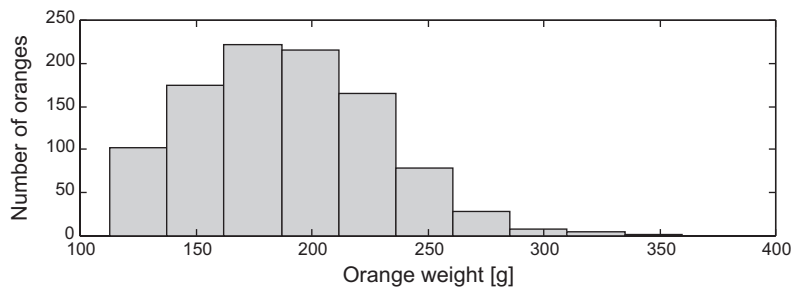


Figure A.3: Histogram of 1000 Valencia orange weights.

For each candidate, category means and variances were estimated from 1000 orange weights, taken as a sample. A histogram of the 1000 orange weights used is shown in Figure A.3. The histogram gives an impression of what the PDF of the orange weights would look like.

Equation (A.1) gave an average cost of R1.48 per box using the thresholds currently in use by Bridge Fruit, namely [138g 164g 200g 248g 287g]. This means that Bridge Fruit are incurring an extra average cost of R1.48 for every box that they pack by using these thresholds.

PBIL, with a population of 200 candidate solutions per generation, a learning rate of  $LR = 0.015$  and a mutation rate of  $MR = 0.020$ , gave an average cost per box of

R0.72. The solution set of threshold weights was [138g 166g 189g 250g 279g]. This result was reached after 346 generations of PBIL. These threshold weights are likely to produce lower costs than the thresholds currently in use by Bridge Fruit. Although there is an unspecified uncertainty associated with the R0.72 cost, R0.72 is the most likely cost given the  $m = 1000$  fruit weight samples. A larger sample of weights could increase or decrease this cost estimate, and would have a lower uncertainty associated with it.

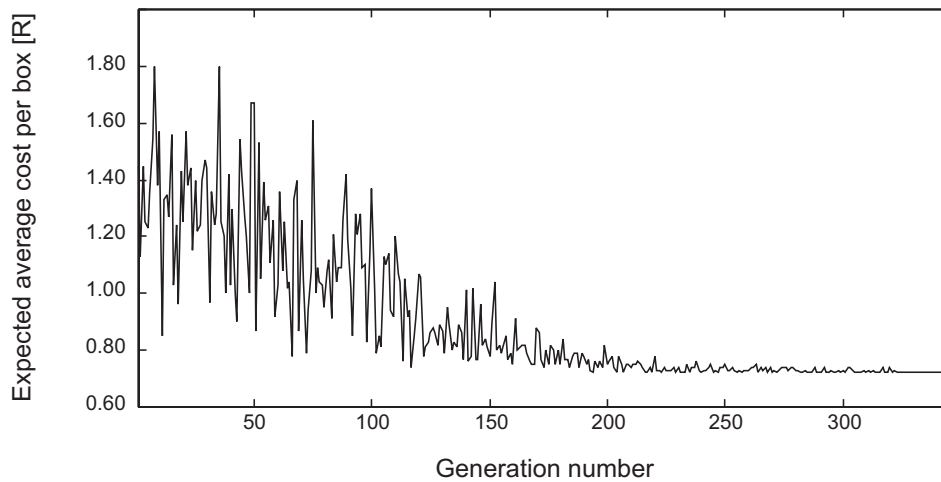


Figure A.4: Best solution in each PBIL generation.

Figure A.4 shows a graph depicting the cost corresponding to the best solution for each generation of the PBIL optimisation. The noisy region on the left of the graph corresponds to the initially random search through the solution space. As one moves to the right, the more significant bits become determined with progressively greater certainty and the range of the search is reduced.

## A.6 Taking weight error into account

The cost function described in Section A.2 makes the assumption that the error on the weighing device is negligible. Any significant error on the weighing device would have the effect of changing the PDF of the measured fruit weights and, more importantly, of increasing the variance and ultimately the packing cost.



