

University of Warwick institutional repository: <http://go.warwick.ac.uk/wrap>

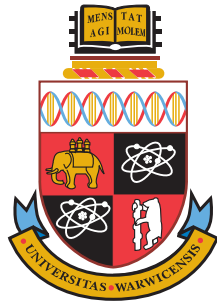
**A Thesis Submitted for the Degree of PhD at the University of Warwick**

<http://go.warwick.ac.uk/wrap/2032>

This thesis is made available online and is protected by original copyright.

Please scroll down to view the document itself.

Please refer to the repository record for this item for information to help you to cite it. Our policy information is available from the repository home page.



# Modelling and Analysis of Plant Image Data for Crop Growth Monitoring in Horticulture

Yu Song, B.Eng.

A doctorate thesis submitted  
in partial fulfilment of the requirements  
for the degree of

Doctor of Philosophy

Warwick HRI & Department of Computer Science

December 19, 2008

THE UNIVERSITY OF  
WARWICK

# Abstract

Plants can be characterised by a range of attributes, and measuring these attributes accurately and reliably is a major challenge for the horticulture industry. The measurement of those plant characteristics that are most relevant to a grower has previously been tackled almost exclusively by a combination of manual measurement and visual inspection. The purpose of this work is to propose an automated image analysis approach in order to provide an objective measure of plant attributes to remove subjective factors from assessment and to reduce labour requirements in the glasshouse.

This thesis describes a stereopsis approach for estimating plant height, since height information cannot be easily determined from a single image. The stereopsis algorithm proposed in this thesis is efficient in terms of the running time, and is more accurate when compared with other algorithms.

The estimated geometry, together with colour information from the image, are then used to build a statistical plant surface model, which represents all the information from the visible spectrum. A self-organising map approach can be adopted to model plant surface attributes, but the model can be improved by using a probabilistic model such as a mixture model formulated in a Bayesian framework. Details of both methods are discussed in this thesis.

A Kalman filter is developed to track the plant model over time, extending the model to the time dimension, which enables smoothing of the noisy measurements to produce a development trend for a crop. The outcome of this work could lead to a number of potentially important applications in horticulture.

# Contents

<b>Contents</b>	<b>i</b>
<b>List of Tables</b>	<b>v</b>
<b>List of Figures</b>	<b>vi</b>
<b>Acknowledgements</b>	<b>ix</b>
<b>Declarations</b>	<b>xi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Modern Horticulture . . . . .	1
1.1.1 Crop Growth Monitoring . . . . .	2
1.1.2 Plant Quality Assessment . . . . .	6
1.2 Digital Imaging Techniques . . . . .	7
1.3 Current Image Analysis Methods in Horticulture . . . . .	8
1.4 Aims of The Thesis . . . . .	11
1.4.1 Crop Growth Monitoring . . . . .	11
1.4.2 Plant Quality Assessment . . . . .	12
1.4.3 Modelling and Analysis . . . . .	12
1.5 Contributions . . . . .	13
1.6 Equipment . . . . .	13
1.7 Thesis Organisation . . . . .	14
<b>2 Measuring Plant Height from Stereo Images</b>	<b>15</b>
2.1 Introduction . . . . .	15



2.2	Approaches to Plant Height Measurement . . . . .	16
2.2.1	Measuring Height from Side Views . . . . .	16
2.2.2	Active 3D Vision . . . . .	18
2.2.3	Passive 3D vision . . . . .	19
2.3	Stereoscopic Vision Setup . . . . .	20
2.4	Measuring Height from Stereoscopic Images . . . . .	26
2.5	A Multiresolution Model for Stereopsis . . . . .	31
2.5.1	The Quadtree Structure . . . . .	31
2.5.2	The Kalman Update . . . . .	33
2.6	Disparity Estimation . . . . .	36
2.6.1	Matching . . . . .	36
2.6.2	Multiresolution Implementation . . . . .	40
2.6.3	Parent Weighting . . . . .	40
2.7	Results . . . . .	43
2.7.1	Stereo Results on a Known-Geometry Object . . . . .	44
2.7.2	Stereo Results for Plants . . . . .	45
2.8	Discussion . . . . .	60
2.9	Summary . . . . .	62
<b>3</b>	<b>A Self-Organising Map Model for Plant Surface Attributes</b>	<b>63</b>
3.1	Introduction . . . . .	63
3.2	Plant Modelling . . . . .	64
3.3	Self-Organising Map: a Neural Network Approach . . . . .	66
3.4	Competitive Learning . . . . .	66
3.5	The Neighbourhood function . . . . .	68
3.6	The Learning Rate . . . . .	69
3.7	Tree-Structured SOM . . . . .	71
3.8	Results . . . . .	73
3.9	Discussion . . . . .	79
3.10	Summary . . . . .	82
<b>4</b>	<b>Statistical Modelling of Plant Surface Attributes</b>	<b>83</b>
4.1	Introduction . . . . .	83

4.2	The K-means Algorithm . . . . .	84
4.3	Gaussian Mixture Model: a Probabilistic Model Approach . .	85
4.4	MGMM: Multiresolution Gaussian Mixture Model . . . . .	86
4.5	Estimation of Mixture Distributions . . . . .	87
4.6	Gibbs Sampling: a Bayesian Approach . . . . .	88
4.7	Model Split and Selection . . . . .	91
4.8	Speed and Efficiency . . . . .	93
4.9	Results . . . . .	94
4.9.1	Modelling Results for a set of Gaussians Data . . . . .	94
4.9.2	Modelling Results for Shape Data . . . . .	106
4.9.3	Modelling Results for Pansy Data . . . . .	108
4.10	Discussion . . . . .	108
4.11	Summary . . . . .	113
<b>5</b>	<b>Monitoring Plant Growth Over Time</b>	<b>115</b>
5.1	Introduction . . . . .	115
5.2	Plant Growth Monitoring . . . . .	116
5.3	The State Space Formation . . . . .	117
5.3.1	The Process Model . . . . .	118
5.3.2	The Observation Model . . . . .	118
5.4	Kalman Filter: State-Space Filtering . . . . .	119
5.5	Parameter Configuration . . . . .	121
5.5.1	The Process Noise Covariance . . . . .	125
5.5.2	The Measurement Noise Covariance . . . . .	127
5.5.3	The Initial State Noise Covariance . . . . .	129
5.6	Kalman Filter for Plant Attributes . . . . .	133
5.6.1	Height . . . . .	134
5.6.2	Population . . . . .	142
5.6.3	Colour . . . . .	143
5.7	Discussion . . . . .	146
5.8	Summary . . . . .	152

<b>6</b>	<b>Conclusions</b>	<b>153</b>
6.1	Summary . . . . .	153
6.1.1	Plant Height Measurement . . . . .	153
6.1.2	Modelling of Plant Attributes . . . . .	155
6.1.3	Tracking Plant Attributes Over Time . . . . .	156
6.2	Limitations and Further Work . . . . .	157
6.2.1	Stereopsis . . . . .	157
6.2.2	Multiresolution Gaussian Mixture Model . . . . .	158
6.2.3	Kalman Filter . . . . .	159
6.2.4	High-Dimensional Data Analysis . . . . .	159
6.2.5	Other Sources of Information . . . . .	160
6.3	Concluding Remarks . . . . .	161
	<b>Appendix</b>	<b>161</b>
<b>A</b>	<b>A Brief Introduction to Bayesian Mixture Modelling</b>	<b>162</b>
A.1	Gaussian Mixture Model . . . . .	162
A.2	Missing Data Structure . . . . .	163
A.3	Conjugate Priors . . . . .	164
A.4	MCMC: Posterior Simulation . . . . .	165
<b>B</b>	<b>Publication</b>	<b>167</b>
	<b>Bibliography</b>	<b>168</b>

# List of Tables

2.1	Summary of the multiresolution stereo algorithm . . . . .	43
2.2	Height comparisons for a known-geometry object . . . . .	45
2.3	Average height comparisons for plants . . . . .	48
2.4	Error comparisons for plants . . . . .	48
3.1	Summary of the tree-structured SOM algorithm . . . . .	72
3.2	Performance comparisons for plants. . . . .	75
4.1	Performance comparisons for data. . . . .	93
4.2	Summary of the Multiresolution Gaussian Mixture Model . . .	95
4.3	Performance comparisons for Gaussian data. . . . .	96
4.4	Performance comparisons for shape data. . . . .	107
4.5	Performance comparisons for <i>Pansy</i> data. . . . .	107
5.1	Summary of the Kalman filter . . . . .	121
5.2	The Kalman filter with different noise covariances . . . . .	133
5.3	Height comparisons at the time $t = 3$ and $t = 6$ . . . . .	147
5.4	Population comparisons at the time $t = 16, t = 17$ and $t = 18$ .	148
5.5	RMS comparisons between curve fitting and Kalman filter. . .	148

# List of Figures

1.1	Aalsmeer flower auction. . . . .	3
1.2	A glasshouse used by Coletta & Tyson Ltd. . . . .	4
1.3	A glasshouse used by DoubleH Ltd. . . . .	4
1.4	A team of assessors in Roundstone Nurseries. . . . .	5
1.5	An <i>Anthurium</i> sorting machine . . . . .	9
1.6	An autonomous robot for rose cutting. . . . .	9
2.1	An illustrative image of <i>Poinsettia</i> and <i>Pansy</i> . . . . .	17
2.2	Plants are grown on a bench during production . . . . .	17
2.3	Parallel stereo camera setup in a glasshouse . . . . .	21
2.4	‘Lightbox’ setup . . . . .	22
2.5	Moving camera rig scanning a crop from above . . . . .	23
2.6	Model of the stereopsis process . . . . .	24
2.7	A stereoscopic pair of the checkerboard images . . . . .	25
2.8	Camera positions using the lightbox setup . . . . .	27
2.9	A pair of stereoscopic images captured using the lightbox . . .	29
2.10	A 2-level model based on the quadtree data structure . . . . .	33
2.11	An illustration of parent weighting . . . . .	42
2.12	Stereo results on a known-geometry object . . . . .	46
2.13	The left views of the 10-pair plant images . . . . .	47
2.14	Stereo results for <i>Pansy</i> , pack 1, stage 1 . . . . .	49
2.15	Reconstructed views of <i>Pansy</i> , pack 1, stage 1 . . . . .	50
2.16	Stereo results for <i>Viola</i> , pack 1, stage 1 . . . . .	51
2.17	Reconstructed views of <i>Viola</i> , pack 1, stage 1 . . . . .	52
2.18	Stereo results for <i>Cyclamen</i> , stage 2 . . . . .	53

2.19	Reconstructed views of <i>Cyclamen</i> , stage 2 . . . . .	54
2.20	Average height measurements for <i>Pansy</i> , pack 1. . . . .	55
2.21	Average height measurements for <i>Pansy</i> , pack 2. . . . .	56
2.22	Average height measurements for <i>Viola</i> , pack 1. . . . .	57
2.23	Average height measurements for <i>Viola</i> , pack 2. . . . .	58
2.24	Average height measurements for <i>Cyclamen</i> . . . . .	59
3.1	Neighbourhood radius $\sigma(t)$ decreases over time. . . . .	70
3.2	Shapes of the two neighbourhood functions. . . . .	70
3.3	Learning rate decreases over time. . . . .	71
3.4	<i>Pansy</i> data and SOM nodes. . . . .	75
3.5	Evolution of the tree-structured SOM over levels. . . . .	76
3.6	Tree-structured SOM at different levels. . . . .	77
3.7	SOM representation for <i>Pansy</i> . . . . .	78
3.8	SOM representation for <i>Dianthus</i> . . . . .	78
3.9	SOM representation for <i>Viola</i> . . . . .	79
3.10	Neighbourhood problem using SOM. . . . .	81
4.1	Changes in labels against sampling iterations . . . . .	97
4.2	log-likelihood values against EM iterations . . . . .	98
4.3	Evolution of the MGMM at the coarsest level . . . . .	99
4.4	Evolution of the MGMM over levels . . . . .	100
4.5	Results of Gibbs sampling for the Gaussian data. . . . .	101
4.6	Results of EM algorithms for the Gaussian data. . . . .	101
4.7	Results of K-means algorithms for the Gaussian data . . . . .	102
4.8	‘Shape’ data . . . . .	103
4.9	Results of the Gibbs sampling for the ‘shape’ data. . . . .	103
4.10	Results of the EM algorithm for the ‘shape’ data. . . . .	104
4.11	Results of the EM algorithm with random starting values. . . . .	104
4.12	Results of the K-means algorithm for the ‘shape’ data. . . . .	105
4.13	<i>Pansy</i> data . . . . .	109
4.14	Components in the MGMM for the <i>Pansy</i> data. . . . .	109
4.15	Classification of the <i>Pansy</i> data using Gibbs sampling. . . . .	110
4.16	Classification of the <i>Pansy</i> data using EM. . . . .	110

4.17	Classification of the <i>Pansy</i> data using K-means. . . . .	111
5.1	2D $x - y$ position data . . . . .	123
5.2	Noisy observations of the 2D position data. . . . .	124
5.3	Effect of varying the process noise covariance. . . . .	126
5.4	Effect of varying the measurement noise covariance. . . . .	128
5.5	Effect of varying the initial state noise covariance. . . . .	131
5.6	Effect by using different initial state noise covariances . . . . .	132
5.7	Days since the early potting stage against the time $t$ . . . . .	135
5.8	A Kalman process model to represent the leaf height over time	136
5.9	Tracking the leaf height by Kalman filter . . . . .	137
5.10	Tracking the leaf height by curve fitting. . . . .	138
5.11	Tracking the leaf height for Plant 1, Plant 2, Plant 3, Plant 4.	139
5.12	Hand measurements on the leaf height . . . . .	140
5.13	Tracking the flower height by Kalman filter . . . . .	141
5.14	Tracking the leaf population by Kalman filter. . . . .	143
5.15	Tracking the population of other objects by Kalman filter. . .	144
5.16	Tracking the leaf colour by Kalman filter. . . . .	145
5.17	Images collected at the time $t = 3$ and $t = 6$ . . . . .	150
5.18	Input images at the time $t = 16, 17, 18$ . . . . .	151

# Acknowledgements

My thanks go first and foremost to my supervisors **Prof. Roland Wilson** and **Mr Rodney Edmondson**. Without their guidance and care, neither my work nor the completion of this thesis would be possible. Roland patiently introduced techniques of computer vision and statistical image analysis to me, and I am grateful for his ideas and suggestions over the years, and for his support during the writing process. Rodney inspired me with practical applications in horticulture, and spent a lot of efforts on the imaging setup. He also arranged all my industrial visits, and encouraged me to present my work to growers.

Thanks to my PhD advisory panel members, Prof. Richard Napier, Dr Debbie Fuller and Dr Nasir Rajpoot, for their comments on my PhD progress both formally and informally. Nasir also acted as a panel member in Computer Science to review my theoretical work. I would also like to express my appreciation for the support of Dr Nick Parsons, for his effort and time to help me.

The members of the Signal and Image Processing Research Group, including Dr Abhir Bhalerao and Dr Chang-Tsun Li, have been supportive to my work. I would like thank all my lab mates, especially Adam Bowen, Andrew Mullins and Adam Hartshorne, for their help to start my research in new areas, and Thomas Popham, for his help to correct my thesis.

Staff at Warwick HRI, especially Dr Steven Adams and Dr Allen Langton, explained the science behind growing plants to me, and gave me an insight into plant attributes. Thanks to Dr Veronica Valdes, for her help on trials at Warwick HRI. Many thanks go to the people I met at Wageningen University, especially Dr Jochen Hemming and Dr Henk Jalink.



Without Mr Martin Holdsworth, there would not be an easy-to-assemble stereo setup I could use for my work. Dr Roger Packwood and Dr Rod Moore discussed how to make a practical setup for stereopsis with me, and Mr Richard Cunningham helped me whenever I ran into trouble with the Linux machine in the office. Thanks to Miss Nikki Glover, for her help on various academic and non-academic matters.

Thanks to people I met at DoubleH Ltd, who kindly helped me during my visits to their glasshouses. In particular, Mr Mike Holmes gave me enormous support, and I am grateful for his advice on my work.

Ms Fay Richardson and Mr Anthony Hawkins gave me opportunities to visit commercial nurseries. Mr Alan Davis, a HDC consultant, and Dr Ruth Finlay, former HDC technical manager for the Protected Crops sector, were both supportive to my work, and helped to develop the idea of image analysis in horticulture.

My friends and house mates made my PhD life an interesting and memorable journey, and I thank all of you for sharing my happiness and helping me whenever I need. Most of all, I thank all of my family for their encouragement, especially my parents, **Mr Song** and **Mrs Xing**. Without their love and support, none of this would have been possible.

This work was funded by the Horticultural Development Company (HDC), formally known as the Horticultural Development Council, and the contract number was CP 37.

# Declarations

I declare that, except where acknowledged, the material contained in the thesis is my own work, and has not been previously published for obtaining an academic degree.

Yu Song

December 19, 2008

# Chapter 1

## Introduction

### 1.1 Modern Horticulture

People unfamiliar with the horticulture industry can easily fail to appreciate the scale of this modern industry. Aalsmeer Flower Auction, a short distance outside Amsterdam, is the world's largest flower auction site, where nineteen million flowers and two million plants are sold every day. It is also the largest trade building in the world, covering an area of almost one million square metres <sup>1</sup>, 1.5 times the area of the Pentagon [86].

In the UK, horticulture is an important industry. In 2007, the combined amenity and production workforce was more than 300,000, and horticulture's contribution to the UK GDP was in excess of £15 billion per year <sup>2</sup>. The UK public spent £1.4 billion on ornamental plants and seeds, over £20 per person per year on average <sup>3</sup>. Horticulture consequently is highly significant

---

<sup>1</sup>Bloemenveiling Aalsmeer, <http://www.aalsmeer.nl/00004.asp>

<sup>2</sup>Figures were obtained from the Institute of Horticulture, UK

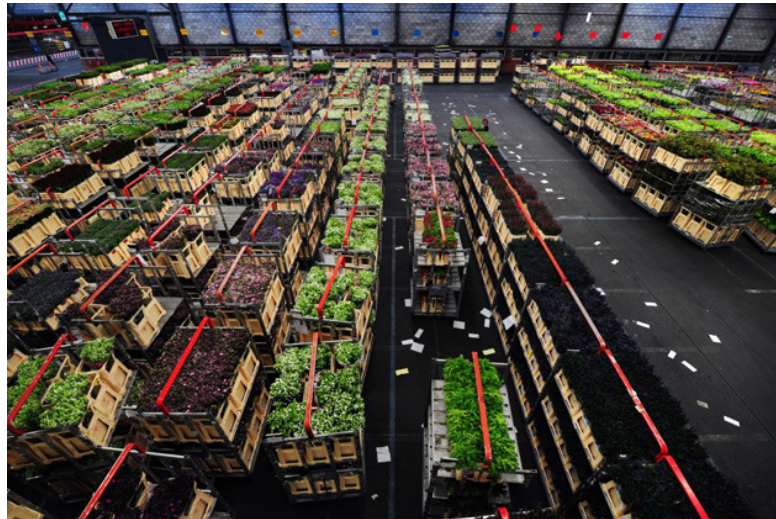
<sup>3</sup>Figures were obtained from the Horticultural Trades Association, UK

for the UK economy.

Despite its rapid growth, the industry is under pressure from reduced profits, environmental constraints and labour issues. The problem of quantifying and measuring plant quality remains a challenge for the industry, and customers and producers could even have different standards for plant quality [39]. The purpose of this thesis is to propose an automated image analysis approach in order to provide an objective and unambiguous measure of plant attributes to remove subjective factors from assessment and to reduce labour requirements in the glasshouse. Two examples of labour-intensive tasks that this work concerns are discussed below.

### 1.1.1 Crop Growth Monitoring

Crop growth monitoring is a demanding task, and crops are monitored throughout growth to ensure that the product is on target for quality and delivery schedule. Modern commercial glasshouses are large and highly automated, but growth monitoring is still largely done manually. Currently, most crop management decisions are based on experience, frequent visual inspection of the crop and fairly crude methods, such as tracking plant heights using hand measurement on samples in growing crops [40]. Moreover, hand measurement by sampling is both time-consuming in large glasshouses and unreliable if done by unskilled labour.



Rows of plants in carts



The auction floor where millions of flowers and plants are sold every day

**Figure 1.1:** Aalsmeer flower auction is the world's largest flower auction site. Image credit: Linda Nyland, Guardian News and Media Ltd.



**Figure 1.2:** A glasshouse used by Coletta & Tyson Ltd, where thousands of *Pansy* plants are produced.



**Figure 1.3:** A glasshouse used by DoubleH Ltd, where *Chrysanthemum* plants are grown in pots on benches.





**Figure 1.4:** A team of assessors in Roundstone Nurseries, who are assessing *Pansy* plants at the marketing stage for adherence to the quality specification.

While direct visual inspection of crops will always be of vital importance in large-scale commercial production, this conventional approach has a number of disadvantages. Certain areas of a crop can be difficult to reach by human hands, for example, in the middle of a ten-metre wide growing bench. This problem is becoming severe because of the increasing pressure to grow more plants in a limited space with a consequent reduction in walkways. Performing hand measurement requires taking plants out from the growing bench, and it is known by growers that frequent human intervention can affect the growing habits of these plant samples and their neighbours. Moreover, these hand measurements can only represent a small number of plant samples, not all the plants in production. Finally, uncertainty and subjective factors are associated with hand measurements. Routine monitoring cannot, therefore, be fully implemented at present, although it is desirable from the crop management point of the view.

### 1.1.2 Plant Quality Assessment

In common with other industries, a quality specification is detailed in a contract between growers and their customers. The specification defines several plant attributes, including size, colour and packaging. The complexity and variability of living plants makes grading of plants in horticulture particularly challenging, especially at the marketing stage, for adherence to the quality specification. Quality assessment is a highly labour-intensive process, given that tens of thousands of plants have to be assessed, and one which is inevitably limited by the subjective assessments of individual observers.



Large-scale plant production is a highly technical operation, involving automated systems, controlled by state-of-the-art computer technology, alongside traditional skills. Automated quality grading in glasshouses can be performed under a highly controlled environment and simple procedures can readily identify attributes such as the maximum plant height or spread [40]. There are some commercial systems discussed in Section 1.3 to measure the plant height, but plant quality depends on more than just the height and spread, which current imaging techniques deal with, and this thesis proposes a more general modelling approach that allows analysis of plant quality attributes such as flower and leaf area, plant uniformity and overall growth habit. In the next section, the use of digital imaging techniques and some examples of its use in the horticulture industry are discussed.

## 1.2 Digital Imaging Techniques

Over the last few years, the rate of development of digital imaging technology has been dramatic. Powerful computers and digital cameras are now available at remarkably low prices [97]. Networking and wireless technology is available which can transfer digital information almost anywhere at any time [105]. Computer vision and image analysis algorithms have been developed that can perform complex tasks on an almost routine basis, and this has led to the use of digital imaging in a wide range of areas, including quality control in industry [73, 113], biometric image analysis for identification purposes [58] and routine monitoring and collection of information for security purposes [110].

The work presented in this thesis is analogous to automated visual inspection applications in factories. Some examples in this area include the measurement of steel bars, inspection of computer screens, colour feature estimation in the ceramic tile industry [73, 113]. According to Vernon [113], automated visual inspection applications in industry are classified as follows:

**Gauging** is concerned with the measurement of dimensional characteristics of parts;

**Inspection** is concerned with performing part verification, that is, determine whether there are any parts missing;

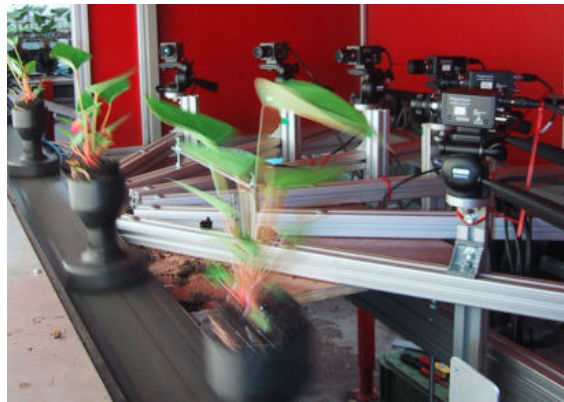
**Sorting** is concerned with the identification and the recognition of parts. Parts are usually on a conveyor-belt system, and they can be simply pushed into an appropriate bin.

### 1.3 Current Image Analysis Methods in Horticulture

In horticulture, some ‘gauging’ and ‘sorting’ applications have been developed, and a commercial system often has both features. However, the number of techniques and systems to perform plant quality assessment is still limited, and methods are fairly crude and simple. Edmondson *et al.* [39] investigated using a single fixed camera to capture packs of *Pansy* and *Viola* plants, and quantify the flower and leaf height from a side view. This is similar to some commercial products. For example, the *Anthurium* sorting machine <sup>4</sup> uses

---

<sup>4</sup>GreenVision, <http://greenvision.wur.nl>



**Figure 1.5:** An *Anthurium* sorting machine developed by GreenVision, which uses six cameras to capture images of individual plants passing through the line. Image credit: GreenVision <http://greenvision.wur.nl>



**Figure 1.6:** An autonomous robot for rose cutting suggested by Noordam and other people at Wageningen University [85]. Image credit: GreenVision <http://greenvision.wur.nl>

six cameras to capture images of individual plants passing through the line, which provides more data from the side than a single view. Another example is the vision system made by WPS Horti Systems <sup>5</sup>, which also places cameras in a controlled environment to capture a side view.

There are some robotics applications for the purpose of harvesting, rather than growth monitoring. Van Henten *et al.* [111] described the concept of an autonomous robot for harvesting cucumbers in greenhouses, which has stereoscopic vision systems for 3D imaging of cucumbers and environment. The vision system identifies a particular cucumber first, and then only matches the features of the cucumber in order to estimate the distance from the robot. An autonomous robot for rose cutting suggested by Noordam *et al.* [85] works in a similar fashion. The complexity of image analysis techniques in these systems lies in the area of object recognition and detection, while the work presented in this thesis concerns with modelling and analysis of attributes extracted from image data.

Visual appearance has been used in assessing the variety or cultivar of crop samples. Davey *et al.* [31] proposed the use of image analysis as a tool for assessing plant uniformity and variety matching. Horgan discussed shape analysis techniques with or without landmarks to summarise the shape of a plant and eigenimage analysis to summarise the colour variation in his review of the statistical analysis of plant appearance [56]. These tools discussed above are used for the purposes of variety classification, rather than crop growth monitoring. A similar problem was approached by Nilsback and Zisserman [83, 84], whose the objective is to classify flowers in images. In

---

<sup>5</sup>WPS Horti Systems, <http://www.wpshortisystems.nl/>

Nilsback and Zisserman’s work [84], an approach to model and combine three different aspects, namely shape, colour and texture, is proposed to distinguish between different flower species. The technique has been demonstrated to work under challenges such as variations in light, view point and scale, and flower deformations, as well as the strong similarities between flower species.

To the best of the author’s knowledge, there is no such system or methodology at all in the area of crop growth monitoring *in situ*. If any cameras are mounted inside the glasshouse, they are used for the purposes of surveillance of the glasshouse rather than of those plants on the growing bench [38].

## 1.4 Aims of The Thesis

Automated systems can be developed for crop growth monitoring and plant quality assessment in a glasshouse using the modelling and analysis techniques proposed in this thesis. The hardware part is ‘off-the-shelf’ digital cameras that can be flexibly mounted somewhere in the glasshouse, for example, on a production grading line or on rails over the growing bench.

### 1.4.1 Crop Growth Monitoring

The monitoring of growing crops in a commercial glasshouse environment is potentially important in a number of areas, including plant scheduling, plant health and growth management. An overhead imaging technique using pairs of stereoscopic images has been proposed in this thesis, and an example of such a system developed at University of Warwick is shown in Chapter 2. This system allows routine and consistent monitoring of the crop *in situ* at

relatively low cost. Monitoring crop development on a day-to-day basis and comparing with the development of similar crops grown previously can give very precise estimates of time to marketing. Immediately prior to marketing, plants can be imaged to estimate quality grades, which can then be used for planning the marketing operation and predicting the labour requirements for a grading line. The information could also be used for estimating the final marketable value of a crop. Since monitoring growing crops *in situ* in a glasshouse has not been implemented yet, there would be more opportunities for horticultural applications when the system becomes available.

#### 1.4.2 Plant Quality Assessment

The system discussed above is also suited to tasks where a large number of plants of the same type pass through a grading line environment, which allows automated assessment of plant quality to be carried out to prevent unsatisfactory products from reaching the customer. In addition, an objective measure can be provided for pots or packs of plants passing through a grading line. The information collected from this process can be used to provide the customer with objective data for the specification, so that unambiguous quality standards can be agreed. This information can also be used for stock control and traceability.

#### 1.4.3 Modelling and Analysis

This thesis provides methods to model uncertainty in image data as discussed by Onyango *et al.* [88]. Image data are modelled in a Bayesian framework,

which is based on mathematical foundations of probability theory. The plant model also allows incorporation of information from sensors other than conventional cameras, since the dimension of the image data can be increased. The methodologies of modelling and analysis developed in this thesis could be applied for a wide range of crops, including those in field, which were examined by Bull *et al.* [20].

## 1.5 Contributions

The contributions of this thesis are therefore:

- A stereo algorithm to estimate plant height from stereoscopic images, which enables the estimation of height information that cannot be easily determined from a single image in an efficient and accurate manner.
- A probabilistic model for representing plant attributes and classifying the image data via a Bayesian scheme, which can incorporate new data into the existing model in a statistical manner.
- A Kalman filter approach for tracking the above probabilistic model over time, which allows the investigation of trends in the image data from a time-series point of view.

## 1.6 Equipment

All the experiments described in this thesis were run on a Red Hat Enterprise Linux machine, which had one Intel Pentium *IV* 3.0G HZ CPU with 1GB

RAM. For C/C++ implementations, the software libraries were OpenCV v0.9.7 and OpenMesh v1.0.0. MATLAB v7.3.0 (R2006b) was also used to develop algorithms. Three imaging setups were used, and they are described in Chapter 2. The cameras used in these setups were *Cannon PowerShot* Digital Camera, *AXIS 210* Network Camera and *IQeye705* Network Camera.

## 1.7 Thesis Organisation

The characteristics of plant quality and growth are based on two basic features: geometry and colour. The colour is readily available from each image, as well as the  $x$ - $y$  position information. Height information cannot be easily determined from a single view, and a stereopsis approach is therefore developed in Chapter 2 for estimating the plant height.

The estimated geometry, together with colour information from the image, are then used to build a plant surface model, which represents all the information from the visible spectrum. To this end, a neural network approach and a probabilistic model approach for plant model construction are discussed in Chapter 3 and Chapter 4 respectively.

In Chapter 5, a Kalman filter is developed to track the plant model over time. This extends the model to the time dimension, which enables smoothing of the noisy measurements to produce a development trend for a crop.

Chapter 6 concludes the thesis and discusses the limitations of the work. Possible applications and research opportunities in the future are also discussed.



## Chapter 2

# Measuring Plant Height from Stereo Images

### 2.1 Introduction

Growers use the height of a growing plant as an important attribute for tracking the development of a crop [80]. Measuring the height in a commercial glasshouse is difficult, as discussed in Chapter 1, and the process is inevitably limited by the subjective factors associated with individuals. This chapter proposes a stereopsis approach for measuring plant height. The proposed approach uses a stereoscopic pair of cameras to capture images of the plants and then estimate the height of their flowers and leaves. Another aim of this work is to estimate the plant geometry, which cannot be easily determined from a single image.