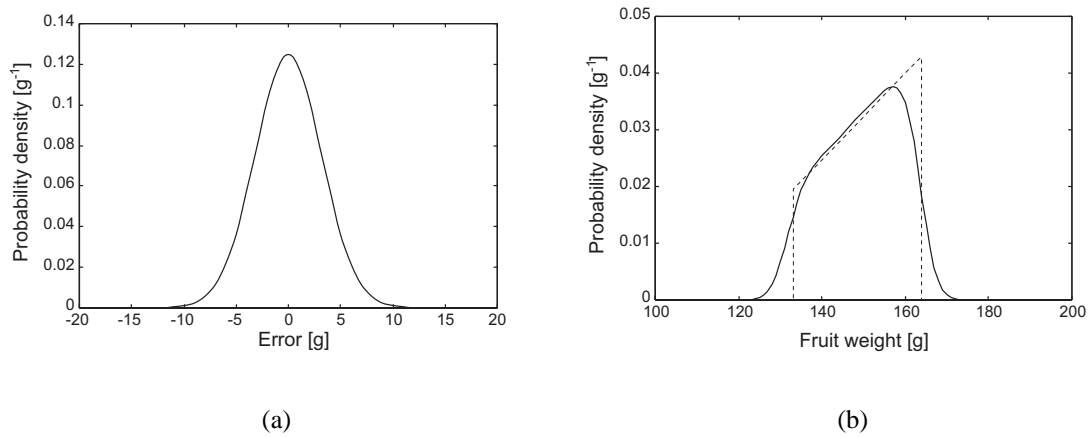


Figure A.5: An example showing (a) an error function for a weighing mechanism, and (b) the actual PDF of a category with the original PDF shown dashed.

Figure A.5(a) shows an example of an error distribution function with RMS error or standard deviation of 3.2g (the weighing mechanism used by Miller [23] for weighing oranges had such an RMS error). The error is assumed to be normally distributed. Since the error distribution PDF is independent of the original PDF, the actual PDF of the box weights can be found by convolving the original PDF (see example in Figure A.1(a)) with the error distribution PDF (such as the example in Figure A.5(a)). Figure A.5(b) shows the result of this convolution. Note how this PDF (drawn as a solid line) has a greater variance than the original PDF (drawn as a dotted line). The new PDF has greater spread since some of the fruit weights may fall outside the category due to measurement error. In the example case in Figure A.5(b), 7.3% of the weights actually fall outside the range defined by the thresholds.

To implement PBIL taking the RMS error on the fruit weights into account, it is necessary to calculate the variance of the new fruit weight PDF, so that the variance,  $\sigma_i^2$ , on the box weight PDFs for each category,  $i$ , can be determined. Since both the original PDF, and the error distribution PDF can only be estimated from sample values, the variance is again calculated using sample values. Sample error distribution values can be obtained when a weighing mechanism or machine vision weight-from-volume estimate is tested for accuracy. The actual fruit weight PDF variance is not a function of the variances of the original PDF and the error function PDF, but is dependent on the shapes of these two distributions. To estimate the variance of the actual fruit weight PDF for the cost function, each weight sample has a randomly selected error distri-

bution sample subtracted from it. This subtraction simulates the error on the weight determination method.

A simple machine vision fruit volume estimator might produce an RMS error of  $7.1\text{cm}^3$  for oranges (as shown in Table 7.4). A histogram of the distribution of volume errors for such a system using the data gathered in this dissertation is shown in Figure A.6(a). A weight estimate can be obtained by multiplying the volume estimate by the mean density of the fruit. During the acquisition stage of this project, oranges were found to have a mean density of  $0.928\text{g}/\text{cm}^3$  (with a standard deviation of  $0.026\text{g}/\text{cm}^3$ ). An RMS error of  $8.7\text{g}$  was obtained when predicting weight from estimated volume. A histogram of the distribution of the weight errors is shown in Figure A.6(b).

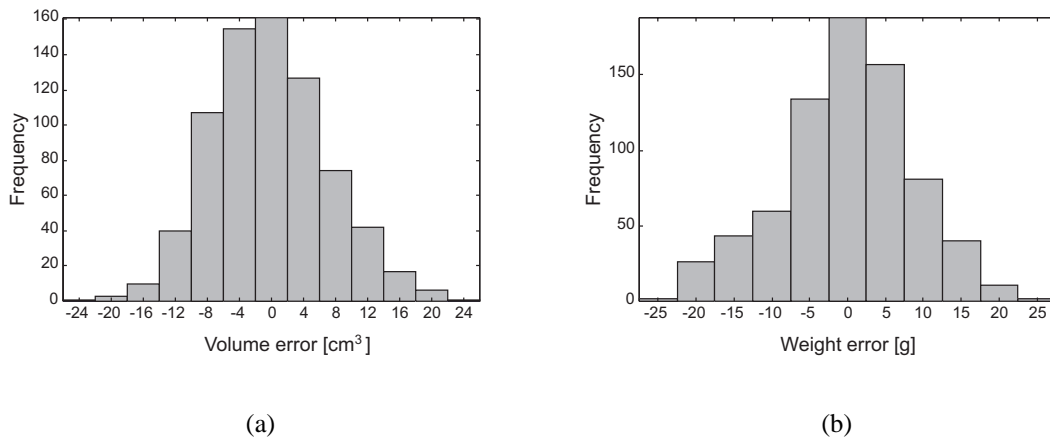


Figure A.6: Histogram of error distribution for (a) orange volume estimation, and (b) orange weight estimation.

To demonstrate the effect of a weight error on the costs of under- or over-packing, PBIL was re-run on the Valencia orange data, this time taking the derived weight error distribution into account. With the PBIL parameters unchanged, a cost of R0.80 was obtained after 310 generations.

Although there are many issues concerning the packing problem that have not been addressed in this appendix (such as the practical issues involved in obtaining a random weight sample from a population of fruit to be packed), it is interesting to note that the results of this appendix indicate that Bridge Fruit could reduce their packing costs from an expected R1.48 per box to an expected R0.72 per box by optimising weight threshold values. It is also interesting to note that this cost would only rise to R0.80

per box (which is still substantially lower than the present R1.48 per box) if a simple weight-from-volume estimator was used to estimate the orange weights from existing digital images of the fruit instead of through the use of an accurate and expensive mechanical weighing device.

# Appendix B

## Volume Estimation Results Tables

For convenience, the results of the volume estimation methods developed in this project are repeated here. This includes error and associated confidence statistics for each of the volume estimation methods. The results are presented for each fruit type, both with the volume distribution of the original data set and with the redistributed volumes (mean of  $200\text{cm}^3$  and standard deviation of  $50\text{cm}^3$ ) for direct comparison of different fruit types and for a better indication of the performance to be expected of different estimation methods used on a typical packing house volume distribution.

Table B.1 gives the results of the linear volume estimation methods, while Table B.2 gives the results of multi-layer perceptron volume estimation methods as described in the body of this dissertation.