## 2.2 Approaches to Plant Height Measurement

### 2.2.1 Measuring Height from Side Views

There are commercial systems currently available for measuring plant height from side views. The *Anthurium* sorting machine shown in Chapter 1 uses six cameras to capture images of individual plants passing through the line. Given six different views from the side, it is possible to recover the plant structure, and therefore, the height can be measured. Another example is the vision system made by WPS Horti Systems <sup>1</sup>, which is similar to the *Anthurium* sorting machine and uses cameras to capture images from a side view.

Although this kind of height sorting machine is available on the market, such systems are only applicable for plants grown in individual pots like *Anthurium* and *Poinsettia*. For those grown in packs, such as *Pansy*, some plants in the pack are occluded by others, and therefore cannot be seen from the side view (see Figure 2.1). Consequently, estimating the characteristics of these flowers is not possible. These systems also fail in situations where it is difficult or impossible to collect horizontal images, for example, when plants are grown on benches during production (see Figure 2.2).

---

<sup>1</sup>WPS Horti Systems, <http://www.wpshortisystems.nl/>



**Figure 2.1:** An illustrative image of *Poinsettia* and *Pansy*. Note that there is only one *Poinsettia* plant grown in a pot, while there are a number of *Pansy* plants grown in a pack.



**Figure 2.2:** *Poinsettia* plants are grown on a bench during production at Pinetops Nurseries. It is impossible to collect horizontal images and measure height from a side view for every plant.

### 2.2.2 Active 3D Vision

#### 3D Laser Scanning

Many of the existing height measurement systems in the horticulture industry concentrate on acquiring and analysing two-dimensional (2D) information [44]. A three-dimensional (3D) image contains more information than any 2D image can possess, and a 3D representation provides more ‘natural’ viewing conditions of objects in the image, as does the output of the human visual system to the brain. Techniques for recovering 3D scene shape can be divided into either ‘active’ or ‘passive’. Active techniques require an external light source to project light onto the scene, the most common being laser scanning. Commercial high resolution 3D laser scanners, such as the one built by the National Research Council Canada [8], can simultaneously capture colour and geometry from real world objects. The scanner has been used for scanning museum objects such as paintings [12] and archaeological collections [8], and its maximum resolution configuration, in the scale of micrometres, is sufficient to examine fine brush stroke details on paintings, as well as tool marks on sculptures and archaeological objects. The laser light does not harm objects and the scanning process is not affected by ambient light. However, apart from the high cost, the use of lasers in glasshouse is subject to safety regulations, since even relatively small amounts of laser light can lead to permanent eye injuries. Moreover, the high resolution scanning process can take hours to complete, and model building from the data using the commercial licensed software may need hours of processing [12].

### Structured Lighting

Another active approach is to make an inexpensive 3D scanner using the structured lighting method [35]. The design is based on consumer electronic technology, and usually includes a video projector to project structured light patterns onto the object to be scanned and a digital still camera to acquire images of the object under the structured light. Although it is less accurate than the high quality laser scanner, the structured light system, with a typical accuracy of 0.1 millimetre, is less expensive and the scanning process is faster [7]. The fundamental problem in the structured light system is shadowing and ambient lighting [34], which is a major problem in the glasshouse environment. Furthermore, the colours of a plant's different surfaces may reflect the light differently, causing unexpected artifacts. With advances in this technique in the future, structured lighting could become another feasible approach to measure plant height in the glasshouse.

#### 2.2.3 Passive 3D vision

Passive approaches do not need a special source of energy to illuminate the scene. Common techniques include single-view, stereo, and multi-view 3D reconstruction. Criminisi *et al.* [29] described how aspects of the affine 3D geometry of a scene may be measured from a single perspective image. The ideas can be seen as reversing the rules for drawing perspective images given by Alberti [4], and it is assumed that images are obtained by perspective projection. However, using only one view for taking absolute measurements, for example, units in the metric system, is not practical in the glasshouse envi-

ronment, due to constraints on image features. Most of the images presented in [29] are scenes containing planes and parallel lines. Although the method is not restricted to these conditions, the structure of each *Pansy* plant (see Figure 2.1) cannot be captured fully with a single view, and therefore this approach is not applicable for the horticulture industry.

One widely used passive vision method is stereopsis, where imaging is done by using a stereoscopic pair of cameras. This method works in a similar fashion to the human visual system and imitates the stereopsis technique. By having two images displaced from each other, the depths of objects can be estimated using geometry if the camera focal lengths are known.

## 2.3 Stereoscopic Vision Setup

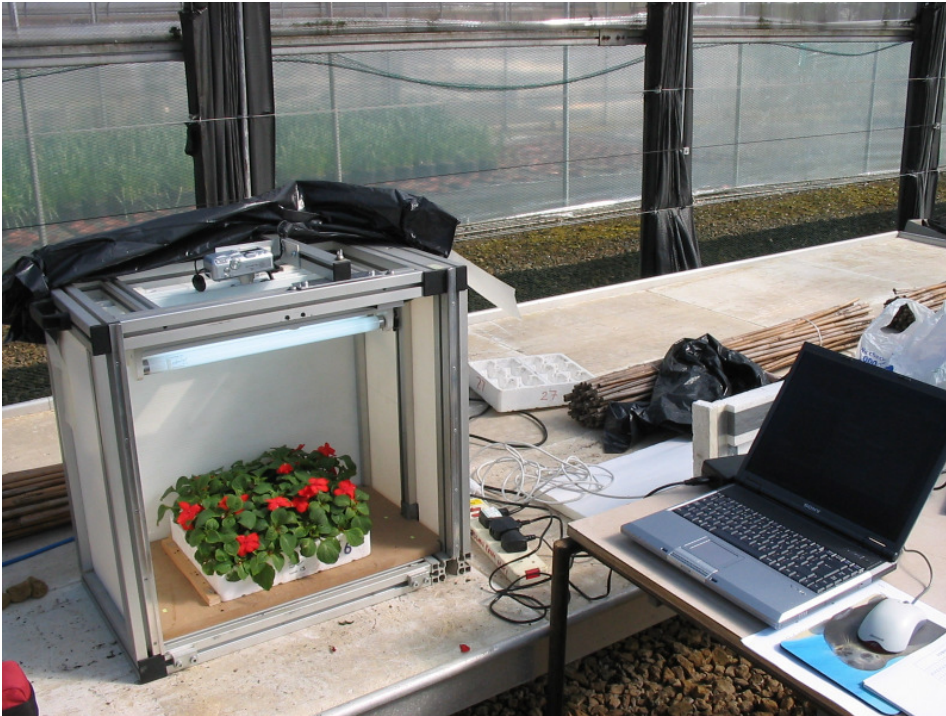
The system requirement for stereoscopic vision is a pair of cameras positioned with overlapping fields of view. These cameras can be arranged in a number of ways, but the most common and simple is placing two identical cameras on a horizontal plane with optical and vertical axes parallel (see Figure 2.3). This setup is known as the standard parallel camera setup [45]. It is also possible to use a single camera to capture one view of a stationary scene and then make another one after moving a short distance horizontally (see Figure 2.4 and 2.5).

The output directly computed from a pair of images is in the form of relative displacements, also called disparities [76]. Let  $(x_1, y_1)$  and  $(x_2, y_2)$  be two image points, and denote  $h^*$  as the distance from the camera lens centre to a world point. The distance between cameras, which is called the



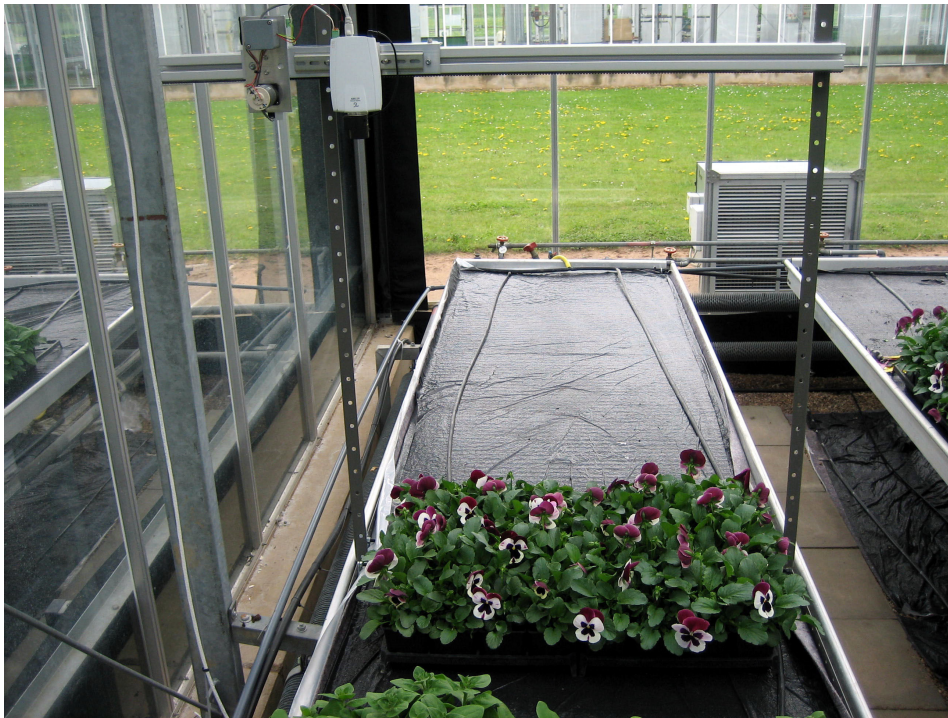
**Figure 2.3:** Parallel camera setup in a glasshouse at DoubleH Ltd. This setup simultaneously captures two views, and this is particularly useful for moving objects, for example, these plants passing through a grading line. This setup can also be flexibly mounted anywhere in the glasshouse.



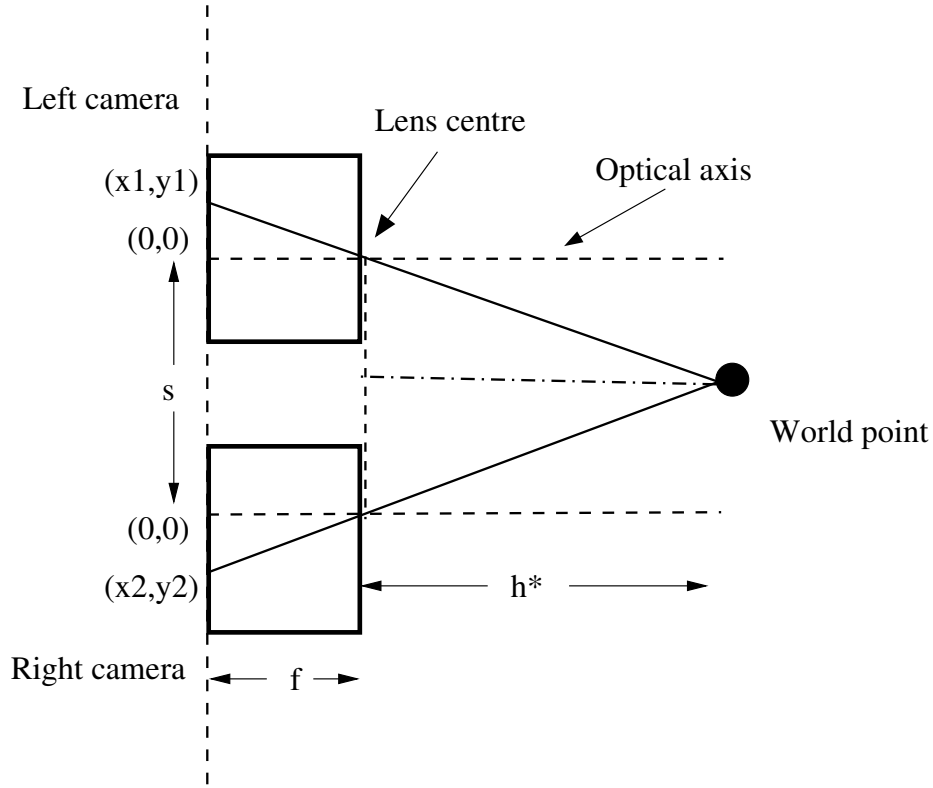


**Figure 2.4:** ‘Lightbox’ setup in a glasshouse at Warwick HRI. This setup shifts a single camera to two different positions for two views, where the *Pansy* plants remain stationary during the capture.





**Figure 2.5:** Moving camera rig scanning a *Pansy* crop from above in a glasshouse at Warwick HRI. This is similar to the idea behind the lightbox setup, and moves a single camera to two different positions for two views.



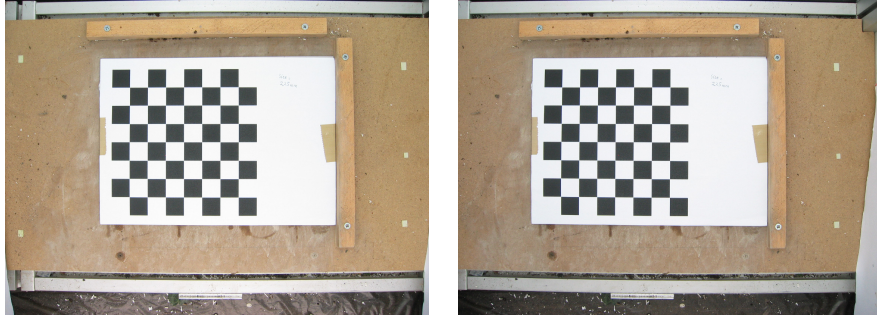
**Figure 2.6:** Model of the stereopsis process for parallel camera setup.

baseline [45], is denoted as  $s$ , while  $f$  represents the camera focal length. As Figure 2.6 shows,

$$\frac{s}{x_2 - x_1} = \frac{h^*}{f} \quad (2.1)$$

to convert disparities into depths, it is necessary to measure the camera focal length  $f$  and baseline  $s$ . The camera focal length in pixels,  $f$ , can be computed using a checkerboard and the camera calibration toolbox [13].

For the parallel camera setup shown in Figure 2.6, the depth map  $\mathbf{H}^*$  can



**Figure 2.7:** A stereoscopic pair of the checkerboard images captured using the lightbox setup.

be computed from:

$$\mathbf{H}^* = f s \mathbf{D}^{-1} \quad (2.2)$$

where  $\mathbf{H}^*$  is the depth map containing distances in centimetres from the camera to objects,  $f$  is the camera focal length in pixels,  $s$  is the baseline in centimetres, and  $\mathbf{D}$  is the disparity map in pixels.

Apart from the noise produced during the disparity computation, the resolution of the depth measurement is also limited by the separation distance between the cameras. Suppose we want to distinguish a point that is  $h_1$  metres away from the camera and another point  $h_2$  metres away, that is, to achieve a resolution of  $(h_1 - h_2)$  metres. To distinguish between the change in depth, let the required disparity difference be 1 pixel.

Rearranging equation 2.2 and substituting all variables,

$$\frac{f s}{h_2} - \frac{f s}{h_1} = 1 \quad (2.3)$$

Let  $\alpha = h_1 h_2$ .  $\Delta h$  is the change in depth between the two points,  $\Delta h =$

$h_1 - h_2$ , which is also the resolution of the depth measurement for a particular stereo setup. Equation 2.3 becomes

$$\Delta h = \frac{\alpha}{s f} \quad (2.4)$$

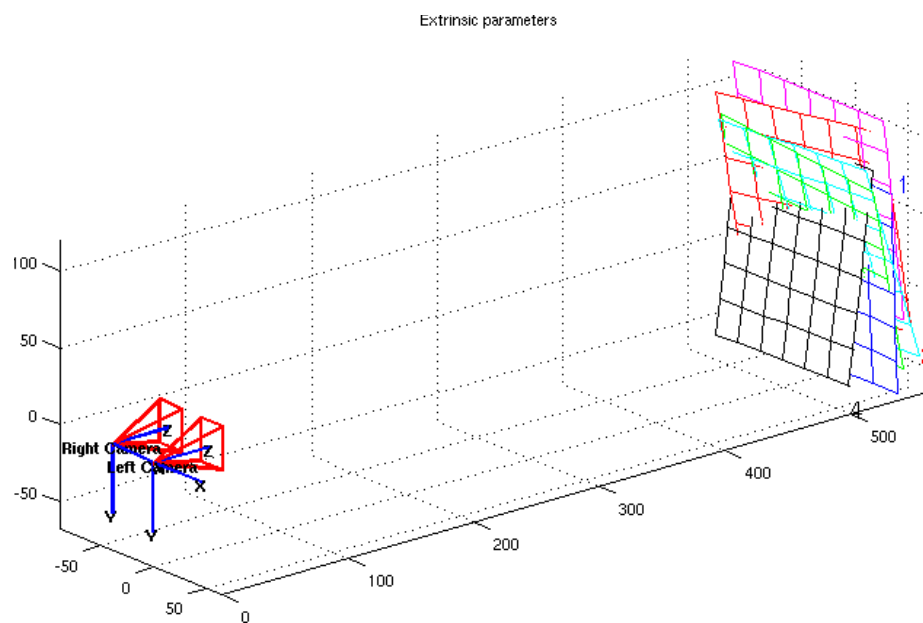
From equation 2.4, the wider displacement between the views, namely a larger  $s$ , results in a smaller change in depth, and therefore the resolution is better. However, the wide displacement between stereoscopic images increases the occurrence of occlusions and missing data, and makes the size of the system impractical, except for very short-range applications.

To measure the absolute height of objects, the depth map  $\mathbf{H}^*$  is subtracted from a reference depth  $h$ . The subtracted depth map  $\mathbf{H}$  contains the height measurements from where  $h$  is located.

$$\mathbf{H} = h - \mathbf{H}^* \quad (2.5)$$

## 2.4 Measuring Height from Stereoscopic Images

The depth from the camera to the objects is implicit in the relative displacements between the left and right views of that scene. By comparing the relative displacements of corresponding pixels in two images, a depth recovery process can be initiated, leading to the reconstruction of the scene [6]. There are therefore two main steps to estimate the depth from stereoscopic



**Figure 2.8:** Camera positions using the lightbox setup. This is estimated by using pairs of the checkerboard images and camera calibration toolbox [13]. Camera focal length in pixels can be also computed.

images:

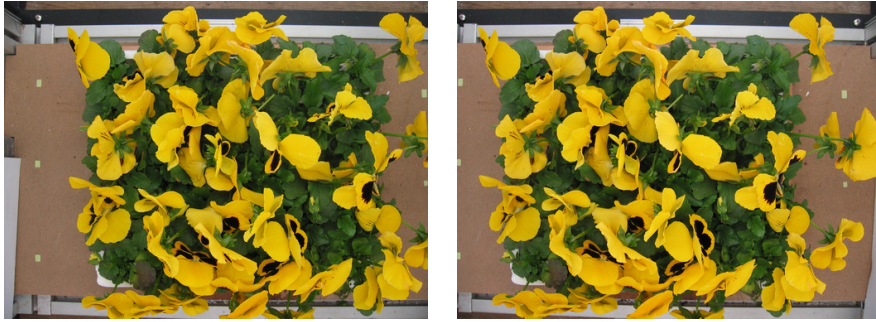
1. *Identifying corresponding points in two stereoscopic images.*
2. *Computing depth, given the geometry of the two camera settings.*

The second step is unambiguous and Section 2.3 has covered the topic. In this section, we discuss the first step in detail.

The task of matching corresponding points is also known as the correspondence problem. This problem has been extensively researched in the vision community since Marr and Poggio's work in 1976 [77]. Grimson extended this feature based approach, and proposed the 'Marr-Poggio-Grimson' algorithm [47]. This algorithm relies on features obtained by the Laplacian of a Gaussian operator and zero-crossing estimation [45]. A coarse to fine strategy to limit the search space of possible matches was also applied. The feature based approach is suitable for images containing random dot stereograms or objects with simple geometry, since a set of features can be easily identified in both images.

Ohta and Kanade [87] proposed a dynamic programming paradigm to locate the corresponding points in a 3D search space. Sun [106] proposed a two-stage dynamic programming and the rectangular subregioning technique. Similar to the feature based approach above, a coarse-to-fine scheme is used.

Figure 2.9 shows an example of stereoscopic images of a pack of *Pansy* plants. The complex structure of small, thin branches and green leaves leads to substantial shading effects and occlusions in the image, as well as many low-texture areas. Much research effort has been devoted to solving these issues [53, 98]. Recently, modelling the disparity image as a Markov Random



**Figure 2.9:** A pair of stereoscopic images capturing a pack of *Pansy* plants using the lightbox setup

Field (MRF) has become popular as a basis for many of the best stereo algorithms in the Middlebury list [98]. MRF inference algorithms have been proposed for stereopsis, which use graph cuts [69] and belief propagation [41]. Tappen and Freeman [108] compared the graph cuts algorithm and the belief propagation algorithm on the same MRF's, which have been created for calculating stereo disparities.

The multiresolution model introduced in the following section is analogous to the Bayesian framework using an MRF, and builds on the multi-scale estimation ideas applied by many authors, for example, Wilson and Knutsson [121], Jepson and Jenkin [60] and Chou *et al.* [27]. The estimates from the coarser representation of the multiresolution model can be viewed as priors to the finer representation and this transition process can be described as a Markov Chain [26]. It was pointed out by Wilson and Li [122] that the MRF framework ensures the local consistency in the estimates through neighbour interactions, whereas the multiresolution model accomplishes this through interactions with the parents in scale. A multiresolution random

field model was therefore introduced by Wilson and Li [122], which accounts for interactions with both planar neighbours and parents in scale. The parent weighting introduced in this chapter provides an alternative to Wilson and Li's model by interactions with the direct parent and the neighbours of the parent. Moreover, the maximum a posteriori (MAP) estimate, which is produced by a Kalman filter [24] in the multiresolution model, can be shown to be equivalent to minimising the energy [108].

The aim of the proposed approach is to retrieve the global geometry, instead of trying to refine depths for individual leaves and branches, as growers require summary statistics for plant growth and marketing purposes. To this end, the proposed stereo algorithm uses a multiresolution structure to reflect the assumption that the plant surface is smooth. The use of the pyramid structure discussed in this chapter also gives a significant speed advantage [1], which is desirable for industry applications, while both graph cuts and belief propagation approaches are computationally demanding [41].

Calway *et al.* [23] suggested a frequency domain approach in a multiresolution framework for disparity estimation. Bowen *et al.* [15] extended this approach by adding a Kalman filter to refine the estimates at each resolution. This thesis proposes a stereo algorithm in spatial domain and a parent weighting technique to interact with the direct parent and the neighbours of the parent.



## 2.5 A Multiresolution Model for Stereopsis

### 2.5.1 The Quadtree Structure

In this section, a multiresolution approach that models the statistical structure of a signal on a hierarchy of scales is discussed. The motivation for following such an approach originates from the observation that objects are composed of different structures at different scales. Clippingdale [28] introduced a multiresolution image model based on the quadtree data structure (see Figure 2.10). In an  $N$ -level multiresolution model representing a two-dimensional (2D) signal, each node in the tree has an associated random variable  $x(i, j, n)$ ,  $n = \{1, \dots, N\}$ , representing relevant information of the signal near the node  $(i, j)$  at that resolution  $n$ . Each node has four children and one parent, and there is only one node  $x(1, 1, 0)$  at the coarsest level 0.

The connection between the parent node  $x(i, j, n - 1)$  and its children  $x(i, j, n)$  is given by a transition operator  $I(i, j, n | n - 1)$ , and the properties of  $x(i, j, n)$  therefore depend on  $I(i, j, n | n - 1)$  and  $x(i, j, n - 1)$ . Chou and Willsky [26] showed that if the coarser representation of the multiresolution model has been estimated, it can provide prior information as an aid to estimating the finer representation. This is analogous to a linear state-space model, in which each node evolves a state throughout different levels of the tree [24].  $x(i, j, n)$  then becomes a state variable of the node  $(i, j)$  at the scale  $n$ , while  $x(i, j, n - 1)$  represents the previous state information at the scale  $n - 1$ . For an  $N$ -level model representing a 2D signal, this process is

expressed as follows,

$$x(i, j, n) = I(i, j, n|n-1)x(i, j, n-1) + v(i, j, n) \quad (2.6)$$

where  $x(i, j, n)$  represents the signal associated with the node  $(i, j)$  at level  $n$ ,  $n = \{1, \dots, N\}$ , and  $x(i, j, n-1)$  is its parent at level  $n-1$ . The operator  $I(i, j, n|n-1)$  controls the input of  $x(i, j, n-1)$ .  $v(i, j, n)$  is the process noise or innovations representing the details added at level  $n$ .

Suppose a linear observation model is used,

$$y(i, j, n) = A(i, j, n)x(i, j, n) + e(i, j, n) \quad (2.7)$$

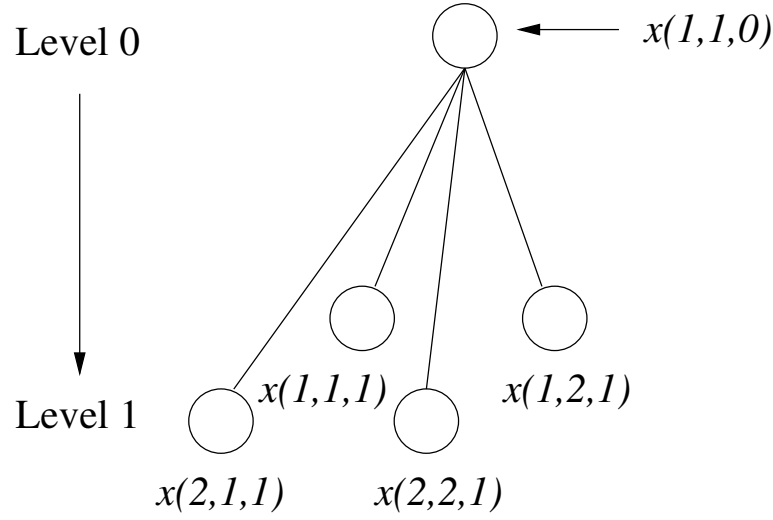
where  $y(i, j, n)$  is the noisy observation, which has the same dimension as  $x(i, j, n)$ .  $A(i, j, n)$  relates the state  $x(i, j, n)$  to the observation  $y(i, j, n)$ .  $e(i, j, n)$  is the measurement noise incurred during the observation [24].

In practice,  $A(i, j, n)$  may vary for different observations. Here  $A(i, j, n)$  is assumed to be a constant, and it is set to 1, which means there is no scaling difference between the noisy observation  $y(i, j, n)$  and the state  $x(i, j, n)$ . The transition operator  $I(i, j, n|n-1)$  is also assumed to be the same across different levels. The process noise  $v(i, j, n)$  and the measurement noise  $e(i, j, n)$  are assumed for simplicity to be additive white Gaussian noise,

$$v(i, j, n) \sim N(0, q(i, j, n)) \quad (2.8)$$

$$e(i, j, n) \sim N(0, w(i, j, n)) \quad (2.9)$$

where  $q(i, j, n)$  and  $w(i, j, n)$  are the process noise variance and measure-



**Figure 2.10:** A 2-level model based on the quadtree data structure

ment noise variance respectively. Under these assumptions, the Kalman update discussed below takes account of all the information contained in the conditional probability density.

### 2.5.2 The Kalman Update

Given the noisy observation and the previous state information, the question remains: how to combine them to produce an ‘optimal’ observation. ‘Optimal’ means that the estimate minimises the expected errors based on prior knowledge. Posing this problem in the Bayesian framework results in a MAP estimator of the state  $x(i, j, n)$  at the scale  $n$ , given the state model shown in equation (2.6) and a set of noisy observations,  $y(i, j, n)$ , described in equation (2.7). For simplicity, the node position  $(i, j)$  is omitted. For example, the state of the node  $(i, j)$  at time  $n$ ,  $x(i, j, n)$ , is expressed in vector

form as  $\mathbf{x}(n)$ .

Applying Bayes' theorem [62], the posteriori density  $p(\mathbf{x}(n)|\mathbf{y}(n))$  can be expressed as follows,

$$p(\mathbf{x}(n)|\mathbf{y}(n)) = \frac{p(\mathbf{y}(n)|\mathbf{x}(n))p(\mathbf{x}(n))}{p(\mathbf{y}(n))} \quad (2.10)$$

$$= \frac{p(\mathbf{y}(n)|\mathbf{x}(n))p(\mathbf{x}(n)|\mathbf{y}(n-1))}{p(\mathbf{y}(n)|\mathbf{y}(n-1))} \quad (2.11)$$

where the likelihood  $p(\mathbf{y}(n)|\mathbf{x}(n))$ , the prior  $p(\mathbf{x}(n)|\mathbf{y}(n-1))$ , and the evidence  $p(\mathbf{y}(n)|\mathbf{y}(n-1))$  are defined below,

$$\begin{aligned} p(\mathbf{x}(n)|\mathbf{y}(n-1)) &= \int p(\mathbf{x}(n)|\mathbf{x}(n-1))p(\mathbf{x}(n-1)|\mathbf{y}(n-1))d\mathbf{x}(n-1) \\ p(\mathbf{y}(n)|\mathbf{y}(n-1)) &= \int p(\mathbf{y}(n)|\mathbf{x}(n))p(\mathbf{x}(n)|\mathbf{y}(n-1))d\mathbf{x}(n) \end{aligned}$$

where  $p(\mathbf{x}(n)|\mathbf{x}(n-1))$  and  $p(\mathbf{y}(n)|\mathbf{x}(n))$  are derived from the state model (equation (2.6)) and the observation model (equation (2.7)) respectively.

Since the process noise and the measurement noise are assumed to be additive white Gaussian, this naturally leads to the Kalman update process [115]. The Kalman update can be viewed as a linear blending of the noisy observation  $\mathbf{y}(n)$  and the prior estimate  $\hat{\mathbf{x}}(n-1)$  [18]. The equation for the Kalman update is shown below,

$$\hat{\mathbf{x}}(n) = \mathbf{I}(n|n-1)\hat{\mathbf{x}}(n-1) + \mathbf{G}(n)\nu(n) \quad (2.12)$$

where  $\nu(n)$  is the difference between the noisy observation  $\mathbf{y}(n)$  and the

prediction from the prior estimate  $\hat{\mathbf{x}}(n-1)$ ,

$$\nu(n) = \mathbf{y}(n) - \mathbf{I}(n|n-1)\hat{\mathbf{x}}(n-1) \quad (2.13)$$

$\nu(n)$  can also be referred as the details to be added from the observation.  $\mathbf{G}(n)$ , known as the Kalman gain, controls the input of  $\nu(n)$ . Given the process noise covariance  $\mathbf{Q}(n)$  and the measurement noise covariance  $\mathbf{W}(n)$ ,  $\mathbf{G}(n)$  is calculated as follows:

$$\hat{\mathbf{P}}(n) = \mathbf{P}(n-1) + \mathbf{Q}(n) \quad (2.14)$$

$$\mathbf{G}(n) = \hat{\mathbf{P}}(n) \left[ \hat{\mathbf{P}}(n) + \mathbf{W}(n) \right]^{-1} \quad (2.15)$$

where  $\mathbf{P}(n-1)$  is the prior estimate error covariance. The posteriori estimate error covariance  $\mathbf{P}(n)$  is calculated by:

$$\mathbf{P}(n) = \left[ \mathbf{I} - \mathbf{G}(n) \right] \hat{\mathbf{P}}(n) \quad (2.16)$$

If the measurement noise is small,  $\mathbf{G}(n) \rightarrow \mathbf{I}$ , which means more input from the observation  $\mathbf{y}(n)$  to the optimal state. On the other hand, if the process noise is small,  $\mathbf{G}(n)$  turns to a zero matrix, which means the previous state  $\hat{\mathbf{x}}(n-1)$  contributes more. In practice,  $0 < |\mathbf{G}(n)| < 1$ , and the optimal state therefore contains a component from the previous state and one from the observation.