Estimation Method	Statistic	Orange ( $n = 31$ )		Pear ( $n = 64$ )		Lemon ( $n = 23$ )		Potato ( $n = 20$ )	
		Orig.	Redist.	Orig.	Redist.	Orig.	Redist.	Orig.	Redist.
Mean volume	RMS%E [%]	23.2	30.4	4.1	33.5	27.6	34.3	18.9	30.7
	RMSE [cm <sup>3</sup> ]	51.7	<b>49.4</b>	7.7	<b>49.7</b>	58.6	<b>48.9</b>	47.3	<b>48.7</b>
	LO95%CI [cm <sup>3</sup> ]	40.7	32.7	5.6	40.3	47.4	29.2	19.0	35.6
	UP95%CI [cm <sup>3</sup> ]	60.7	61.6	9.4	57.6	68.0	62.6	64.1	58.9
	MSE [cm <sup>6</sup> ]	2674.1	2435.5	59.6	2469.3	3438.0	2386.6	2236.6	2370.6
	STD(MSE) [cm <sup>6</sup> ]	2881.4	3876.1	114.7	3456.9	2749.1	3754.1	4005.5	2353.9
Area	RMS%E [%]	3.7	4.0	3.7	7.2	8.4	8.5	12.3	14.3
	RMSE [cm <sup>3</sup> ]	8.6	<b>8.0</b>	7.0	<b>14.8</b>	18.5	<b>16.3</b>	31.3	<b>27.0</b>
	LO95%CI [cm <sup>3</sup> ]	7.6	6.8	5.5	13.0	16.5	14.5	20.9	24.0
	UP95%CI [cm <sup>3</sup> ]	9.5	9.1	8.2	16.5	20.3	17.9	38.9	29.7
	MSE [cm <sup>6</sup> ]	73.6	64.5	49.0	220.3	342.6	265.2	976.3	727.9
	STD(MSE) [cm <sup>6</sup> ]	46.5	52.6	76.1	207.0	161.1	129.5	1152.3	328.2
Area <sup>3/2</sup>	RMS%E [%]	3.7	3.8	3.7	7.0	8.2	9.6	12.1	14.3
	RMSE [cm <sup>3</sup> ]	8.7	<b>7.7</b>	7.0	<b>14.2</b>	18.4	<b>16.7</b>	30.7	<b>27.3</b>
	LO95%CI [cm <sup>3</sup> ]	7.6	6.6	5.5	12.8	16.2	15.2	21.1	23.8
	UP95%CI [cm <sup>3</sup> ]	9.7	8.6	8.2	15.5	20.3	18.2	37.9	30.4
	MSE [cm <sup>6</sup> ]	75.3	59.1	49.1	202.7	338.0	280.0	939.6	744.1
	STD(MSE) [cm <sup>6</sup> ]	51.1	44.1	76.8	155.7	173.3	116.9	1057.4	378.7
Area with $K$ -estimator	RMS%E [%]	3.4	3.4	5.4	5.4	5.0	5.0	16.2	16.2
	RMSE [cm <sup>3</sup> ]	8.4	<b>7.1</b>	10.3	<b>10.9</b>	11.9	<b>10.2</b>	40.5	<b>35.3</b>
	LO95%CI [cm <sup>3</sup> ]	7.2	6.1	9.5	9.9	9.1	8.5	24.0	12.4
	UP95%CI [cm <sup>3</sup> ]	9.3	7.9	11.1	11.8	14.2	11.6	52.0	48.3
	MSE [cm <sup>6</sup> ]	69.7	50.0	106.8	117.8	141.3	103.0	1638.4	1243.7
	STD(MSE) [cm <sup>6</sup> ]	49.5	36.0	67.1	83.0	137.3	72.6	2273.0	2329.1
2 images from mirrors at 90°	RMS%E [%]	2.2	2.2	3.4	3.4	3.9	3.9	9.8	9.8
	RMSE [cm <sup>3</sup> ]	5.6	<b>4.5</b>	6.6	<b>7.0</b>	9.1	<b>7.8</b>	24.9	<b>22.1</b>
	LO95%CI [cm <sup>3</sup> ]	4.6	3.9	5.8	6.1	6.4	6.1	12.7	3.9
	UP95%CI [cm <sup>3</sup> ]	6.4	5.0	7.3	7.8	11.1	9.2	32.9	31.0
	MSE [cm <sup>6</sup> ]	31.0	20.3	43.3	48.9	82.8	60.6	620.4	488.3
	STD(MSE) [cm <sup>6</sup> ]	28.4	13.6	37.9	45.9	95.6	54.7	980.8	1011.4
All 4 mirror chamber images	RMS%E [%]	1.5	1.5	2.4	2.4	3.6	3.6	6.3	6.3
	RMSE [cm <sup>3</sup> ]	4.0	<b>3.2</b>	4.7	<b>5.1</b>	8.4	<b>7.2</b>	16.1	<b>14.5</b>
	LO95%CI [cm <sup>3</sup> ]	3.1	2.6	3.7	4.1	5.7	5.3	3.0	0.0
	UP95%CI [cm <sup>3</sup> ]	4.8	3.6	5.4	5.9	10.5	8.6	22.6	21.6
	MSE [cm <sup>6</sup> ]	16.1	10.0	21.6	25.5	70.7	51.1	259.7	208.9
	STD(MSE) [cm <sup>6</sup> ]	18.9	9.0	31.2	36.2	89.7	52.4	535.0	548.6
All 24 images of the fruit	RMS%E [%]	0.9	1.1	1.8	1.9	3.3	3.2	4.0	4.5
	RMSE [cm <sup>3</sup> ]	2.2	<b>2.3</b>	3.4	<b>3.9</b>	7.3	<b>6.3</b>	10.4	<b>10.1</b>
	LO95%CI [cm <sup>3</sup> ]	1.4	1.5	2.3	2.8	5.2	6.0	4.6	2.4
	UP95%CI [cm <sup>3</sup> ]	2.8	2.8	4.3	4.8	9.0	7.8	14.0	14.0
	MSE [cm <sup>6</sup> ]	4.9	5.1	11.6	15.4	53.5	39.2	108.5	101.4
	STD(MSE) [cm <sup>6</sup> ]	8.3	8.0	26.7	31.4	61.6	51.0	187.0	204.2

Table B.1: Errors on volume estimates using linear regression methods.

Estimation Method	Statistic	Orange ( $n = 31$ )		Pear ( $n = 64$ )		Lemon ( $n = 23$ )		Potato ( $n = 20$ )	
		Orig.	Redist.	Orig.	Redist.	Orig.	Redist.	Orig.	Redist.
Area and shape feature	RMS%E [%]	3.3	3.5	3.6	6.1	4.5	4.9	11.8	10.4
	RMSE [ $\text{cm}^3$ ]	7.9	<b>6.8</b>	6.7	<b>10.7</b>	9.9	<b>9.4</b>	29.8	<b>26.4</b>
	LO95%CI [ $\text{cm}^3$ ]	6.8	6.1	5.2	9.7	8.1	8.0	20.8	17.8
	UP95%CI [ $\text{cm}^3$ ]	8.9	7.5	7.9	11.7	11.3	10.6	36.7	32.8
	$\overline{\text{MSE}}$ [ $\text{cm}^6$ ]	62.8	46.8	44.6	115.3	97.1	88.3	888.8	697.0
	STD(MSE) [ $\text{cm}^6$ ]	48.6	26.7	71.4	90.7	71.8	57.8	972.0	810.2
2 randomly selected images	RMS%E [%]	2.3	2.6	3.1	3.8	3.7	3.5	11.2	14.6
	RMSE [ $\text{cm}^3$ ]	5.5	<b>5.1</b>	5.9	<b>7.6</b>	7.6	<b>6.6</b>	29.4	<b>23.8</b>
	LO95%CI [ $\text{cm}^3$ ]	4.8	4.6	4.8	6.8	6.7	5.8	17.8	15.9
	UP95%CI [ $\text{cm}^3$ ]	6.2	5.6	6.9	8.4	8.5	7.4	37.6	29.6
	$\overline{\text{MSE}}$ [ $\text{cm}^6$ ]	30.3	25.9	35.0	58.4	57.9	43.6	865.8	565.9
	STD(MSE) [ $\text{cm}^6$ ]	21.1	14.6	50.9	52.0	31.3	24.3	1172.4	668.3
2 images from mirrors at $90^\circ$	RMS%E [%]	2.1	2.3	2.5	3.2	2.4	2.4	8.7	9.2
	RMSE [ $\text{cm}^3$ ]	4.8	<b>4.5</b>	4.8	<b>6.4</b>	5.4	<b>4.8</b>	21.6	<b>20.5</b>
	LO95%CI [ $\text{cm}^3$ ]	4.2	3.9	4.2	5.7	4.3	3.9	16.3	13.4
	UP95%CI [ $\text{cm}^3$ ]	5.3	4.9	5.4	7.1	6.2	5.6	25.8	25.8
	$\overline{\text{MSE}}$ [ $\text{cm}^6$ ]	23.0	19.8	23.3	41.3	28.7	23.5	465.0	421.5
	STD(MSE) [ $\text{cm}^6$ ]	15.4	12.4	24.7	37.9	23.7	19.1	429.2	518.3
4 randomly selected images	RMS%E [%]	1.8	2.2	2.6	2.9	3.3	3.3	9.1	9.5
	RMSE [ $\text{cm}^3$ ]	4.2	<b>4.1</b>	5.0	<b>5.8</b>	7.1	<b>6.8</b>	23.9	<b>21.2</b>
	LO95%CI [ $\text{cm}^3$ ]	3.6	3.7	4.1	5.0	5.9	4.0	14.9	14.6
	UP95%CI [ $\text{cm}^3$ ]	4.7	4.5	5.7	6.5	8.1	8.8	30.3	26.3
	$\overline{\text{MSE}}$ [ $\text{cm}^6$ ]	17.5	17.0	24.6	33.4	49.9	46.8	569.4	451.3
	STD(MSE) [ $\text{cm}^6$ ]	13.1	8.7	32.5	36.2	34.5	70.6	742.2	507.9
All 4 mirror chamber images	RMS%E [%]	1.3	1.7	1.9	2.0	1.6	1.7	5.3	5.6
	RMSE [ $\text{cm}^3$ ]	2.9	<b>3.2</b>	3.7	<b>4.1</b>	3.2	<b>3.6</b>	14.1	<b>12.6</b>
	LO95%CI [ $\text{cm}^3$ ]	2.5	2.7	2.9	3.2	2.7	2.0	8.2	2.2
	UP95%CI [ $\text{cm}^3$ ]	3.3	3.5	4.3	4.8	3.7	4.8	18.2	17.6
	$\overline{\text{MSE}}$ [ $\text{cm}^6$ ]	8.5	10.0	13.7	16.5	10.5	13.2	198.9	158.2
	STD(MSE) [ $\text{cm}^6$ ]	6.2	7.4	21.0	26.3	7.9	21.8	280.1	327.2