In the field of stereopsis, the structure of disparities between stereoscopic images can be captured through such a scale-recursive statistical model, in

which  $\mathbf{x}(n)$  represents the actual disparity and  $\mathbf{y}(n)$  represents the measured disparity. The next section discusses how to perform the disparity measurement given a pair of stereoscopic images.

## 2.6 Disparity Estimation

### 2.6.1 Matching

An important cue to depth perception - disparities between stereoscopic images - can be measured by finding matching pixels, rather than relying on features. The common matching cost functions are Sum-of-Squared Differences (SSD) and Sum-of-Absolute Differences (SAD) [98]. Denote  $l(i, j)$  and  $r(i, j)$  as two points in the left and right gray-scale images respectively, SSD and SAD between the two images are computed as:

$$SSD = \sum_{i,j} \left[ l(i, j) - r(i, j) \right]^2 \quad (2.17)$$

$$SAD = \sum_{i,j} |l(i, j) - r(i, j)| \quad (2.18)$$

The minimal value in SSD or SAD yields the corresponding point. A noticeable difference between SSD and SAD is that SSD magnifies and penalises larger errors while SAD does not.

Normalised cross-correlation [49] can also be used for matching. An efficient way to compute cross correlation is based in the Fourier domain [89]. Denote  $\mathbf{l}(i, j)$  and  $\mathbf{r}(i, j')$  as two blocks with the same size in the left and right gray-scale images respectively, the correlation matrix  $\mathbf{C}$  can be ob-

tained from:

$$\mathbf{C} = \mathfrak{F}^{-1}(\mathfrak{F}(\mathbf{l}(i, j))\mathfrak{F}(\mathbf{r}^*(i, j')))) \quad (2.19)$$

where  $\mathfrak{F}$  is the Fourier transform,  $\mathfrak{F}^{-1}$  is the inverse Fourier transform and  $*$  is the complex conjugate. The maximal value in  $\mathbf{C}$  yields the corresponding displacement. The corresponding position inferred from the coefficient matrix needs to be adjusted due to wrap-around effects [90]. The images should also be prefiltered by a Laplacian operator to whiten the image spectrum [45].

Phase correlation [72] computes correlation based on phase information in Fourier domain and therefore it is not sensitive to changes in image intensity. Denote the phase difference between two images by  $\Delta\phi$ , the correlation matrix  $\mathbf{C}$  can be computed as:

$$\mathbf{C} = \mathfrak{F}^{-1}(e^{j\Delta\phi}) \quad (2.20)$$

The maximal value in  $\mathbf{C}$  gives the corresponding point. A more robust phase correlation method by restricting the position of the peak value was proposed by Keller *et al.* [65].

For a more detailed discussion on the subject of matching, the reader is referred to Scharstein and Szeliski's paper [98], and the work by Hirschmuller and Scharstein [53]. In this thesis, the matching cost function used for each colour channel to compute dissimilarity  $s_k(i, j), k \in \{R, G, B\}$  is identical to the one proposed by Birchfield and Tomasi [10]. Let  $r_l(i, j)$  and  $r_r(i, j)$  represent linearly interpolated intensity values around a pixel in the right

image  $r(i, j)$ ,

$$\begin{aligned} r_l(i, j) &= \frac{1}{2} \left[ r(i, j) + r(i, j - 1) \right] \\ r_r(i, j) &= \frac{1}{2} \left[ r(i, j) + r(i, j + 1) \right] \end{aligned}$$

The pixel dissimilarity computed for colour channel  $k$ ,  $s_k(i, j)$ , is then obtained by the following equation:

$$s_k(i, j) = \max \left\{ 0, \delta_{\min}(i, j), \delta_{\max}(i, j) \right\} \quad (2.21)$$

where  $\delta_{\min}(i, j)$  and  $\delta_{\max}(i, j)$  are defined as follows:

$$\begin{aligned} \delta_{\min}(i, j) &= l(i, j) - \min \left\{ r_l(i, j), r(i, j), r_r(i, j) \right\} \\ \delta_{\max}(i, j) &= \max \left\{ r_l(i, j), r(i, j), r_r(i, j) \right\} - l(i, j) \end{aligned}$$

where  $l(i, j)$  represents a pixel in the left image. The maximum matching cost,  $s_{\max}$ , is set to be 30, and this enables normalisation of the cost of matching to an error score ranging between 0 and 1. The normalised error score represents uncertainty of the matching, that is, how good the estimated disparities are. Finally, the RGB pixel dissimilarities are summed for the overall cost as follows:

$$s(i, j) = s_R(i, j) + s_G(i, j) + s_B(i, j) \quad (2.22)$$

Although a colourspace transformation such as CIE L\*a\*b\* is assumed to



improve the similarity matching, we have not observed noticeable changes in the results. Similar findings regarding a colourspace transformation were presented in [48, 101, 124], and Yang *et al.* [124] also used Birchfield and Tomasi's matching method.

For the parallel camera setup shown in Figure 2.6, the corresponding points in both images lie on the same horizontal scan-line. The multiresolution model therefore expresses the connection between the point  $(i, j)$  in the left image  $\mathbf{l}$  and the corresponding one in the right image  $\mathbf{r}$  as follows:

$$l(i, j) = r(i, j - \hat{x}(i, j)) + v(i, j) \quad (2.23)$$

where  $\hat{x}(i, j)$  is the estimated disparity between the two points, and  $v(i, j)$  is the residual error between the two images.

In practical applications, image noise and illumination changes are inevitable, and pixel-based matching is not robust under such circumstances [70]. One way to improve this is to match corresponding regions instead of individual pixels. Assuming pixels in a small region have the same disparity value, the input image is divided into a number of blocks overlapped by 50% and these blocks then match in the same way as the pixel-based methods. Equation 2.23 then becomes,

$$l(i', j') = r(i', j' - \hat{x}(i', j')) + v(i', j') \quad (2.24)$$

where  $(i', j')$  denotes the position of a block.

### 2.6.2 Multiresolution Implementation

A pyramid scheme can be used to speed up the matching. In this thesis, a Gaussian pyramid [22] is used as a representation of the pair of stereoscopic images, and a bilinear interpolator acts as the transition operator. The Gaussian pyramid is a hierarchy of low-pass filtered images of the original. An  $N$ -level Gaussian pyramid,  $x(i, j, n), n = \{1, \dots, N\}$ , for an image  $\mathbf{M}$  is defined as follows,

$$\begin{aligned} x(i, j, N) &= M(i, j) \\ x(i, j, n) &= \sum_{\substack{-2 \leq i' \leq 2 \\ -2 \leq j' \leq 2}} \Delta(i', j') x(2i + i', 2j + j', n + 1), 0 < n \leq N \end{aligned}$$

where  $\Delta$  is a Gaussian kernel, of which an example is given below,

$$\Delta = \frac{\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \end{bmatrix}^T \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \end{bmatrix}}{16 \times 16} \quad (2.25)$$

### 2.6.3 Parent Weighting

Using the pyramid structure, the size of the search window remains constant throughout different levels. Although this can speed up stereo matching, poor results could be propagated by direct parent nodes through levels, leading to false matches between images. To prevent this, a method to weight a set of parent nodes to form an optimal predictor  $\hat{x}(i, j, n - 1)$  is applied. The proposed method takes account of inputs from neighbouring parent nodes as well as the direct parent node. It initialises the weights  $\omega(i, j, n - 1)$  for

block  $(i, j)$  at level  $n$  by selecting ‘good’ parents from 8 nodes around the direct parent, and the direct parent itself. This selection process can be done by median filtering the errors associated with each parent node, which is  $s(x, y, n - 1)$ ,  $x \in \{i - 1, i, i + 1\}$ ,  $y \in \{j - 1, j, j + 1\}$  for the child block  $(i, j)$ . After that, the weights are calculated as follows:

$$\begin{aligned} \omega(x, y, n - 1) &= s_{max} - s(x, y, n - 1) \\ x &\in \{i - 1, i, i + 1\}, y \in \{j - 1, j, j + 1\} \end{aligned} \quad (2.26)$$

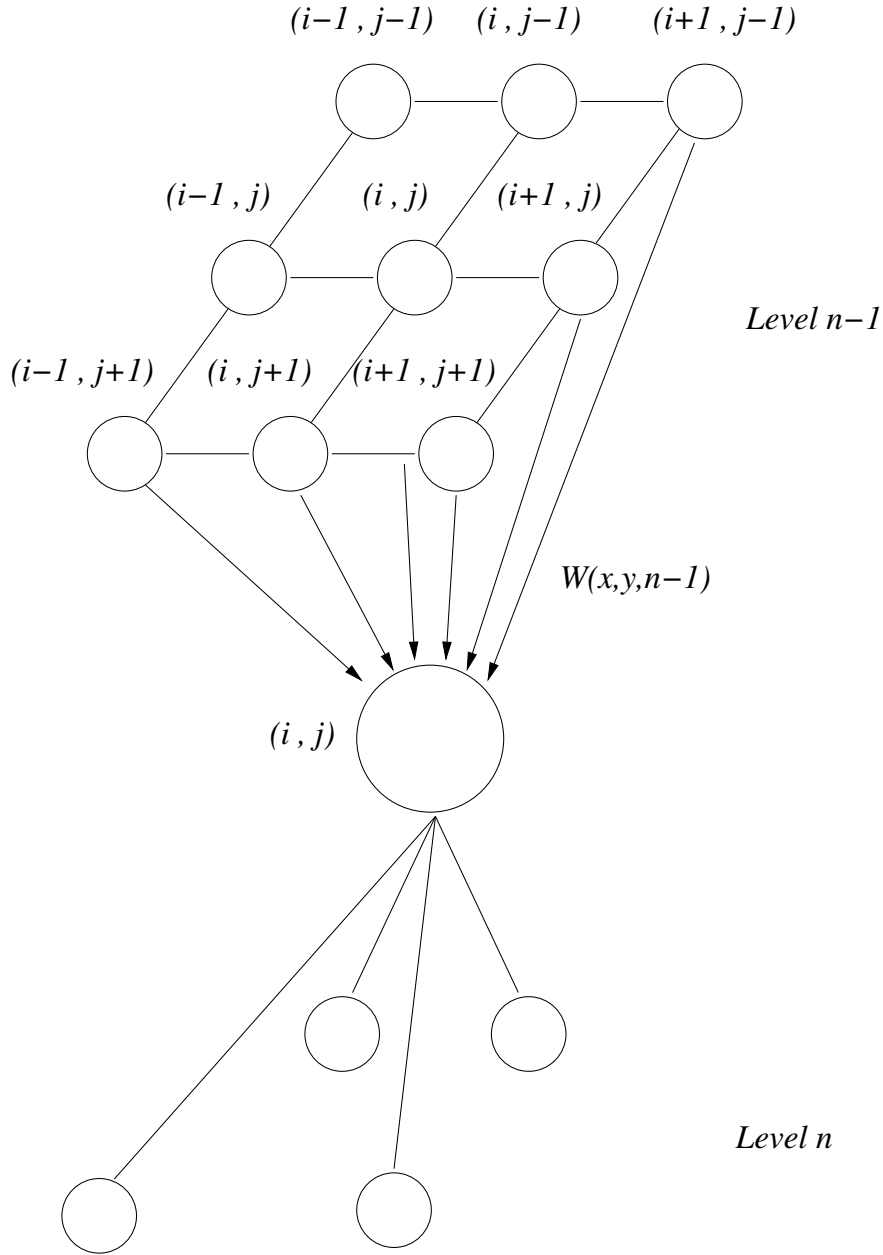
$$\omega(x, y, n - 1) = \frac{\omega(x, y, n - 1)}{\sum_{x,y} \omega(x, y, n - 1)} \quad (2.27)$$

where  $s(x, y, n - 1)$  is the error from level  $n - 1$  for one of the selected ‘good’ parents, and  $s_{max}$  is the maximum error in these ‘good’ parent nodes. The weighted parent node is then derived as follows:

$$\hat{x}(i, j, n - 1) = \sum_{x,y} \omega(x, y, n - 1) \hat{x}(x, y, n - 1) \quad (2.28)$$

where  $\hat{x}(i, j, n - 1)$  is the disparity of the weighted parent node for the child block  $(i, j)$ .

The weighting technique can be seen as an improved transition operator,  $I(i, j, n|n - 1)$ , which controls the input of the direct parent node. In effect, the value in  $I(i, j, n|n - 1)$  is modified in a data dependent way: if the direct parent is not ‘good’, the neighbouring parents contribute to form a ‘good’ one accordingly.



**Figure 2.11:** An illustration of parent weighting. Parent weighting can be seen as an improved state transition operator, which is modified in a data dependent way.

**Table 2.1:** Summary of the multiresolution stereo algorithm

1. Compute  $N$ -level Gaussian pyramid pairs for the left and right images.
2. For the coarsest level 0, divide the left image into a number of blocks and find the corresponding blocks in the right image.
3. For each block at level  $n$  where  $n = \{1 \dots N\}$ ,
  - (a) Find the weighted parent from level  $n - 1$  to define the centre of a small search window
  - (b) Find the corresponding block in this small search window
  - (c) Apply Kalman update for the optimal estimate
4. Interpolate disparities estimated to the original image size.
5. Convert the disparities into depths by triangulation.

## 2.7 Results

This section presents some of the results obtained by the multiresolution stereo algorithm discussed in this chapter. The algorithm used a 5-level Gaussian pyramid with blocks sized  $16 \times 16$  overlapped by 50%. The search range was 5 as the maximum disparity was 75. The size of input images was  $512 \times 384$ , and all these images were rectified using the calibration procedures [13].

Figure 2.13 shows the left views of two packs of *Pansy* and *Viola*, and one pack of *Cyclamen*, captured during two growing stages. Also, an object with known geometry, an inverted flowerpot, is displayed in Figure 2.12. The proposed stereo algorithm (MR) is compared with standard absolute

differencing method (AD), the belief propagation algorithm (BP) by Felzenszwalb and Huttenlocher [41] and the graph-cut method (GC) discussed in Kolmogorov and Zabih's paper [69]. All stereo algorithms were implemented in C/C++.

The AD method adopted a Gaussian pyramid implementation without Kalman update and parent weighting, which is useful to demonstrate the importance of neighbourhood and parent-level interactions used in the MR method. Both BP and GC methods were developed in a MRF framework, and minimised the energy to produce a MAP estimate (see Section 2.4). The BP method [41] used two parameters for data cost and discontinuity cost respectively to formulate the energy function, while the GC method [69] used one parameter to achieve energy regularisation.

### 2.7.1 Stereo Results on a Known-Geometry Object

In this section, the four stereo algorithms were tested by processing images featuring a known-geometry object (Figure 2.12). The expected disparity result should clearly show a near-cylindrical object slightly tapering to the top. The AD method used the same window settings as the MR method, and the search range was 75. Parameters for the BP method and the GC method were empirically chosen after a number of experiments. The BP method used 30.0 as the truncation of data cost (DATA\_K), and 15.0 for the truncation of discontinuity cost (DISC\_K). The regularisation parameter  $\lambda$  for the GC method was 50, and the maximum iteration count was 3. Disparity results are presented in Figure 2.12, and a summary of running time and height

**Table 2.2:** Known-geometry object: comparisons of height and running time

	AD	BP	GC	MR
Height (in cm)	29.4	29.2	29.4	29.2
Error (in cm)	0.2	<b>0.0</b>	0.2	<b>0.0</b>
Time (in sec.)	2.506	8.601	250.66	<b>0.679</b>

The hand measurement for the height of the inverted flowerpot was 29.2 cm

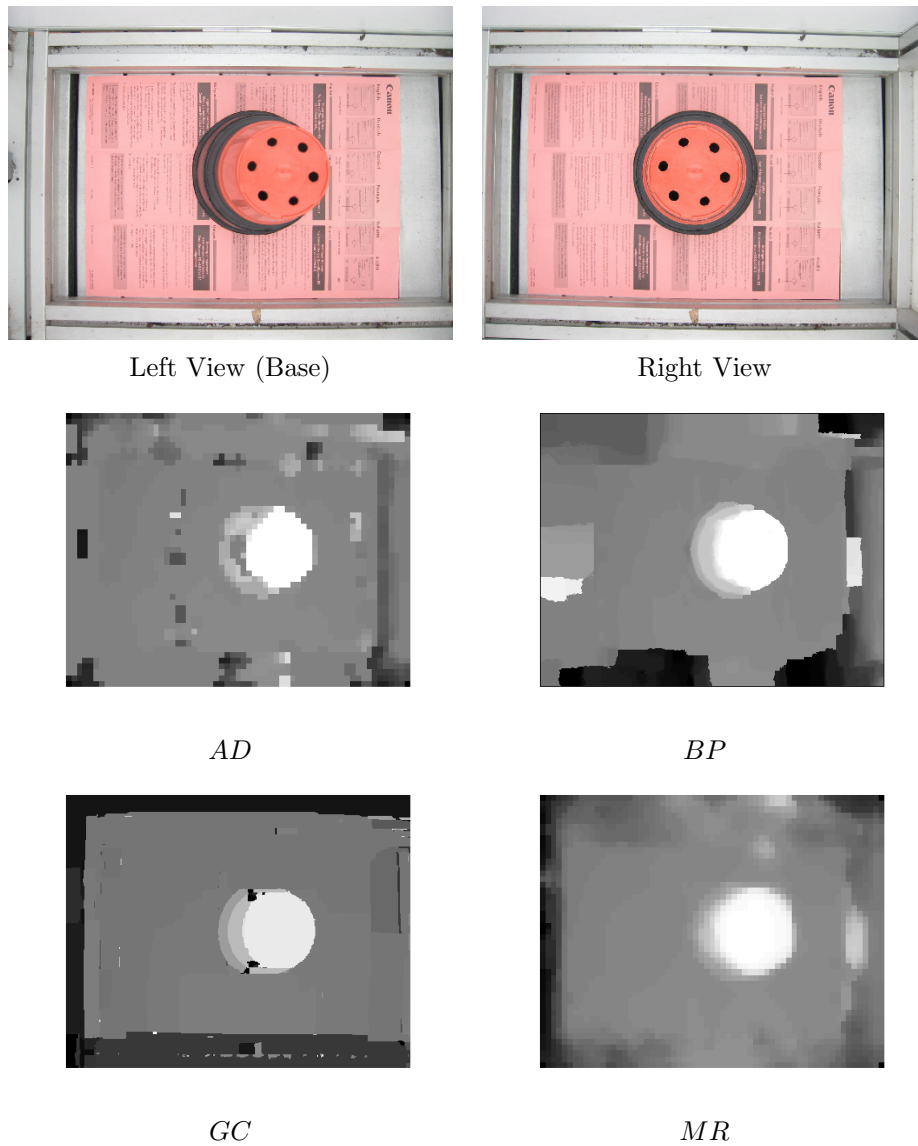
derived is presented in Table 2.2.

### 2.7.2 Stereo Results for Plants

For the plant images (Figure 2.13), the settings for all four stereo algorithms were identical to those used earlier for the known-geometry object. As the boundary information for the pack was available, that is, the position and size of the pack, the background was cropped outside the boundary region. Disparity results for all the flowers in Figure 2.13 are shown in Figures 2.14-2.18. Height measurements  $\mathbf{H}$  are computed from the disparity results using equation (2.4) with the intrinsic and extrinsic parameters computed by the calibration procedures [13]. The average flower height  $\hat{h}$  is then computed by the following equation:

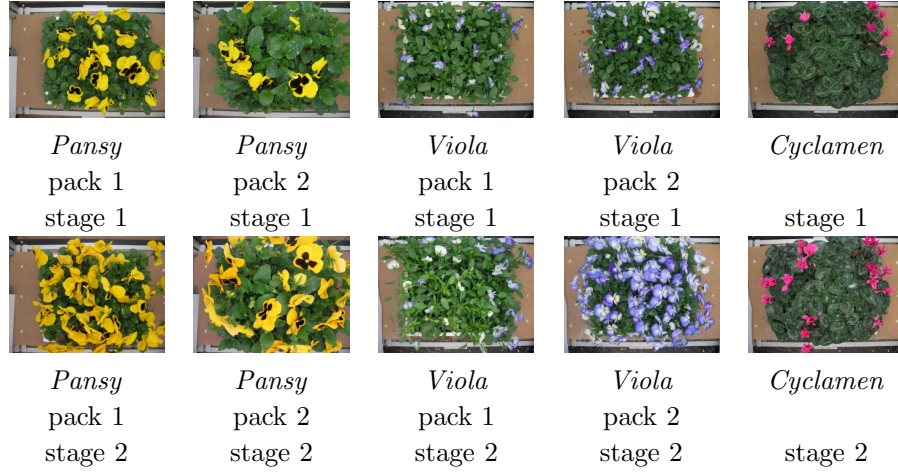
$$\hat{h} = \frac{1}{|\lambda_f|} \sum_{x \in \lambda_f} \mathbf{H}(x) \quad (2.29)$$

$\lambda_f$  is a binary mask labelling each flower, and it is obtained by a supervised multi-thresholding image segmentation method [113]. Figures 2.20-2.24 illustrate the average flower height measurements  $\hat{h}$  converted for each stereoscopic pair. The error  $E$  between hand measurements  $h_G$  and that estimated



**Figure 2.12:** Stereoscopic images and results using four stereo methods on an inverted flowerpot with known geometry.





**Figure 2.13:** Left views in the ten pairs of stereoscopic images for five different packs of plants.

by the stereo algorithm  $\hat{h}$  is computed by:

$$E = \left| h_G - \hat{h} \right| \quad (2.30)$$

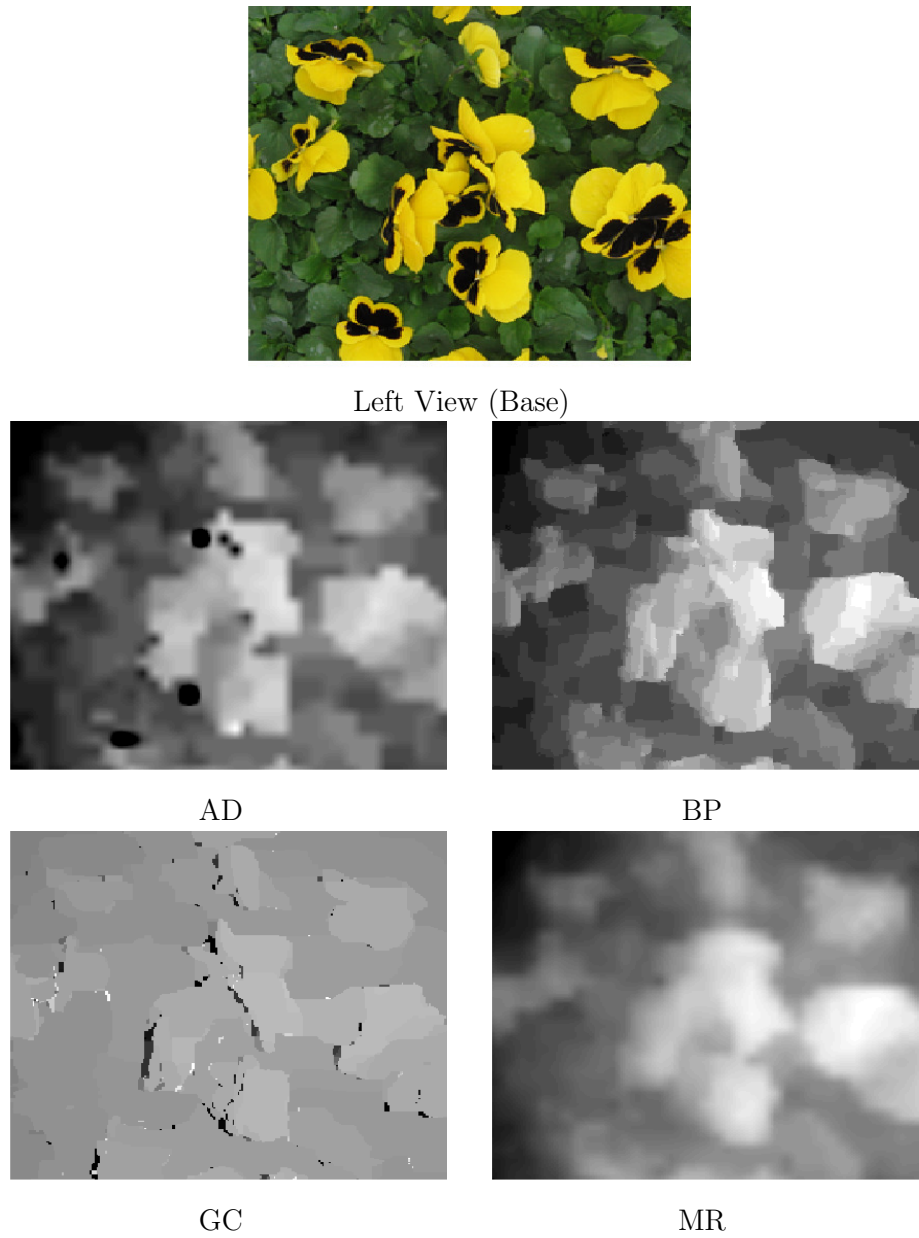
Table 2.4 presents the average absolute error,  $\bar{E}$ , standard deviation of the error,  $\sigma$ , maximum error value  $E_{max}$ , and the average running time required to process all ten stereoscopic pairs.

**Table 2.3:** Plant packs: comparisons of average height measurements in cm.

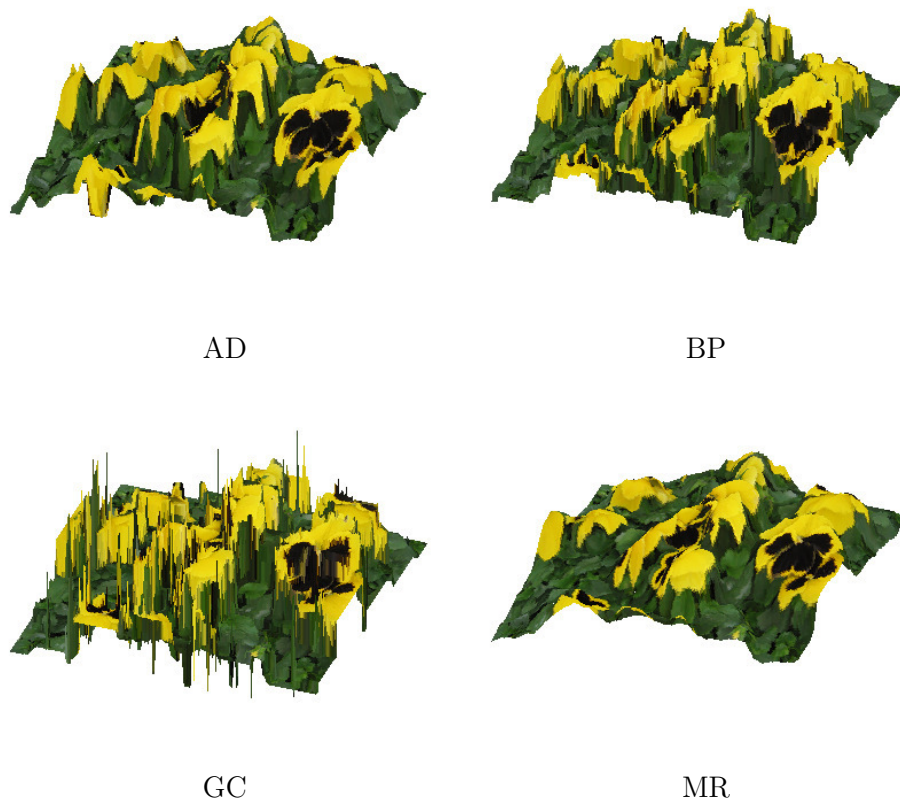
	Human	AD	BP	GC	MR
<i>Pansy</i> , pack 1, stage 1	8	10.34	12.68	13.75	<b>9.51</b>
<i>Pansy</i> , pack 1, stage 2	11	7.84	12.77	13.77	<b>10.55</b>
<i>Pansy</i> , pack 2, stage 1	15	4.53	<b>15.40</b>	16.69	15.70
<i>Pansy</i> , pack 2, stage 2	16	10.47	<b>16.10</b>	17.25	16.16
<i>Viola</i> , pack 1, stage 1	13	14.68	13.46	14.84	<b>13.44</b>
<i>Viola</i> , pack 1, stage 2	16	<b>16.56</b>	13.38	14.69	13.89
<i>Viola</i> , pack 2, stage 1	12	13.06	<b>12.67</b>	13.94	10.33
<i>Viola</i> , pack 2, stage 2	14	14.89	12.60	<b>13.88</b>	<b>14.12</b>
<i>Cyclamen</i> , stage 1	11	<b>11.43</b>	11.79	14.47	11.58
<i>Cyclamen</i> , stage 2	13.5	11.05	12.59	14.48	<b>12.88</b>

**Table 2.4:** Plant packs: comparisons of the average error  $\bar{E}$ , standard deviation of the error  $\sigma$ , maximum error value  $E_{max}$ , and the average time required to process all ten pairs of images by four stereo methods.