Table B.2: Errors on volume estimates using MLP function approximation.

# Bibliography

- [1] Autoline Inc. Fruit Sorting Systems. <http://www.autolinesorters.com>.
- [2] Shumeet Baluja. Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning. Technical report, Carnegie Mellon University, June 1994.
- [3] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.
- [4] E. R. Davies. *Machine Vision*. Academic Press, 2nd edition, 1997.
- [5] Stanley M. Dunn, Richard L. Keizer, and Jongdaw Yu. Measuring the area and volume of the human body with structured light. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(6), 1989.
- [6] Ellips, Electronic Fruit Sizers. <http://www.ellips.nl/fruit.html>.
- [7] K. A. Forbes and G. M. Tattersfield. An investigation into the volumetric determination of apples using machine vision techniques. In D. M. Weber, B. Herbst, and J. A. du Preez, editors, *Proceedings of the Ninth Annual South African Workshop on Pattern Recognition*, pages 55–59. PRASA, November 1998.
- [8] K. A. Forbes and G. M. Tattersfield. Estimating fruit volume from digital images. In *Africon*, volume 1, pages 107–112. IEEE, September 1999.
- [9] K. A. Forbes and G. M. Tattersfield. Volumetric determination of apples using machine vision techniques. *Elektron*, 16(3):14–17, March 1999.
- [10] Rafael C. Gonzalez and Paul Wintz. *Digital Image Processing*. Addison-Wesley Publishing Company, Inc., 1979.