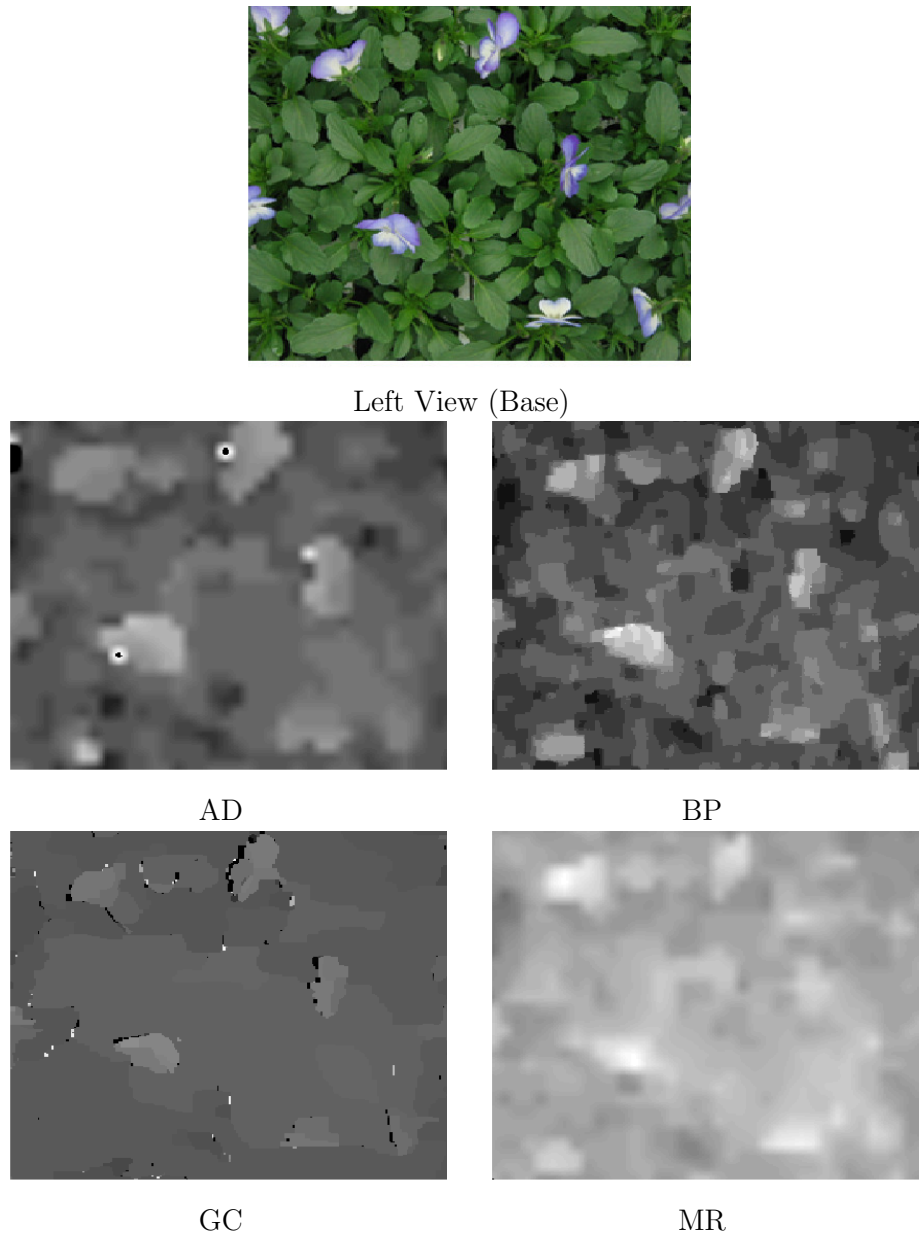
	AD	BP	GC	MR
Time (in sec)	2.4503	9.8697	194.0180	<b>0.6168</b>
$\bar{E}$ (in cm)	2.86	1.38	2.11	<b>0.84</b>
$\sigma$ (in cm)	3.08	1.38	1.58	<b>0.68</b>
$E_{max}$ (in cm)	10.47	4.68	5.75	<b>2.11</b>



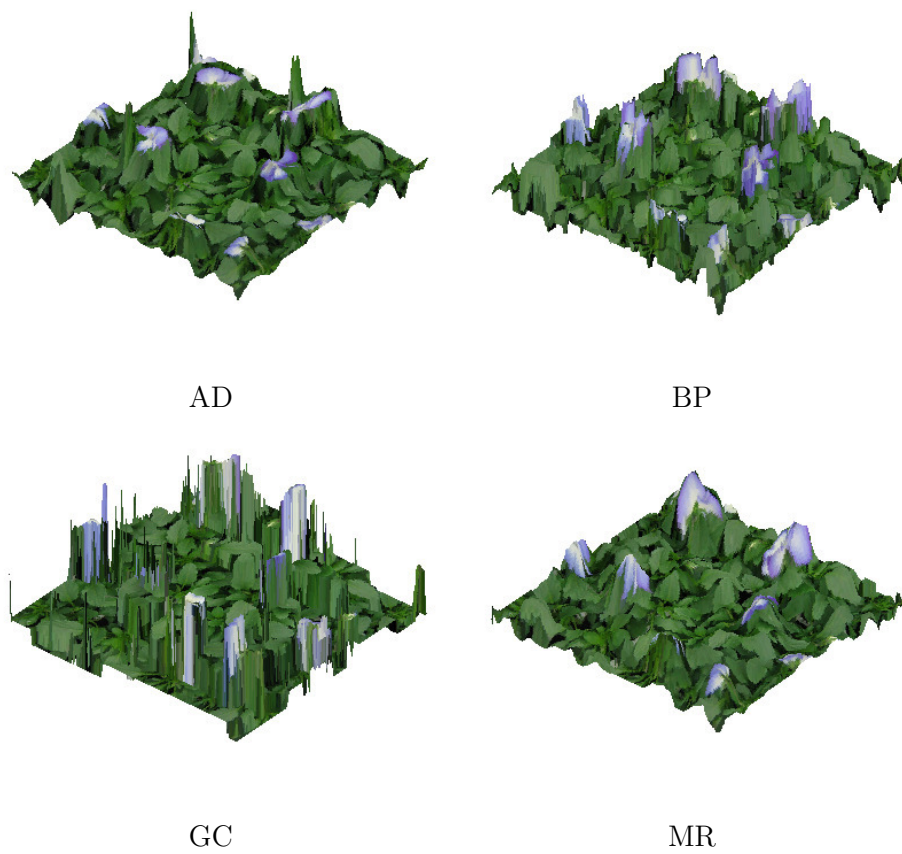
**Figure 2.14:** Stereo results for *Pansy*, pack 1, stage 1



**Figure 2.15:** Reconstructed views of *Pansy*, pack 1, stage 1



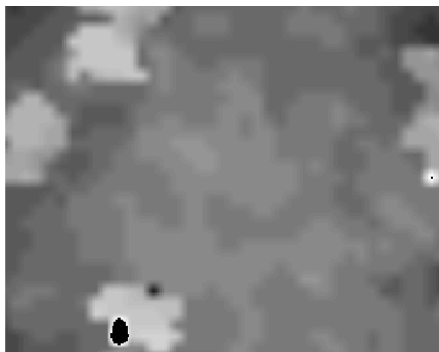
**Figure 2.16:** Stereo results for *Viola*, pack 1, stage 1



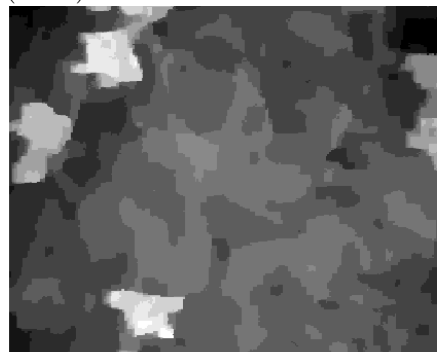
**Figure 2.17:** Reconstructed views of *Viola*, pack 1, stage 1



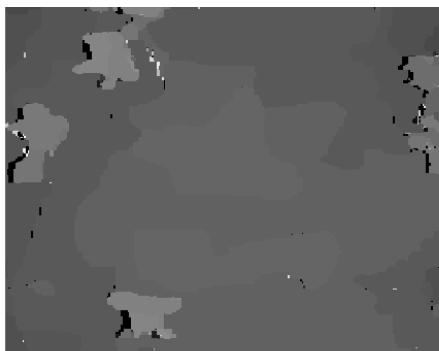
Left View (Base)



AD



BP

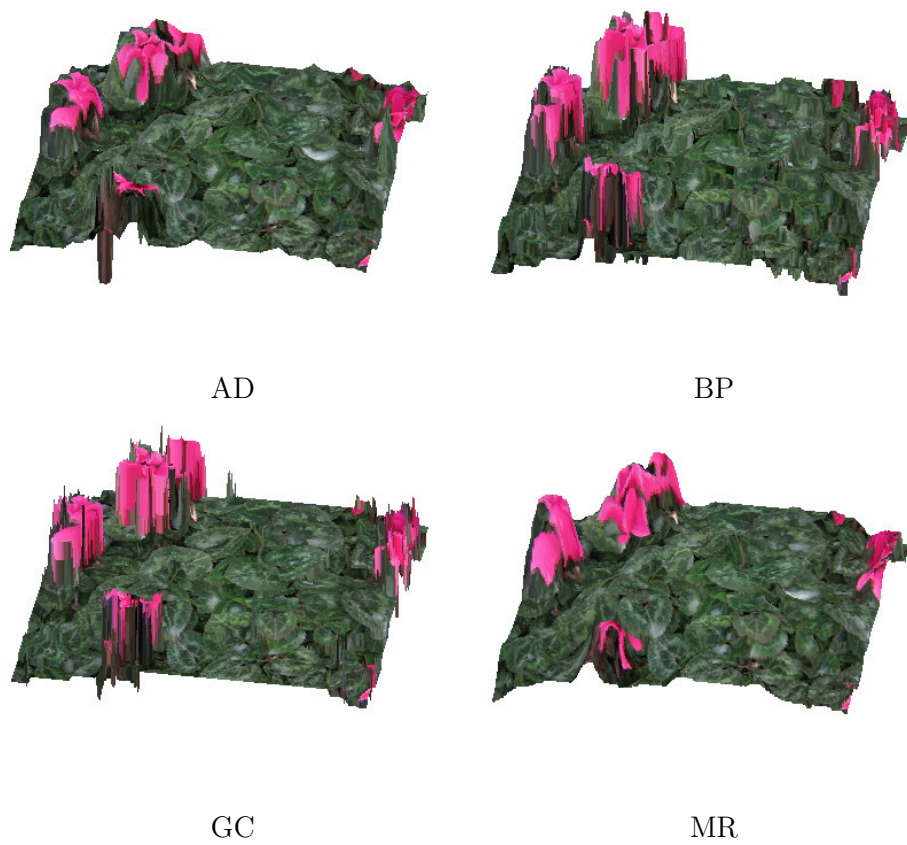


GC



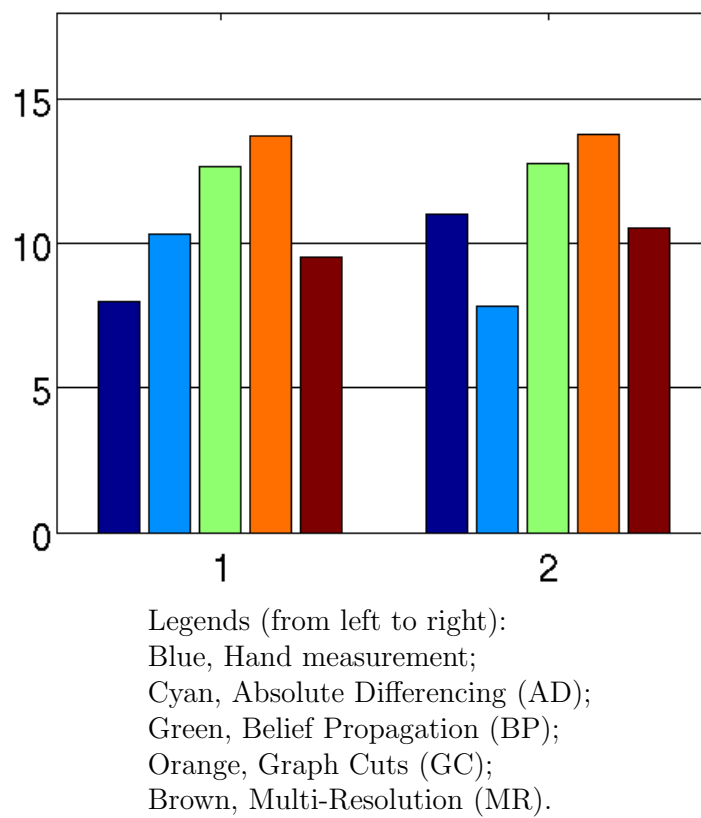
MR

**Figure 2.18:** Stereo results for *Cyclamen*, stage 2

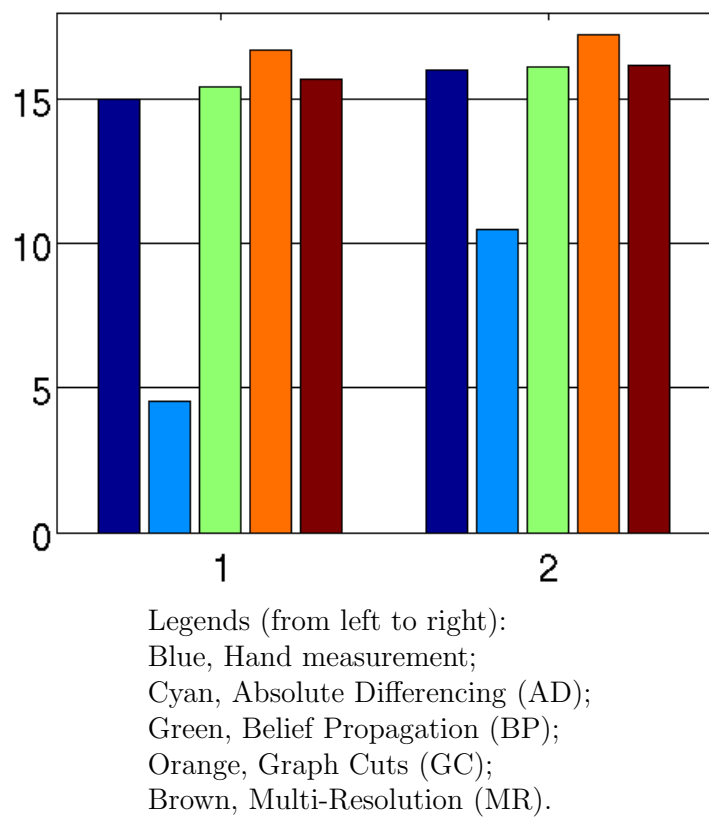


**Figure 2.19:** Reconstructed views of *Cyclamen*, stage 2

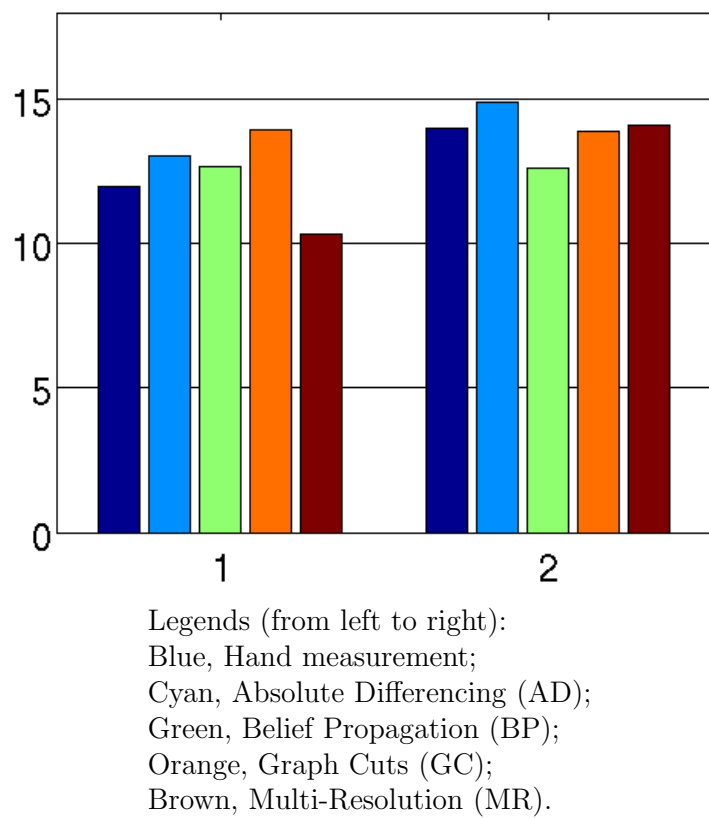




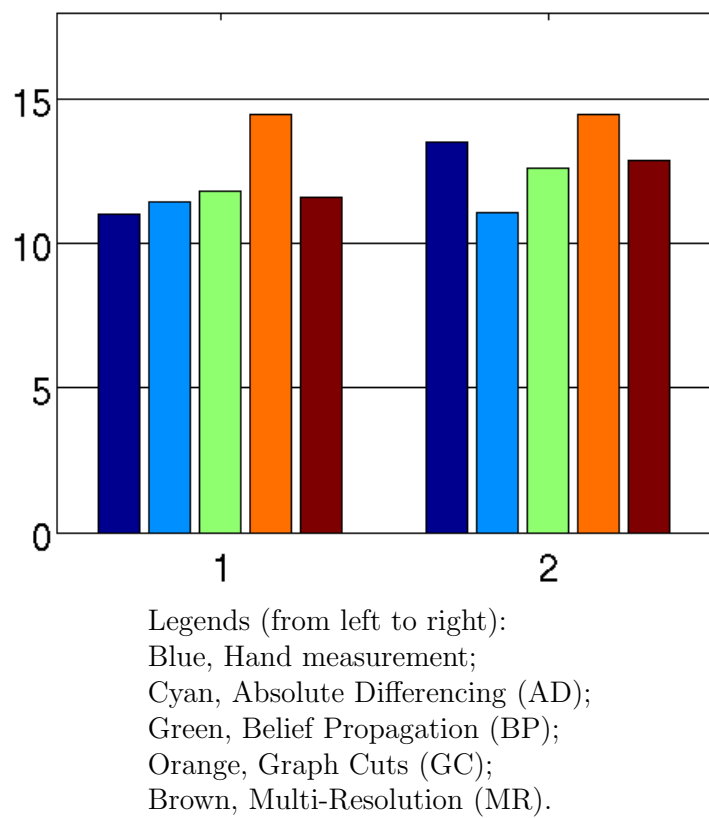
**Figure 2.20:** Average height measurements in cm for *Pansy*, pack 1. The figure on the left is for stage 1, and the figure on the right is for stage 2.



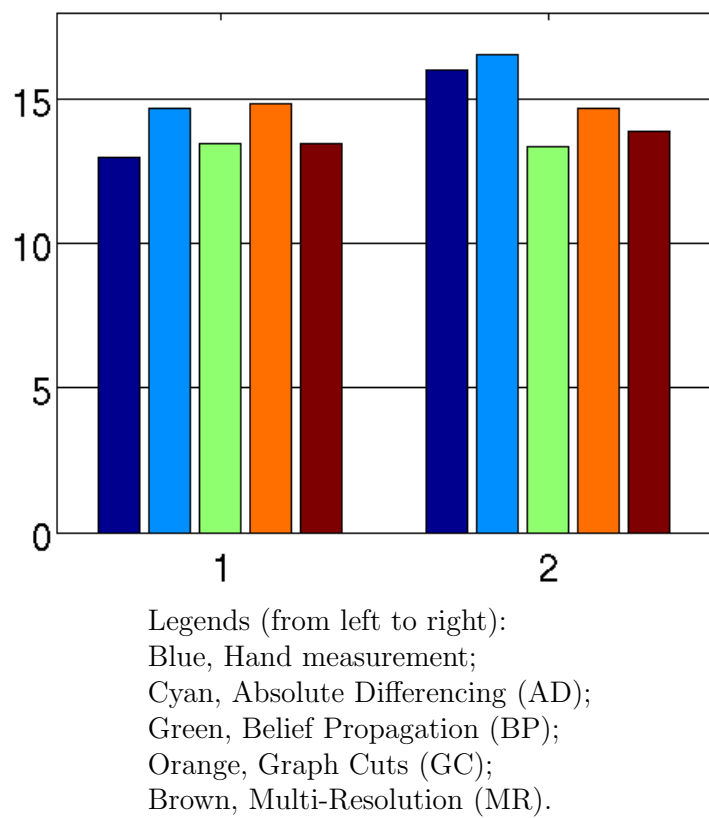
**Figure 2.21:** Average height measurements in cm for *Pansy*, pack 2. The figure on the left is for stage 1, and the figure on the right is for stage 2.



**Figure 2.22:** Average height measurements in cm for *Viola*, pack 1. The figure on the left is for stage 1, and the figure on the right is for stage 2.



**Figure 2.23:** Average height measurements in cm for *Viola*, pack 2. The figure on the left is for stage 1, and the figure on the right is for stage 2.



**Figure 2.24:** Average height measurements in cm for *Cyclamen*. The figure on the left is for stage 1, and the figure on the right is for stage 2.

## 2.8 Discussion

The proposed MR stereo algorithm produced the expected smooth disparity map and was able to give a height estimation with no practical difference from the ground truth (Table 2.2) for the stereo results featuring an object with known geometry (Figure 2.12). Similar results were recorded by the belief propagation method [41] (BP). The absolute-differencing method (AD) produced noisy results for the sloping side, where occluded regions lie, while the graph-cut algorithm [69] (GC) gave better estimates for these regions. For the stereo results featuring flowers (Figure 2.14), GC produced noisy results where occlusions occur. Although GC can produce smoother results with a larger regularisation parameter, an increase in  $\lambda$  would lead to fusing different depth layers together shown in the work by Song *et al.* [104]. The AD method was not reliable, and some results were too noisy (Figure 2.16). Better and more reliable disparity results were obtained by the BP and the proposed MR method (Figure 2.17). The BP method delivered accurate estimates for occluded regions, such as regions near the flower edges, while the proposed MR method produced less sharp results. On the other hand, BP produced isolated noise occasionally in low-texture areas, while the proposed MR method produced smooth results.

The average errors to process all 10 pairs of stereoscopic images shown in Table 2.4 also indicate that the BP and MR method were more reliable than the GC and AD method. Chapter 1 have discussed the subjective factors associated with human operators. Since there was only one height reading provided for plants grown in packs in this thesis, it was not clear that which

plant the measurement was exactly taken. In addition to this, there was a lack of multiple operators performing the same hand measurements, which is similar in the commercial growing environment. Despite of these limitations, this is only available reference to compare with.

Comparing the total running time required, the proposed MR method was by far the most efficient approach. The average time required by the MR method for a single set of stereoscopic images was only 616.8ms shown in Table 2.4. This is 4 times faster than the AD method. The BP and GC methods were much slower, requiring 9.870 and 194.018 seconds respectively on average for a single pair of stereoscopic images.

The choice of parameters was fixed throughout the experiment. Although parameters could be tuned for each stereoscopic pair, more human intervention is required, which is difficult in horticulture industry. Compared with the GC and BP methods, the proposed multiresolution stereo algorithm required tuning of at most three parameters (search range, window size and pyramid levels), and these parameters can be easily configured once for a particular camera setup to process different types of plant images. The GC method required tuning of 6 parameters, of which 4 are regularisation parameters and 2 are common parameters (search range, the number of iterations). The BP method required 3 common parameters (search range, the number of iterations, the number of levels) and 4 regularisation parameters.

## 2.9 Summary

An efficient stereo algorithm to estimate plant height has been discussed in this chapter. The proposed algorithm uses a Gaussian pyramid to represent input images at different resolutions. For each block in each resolution, a weighting method developed to prevent the direct parent node from propagating poor results. By using a Kalman filter updating at each level together with the weighting method, results from neighbours and scales are used to correct isolated wrong estimates. After that, the disparity results are converted into depths by triangulation. It has been demonstrated that the proposed stereo algorithm produces smooth depth maps for different plant types using the same parameter settings. Among all the algorithms tested, the proposed method was the most efficient, in terms of the running time, and the most accurate, when compared with hand measurements.



## Chapter 3

# A Self-Organising Map Model for Plant Surface Attributes

### 3.1 Introduction

The modern horticulture industry is scientifically managed, but the problem of quantifying and measuring plant quality remains a challenge for the industry [112]. The relative quality ranking of a few plants, in other words, “Is this plant ‘better’ than that?”, can be made, but the absolute quality, “Does this plant have a quality score of 4 out of 5?”, can be difficult to define for a particular plant without reference to a standard. Customers and producers could even have different standards for plant quality [39]. Despite this ambiguous definition, a set of plant attributes such as plant height and the amount of flowers can be quantified, and these attributes have been used as marketing specifications by commercial growers. In the previous chapter, a stereopsis approach to measure the plant height has been discussed. However, the

measurement collected by the stereo algorithm contains noise. Furthermore, although a naive multi-thresholding segmentation method [113] was used to distinguish between flowers and leaves, little effort was made to model the relevant plant attributes. This chapter presents details of a neural network approach to address these two problems, and discusses how to model the plant surface attributes.

Before discussing how to model the attributes, it is necessary to explore the available plant data. From a stereoscopic pair of images, the colour data and 2D position data are readily obtainable. The height data can be estimated by a stereo algorithm, as discussed in Chapter 2. These measurements can therefore be seen as a set of high dimensional surface points in a 3D coordinate system, in which each point is associated with a 6D vector as follows,

$$\begin{array}{ll} x, y, z & - \text{ 3D geometry} \\ r, g, b & - \text{ Colour} \end{array}$$

This is similar to the point clouds created by 3D scanners [66]. Since the height,  $z$ , is estimated and may be noisy, there is an uncertainty associated with every point, which is derived from the error score produced by the stereo algorithm.

## 3.2 Plant Modelling

This section reviews some previous work in the area of plant modelling. Anastacio *et al.* [5] introduced the idea of modelling the plant structure based on user sketches. The proposed system can automatically construct

a 3D plant model, given some user sketches for the main plant body and lateral organs. This provides an easy-to-use interface for digital illustration production that creates 3D representations from preliminary drawings.

Quan *et al.* [91] proposed an image-based approach to model plants directly from dozens of images. The semi-automatic modelling system can reconstruct a realistic shape and capture the complexity of a real plant. Results for plants like *Poinsettia* and *Nephthytis* have been presented to demonstrate the realism of the modelling. Tan *et al.* [107] followed a similar image-based modelling approach to modelling trees. There are a few improvements to the work by Quan *et al.* [91], including a partial coverage of the tree instead of a full 360° capture and pattern-based interpolation in occluded areas. Xu *et al.* [123] presented a tree model constructed from sparse point clouds produced by laser scanning, while image-based approaches [91, 107] recover 3D points from a set of images [75]. The image-based approach is also widely used in both cinema and television [14]. Among others, Mullins and Bowen [81] have shown that novel viewpoints can be generated from a fixed set of views.

Most of the approaches discussed above focus on the virtual realism of plants for visualisation purposes, which somewhat departs from the aim of this thesis. It is inappropriate to capture the details of every plant in a commercial glasshouse. This would not only increase the complexity of analysis and comparison, but also decrease the efficiency of automated assessment. The ideal model in this thesis therefore should represent the plants reasonably well and be relatively simple in terms of the model complexity.

### 3.3 Self-Organising Map: a Neural Network Approach

The Self-Organising Map (SOM) is one type of artificial neural network model, introduced by Kohonen [67]. The basic configuration of the SOM is a two-dimensional grid of cells denoted by  $\{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_K\}$ . Each cell contains a weight vector, which has six dimensions in this work,  $\{r, g, b, x, y, z\}$ , corresponding to the input data. Weight vectors are also known as neurons from a neural network point of view. Every cell is connected to a number of neighbours.

Yu [126] suggested the use of a SOM to reconstruct a surface from point clouds. However, his method was also used for visualisation purposes as discussed in Section 3.2. In this thesis, the purposes of applying the SOM are therefore twofold:

1. Eliminating the noise in height estimates,
2. Representing essential plant surface attributes.

In effect, the SOM models the topological skeleton of the surface shape, which can be used for representation and comparison purposes.

### 3.4 Competitive Learning

The SOM evolves by learning randomly-selected samples of input data over a period of time. There are two main steps to form the basis of this self-organising process:

1. Locate the best-matching cell,
2. Update the weights of this cell and its neighbours towards the value of the input data.

The Euclidean distance is commonly used to locate the best-matching cell. Consider at the time  $t$ , a SOM with  $K$  cells  $\mathbf{c}(j, t)$ ,  $j = \{1, 2, \dots, K\}$  and the input data sample  $\mathbf{x}(t)$ . The best-matching cell  $\mathbf{c}(b, t)$  can be located by the following equation:

$$b = \underset{j}{\operatorname{argmin}} \{ \|\mathbf{c}(j, t) - \mathbf{x}(t)\| \} \quad (3.1)$$

where  $\|\cdot\|$  denotes Euclidean distance, and  $b$  and  $j$  are positions of the cells  $\mathbf{c}(b, t)$  and  $\mathbf{c}(j, t)$  in the SOM respectively. Under this competitive learning scheme, every cell competes for a particular sample of the input data, and the winner cell is given by equation (3.1). This is also known as the ‘Winner-Take-All’ principle [64].

Having located the best-matching cell,  $\mathbf{c}(b, t)$ , all the cells in the SOM,  $\mathbf{c}(j, t)$ ,  $j = \{1, 2, \dots, K\}$ , are updated according to the update rule shown below:

$$\mathbf{c}(j, t+1) = \begin{cases} \mathbf{c}(j, t) + \alpha(t) [\mathbf{x}(t) - \mathbf{c}(j, t)] & \text{if } j \in h(j, b, t) \\ \mathbf{c}(j, t) & \text{otherwise} \end{cases} \quad (3.2)$$

where  $\alpha(t)$  is the learning rate and  $h(j, b, t)$  is the neighbourhood function. Equation (3.2) shows that the SOM updates the winner cell  $\mathbf{c}(b, t)$  as well as its neighbours under the constraint  $h(j, b, t)$ . This learning process repeats

until each dimension of the SOM has a sufficiently accurate representation for the input data.

### 3.5 The Neighbourhood function

The neighbourhood function controls how many cells around the winner update at a given time. The larger the area where the neighbourhood function has high values, the more rigid is the map. ‘Bubble’ is the simplest neighbourhood function [67], which is constant over the neighbourhood of the winner cell and zero elsewhere. Denote  $j$  and  $b$  as the positions of the cell  $\mathbf{c}(j, t)$  and the best-matching cell  $\mathbf{c}(b, t)$  at the time  $t, t = \{1, 2, 3, \dots, \infty\}$  respectively,

$$h(j, b, t) = \begin{cases} 1 & \text{if } b - \sigma(t) < j < b + \sigma(t) \\ 0 & \text{otherwise} \end{cases} \quad (3.3)$$

A more smooth neighbourhood function incorporates a Gaussian kernel,

$$h(j, b, t) = \exp \left\{ -\frac{(j - b)^2}{2\sigma^2(t)} \right\} \quad (3.4)$$

where  $\sigma(t)$  represents the neighbourhood radius, and it typically decreases with time. Since large neighbourhood radius makes the SOM more rigid, it is usually used when  $t$  is small, that is at the start, and then is gradually decreased to the final radius, which is usually set to 1. In this study,  $\sigma(t)$  is

calculated as follows,

$$\sigma(t) = \frac{1}{\log_{10}(1+t)} + 1 \quad (3.5)$$

### 3.6 The Learning Rate

The learning rate controls how fast the SOM learns, and is a decreasing function of time  $t$ . At the start, the network is not well structured and the SOM therefore quickly adapts to input data. In the later stages, since the network has a reasonable structure, the input data cause less changes than in the early stage. At a particular time  $t, t \in \{1, 2, 3, \dots, \infty\}$ , the learning rate is calculated as follows,

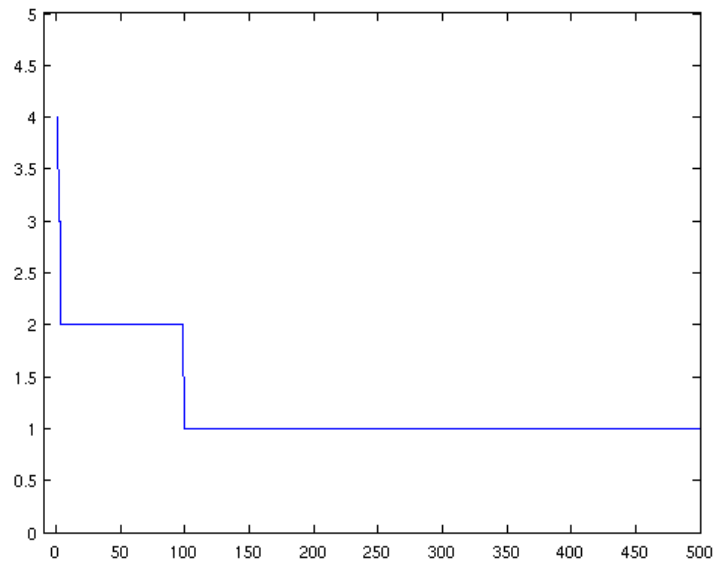
$$\alpha(t) = \frac{\alpha_0}{\log_{10}(1+t)} \quad (3.6)$$

where the initial value of  $\alpha$ ,  $\alpha_0$ , is set to 0.01.

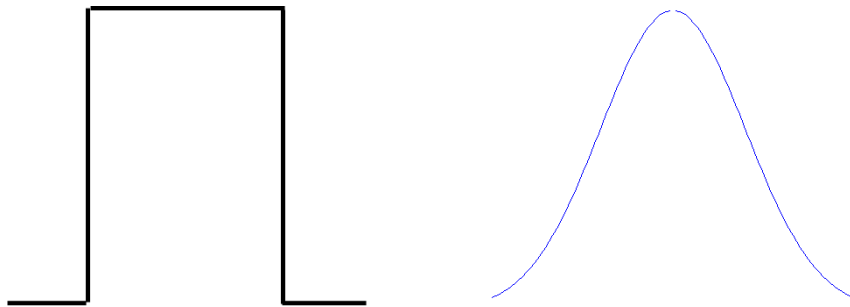
In this work, the learning rate also takes account of uncertainty. Since the estimates are noisy, the SOM learns these ‘bad’ samples slowly. From equation (3.6),

$$\alpha(t) = \frac{\alpha_0 \times [1 - s_e(t)]}{\log_{10}(1+t)} \quad (3.7)$$

where  $s_e(t)$  is the uncertainty for the input data sample  $\mathbf{x}(t)$ .

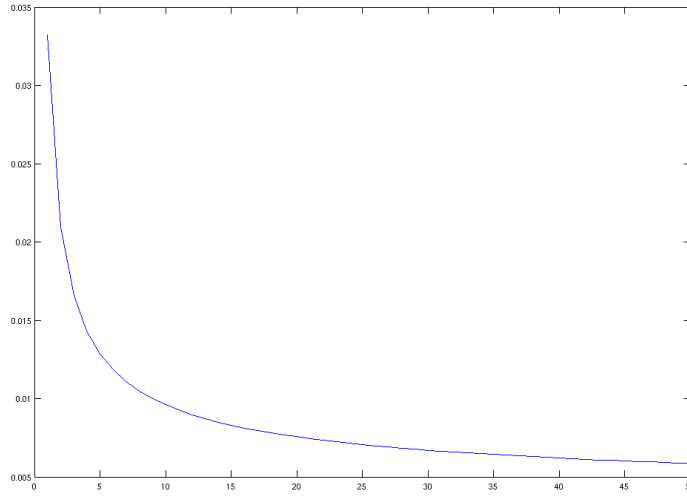


**Figure 3.1:** Neighbourhood radius  $\sigma(t)$  decreases over time.



**Figure 3.2:** Shapes of the two neighbourhood functions. Left: ‘Bubble’ function, Right: Gaussian function.





**Figure 3.3:** Learning rate decreases over time.

### 3.7 Tree-Structured SOM

The sequential learning process discussed above is very time consuming. Kohonen [67] discussed the batch version of the sequential SOM, since all the data samples  $\mathbf{x}(t)$ ,  $t = \{1, \dots, N\}$ , are available prior to computation. The ‘batch map’ is similar to the K-means algorithm. The nodes move towards the centre of a portion of input data samples at each iteration, and a few iterations usually suffice. Since there is no learning rate parameter, this approach is particularly effective if the initial SOM is roughly ordered.

Another approach to speed up learning is the hierarchical tree structure scheme [68, 109], which is analogous to the multiresolution model introduced in Chapter 2. Under this tree scheme, the learning starts at the top of tree, where there are only few cells. This can be seen as forming a codebook

**Table 3.1:** Summary of the tree-structured SOM algorithm

1. Start the SOM with a  $2 \times 2$  network.
2. Assign each data sample to a cell.
3. Each cell is divided into 4 children.
4. In the next level, for each data sample,
  - (a) Find the best-matching cell from data samples assigned in the parent cell,
  - (b) Update the best-matching cell and its neighbours,
  - (c) Re-assign the data sample to child cells.
5. Go back to Step 3

for each cell in the lower but larger SOM's, and the subsequent networks therefore only need to search the winner cell in the codebook instead of the whole SOM. This approach provides a more natural progression, since it only changes the way of searching the best-matching cell and does not change any step in the sequential learning.

The tree structure also solves the initialisation problem. In theory, the SOM can be initialised using arbitrary values for each node, and can be organised within a few hundred initial iterations [67]. It is easy to construct a good small SOM, since enough updates will be given to each node. This takes place at the top level under the tree scheme, where there are few nodes, like a  $2 \times 2$  network for example. Such a well-ordered SOM from the above level can be viewed as the initialisation to the SOM in the next level.

### 3.8 Results

This section presents some results using the tree-structured SOM. The SOM had  $32 \times 32$  cells on a rectangular-grid, and the Gaussian neighbourhood function discussed in this chapter was used. The number of iterations for the learning process is 10000 at each level.

The input data have 65536 samples. For the *Pansy* data, Figures 3.4 (a), 3.4 (b) and 3.4 (c) show the  $r, g, b$  colour, the height data  $z$ , and uncertainty  $s_e$  respectively. Figure 3.4 (d) shows the  $x, y$  positions of all the units in SOM from the top view. Figure 3.5 and 3.6 show the  $x, y, z$  positions and demonstrates how the tree-structured SOM evolves over different levels. Figure 3.7 presents the SOM with colour data in a 3D plot. Figures 3.8 and 3.9 show the SOM representation for two other plants, *Dianthus* and *Viola* respectively.

The log-likelihood value is computed for each SOM representation. The SOM can be seen as a 1024-component Gaussian mixture model [71], in which each component is a cell in the  $32 \times 32$  SOM. Let  $\mathbf{x}_i, i = \{1, 2, \dots, N\}$ , be data samples, and assume the component to which each data sample belongs is given by a random variable  $\mathbf{y}_j, j = \{1, 2, \dots, K\}$ . The probability density function for  $\mathbf{x}_i$  given  $\mathbf{y}_j$  is shown below:

$$p_{\mathbf{x}_i|\mathbf{y}_j}(\mathbf{x}_i|\theta) = \frac{1}{(2\pi)^{D/2}} |\Sigma_j|^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} (\mathbf{x}_i - \mu_j)^T \Sigma_j^{-1} (\mathbf{x}_i - \mu_j) \right\} \quad (3.8)$$

where  $\mu_j$  and  $\Sigma_j$  are the mean and covariance for the component  $j$ , and  $D$  is the dimension of the input data. In this thesis, each SOM cell is treated as a component, and the mean  $\mu_j$  is therefore equal to the weight vector

$\mathbf{c}_j, j = \{1, \dots, K\}$ . The covariance  $\Sigma_j$  is computed from the data assigned to each cell  $\mathbf{c}_j$ .  $\theta$  denotes the set of parameters,  $\theta = \{\mu_1 \dots \mu_K, \Sigma_1 \dots \Sigma_K\}$ .  $D$  is the data dimension.

Since the random variable  $\mathbf{y}_j$  is unknown, the density function for  $\mathbf{x}_i$  can be computed by applying the definition of conditional probability and summing over  $j$ ,

$$p_{\mathbf{x}_i}(\mathbf{x}_i|\theta) = \sum_{j=1}^K \pi_j p_{\mathbf{x}_i|\mathbf{y}_j}(\mathbf{x}_i|\theta) \quad (3.9)$$

where  $\pi_j$  represents the mixture weight. The log-likelihood of the entire data  $\mathbf{x}_i, i = \{1, \dots, N\}$  is then given by:

$$\log(p_{\mathbf{x}}(\mathbf{x}|\theta)) = \log\left(\sum_{i=1}^N p_{\mathbf{x}_i}(\mathbf{x}_i|\theta)\right) \quad (3.10)$$

The log-likelihood value is negative, since  $0 \leq p_{\mathbf{x}}(\mathbf{x}|\theta) \leq 1$ , and values that are more close to zero imply a better data fit. The Gaussian mixture model is discussed in detail in Section 4.3.

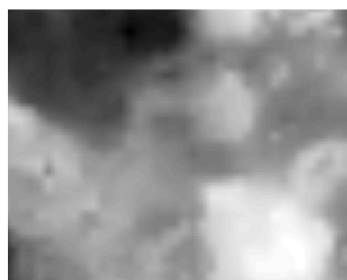
Table 3.2 presents the log-likelihood value and the running time for the plants *Pansy*, *Dianthus* and *Viola*. The variation in the log-likelihood values indicates that the SOM fits the *Pansy* data better than *Dianthus* or *Viola*. This is caused by the more arbitrary shape in *Dianthus* and *Viola* data, which is represented as a smooth surface in the SOM.

**Table 3.2:** Comparisons of the log-likelihood and the running time for *Pansy*, *Dianthus* and *Viola*.

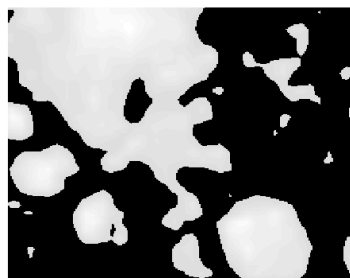
Plant	Log-likelihood Value	Time in Seconds
<i>Pansy</i>	-1017960	6.089
<i>Dianthus</i>	-1153788	6.688
<i>Viola</i>	-1203239	6.203



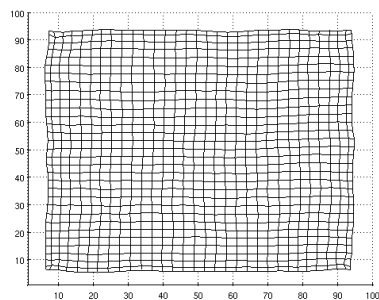
(a) Colour data

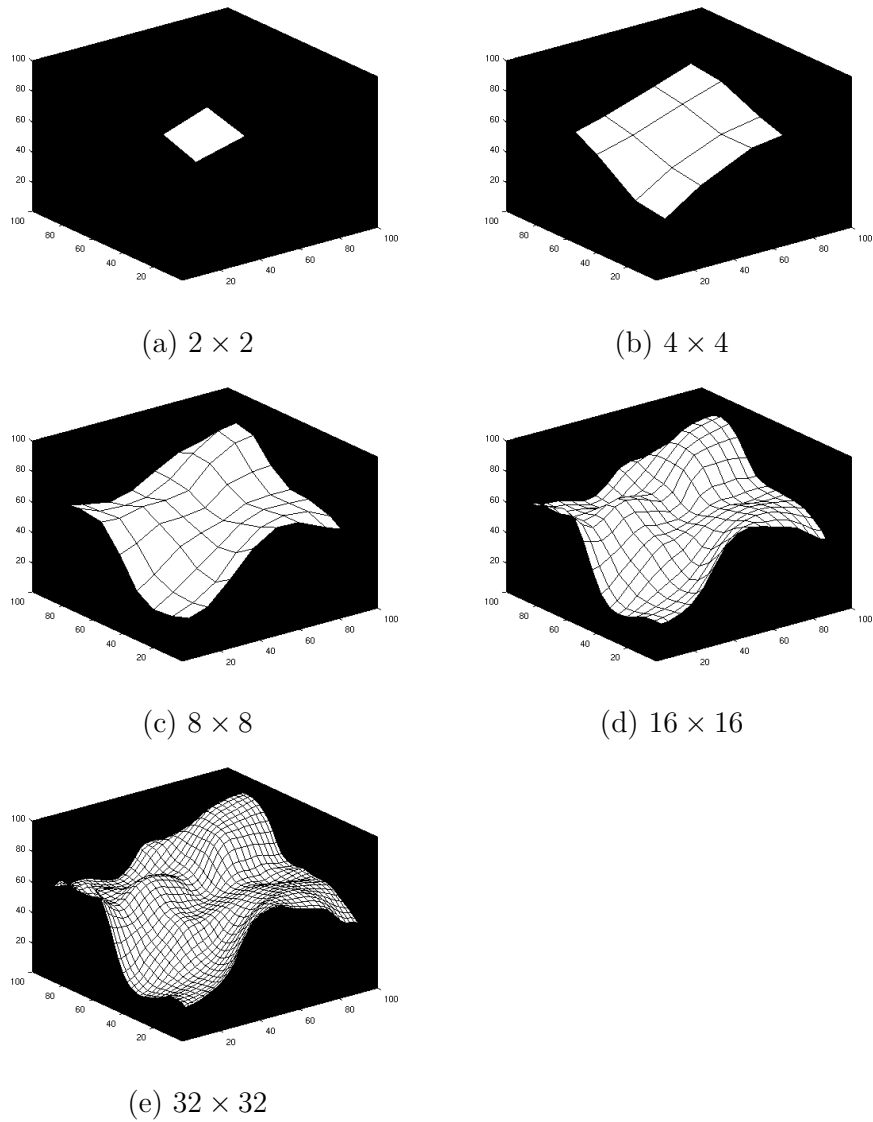


(b) Height data

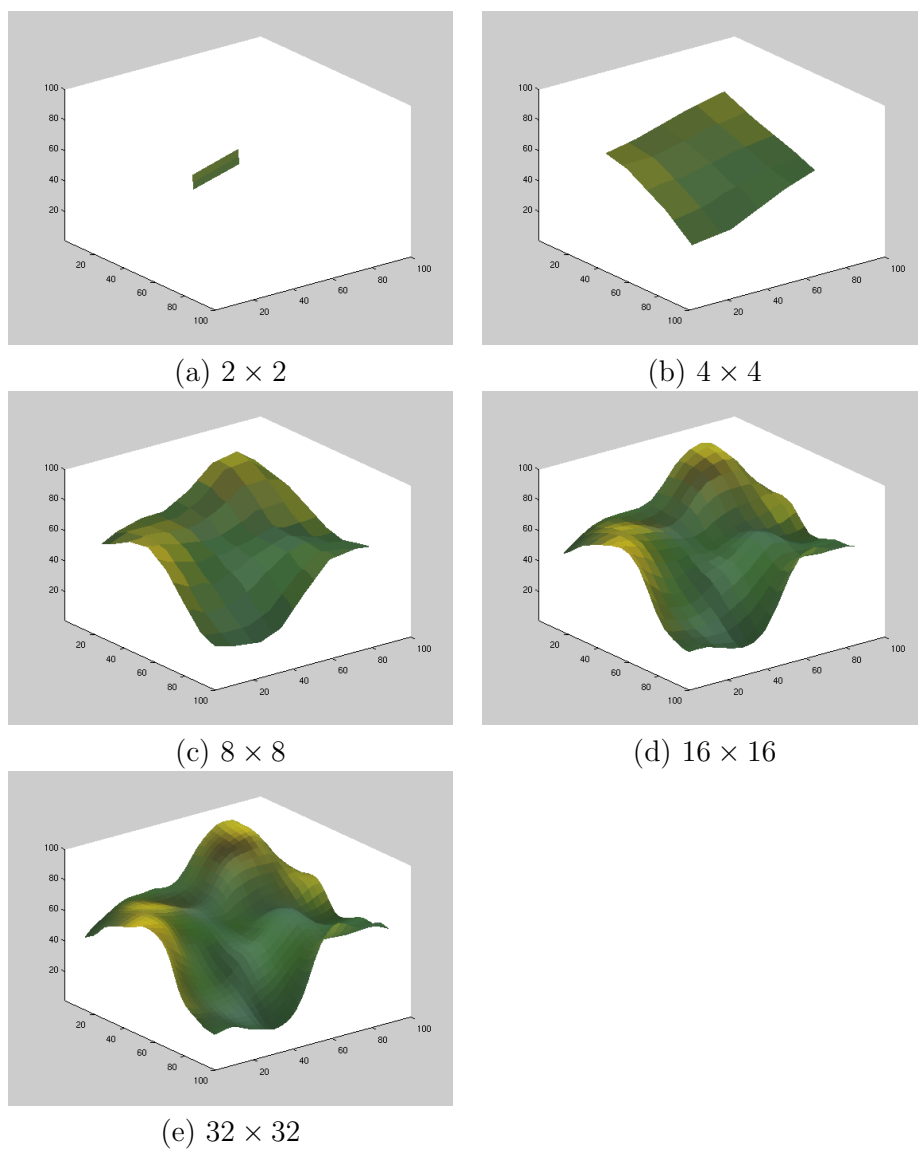


(c) Uncertainty

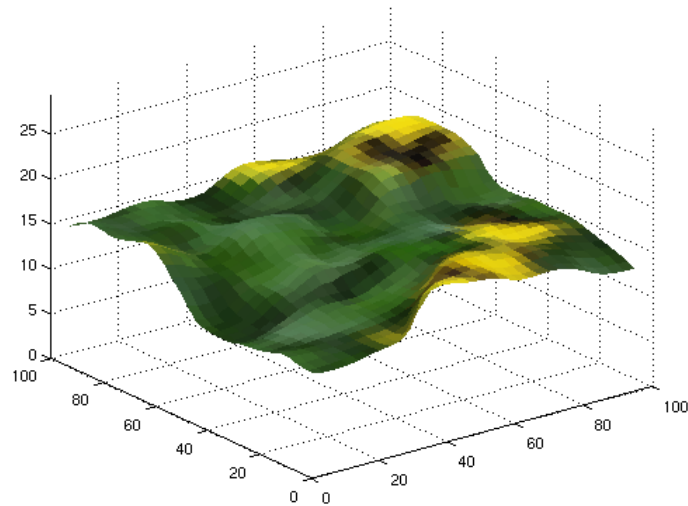
(d) SOM nodes  
from the top view**Figure 3.4:** *Pansy* data and SOM nodes.



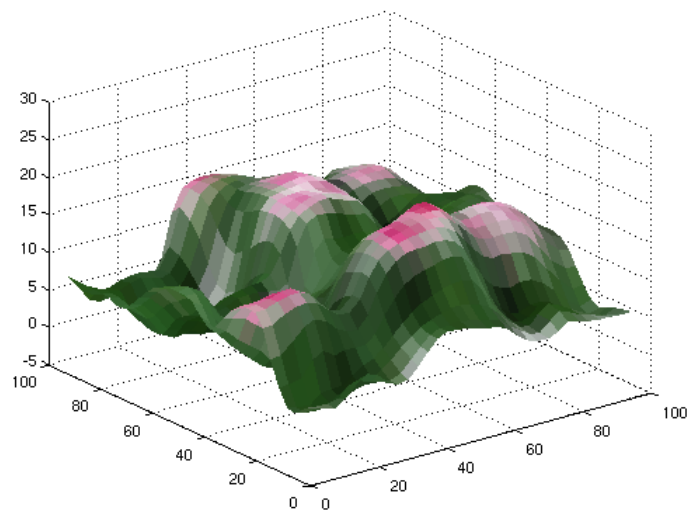
**Figure 3.5:** Evolution of the tree-structured SOM over levels shown as a 3D surface plot.



**Figure 3.6:** Tree-structured SOM at different levels shown as a 3D surface plot. The surface is textured with the mean colour for each cell.

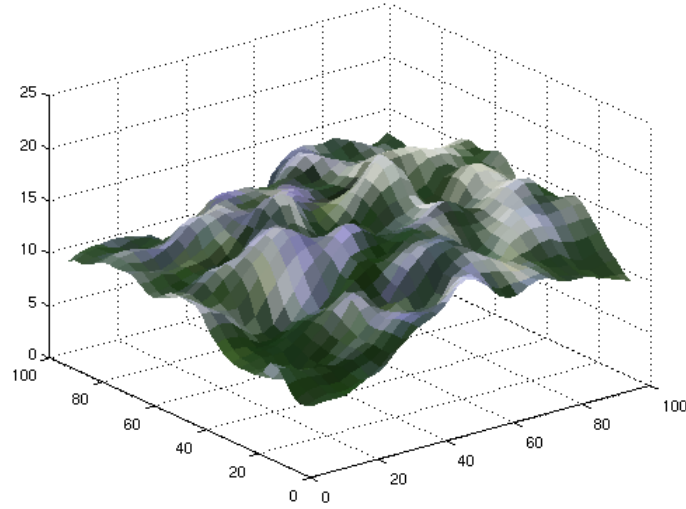


**Figure 3.7:** SOM representation for *Pansy*. The surface is textured with the mean colour for each cell.



**Figure 3.8:** SOM representation for *Dianthus*. The surface is textured with the mean colour for each cell.





**Figure 3.9:** SOM representation for *Viola*. The surface is textured with the mean colour for each cell.

### 3.9 Discussion

Applications using the SOM can be classified into two areas [125]:

1. Dimensionality-reducing mapping that is topology-preserving;
2. Classification of input data.

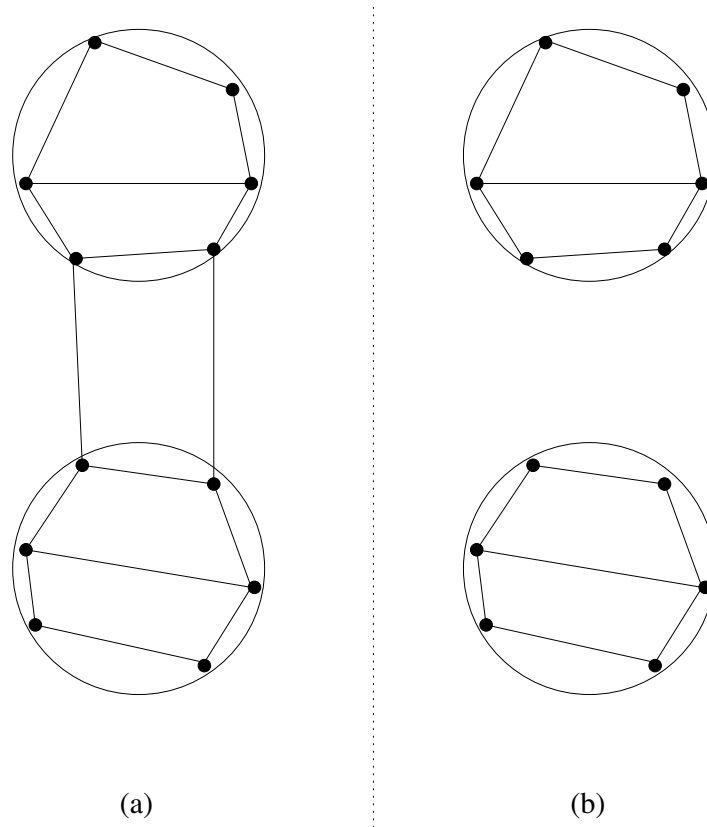
The results in the previous section have demonstrated that the SOM is good as a mapping tool, since it preserves the global geometry as well as colour information of the objects featured in the images. Incorrect heights estimated by the stereo algorithm can be effectively eliminated and the complex visual appearance is also simplified.

However, there are some issues when applying the SOM to classify input data. The output of the SOM is identical to the K-means algorithm

if the neighbourhood radius is 0, that is only the winner cell is updated. The quality of the classification depends largely on the initial set of clusters, and could be much poorer than the global optimum in practice [17]. The neighbourhood connection could also contribute to undesirable results, and Figure 3.10 (a) presents such a case. Suppose there are two clusters in the data, and samples are randomly drawn from the two distributions. The SOM might have some cells outside the target distributions, because of the connection between distant neighbours. If the SOM structure is adaptive, the connection between neighbours can be broken. This scheme, known as dynamically defined topologies [19, 67], would lead to two completely separated SOMs in this example shown in Figure 3.10 (b), which describes the two clusters better than the fixed topology does.

Another problem with SOM is the ‘Winner-Take-All’ learning principle. If there are significant differences in variance of the data, some nodes may win too much, and thus dominate selection and response. The result is that some nodes may never win. Ahalt *et al.* [2] discussed the frequency-sensitive competitive learning approach, which keeps a count of how frequently each unit is the winner. Kohonen discussed the use of adaptive tensorial weights [67] instead of the unweighted Euclidean metric, and some results presented have shown that this can force each cell to be updated approximately as often as the others.

Determining the final size of a SOM is ambiguous. In this thesis, the tree structure is ‘grown’ from a  $2 \times 2$  network. Although this does not require any effort to configure at the start, the final size was determined after a number of tests. The quality of the SOM for each level on the tree can be



**Figure 3.10:** Neighbourhood problem using SOM. Two separated SOMs describe two clusters better than a single SOM with fixed topology does

quantified in a number of ways [67], but the SOM might not be optimal in certain regions. To address this problem, the SOM can be treated as a mesh, and some mesh optimisation techniques [54, 55] can therefore apply.

SOM cannot form optimal Bayesian classifications unless the data are uniformly distributed or well separated [125]. The classification can only be binary for each cluster, since SOM is not a probabilistic model. In practice, the distributions usually overlap, and their joint distribution can be described by a mixture distribution [57]. In this case, data samples could belong to any of the distributions, and a non-probabilistic model therefore is not applicable.

### 3.10 Summary

This chapter presents a neural network approach to modelling plant surface attributes. The SOM can preserve the global geometry and colour information in the image, and incorrect height estimates can be eliminated. However, the fixed structure can cause issues in optimally representing the data, and the choice of the final size is ambiguous. Furthermore, the SOM cannot form optimal classification when overlapped distributions are present, since it is not a probabilistic model and is similar to the K-means algorithm if no neighbourhood update is used.

## Chapter 4

# Statistical Modelling of Plant Surface Attributes

### 4.1 Introduction

Heuristic procedures, such as the self-organising map, can be used to model plant surface attributes; nevertheless, the lack of a statistical basis appears to be a major limitation. For example, classical questions, such as the number of clusters, are theoretically impossible to solve. Using heuristic algorithms, it is difficult to know which metric to apply, especially for high dimensional data such as the plant image data presented in Chapter 3. Above all, these methods do not offer a probabilistic model of the data, and it is therefore hard to know how ‘good’ the result is, for comparisons with other models, and for making predictions and incorporating new data into the existing model. This chapter presents a statistical approach to overcome these limitations.

## 4.2 The K-means Algorithm

The K-means algorithm is a simple approach to modelling high dimensional data, in which each centroid represents a number of samples. The algorithm places  $K$  initial centroids into the data, and assigns each data sample to the closest centroid. When all samples have been assigned, the  $K$  centroids are updated according to a distance measure such as the Euclidean distance. This process repeats until all the centroids no longer change significantly, and this produces a separation of the data into clusters.

Although the K-means algorithm is simple and fast to iterate, it is a gradient descent method and therefore incapable of finding a global optimum. In other words, it will converge on a solution that minimises the distance between centroids and data defined by the used distance measure. However, due to the fact that the distance measure might be inadequate, the solution is not guaranteed to be globally optimal. The K-means algorithm is sensitive to the initial centroids, and without careful selection of initial centroids, the result could be much poorer than the global optimum [17]. Since the algorithm is fast, a common method is to run the algorithm several times from different initial conditions and then determine the best solution.

The K-means algorithm and Self-Organising Map (SOM) are closely related. As discussed in Chapter 3, SOM produces identical results to the K-means algorithm if no neighbourhood update is used. Since data are assigned in a ‘hard’ way by both methods, in other words, each data sample either belongs to a cluster or not, optimal Bayesian classifications cannot be produced unless the data are well separated [125].

A Gaussian Mixture Model (GMM), discussed below, can resolve many of these issues. It has been proven that a Gaussian mixture can approximate arbitrary probability densities for general cases [120]. The GMM can be readily updated in a Bayesian framework, and it can therefore allow the current estimates to be conditioned on the previous. Since it is a probabilistic model, it is possible to derive the probability that a particular data sample belongs to a given cluster.

### 4.3 Gaussian Mixture Model: a Probabilistic Model Approach

A probabilistic model offers a principled alternative to the K-means algorithm and SOM. In the context of probabilistic modelling, the input data are viewed as coming from a mixture of probability distributions. Compared with the K-means algorithm, a mixture model offers a ‘soft’ assignment option. In other words, the estimates are associated with a probability, which can determine how well the data fit in a cluster.

Using the Gaussian Mixture Model (GMM), each cluster could be represented by a Gaussian distribution, and the entire data set is modelled by a mixture of these distributions. The true density  $g(\mathbf{x})$  of a mixture can be approximated as follows:

$$g(\mathbf{x}) \simeq \hat{g}(\mathbf{x}) = \sum_{j=1}^K \pi_j f_j(\mathbf{x}|\theta_j) \quad (4.1)$$

where the mixture weights  $\pi_j$  satisfy  $\sum_{j=1}^K \pi_j = 1$ , and  $\theta_j$  denotes model pa-

rameters. An individual distribution used to model a specific cluster is often referred to as a component distribution. Suppose there are  $K$  components in the input data  $\mathbf{x}_i$ ,  $i = \{1, \dots, N\}$ , and each component is a Gaussian parameterised by its mean vector  $\mu_j$  and covariance matrix  $\Sigma_j$ ,  $j = \{1, \dots, K\}$ . The density of component  $j$  is

$$f_j(\mathbf{x}|\theta_j) = N_v(\mathbf{x}|\mu_j, \Sigma_j) \quad (4.2)$$

$$= \frac{1}{(2\pi)^{D/2}} |\Sigma_j|^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \mu_j)^T \Sigma_j^{-1} (\mathbf{x} - \mu_j) \right\} \quad (4.3)$$

where  $D$  is the dimension of the input data, which is 6 in this thesis. A sample of input data  $\mathbf{x}_i$ ,  $i = \{1, \dots, N\}$ , can therefore be expressed as

$$\mathbf{x}_i \sim N_v(\mu_j, \Sigma_j) \quad (4.4)$$

## 4.4 MGMM: Multiresolution Gaussian Mixture Model

The Multiresolution Gaussian Mixture Model (MGMM) [120] discussed in this section is analogous to the tree-structured SOM presented in Chapter 3. The idea of using a MGMM comes from the assumption that the density of interest is unknown at the start, but it can be estimated from some data. For example, a *Pansy* can be modelled by a two-component mixture model, in which one represents the flower and the other represents the leaf. The sample mean and covariance can therefore be computed for this two-component mixture model. Then, a measure of goodness-of-fit can be applied to check



if the model fits the data adequately. In some situations, this model would fit well. If not, the component with poorer fit to the data can be split and the goodness-of-fit test can be applied again to the new model. This recursion forms a tree of Gaussians of decreasing variance, which has the property of approximating the desired density of the data as the number of levels increases [120].

The idea of hierarchical mixture modelling was also explored by many authors [52, 92, 119]. Rasmussen [92] thoroughly analysed a Dirichlet process mixture model with Gaussian components, and presented an implementation using Gibbs sampling. Williams [119] described a hierarchical mixture model in a tree-structured manner, and demonstrated the results on univariate data using the Metropolis sampling algorithm. Heller and Ghahramani [52] presented prediction and hyperparameter optimisation procedures for a Bayesian hierarchical model based on Dirichlet process mixtures, as a replacement of the sampling technique, which, for example, is used in Williams's work [119].

## 4.5 Estimation of Mixture Distributions

There are a number of approaches for estimating mixture distributions, of which the well-known ones are maximum likelihood and Bayesian methods. McLachlan and Peel [79] reviewed the literature on different approaches to this problem, and gave detailed examination of each technique.

Dempster *et al.* [33] drew the attention to the use of the Expectation-Maximisation (EM) algorithm to the problem of fitting mixture models by maximum likelihood. The idea of the EM algorithm is to associate a com-

plete data model to the incomplete structure that is observed in order to simplify the computation of maximum likelihood estimates, and the problem of fitting mixture models is a classic example that is simplified considerably by EM's maximum likelihood estimation from the missing data structure. The EM algorithm has two steps, namely an 'E' step and an 'M' step. The E step is to estimate the expected value of the unknown variables, given the estimates of distribution parameters, while the M step is to estimate the distribution parameters to maximise the likelihood of the data, given the expected estimates of the unknown variables.

However, the EM algorithm is sensitive to starting values, and Seidel *et al.* [100] have demonstrated how different starting strategies can lead to different estimates using the EM algorithm. A more robust approach is to apply stochastic techniques, such as Gibbs sampling, discussed below. The EM algorithm can be viewed as a deterministic version of Gibbs sampling [94], and both EM and Gibbs sampling are used for approximation with incomplete data. EM operates on unknown variables using sufficient statistics instead of sampling unknown variables as a Gibbs sampler does.