- [11] J. R. Greene. Generalisation in neural networks with monotonic activation functions and fixed, randomly-assigned hidden weights. In *Proceedings of the Eighth Annual South African Workshop on Pattern Recognition*, pages 98–104. PRASA, 1997.
- [12] Kevin Gurney. *An Introduction to Neural Networks*. UCL Press, 1997.
- [13] Simon Haykin. *Neural Networks A Comprehensive Foundation*. Tom Robbins, 2nd edition, 1999.
- [14] Karen Henry. An investigation of a pattern recognition system to analyse and classify dried fruit. Master's thesis, University of Cape Town, 1996.
- [15] Horticultural Automation Limited. <http://www.hortauto.co.nz>.
- [16] Bernd Jähne. *Digital Image Processing*. Springer-Verlag, 2nd edition, 1993.
- [17] Ramesh Jain, Rangachar Kasturi, and Brian Schunck. *Machine Vision*. McGraw-Hill, Inc., 1995.
- [18] Gary R. Kay. Colour analysis and the classification of fruit. Master's thesis, University of Cape Town, 1992.
- [19] A. Laing, Q. Smit, B. Mortimer, and J. Tapson. Ultrasonic fruit sizing device. *Journal of the South African Acoustics Institute*, 6:60–65, 1995.
- [20] Alberto Leon-Garcia. *Probability and Random Processes for Electrical Engineering*. Addison-Wesley, second edition, May 1994.
- [21] J. A. Marchant. A mechatronic approach to produce grading. In *Mechatronics: Designing intelligent machines*, pages 159–164, 1990.
- [22] Irwin Miller, John E. Freud, and Richard A. Johnson. *Probability and Statistics for Engineers*. Prentice-Hall, fourth edition, 1990.
- [23] William M. Miller, Kalman Peleg, and Patrick Briggs. Automatic density separation for freeze-damaged citrus. *Applied Engineering in Agriculture*, 4(4):344–348, December 1988.
- [24] Norman Morrison. *Introduction to Fourier Analysis*. John Wiley and Sons, Inc., 1994.



- [25] Alan V. Oppenheim and Ronald W. Schaffer. *Discrete-Time Signal Processing*. Prentice-Hall, Inc., 1989.
- [26] Dan W. Patterson. *Artificial Neural Networks Theory and Applications*. Prentice Hall, 1996.
- [27] F. Pla and F. Juste. A thinning-based algorithm to characterize fruit stems from profile images. *Computers and Electronics in Agriculture*, 13(4):301–314, 1995.
- [28] Theresa Pratt. From photo to 3D model. *IEE Review*, 46(1):9–12, January 2000.
- [29] H. K. Purwadari and I. W. Budiastira. Computer-controlled on-line system for mango grading using image processing and NIR measurement. In I. Farkas, editor, *Second International Symposium on Mathematical Modelling and Simulation in Agriculture and Bio-Industries*, pages 259–264, May 1997.
- [30] M. Recce, A. Plebe, J. Taylor, and G. Tropicano. Video grading of oranges in real-time. *Artificial Intelligence Review*, 12:117–136, 1998.
- [31] G. E. Rehkugler and J. A. Throop. Apple sorting with machine vision. *Transactions of the American Society of Agricultural Engineers*, 29(5):1388–1397, 1986.
- [32] J. C. Russ. *The Image Processing Handbook*. CRC Press, Inc. and IEEE Press, 1995.
- [33] Paul Stanford. Bridge Fruit, postal correspondance.
- [34] James Stewart. *Calculus*. Brooks/Cole Publishing Company, second edition, 1991.
- [35] B. Sutton. *Solution of Problems in Soil Mechanics*. Pitman, 1975.
- [36] Yang Tao. Spherical transform of fruit images for on-line defect extraction of mass objects. *Optical Engineering*, 35(2):344–350, February 1996.
- [37] R. D. Tillet, C. M. Onyango, P. F. Davis, and J. A. Marchant. Image analysis for biological objects. In *Third International Conference on Image Processing and its Applications*, pages 207–211, July 1989.
- [38] Bernard van Zyl. Vizier Systems Pty. Ltd., personal communication.

- [39] W. F. Wardowski, W. M. Miller, and W. Grierson. Separation and grading of freeze-damaged citrus fruits. Circular 372, Institute of Food and Agricultural Sciences, University of Florida, October 1997.
- [40] Arthur R. Weeks. *Fundamentals of Electronic Image Processing*. IEEE Press, 1996.
- [41] Q. Yang. Finding stalk and calyx of apples using structured lighting. *Computers and Electronics in Agriculture*, 8:31–42, 1993.