## 4.6 Gibbs Sampling: a Bayesian Approach

With the low cost of computation and rapid development in posterior simulation techniques, such as Markov Chain Monte Carlo (MCMC), Bayesian methods for the analysis of data have become increasingly popular [79]. Given appropriate prior distributions, Bayes estimators for mixture models are well defined [36]. In a Bayesian framework, uncertainty concerning

the model parameters is expressed by means of a probability distribution over the possible values. The distribution is updated as new information becomes available, and a MCMC sampling scheme can then be applied to draw samples from the distribution in a stochastic manner.

Gelfand and Smith [43] discussed the practical implementation of the Bayesian approach for estimation of mixture distributions in a wide variety of statistical problems, and the use of Gibbs sampling in particular. Consider a data sample from  $\mathbf{x}_i$ ,  $i = \{1, \dots, N\}$ , from the Gaussian mixture distribution:

$$\mathbf{x}_i \sim N_v(\mathbf{x} | \mu_{z_i}, \Sigma_{z_i}) \quad (4.5)$$

Robert [93] proposed that the class labels  $z_i, z_i \in \{1, \dots, K\}$ , which identify the component to which each data sample belongs, can be seen as the missing data part of the sample. Simulation conditional on  $z_i$  is possible, if the so called ‘conjugate structure’ is preserved [95], which means posterior has the same form of distribution as prior. This implies using the conjugate densities for the Gaussian mixture: Dirichlet for the mixture weights, Gaussian for the means and inverse Wishart for the covariances. Appendix A provides further details on this subject.

Given the missing data structure, the mixture model can be identified from samples from posteriors by assuming conjugate priors. In the Gibbs sampler described below, the missing data and model parameters are part of a missing-data chain and a parameter chain, both of which are Markov. The Gibbs sampler can therefore be implemented in two steps. In the first step, the class labels are fixed and samples are drawn from the conditional

distributions of the mixture parameters.

Step 1. Simulate

$$\pi_j \sim D(\alpha_j^*) \quad (4.6)$$

$$\mu_j \sim N_v(\epsilon_j^*, \tau_j^* \Sigma_j) \quad (4.7)$$

$$\Sigma_j^{-1} \sim W_v(\omega_j^*, \lambda_j^*) \quad (4.8)$$

$N_v$  denotes the multivariate normal distribution,  $W_v$  denotes the Wishart distribution, and  $D$  denotes the Dirichlet distribution.

$$\alpha_j^* = \alpha_j + n_j \quad (4.9)$$

$$\epsilon_j^* = \frac{n_j \bar{\mathbf{x}}_j + \epsilon_j \tau_j^{-1}}{n_j + \tau_j^{-1}} \quad (4.10)$$

$$\tau_j^* = \frac{\tau_j}{\tau_j n_j + 1} \quad (4.11)$$

$$\lambda_j^* = \lambda_j + n_j \quad (4.12)$$

$$\omega_j^* = (\omega_j^{-1} + S_j + \frac{n_j}{\tau_j n_j + 1} (\bar{\mathbf{x}}_j - \epsilon_j)(\bar{\mathbf{x}}_j - \epsilon_j)^T)^{-1} \quad (4.13)$$

where  $\alpha_j, \epsilon_j, \tau_j, \lambda_j, \omega_j$  are parameters of the prior distributions for the  $j$ th component. Parameters with a \* denote the posterior parameters.  $n_j$  is the number of observations in the  $j$ th component,  $j = \{1, \dots, K\}$ .  $\bar{\mathbf{x}}_j$  is the sample mean for the  $j$ th component, and  $S_j$  is the sample covariance,

$$S_j = \sum_{i=1}^j (\mathbf{x}_i - \bar{\mathbf{x}}_j)(\mathbf{x}_i - \bar{\mathbf{x}}_j)^T \quad (4.14)$$

In the second step, the mixture parameters are fixed, and the class labels are drawn according to them.

Step 2. Derive the missing data  $z_i$ ,  $i = \{1, \dots, N\}$ , from a simulated standard uniform random variable  $u_i$ ,  $z_i = j$ , if  $\pi_{i1} + \dots + \pi_{i(j-1)} < u_i \leq \pi_{i1} + \dots + \pi_{ij}$ , where

$$\pi_{ij} \propto \pi_j L_{ij} \quad (4.15)$$

$L_{ij}$  is the likelihood for each observation with  $\mu_j$  and  $\Sigma_j$  for each class  $j$ ,  $j = \{1, \dots, K\}$

$$L_{ij} = \frac{1}{(2\pi)^{D/2}} |\Sigma_j|^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} (\mathbf{x}_i - \mu_j)^T \Sigma_j^{-1} (\mathbf{x}_i - \mu_j) \right\} \quad (4.16)$$

Note that the posterior estimates are not just the sample means, but include the effect of the priors, avoiding some of the problems associated with maximum likelihood estimation.

## 4.7 Model Split and Selection

The component  $j$ ,  $j = \{1, \dots, K\}$ , with the worst data fit, for example, the log-likelihood value, is assumed to contain more than one Gaussian, and this component is therefore divided into sub-components. In effect, the data allocated to the component  $j$  is a mixture of Gaussians. A simple yet effective way of splitting the component  $j$  is shown below:

1. Initialise  $\mu_j^*$  and  $\Sigma_j^*$  for sub-components by varying the mean and covariances of the component  $j$ .
2. Calculate the probabilities under  $\mu_j^*$  and  $\Sigma_j^*$  for the data allocated to the component  $j$ .

In this thesis, the number of sub-components is set to 2, that is, the component is divided into two for each split.

After splitting the model, a new model is available, and the following question immediately arises, “Does the new model represent the data sufficiently better than the original model?” In this thesis, the original model splits itself, and the new model therefore always has more components to fit the data. In this case, the data fit improves a certain amount. However, this does not mean the new model is always better, since there are more parameters and the complexity is increased. A criterion is therefore required to balance the data fit and the model complexity.

This is well researched in the literature [21]. One approach is the Akaike Information Criterion (AIC) developed by Akaike in 1974 [3], which is based on the Kullback-Leibler information loss [74]. Another widely used alternative to AIC is the Bayesian Information Criterion (BIC) adopted by Schwarz in 1978 [99]. The BIC can be seen as an approximation to the Bayes factors for Bayesian model choice, although exact Bayesian model selection can be much more complicated than BIC [42]. Wilson [120] discussed the use of a log-Bayes factor criterion and minimum information criterion.

The formulas for the AIC and BIC are:

$$AIC = 2\theta - 2\log(L) \quad (4.17)$$

$$BIC = \log(M)\theta - 2\log(L) \quad (4.18)$$

where  $L$  is the likelihood of observing the data given the model;  $\theta$  is the number of parameters in the model, which describes the complexity of the model;

**Table 4.1:** Comparisons of the log-likelihood and the model complexity for BIC criterion and BIC with a  $\log(K)$  penalty.

	Number of Components	Log-likelihood Value
BIC	36	−1368832
BIC with $\log(K)$ penalty	18	−1438996

$M$  represents the dimensionality and the number of data points. Lower values in both criteria imply the preferred model. It is also shown in equation (4.17) and equation (4.18) that the AIC penalises model parameters less than the BIC does.

The goodness-of-fit test used in this thesis is derived from the BIC, since the BIC appears to favour the data fit and results in making too many components in a number of experiments. A penalty,  $\log(K)$ , where  $K$  is number of components, was added to avoid splitting into too many components,

$$\alpha = \log(M) \log(K) \theta - 2 \log(L) \quad (4.19)$$

In effect, this penalty gives more weight to model complexity than BIC does. Results using this criterion are shown in Table 4.1 and Section 4.9.

## 4.8 Speed and Efficiency

Selectively updating using neighbouring components can improve the speed of likelihood computation of the observations. For each data point  $\mathbf{x}_i$ , some components, usually ‘far away’ from  $\mathbf{x}_i$ , have little or no connection at all. Therefore, the likelihoods of  $\mathbf{x}_i$  for these components can be approximated

to zero. In this study, the probability of each component  $j$  for  $\mathbf{x}_i$  is used as a distance metric. For example, if  $\mathbf{x}_i$  has a probability of 0% to component  $j$ , this means component  $j$  has no relationship with  $\mathbf{x}_i$ . In this study, the threshold of probability is set to 0.001%, which keeps connections to possible neighbouring components and discards ‘far away’ ones.

Determining the number of components is a very expensive operation at the full scale. In this study, the number of components as well as  $\mu_j$  and  $\Sigma_j$  for each component  $j$  are estimated using a small portion of the data. Then, only neighbouring Gaussians are used to selectively update at the full scale. However, a lack of data can lead to weak likelihoods, which would not correct some badly-estimated priors. Hence, in this study, an intermediate scale is introduced to update the model structure and parameters.

Table 4.2 presents a summary of the Multiresolution Gaussian Mixture Model used in this thesis.

## 4.9 Results

### 4.9.1 Modelling Results for a set of Gaussians Data

This section presents results of applying the Multiresolution Gaussian Mixture Model (MGMM) to synthetic data. The data contain six 2D Gaussians. Four of the Gaussians are placed at corners in the  $x$ - $y$  image domain, while two are in the middle and have the same mean but different covariances (a ‘cross’ shape). The total number of data samples is 65536, with around 10000 samples for each Gaussian.



**Table 4.2:** Summary of the Multiresolution Gaussian Mixture Model**Step 1: Initial Guess**

The model parameters  $\mu_j$  and  $\Sigma_j$  for each class  $j, j = \{1, \dots, K\}$  are initialised by K-means or training procedures.

**Step 2: Gibbs Sampling**

Apply the Gibbs sampling method to update model parameters.

**Step 3: Model Split**

Split the cluster which has a bad fit to the data.

**Step 4: Model Selection**

Compute the goodness-of-fit for the post-split model  $\alpha_{new}$  and the pre-split model  $\alpha_{ori}$ . If  $\alpha_{new} > \alpha_{ori}$ , go back to Step 2; Else, return the pre-split model as the output.

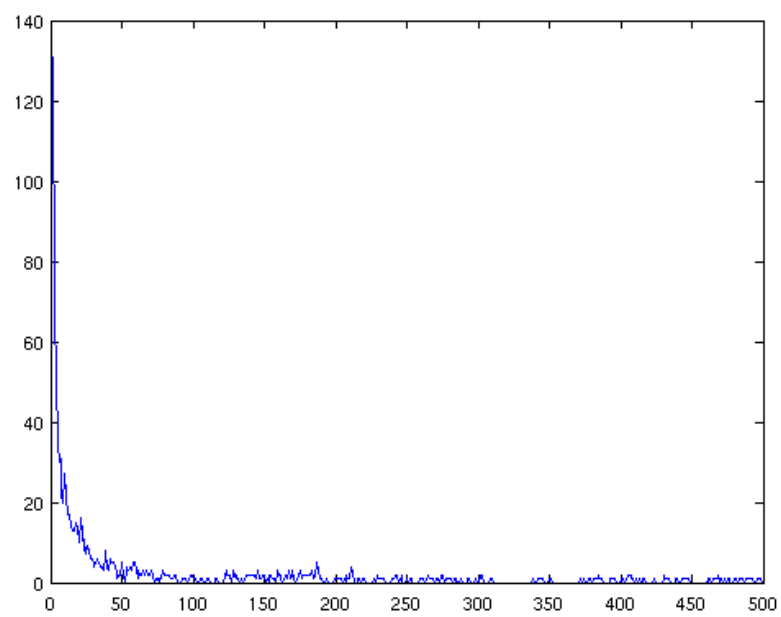
The MGMM was configured to start as a two-component model, with both components starting from the location of the ‘cross’ in the middle. As discussed in this chapter, a new component was added into the model if the goodness-of-fit test on the model at that resolution was not satisfied. After a number of empirical tests,  $\alpha_j$ ,  $\epsilon_j$ , and  $\omega_j$  were set according to components in the parent model.  $\tau_j$  and  $\lambda_j$  were set to 1 and 8 respectively [93]. The number of Gibbs sampling iterations was set to 300, and Figure 4.1 presents the number of changes in class labels against the number of Gibbs sampling iterations. It was found in experiments that the number of changes in class labels does not vary significantly beyond 200 iterations, and 300-iteration ensures the convergence of the Gibbs sampler. The number of EM algorithm was set to 30, and Figure 4.2 plots the log-likelihood values against the number of EM iterations.

**Table 4.3:** Comparisons of the log-likelihood and the running time for Gaussians data. All four results had the same starting values, and two variations of K-means algorithm were used to illustrate the ‘cross’ problem.

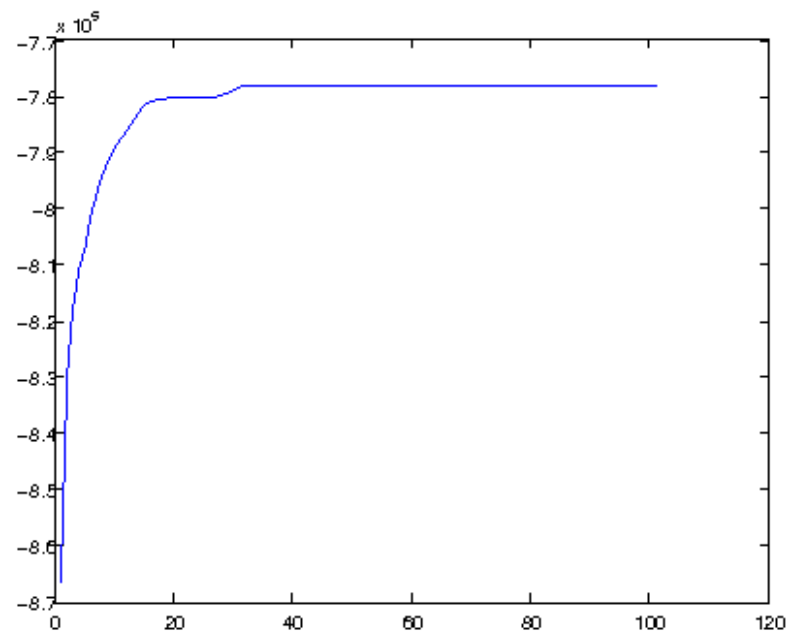
	Number of Components	Log-likelihood Value	Time in Seconds
Gibbs sampling	6	−253818	37.220
EM	6	−256188	3.872
K-means	5	−291037	0.759
K-means	6	−296737	0.793

Figure 4.3 shows how the MGMM works to identify the number of components at the coarsest resolution. Figure 4.4 presents how the MGMM develops over 3 resolutions. Figures 4.5 and 4.6 present the results using Gibbs sampling and the EM algorithm respectively, while Figure 4.7 shows the comparisons with the K-means algorithm. The K-means algorithm was configured to have 6 centroids, identical to the number of Gaussians. It is clear that the K-means with 6 centroids could not locate all components in the data, which was caused mainly by the ‘cross’ problem. A better result was obtained by using the K-means with 5 centroids, but there was one Gaussian missing.

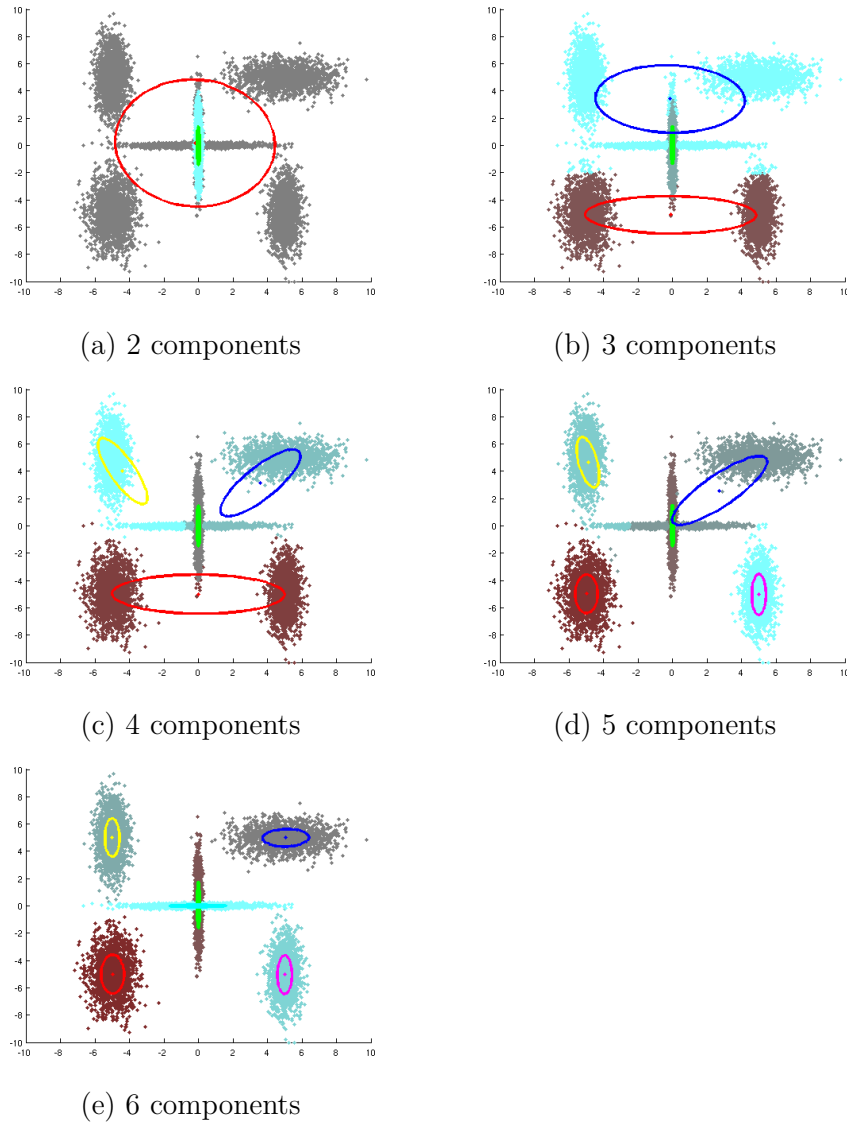
Table 4.3 presents the log-likelihood value and the running time using the four implementations. The procedure to compute the log-likelihood value for the K-means is identical to the SOM discussed earlier, in which each centroid represents the mean in the Gaussian mixture model and the covariance is computed from the data assigned to each centroid.



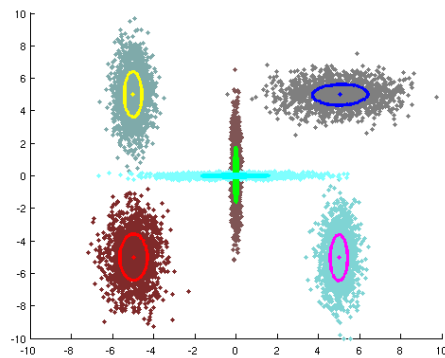
**Figure 4.1:** A plot of the number of changes in class labels against the number of Gibbs sampling iterations for 2D ‘shape’ data.



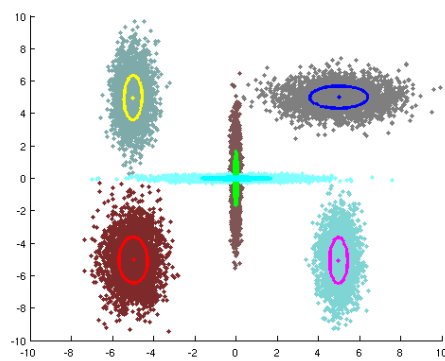
**Figure 4.2:** A plot of log-likelihood values against the number of EM iterations for 2D ‘shape’ data.



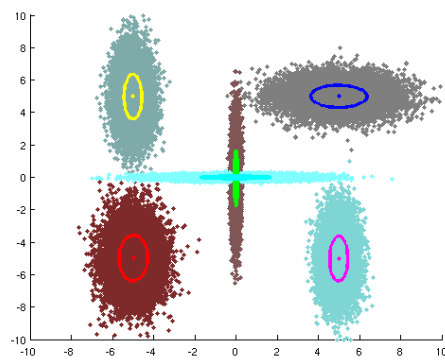
**Figure 4.3:** Evolution of the MGMM over different numbers of components at the coarsest level. The dot and the circle represents the mean and covariance respectively.



The coarsest level

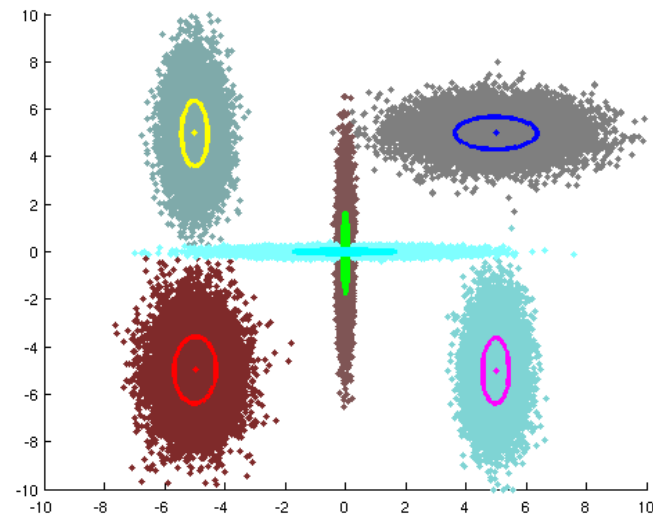


The intermediate level

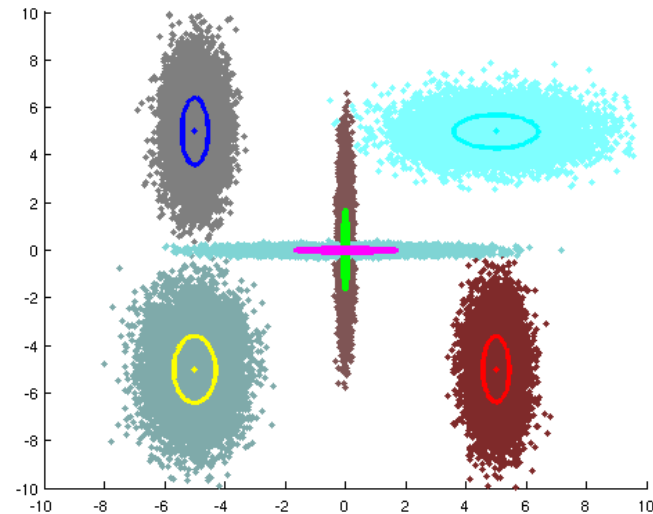


The finest level

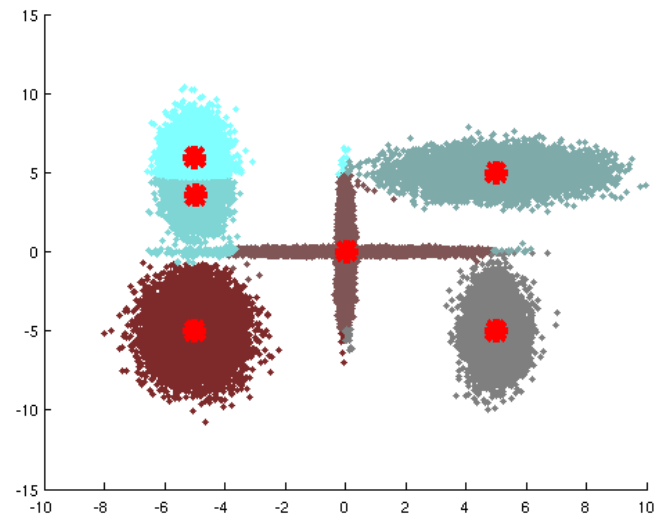
**Figure 4.4:** Evolution of the MGMM over levels. The dot and the circle represents the mean and covariance respectively.



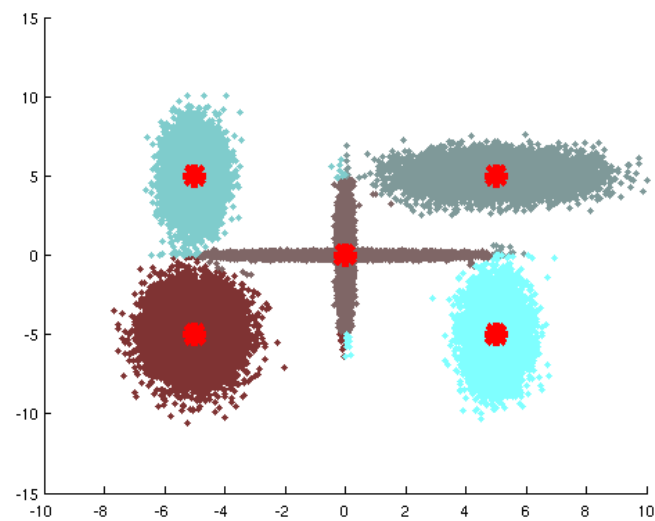
**Figure 4.5:** Results of Gibbs sampling for the Gaussian data. The dot and the circle represents the mean and covariance respectively.



**Figure 4.6:** Results of EM algorithms for the Gaussian data. The dot and the circle represents the mean and covariance respectively.



6 centroids



5 centroids

**Figure 4.7:** Results of K-means algorithms with different centroids for the Gaussian data. The red dot represents the location of a centroid.





Figure 4.8: 'Shape' data

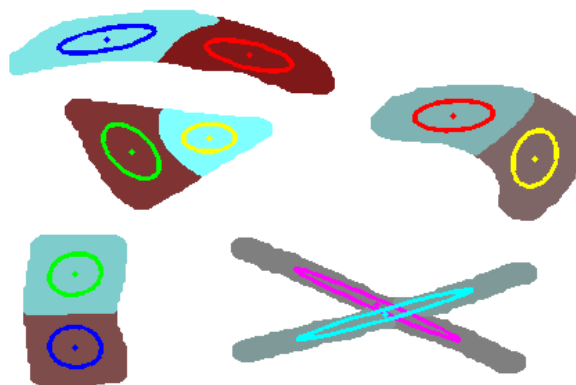
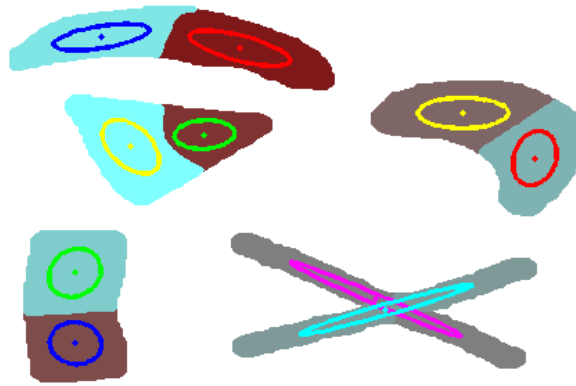
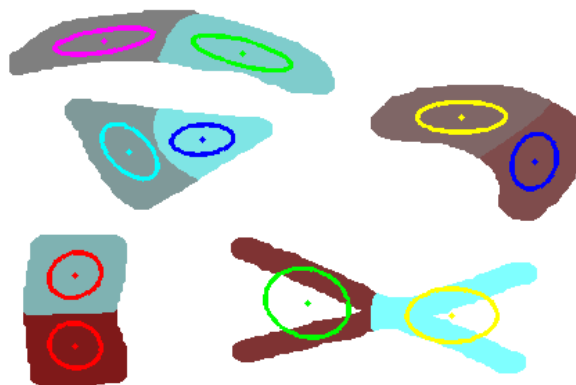


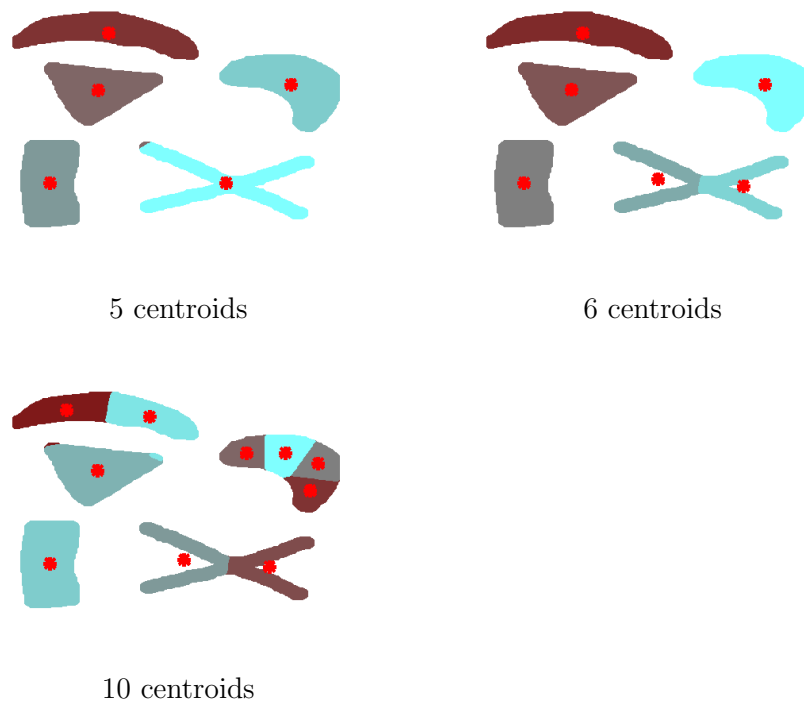
Figure 4.9: Results of the Gibbs sampling for the 'shape' data. The dot and the circle represents the mean and covariance respectively.



**Figure 4.10:** Results of the EM algorithm for the ‘shape’ data.



**Figure 4.11:** Results of the EM algorithm with random starting values. The dot and the circle represents the mean and covariance respectively.



**Figure 4.12:** Results of the K-means algorithm for the ‘shape’ data. The red dot represents the location of a centroid.

### 4.9.2 Modelling Results for Shape Data

This section presents results of an example of more complicated synthetic data. The 2D ‘shape’ data is no longer a set of Gaussians, which would require more components from the model to fit. The ‘cross’ is featured, together with a few other regular shapes. The total number of data samples is 66593.

The MGMM was configured to start as a six-component model, in which each shape had one component. The ‘cross’ is considered to have two components, but overlapped with each other. Figure 4.9 presents the final formation of the MGMM using Gibbs sampling, which had 2 components for each shape except the ‘cross’. The need for more components for each shape is due to the fact that the shapes are not Gaussians, although they can be approximated by a number of Gaussians, which is demonstrated in Figure 4.9.

Figure 4.10 presents the formation of the MGMM using the EM algorithm, which had the same initial configuration as Gibbs sampling. However, to demonstrate the EM algorithm’s sensitivity to starting values, a random configuration was applied and the result is shown in Figure 4.11. The random configuration was applied to the MGMM using Gibbs sampling, and the result did not vary significantly from the one shown in Figure 4.9.

The K-means algorithm with 6 centroids (Figure 4.7) could not locate the exact position of the ‘cross’. The 5-centroid K-means algorithm (see Figure 4.7) fitted each shape by one centroid by treating the ‘cross’, which comes from two distributions, as a single distribution. Table 4.4 presents the log-likelihood value for the two formations of the K-means and the MGMM.

**Table 4.4:** Comparisons of the log-likelihood and the running time for 2D ‘shape’ data. Gibbs sampling method, EM and K-means with 6 centroids had the same starting values. K-means with 5 centroids only used one centroid instead of two for the ‘cross’ shape. Gibbs sampling and EM with random starting values were also tested. K-means with 10 centroids also had random starting values.

	Number of Components	Log-likelihood Value	Time in Seconds
Gibbs sampling	10	−762844	46.327
EM	10	−759701	11.990
K-means	6	−785153	0.789
K-means	5	−782128	0.764
Gibbs (Random)	10	−760425	46.142
EM (Random)	10	−769999	15.468
K-means (Random)	10	−772640	0.961

**Table 4.5:** Comparisons of the log-likelihood and the running time for *Pansy* data. All four methods started with 3 components. K-means cannot change its model structure, and SOM increased model complexity at a fixed rate for each resolution. Gibbs sampling and EM automatically evolved to balance model complexity and data fit.

	Number of Components	Log-likelihood Value	Time in Seconds	Number of Iterations
Gibbs sampling	18	−1435433	104.600	300
EM	18	−1435698	67.823	30
K-means	3	−1624527	2.616	17
SOM	1024	−1017960	6.089*	10000

\* SOM was implemented in C++, while the other three were in MATLAB.

### 4.9.3 Modelling Results for Pansy Data

This section presents results for a pack of *Pansy* plants. The initial configuration of components for the MGMM are: one component for the yellow part of the *Pansy* flower, one component for the black part of the flower and one component for the leaves. The MGMM in the end had 18 components in total, five of which represent the flowers; two components represent the yellow flower part and three components represent black flower part. Components developed from the yellow and black part of the *Pansy* are shown as red. The other 13 components represent leaves, which are shown as green.

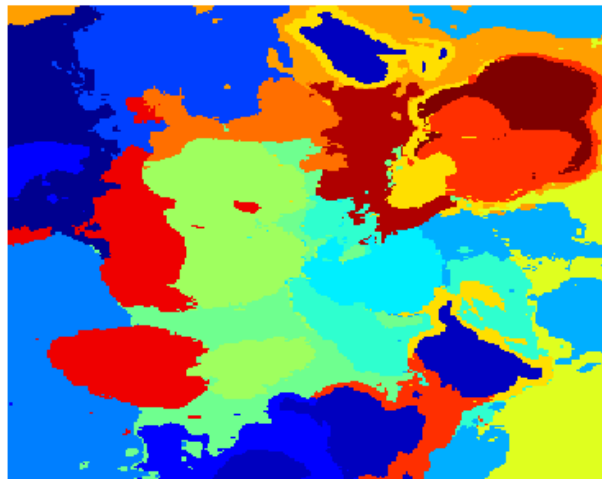
Figures 4.15 and 4.16 present the result using Gibbs sampling and the EM algorithm respectively. Figure 4.17 presents the results of the K-means algorithm. Three components were used, identical to the initial configuration for the MGMM, and the model complexity remained the same since the K-means algorithm cannot evolve itself. The result obtained by Gibbs sampling was the best of the three, and the K-means algorithm produced the most noisy estimates. Table 4.5 presents the log-likelihood value and the running time using the three methods and the SOM results from Chapter 3. Although the EM algorithm produced similar log-likelihood value as the sampling method, a large part of the flower component is misclassified into leaves.

## 4.10 Discussion

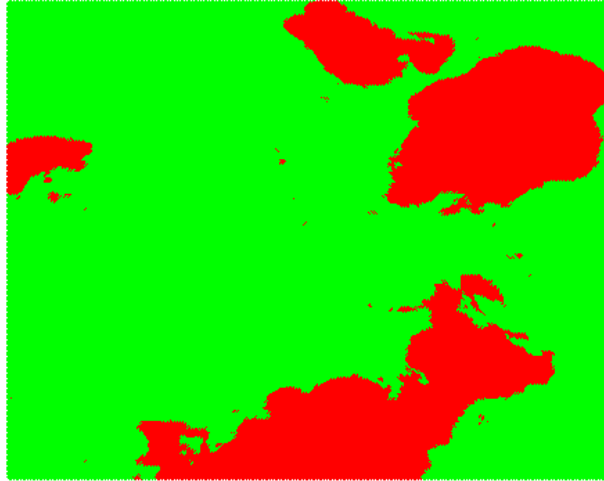
The MGMM can produce good results in the case of overlapped distributions. The result for the ‘cross’ in Figure 4.4 has demonstrated that the MGMM’s ability to locate two distributions with the same mean but different covari-



**Figure 4.13:** *Pansy* data



**Figure 4.14:** 18 Components in the MGMM using Gibbs sampling for the *Pansy* data.

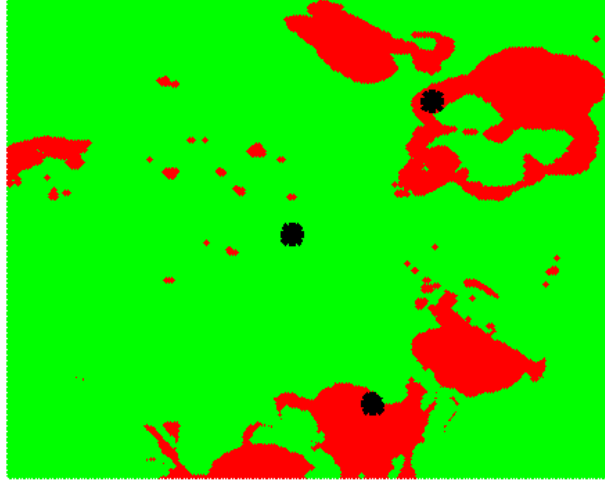


**Figure 4.15:** Classification of the *Pansy* data using Gibbs sampling. Red colour represents the flowers, and green colour represent the leaves.



**Figure 4.16:** Classification of the *Pansy* data using EM.





**Figure 4.17:** Classification of the *Pansy* data using K-means. The black dot represents the location of a centroid.

ances. The K-means method does not take account of the covariance, and it therefore could not identify all the Gaussians in the data. The result for the ‘shape’ data has demonstrated the MGMM can approximate arbitrary probability densities, not only for the Gaussian case. The goodness-of-fit test has also been shown to work, and it balances the data fit and the model complexity.

The *Pansy* result has demonstrated the potential that the MGMM is a tool to model the plant attributes for monitoring and grading purposes. Each component in the MGMM represents a region of interest for a plant. The mean  $\mu_j$  for each component contains not only the geometry measurement but also the colour measurement. The covariance  $\Sigma_j$  for each component describes how variable the measurements in the component are. The pop-

ulation,  $\pi_j$ , represents how large the region of interest is, and an example is the amount of *Pansy* flowers. The MGMM models the plant complexity automatically, given simple training or initialisation at the start.

It can be seen from Tables 4.3 - 4.5 that the Gibbs sampling method can give much better estimates of probability densities than the K-means. There is a 13% increase in terms of the log-likelihood value for the Gaussian data as well as the *Pansy* data, and a 3% increase for the ‘shape’ data. Although the increased number of components in the mixture model should result in a better data fit, this does not apply to the K-means. The K-means algorithm cannot guarantee the convergence to the optimal data fit, since this depends largely on the initial set of clusters [17]. The EM algorithm can produce similar results to the Gibbs sampling method given a good initial configuration, and the log-likelihood value did not vary significantly. However, Gibbs sampling is more robust than the EM algorithm, since the random configuration affected the EM algorithm more than the sampling method, which resulted in a 2.7% decrease in the log-likelihood value in Table 4.4. The SOM is expected to produce the best data fit, since it was treated as a 1024-component model, which is a lot larger than the others. In terms of the running time, the K-means is the most efficient approach. The time required for running Gibbs sampling was much longer than the EM algorithm.

The undesirable noise shown in the *Pansy* example is caused by incorrect height estimates. This occurs in regions where the change of height is significant, and occlusions are inevitable. Also in some of these regions, the colour data cannot contribute much to noise removal, since the shadow has the same

visual appearance as the black part of the flower for example. In this thesis, the uncertainty  $s_e$  associated with each point did not have a noise model built into the MGMM, unlike the learning rate in the SOM, and the height data are therefore assumed to be 100% correct. The results could be improved by taking the associated uncertainty into account by introducing a noise model for height data. Another notable difference is that the MGMM only takes account of the prior from the parent level, while the SOM is constructed from the neighbours and the parents.

As discussed above, the sampling method is more robust than the EM algorithm, however, the EM algorithm is much more efficient. The trade-off between reliability and efficiency therefore should be determined for a particular horticultural application. For example, the EM algorithm can be used to produce rough estimates for a crop in a relatively short time. When more precise estimation is required, the sampling method is a better choice.

## 4.11 Summary

The probabilistic model approach, MGMM, can solve many problems faced by heuristic procedures, such as the self-organising map and the K-means algorithm. It has been demonstrated in this chapter to approximate arbitrary probability densities, and evolve itself to a certain size to balance the data fit with the model complexity. Surface attributes, such as plant height and the amount of flowers, are readily obtainable from the components in the MGMM. Although much work in prior and likelihood construction is still required to tune the MGMM, for example, to incorporate a noise model for

height data, it has been shown the potential as a useful tool to model the plant attributes for monitoring and grading purposes in the horticulture industry. As demonstrated in this chapter, the sampling method is more reliable while the EM algorithm is more efficient. Different horticultural applications have their preference for reliability and efficiency, and the choice between the sampling method and the EM algorithm should therefore be determined on an individual basis.

# Chapter 5

## Monitoring Plant Growth Over Time

### 5.1 Introduction

In Chapters 3 and 4, approaches to modelling plant attributes at a particular time point have been discussed, but the relationship between plant attributes at different times was not discussed. In this chapter, a Kalman filter approach is presented for tracking a set of attributes over time, in order to characterise the plant growth.

There are two reasons for examining this relationship. First, tracking a model over time can help to investigate trends in the data from a time-series point of view, and the model can be used to explain the behaviour of the measurements, since the underlying plant growth process is incompletely understood. In particular, growers often want to know if the measurements on plants exhibit an increasing or decreasing trend, as well as when the plants

are ready for marketing.

Secondly, tracking the plant model over time can be used to predict future measurements from past ones. The prediction can be seen as prior knowledge input into a Bayesian framework to form an optimal estimate of plant growth at a particular point in time.

## 5.2 Plant Growth Monitoring

Plant growth monitoring is conventionally performed by individuals who have special knowledge of a particular plant. However, in large-scale commercial production, this conventional approach has a number of disadvantages as discussed in Chapter 1.

*Poinsettia* growers, for example, are required to produce plants that adhere to tight specifications, and one of the most rigorous specifications is plant height [80]. A *Poinsettia* tracker [51] was developed in 2001 for crop scheduling, where the actual plant state is plotted against an ideal target curve which encapsulates all crop and environmental characteristics. However, this tool is only used for the purposes of presentation and record. It does not solve the problems of human hand measurement outlined above, and the tool itself cannot produce measurements given a set of images, unlike the approach proposed in this thesis. It is worth pointing out that there are no other methods apart from the curve fitting approach used by the *Poinsettia* tracker and the Kalman filter approach proposed in this thesis to compare with.

A general statistical model for plant growth over time can be formulated

as a regression model, in which the variables are functions of time and the parameters are time-varying. The key to handling such a time-series model is the state-space formation, with each state representing the variables and parameters at a given time [37]. Since observations often contain a certain amount of noise, a smoothing operation is required to minimise the errors in the data. This problem can be tackled by following a moving average approach [16] or a Wiener filtering approach [118]. Both approaches require a windowing operation on the measurements, and then weighting the past values to determine the present value.

Another approach is recursive processing, in which the key element is the use of the results in the previous state to help estimate the result for the current state. The best-known recursive method is Kalman filtering [63], which is discussed in Section 5.4. The Kalman Update, briefly discussed in Chapter 2, forms part of a scalar Kalman filter, where the prediction process is replaced by the propagated estimates from the parent level. The discussion of the Kalman filter in this chapter considers a more general case, and both the process and the observation are modelled in vector form.

### 5.3 The State Space Formation

Assume that the development over time is determined by a set of states  $\mathbf{x}(t), t = \{1, 2, \dots, \infty\}$ , with which are associated a set of observations  $\mathbf{y}(t), t = \{1, 2, \dots, \infty\}$ . The aim of state-space analysis is to estimate the properties of the states, given knowledge of the observations [18]. A process model is used to describe the transition between two successive states, and

the relationship between the states and the observations is specified by an observation model.

### 5.3.1 The Process Model

The process model describes the state transition over time. A linear model is shown below:

$$\mathbf{x}(t) = \mathbf{A}(t)\mathbf{x}(t-1) + \mathbf{b}(t) + \mathbf{v}(t) \quad (5.1)$$

where  $\mathbf{x}(t)$  is a vector, which contains all values in the state at the time  $t$ .  $\mathbf{A}(t)$  is a matrix representing the state transition, while the vector  $\mathbf{b}(t)$  represents the trend. The vector  $\mathbf{v}(t)$  is the process noise, and is assumed to be drawn from a multivariate Gaussian distribution shown below,

$$\mathbf{v}(t) \sim N_v(0, \mathbf{Q}(t)) \quad (5.2)$$

where  $\mathbf{Q}(t)$  is the process noise covariance.

In general, the ‘true’ state,  $\mathbf{x}(t)$ , cannot be observed directly, although it is considered to be generated by a Markov process [24]. The output of a state  $\mathbf{x}(t)$  can be observed, as discussed below.

### 5.3.2 The Observation Model

The observation model, also known as the measurement model, describes the process of observing the output of the state. Similar to the process model used in this thesis, a linear observation model is used. If a vector  $\mathbf{y}(t)$  is the



noisy observation for a state  $\mathbf{x}(t)$ , the model equation is shown below:

$$\mathbf{y}(t) = \mathbf{H}(t)\mathbf{x}(t) + \mathbf{e}(t) \quad (5.3)$$

where the matrix  $\mathbf{H}(t)$  maps the state space into the observation space. The vector  $\mathbf{e}(t)$  is the measurement noise, which is drawn from a zero-mean white Gaussian process.

$$\mathbf{e}(t) \sim N_v(0, \mathbf{W}(t)) \quad (5.4)$$

where  $\mathbf{W}(t)$  is the measurement noise covariance.

$\mathbf{H}(t)$  in equation (5.3) accounts for certain variables in the state vector that cannot be observed. However, in this thesis, it is assumed that each state can be completely observed, and there is no scaling difference between each state and its observation. The observation model is therefore simplified into the following form,

$$\mathbf{y}(t) = \mathbf{x}(t) + \mathbf{e}(t) \quad (5.5)$$

Using the state-space formation, the Kalman filter discussed below is a recursive technique to solve the minimum mean-square error filtering problem.

## 5.4 Kalman Filter: State-Space Filtering

The Kalman filter has been the subject of extensive research since *Rudolf Kalman* published his paper in 1960 [63]. It has been used successfully in a wide range of applications, from financial forecasting [117] and economic applications [50], to GPS navigation [46], missile guidance [103] and helicopter

control [96]. In effect, the Kalman filter is a set of equations that enables recursive estimation of the state, given a process model, an observation model and a series of noisy measurements. The process model is assumed to be driven by additive Gaussian noise, while the measurements are also assumed to be corrupted by additive Gaussian noise with known covariance. The objective of the Kalman Filter is to update knowledge every time an observation becomes available. The filter operates in a ‘Predict-Update’ cycle, in which the state is predicted using the process model and the predicted state is then updated with a noisy measurement.

In the ‘Predict’ stage, the state estimate  $\hat{\mathbf{x}}(t)$  and its noise covariance  $\mathbf{P}(t)$  at the time  $t$  are predicted from the state at the time  $t - 1$ .

$$\hat{\mathbf{x}}(t) = \mathbf{A}(t)\hat{\mathbf{x}}^*(t-1) + \mathbf{v}(t) \quad (5.6)$$

$$\mathbf{P}(t) = \mathbf{A}(t)\mathbf{P}^*(t-1)\mathbf{A}^T(t) + \mathbf{Q}(t) \quad (5.7)$$

where  $\hat{\mathbf{x}}(t)$  and  $\mathbf{P}(t)$  are the predicted state and covariance respectively. Values with  $*$  denote the posteriors.

The ‘Update’ stage can be viewed as a linear blending of the noisy observation  $\mathbf{y}(t)$  and the predicted estimate  $\hat{\mathbf{x}}(t)$ . The equations to compute the posterior estimate  $\hat{\mathbf{x}}^*(t)$  and its noise covariance  $\mathbf{P}^*(t)$  are shown below:

$$\hat{\mathbf{x}}^*(t) = \hat{\mathbf{x}}(t) + \mathbf{G}(t) \left[ \mathbf{y}(t) - \hat{\mathbf{x}}(t) \right] \quad (5.8)$$

$$\mathbf{P}^*(t) = \left[ \mathbf{I} - \mathbf{G}(t) \right] \mathbf{P}(t) \quad (5.9)$$

where  $\mathbf{G}(t)$ , known as the Kalman gain, controls the input of the innovation,

**Table 5.1:** Summary of the Kalman filter

The ‘Predict’ Stage:

$$\begin{aligned}\hat{\mathbf{x}}(t) &= \mathbf{A}(t)\hat{\mathbf{x}}^*(t-1) + \mathbf{v}(t) \\ \mathbf{P}(t) &= \mathbf{A}(t)\mathbf{P}^*(t-1)\mathbf{A}^T(t) + \mathbf{Q}(t)\end{aligned}$$

The ‘Update’ Stage:

$$\begin{aligned}\mathbf{G}(t) &= \mathbf{P}(t) [\mathbf{P}(t) + \mathbf{W}(t)]^{-1} \\ \hat{\mathbf{x}}^*(t) &= \hat{\mathbf{x}}(t) + \mathbf{G}(t) [\mathbf{y}(t) - \hat{\mathbf{x}}(t)] \\ \mathbf{P}^*(t) &= [\mathbf{I} - \mathbf{G}(t)] \mathbf{P}(t)\end{aligned}$$

which is the difference between the observation  $\mathbf{y}(t)$  and the prediction  $\hat{\mathbf{x}}(t)$ .

Given the measurement noise covariance  $\mathbf{W}(t)$ ,  $\mathbf{G}(t)$  is calculated as follows:

$$\mathbf{G}(t) = \mathbf{P}(t) [\mathbf{P}(t) + \mathbf{W}(t)]^{-1} \quad (5.10)$$

If  $\mathbf{G}(t) \rightarrow \mathbf{I}$ , more input comes from the observation  $\mathbf{y}(t)$  to the state estimate, while the prediction  $\hat{\mathbf{x}}(t)$  contributes more to the estimate if  $\mathbf{G}(t)$  becomes a zero matrix. Like the ‘scale-space’ Kalman filter discussed in Chapter 2, the Kalman gain is a matrix, which can be used to take account of correlations between state variables.

## 5.5 Parameter Configuration

This section presents a simple example of synthetic data using the Kalman filter and demonstrates the effect of changing parameters of the filter. The core parameters that affect the filter performance are: the process noise

covariance  $\mathbf{Q}(t)$  in equation (5.7), the measurement noise covariance  $\mathbf{W}(t)$  in equation (5.10) and the initial state covariance  $\mathbf{P}^*(0)$ . It is common practice to assume the noise covariance matrix to be diagonal, due to the lack of sufficient information to evaluate its off-diagonal terms [116].

From equation (5.7), the state noise covariance  $\mathbf{P}(t)$  is large if  $\mathbf{Q}(t)$  is large, which indicates an inadequate process model. For large  $\mathbf{P}(t)$ , the Kalman gain  $\mathbf{G}(t) \rightarrow \mathbf{I}$  and the inadequate process model therefore has less input to the state estimate. Due to this connection between  $\mathbf{P}(t)$  and  $\mathbf{Q}(t)$ , the Kalman gain can be expressed as a ratio of process to measurement noise, which is shown below,

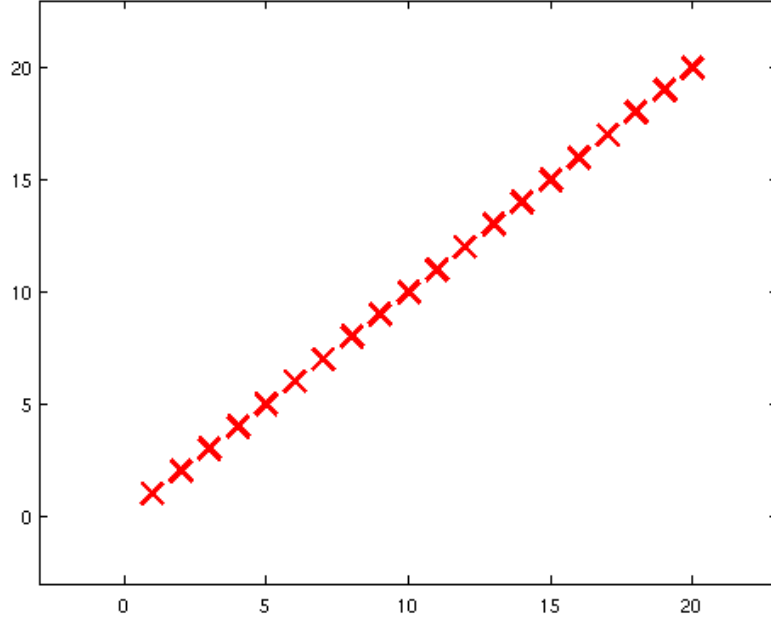
$$\mathbf{G}(t) \propto \mathbf{Q}(t)\mathbf{W}^{-1}(t) \quad (5.11)$$

To demonstrate the effect of these noise covariances on the performance of the Kalman filter, consider a simple example of synthetic data. Although some authors [24, 115] have presented 1D scalar examples, this thesis presents an example in vector form on 2D  $x-y$  position data. The series has 20 samples in total, and the state equation for this series is shown below,

$$\mathbf{x}(t) = \begin{cases} \begin{bmatrix} 1 & 1 \end{bmatrix}^T, & \text{if } t = 1 \\ \mathbf{x}(t-1) + \begin{bmatrix} 1 & 1 \end{bmatrix}^T, & \text{otherwise} \end{cases} \quad (5.12)$$

Figure 5.1 presents the values of the ‘true’ states  $\mathbf{x}(t)$  to be estimated in the  $x-y$  domain.

The ‘true’ value of each state allows computation of the Root Mean Square



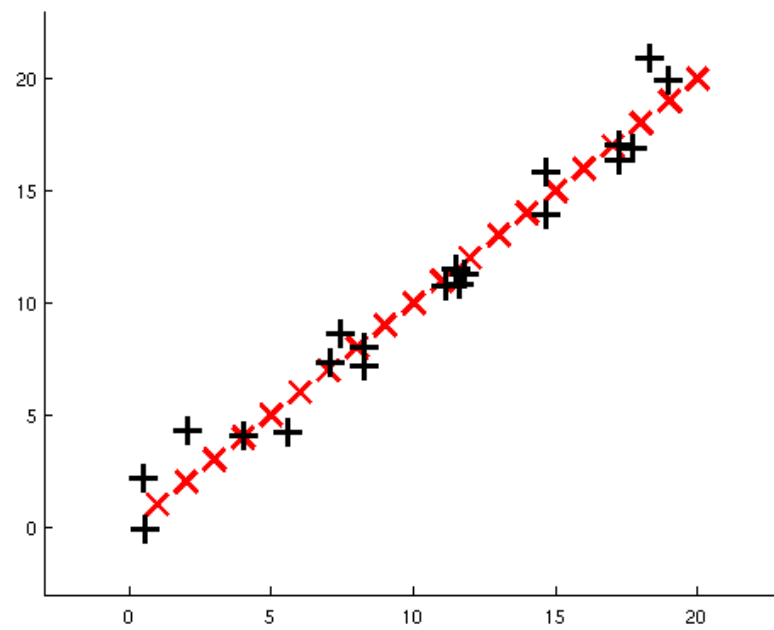
**Figure 5.1:** 2D  $x - y$  position data

(RMS) difference, which indicates the quality of the estimates.

$$RMS = \sum_{i=1}^D \sqrt{\frac{1}{T} \sum_{t=1}^T \left[ \hat{\mathbf{x}}(t) - \mathbf{x}(t) \right]^2} \quad (5.13)$$

In practice, this kind of ground truth data is not available, nor is the process model; nevertheless, it is useful for the toy example here.

Figure 5.2 presents the noisy observations  $\mathbf{y}(t)$ , which is obtained according to the equation (5.5). The process noise  $\mathbf{e}(t)$  is synthetically generated from a Gaussian distribution with mean  $\mu = 1$  and standard deviation  $\sigma = 1$ .



**Figure 5.2:** Noisy observations of the 2D position data. Black cross: the noisy observations, Red letter X: the ‘true’ states.

### 5.5.1 The Process Noise Covariance

The process noise determines the characteristics of the uncertainty in the process model. In this example, three values are selected for the process noise covariance  $\mathbf{Q}(t)$ .

$$\mathbf{Q}_1(t) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad (5.14)$$

$$\mathbf{Q}_2(t) = \begin{pmatrix} 0.01 & 0 \\ 0 & 0.01 \end{pmatrix} \quad (5.15)$$

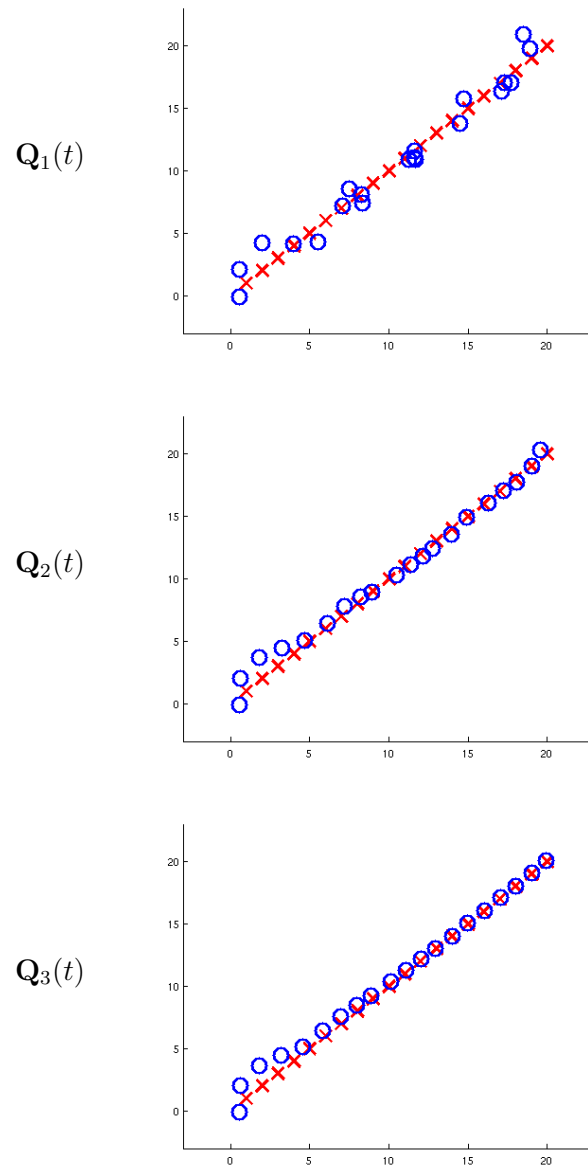
$$\mathbf{Q}_3(t) = \begin{pmatrix} 0.000001 & 0 \\ 0 & 0.000001 \end{pmatrix} \quad (5.16)$$

The other two parameters are set as follows,

$$\mathbf{W}(t) = \begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix} \quad (5.17)$$

$$\mathbf{P}^*(0) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad (5.18)$$

Figure 5.3 presents the results. Table 5.2 presents the RMS values with different  $\mathbf{Q}(t)$ . For large  $\mathbf{Q}(t)$ , there is more input from the observations, and the state estimates are therefore more noisy. The estimates become smoother as the process noise decreases, which is a result of more input from the predicted estimates using the known process model.



**Figure 5.3:** Effect of varying the process noise covariance. Blue circle: the Kalman estimates, Red letter X: the ‘true’ states.



### 5.5.2 The Measurement Noise Covariance

The measurement noise quantifies the uncertainty in the observations. Three values are used for  $\mathbf{W}(t)$ .

$$\mathbf{W}_1(t) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad (5.19)$$

$$\mathbf{W}_2(t) = \begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix} \quad (5.20)$$

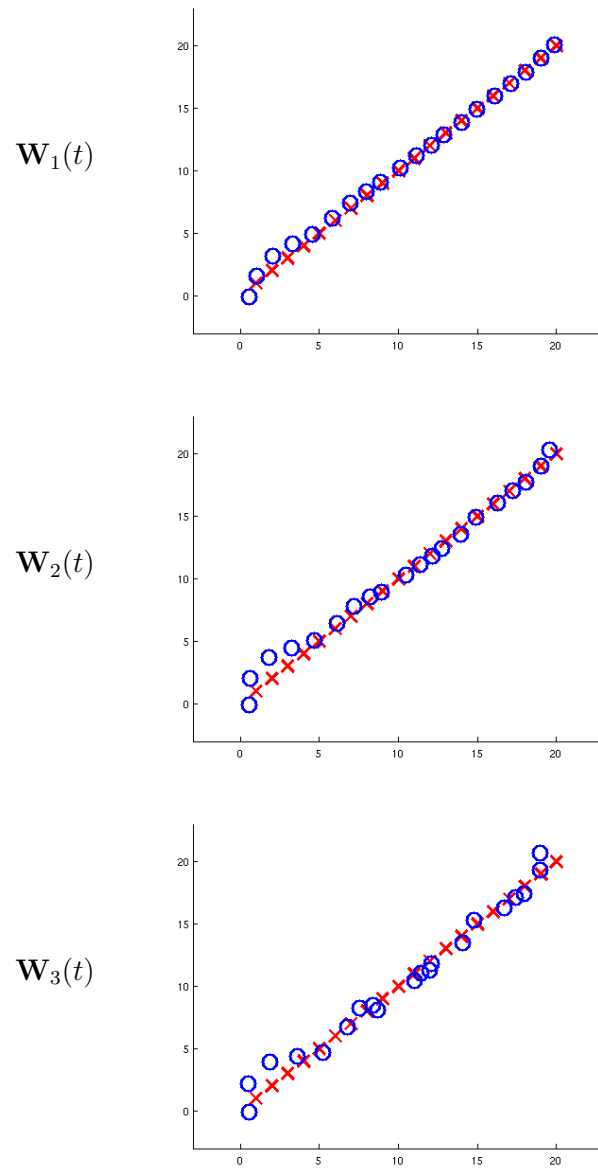
$$\mathbf{W}_3(t) = \begin{pmatrix} 0.01 & 0 \\ 0 & 0.01 \end{pmatrix} \quad (5.21)$$

The other two parameters are set as follows,

$$\mathbf{Q}(t) = \begin{pmatrix} 0.01 & 0 \\ 0 & 0.01 \end{pmatrix} \quad (5.22)$$

$$\mathbf{P}^*(0) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad (5.23)$$

Figure 5.4 shows the results, and Table 5.2 presents the RMS values. The effect of changing the measurement noise covariance  $\mathbf{W}(t)$  is opposite to the one that varies the process noise covariance  $\mathbf{Q}(t)$ . For small  $\mathbf{W}(t)$ , the effect is similar to the one with large  $\mathbf{Q}(t)$ , in which there are more noisy estimates, although the variation in  $\mathbf{Q}(t)$  affects less than  $\mathbf{W}(t)$  does.



**Figure 5.4:** Effect of varying the measurement noise covariance. Blue circle: the Kalman estimates, Red letter X: the 'true' states.

### 5.5.3 The Initial State Noise Covariance

The state noise represents the uncertainty in the state estimate, and it can be used to derive a scalar value indicating the overall performance of the filter [24]. This performance criterion is shown below,

$$J = \frac{1}{T} \sum_{t=1}^T tr(\mathbf{P}^*(t)) \quad (5.24)$$

where  $tr$  represents the trace of the covariance matrix  $\mathbf{P}^*(t)$ , and  $T$  is the total number of states. Lower values indicate better filter performance, and the state estimate therefore always has a greater value than the ‘true’ state.

$$J_{true} < J_{estimate} \quad (5.25)$$

Jazwinski [59] has shown that the effect of initial state noise can be neglected as more data are processed. The state noise therefore approaches to zero given enough data, which implies that the estimate converges to the ‘true’ value. To demonstrate this, let

$$\mathbf{P}_1^*(0) = \begin{pmatrix} 10 & 0 \\ 0 & 10 \end{pmatrix} \quad (5.26)$$

$$\mathbf{P}_2^*(0) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad (5.27)$$

$$\mathbf{P}_3^*(0) = \begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix} \quad (5.28)$$

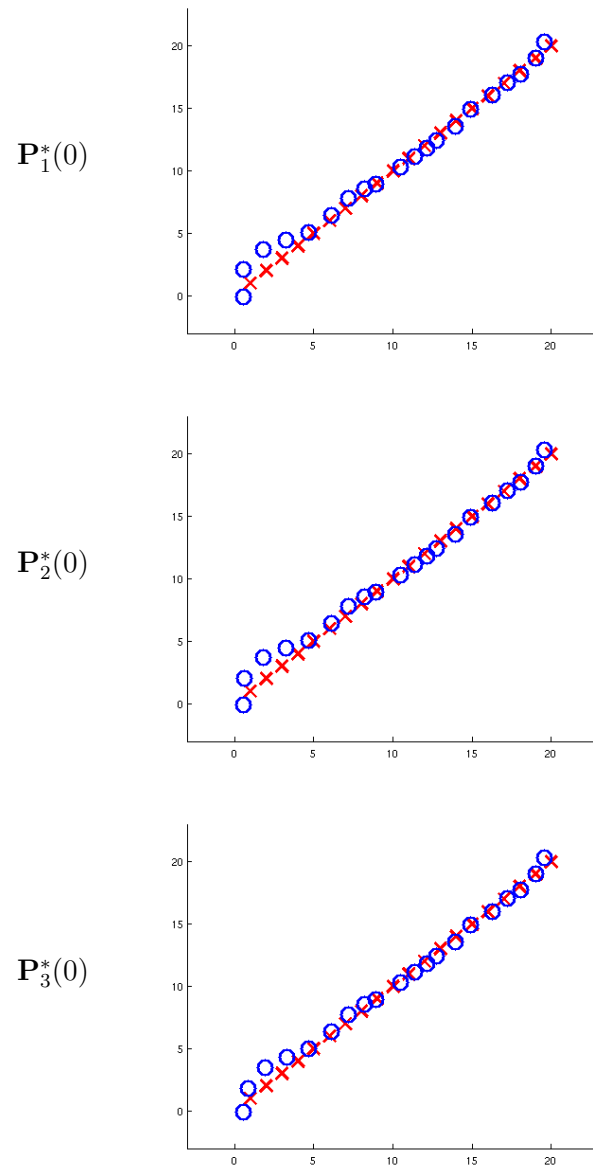
and

$$\mathbf{Q}(t) = \begin{pmatrix} 0.01 & 0 \\ 0 & 0.01 \end{pmatrix} \quad (5.29)$$

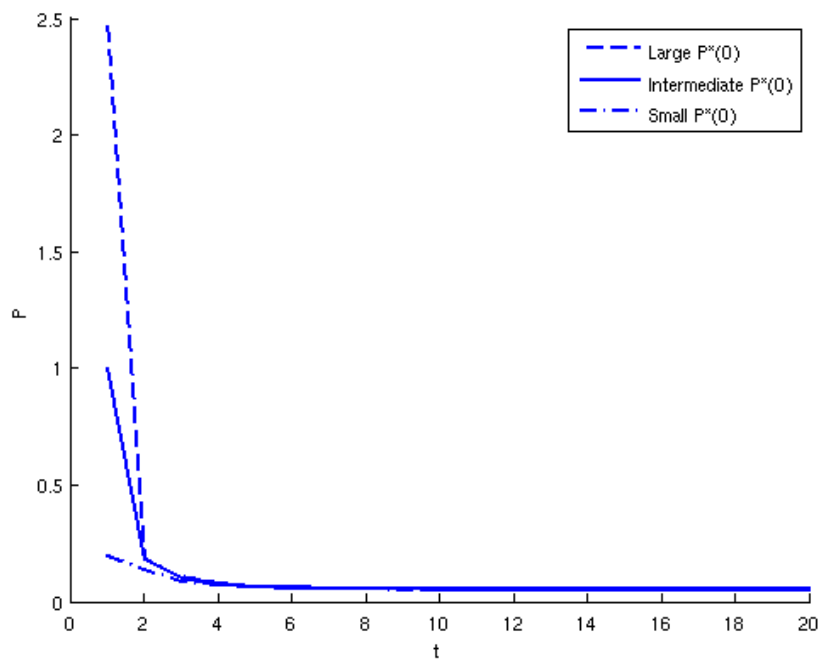
$$\mathbf{W}(t) = \begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix} \quad (5.30)$$

The effect of different values for  $\mathbf{P}^*(0)$  is shown in Figure 5.5 and Table 5.2. The initial state noise makes a minor difference in the early part of the processing, where large  $\mathbf{P}^*(0)$  leads to more input from the observations and less from the predicted estimates. However, as shown in Figure 5.5, after processing three data samples, the effect of different values in  $\mathbf{P}^*(0)$  becomes negligible, and the state noise covariance  $\mathbf{P}^*(t)$  converges to the same steady-state value as expected. This is important, since the initial statistics of the state noise are rarely known in practice.

Both the process model and the ‘true’ states are known in this example, but they could be difficult to determine in practice. In the next section, the Kalman filter for tracking plant attributes is developed.



**Figure 5.5:** Effect of varying the initial state noise covariance. Blue circle: the Kalman estimates, Red letter X: the ‘true’ states.



**Figure 5.6:** Effect on the state noise by using different initial state noise covariances. Dashed line:  $\mathbf{P}_1^*(0)$ , large  $\mathbf{P}^*(0)$ . Solid line:  $\mathbf{P}_2^*(0)$ , intermediate  $\mathbf{P}^*(0)$ . Dash-dot line:  $\mathbf{P}_3^*(0)$ , small  $\mathbf{P}^*(0)$ .

**Table 5.2:** RMS performance comparisons of the Kalman filters with different noise covariances

Conditions	Variables	RMS
fixed $\mathbf{W}(t)$ and $\mathbf{P}^*(0)$	$\mathbf{Q}_1(t)$	1.7585
fixed $\mathbf{W}(t)$ and $\mathbf{P}^*(0)$	$\mathbf{Q}_2(t)$	0.9352
fixed $\mathbf{W}(t)$ and $\mathbf{P}^*(0)$	$\mathbf{Q}_3(t)$	0.8596
fixed $\mathbf{Q}(t)$ and $\mathbf{P}^*(0)$	$\mathbf{W}_1(t)$	0.6874
fixed $\mathbf{Q}(t)$ and $\mathbf{P}^*(0)$	$\mathbf{W}_2(t)$	0.9352
fixed $\mathbf{Q}(t)$ and $\mathbf{P}^*(0)$	$\mathbf{W}_3(t)$	1.3353
fixed $\mathbf{W}(t)$ and $\mathbf{Q}(t)$	$\mathbf{P}_1^*(0)$	0.9539
fixed $\mathbf{W}(t)$ and $\mathbf{Q}(t)$	$\mathbf{P}_2^*(0)$	0.9352
fixed $\mathbf{W}(t)$ and $\mathbf{Q}(t)$	$\mathbf{P}_3^*(0)$	0.8552

## 5.6 Kalman Filter for Plant Attributes

This section presents the results of using the Kalman filter to track a set of plant attributes over time. This set of image data captures a number of growing *Pansy* plants twenty-one times over two months, three times a week, starting from the early potting stage to the marketing stage. The hand measurements on the leaves' and flowers' heights were carried out twice a week, and cover two-thirds of the captures. The image data were collected using the moving camera rig described in Chapter 2, and all the visual characteristics of the plants were captured. The attributes are extracted using the MGMM approach discussed in Chapter 4 for three main classes, which are 'flowers', 'leaves' and 'others' (soil and box edge, for example). Each class includes a set of attributes: height, population (the amount of flowers or leaves, in other words) and colour. In the early growing stage, *Pansy* has only leaves, with flowers appearing near the marketing stage.

Figure 5.7 presents the relationship between the number of days since the

early potting stage and the time  $t$  used in this thesis. To apply the Kalman filter, all the attributes for each main class at time  $t = \{1, \dots, 21\}$  are formed as a  $5 \times 1$  vector of measurements, which contains height, population and colour (R,G,B). However, for simplicity of presentation, each attribute is discussed separately below. All the parameters of the Kalman filter can be incorporated into a profile for a particular plant, which can be re-used. In this thesis, the parameters were derived from the result of one *Pansy* plant (also known as Plant 1), and they were then used for processing three other plants (also known as Plant 2, Plant 3 and Plant 4). The process, measurement, and initial state noise were determined through a series of experiments similar to those in Section 5.5.

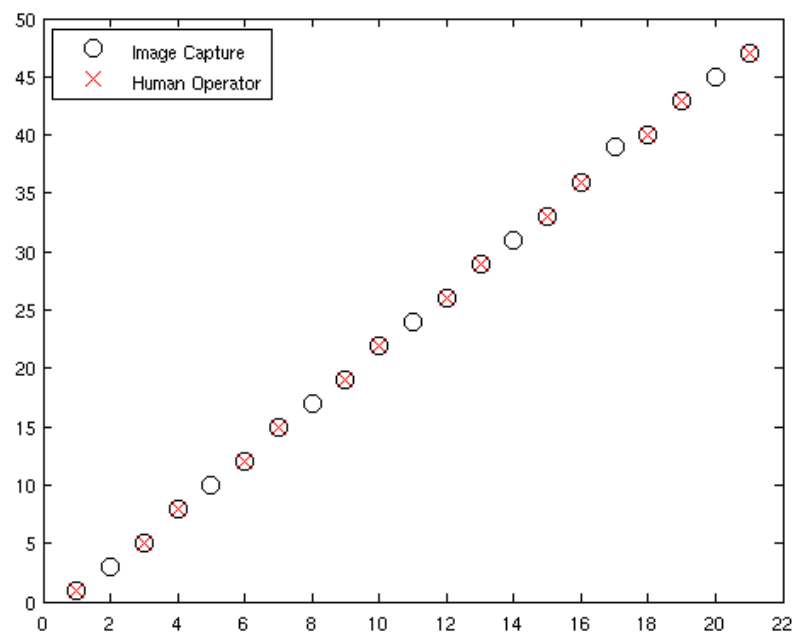
### 5.6.1 Height

The leaf height was computed by averaging all the mean heights in the components of ‘leaves’ in the Gaussian mixture model, which is analogous to what an expert does to measure the height by hand. The characteristics of plant height variations over time can be modelled using a linear Kalman process model. Figure 5.8 presents a set of hand measurements on the leaf height of a *Pansy* plant (Plant 1), and a linear Kalman process model to fit the measurements. The equation of the Kalman process model is shown below:

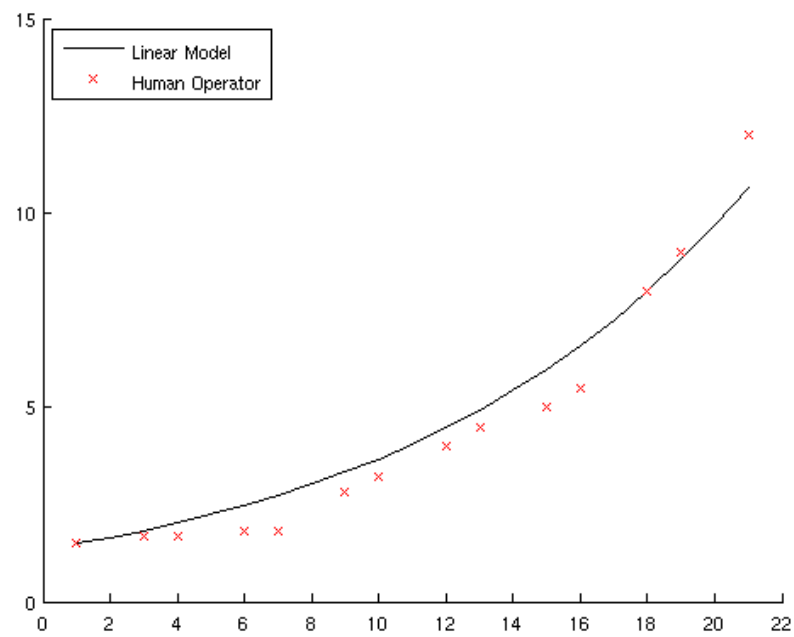
$$x(t) = 1.1x(t-1) + 0.01 \quad t = 1, 2, \dots, 21 \quad (5.31)$$

It can be seen in the Figure 5.8 that this process model sufficiently describes the evolution of the plant height.

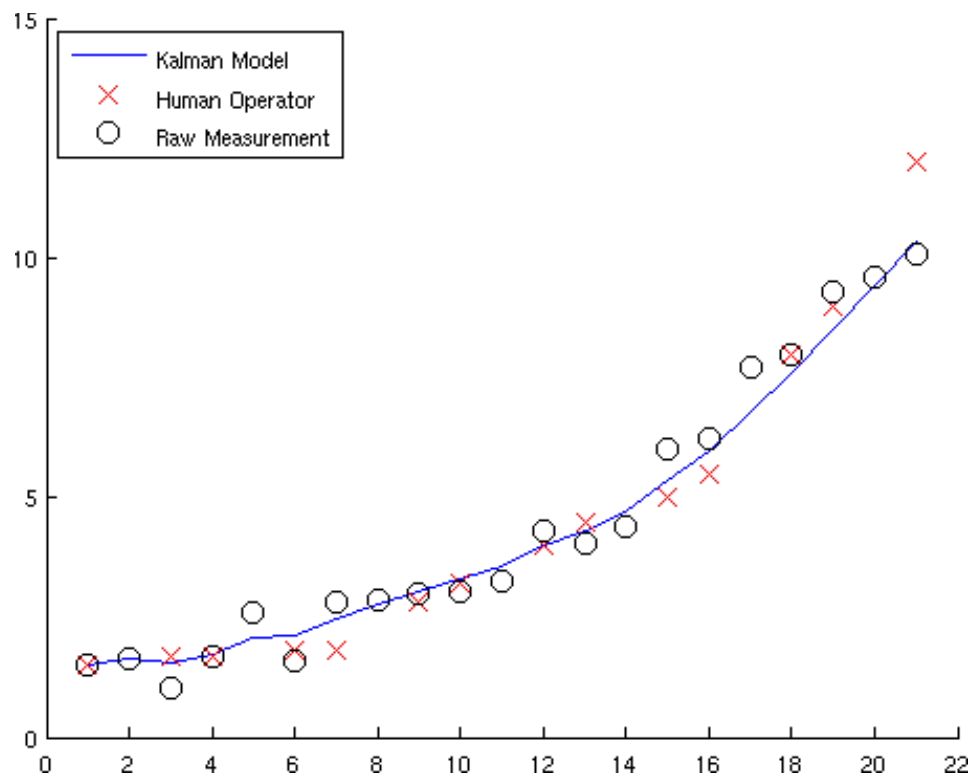




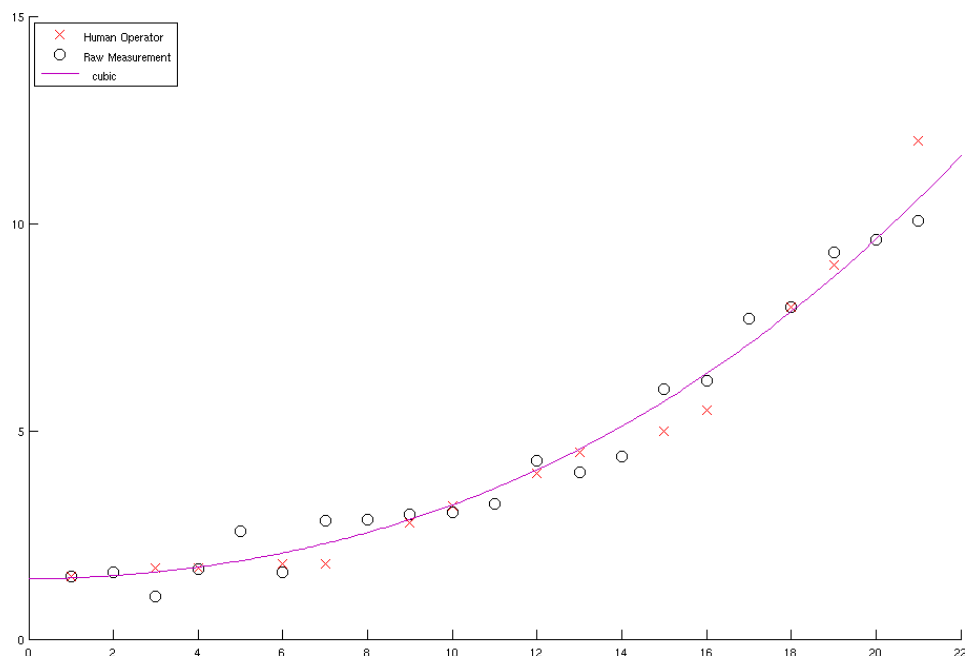
**Figure 5.7:** A plot of days since the early potting stage against the time  $t$ . Black circle: the image measurements. Red letter X: the hand measurements.



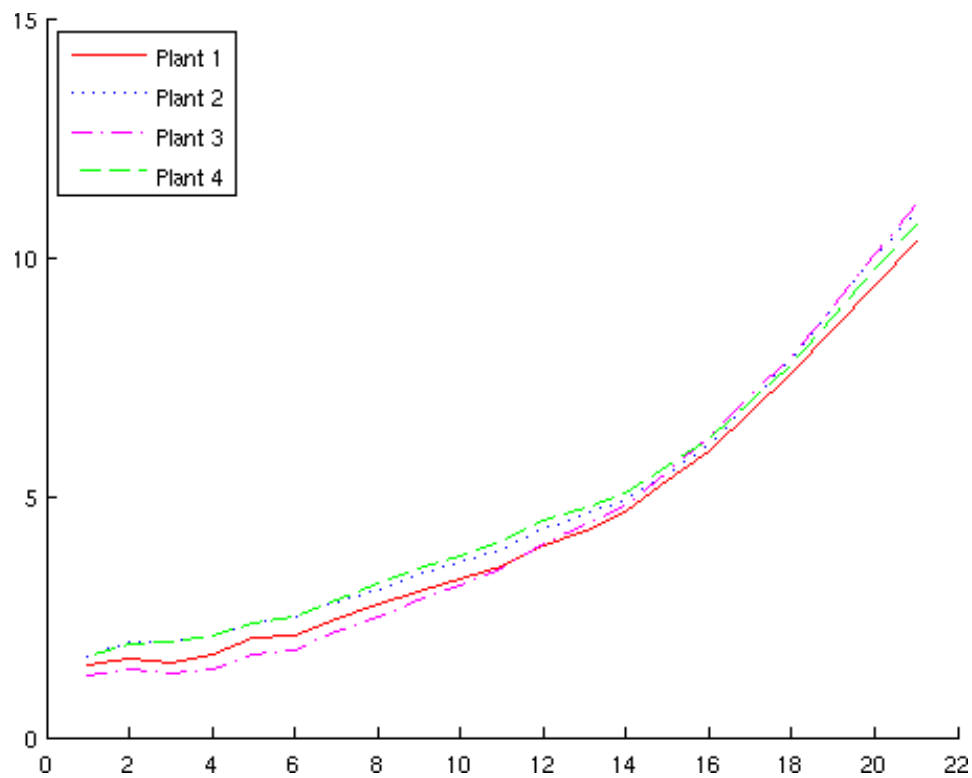
**Figure 5.8:** A linear Kalman process model to represent the leaf height over time. Black curve: the linear model, Red letter X: the hand measurements.



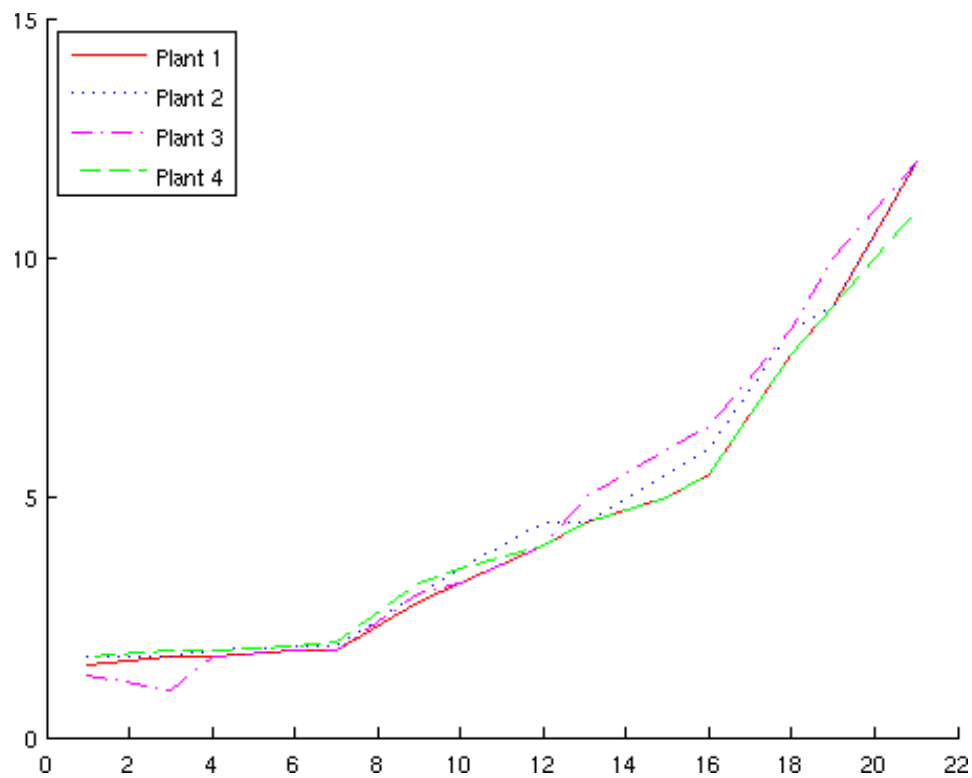
**Figure 5.9:** Tracking the leaf height over time for Plant 1 by Kalman filter. Blue curve: the tracked height. Black circle: the noisy observations. Red letter X: the hand measurements.



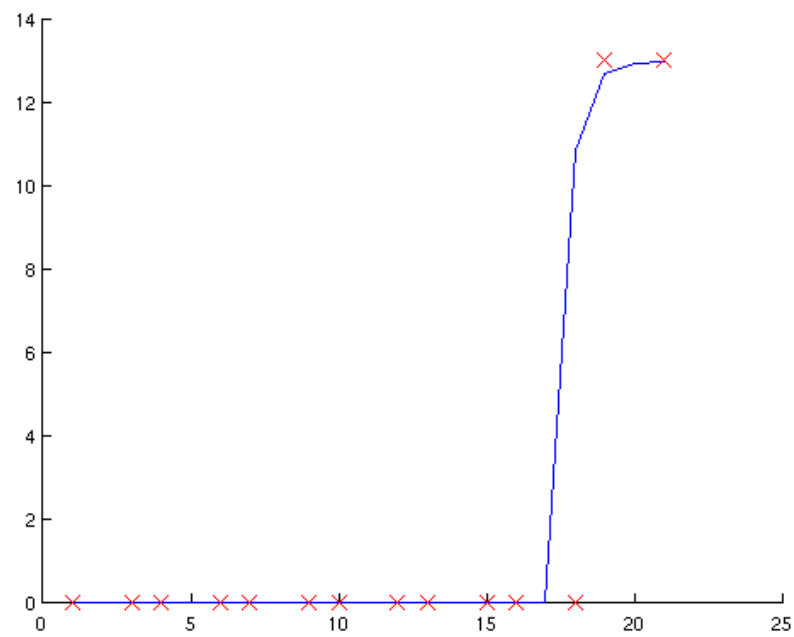
**Figure 5.10:** Tracking the leaf height over time for Plant 1 by curve fitting. Purple curve: the tracked height. Black circle: the noisy observations. Red letter X: the hand measurements.



**Figure 5.11:** Tracking the leaf height over time for Plant 1, Plant 2, Plant 3, Plant 4 by Kalman filter. Red solid line: Plant 1, Blue dotted line: Plant 2, Magenta dash-dot line: Plant 3, Green dashed line: Plant 4.



**Figure 5.12:** Hand measurements on the leaf height for Plant 1, Plant 2, Plant 3, Plant 4. Red solid line: Plant 1, Blue dotted line: Plant 2, Magenta dash-dot line: Plant 3, Green dashed line: Plant 4.



**Figure 5.13:** Tracking the flower height by Kalman filter. Blue curve: the tracked height, Red letter X: the hand measurements.

Figure 5.9 presents how the Kalman filter tracks the leaf height extracted with the model shown in (5.31). A cubic curve fitting procedure was used to fit the data in a similar manner as a *Poinsettia* tracker [51] does, and Figure 5.10 presents the results. Table 5.5 shows the RMS values for the hand measurements and the MGMM measurements.

The same Kalman filter procedures were applied to three other *Pansy* plants (Plant 2, Plant 3 and Plant 4), and Figure 5.11 presents the results. The corresponding hand measurements are shown in Figure 5.12.

The number of measurements on the flower height is four, which is too small to apply the Kalman filter. The process model is therefore considered as a bad model with a larger noise, which is 50 times more than the leaf height. Figure 5.13 presents the results.

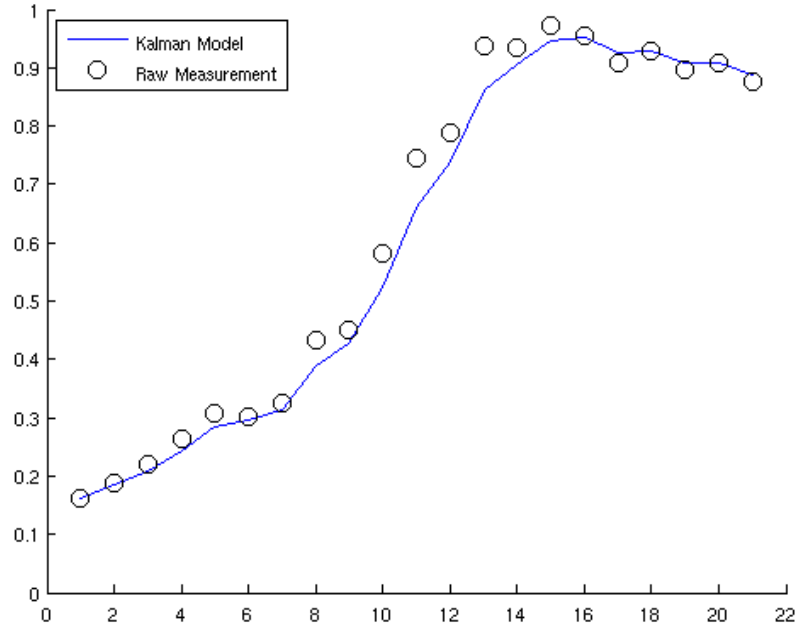
### 5.6.2 Population

The leaf population is calculated by summing the mixture weights of the components for ‘leaves’. The model used here assumes that there is minor change of population between the time  $t$  and  $t - 1$ . This allows population to either increase or decrease, due to the fact that the population is a percentage, not an absolute measurement, unlike the height. The equation of the linear process model is shown below:

$$x(t) = x(t - 1) \quad t = 1, 2, \dots, 21 \quad (5.32)$$

Figure 5.14 presents tracking the leaf population with the model shown in (5.32), while Figure 5.15 presents tracking the population of ‘other’ objects



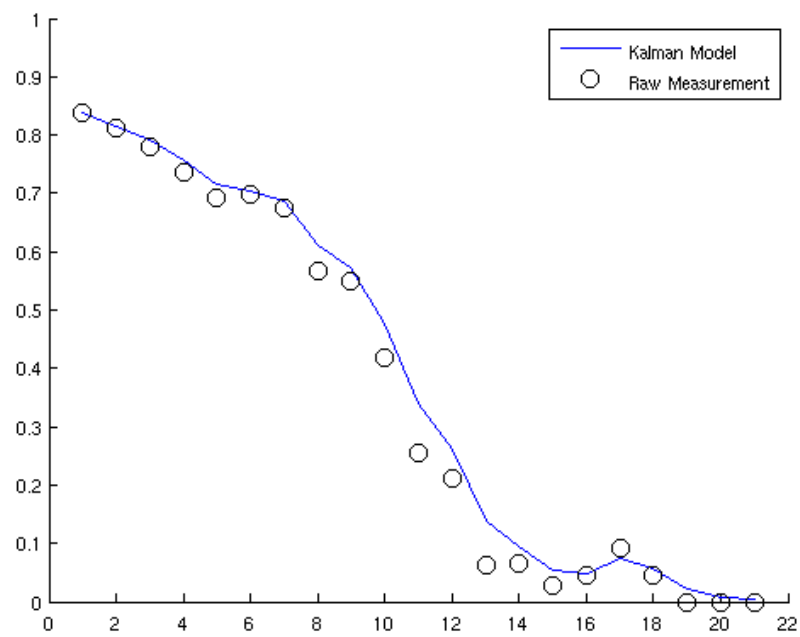


**Figure 5.14:** Tracking the leaf population by Kalman filter. Blue curve: the tracked population, Black circle: the noisy observations.

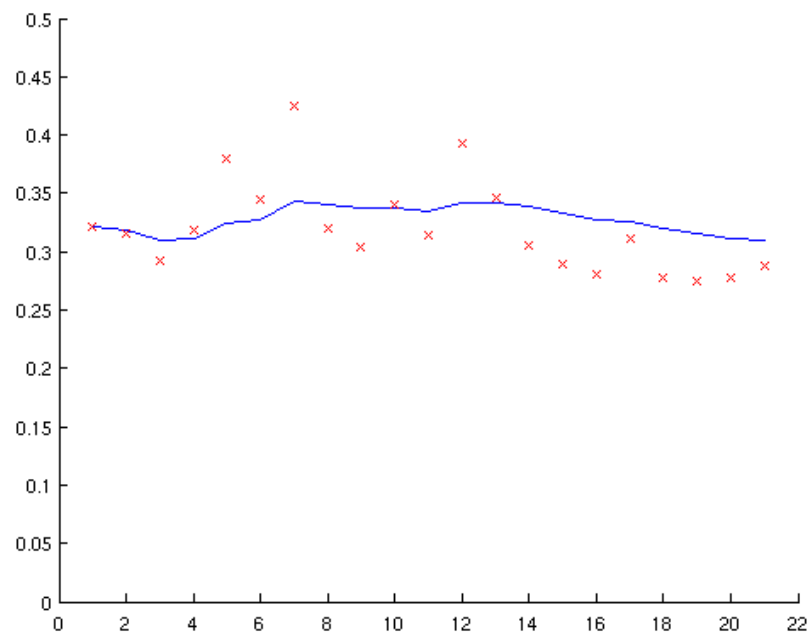
(for example, soil and box) with the same model.

### 5.6.3 Colour

In this thesis, each input image is contrast-enhanced by applying a Gamma correction procedure [45]. As a result, the colour data are different from the original observation. To address this problem, the RGB (Red, Green, Blue) colour vectors were converted into the HSV (Hue, Saturation, Value) colour space [45]. The linear model used here is the identical to the one shown in Equation (5.32). The Hue value refers to a pure colour without lighting



**Figure 5.15:** Tracking the population of other objects by Kalman filter. Blue curve: the tracked population, Black circle: the noisy observations.



**Figure 5.16:** Tracking the leaf colour by Kalman filter. Blue curve: the tracked colour, Red letter X: the noisy observations.

or shading, and the change of the Hue therefore represents the variation of the pure colour. Figure 5.16 presents the change of the Hue for ‘leaves’ over time. This result is used to show the potential use of the colour attribute for applications in plant growth monitoring, which is discussed in the next section.

## 5.7 Discussion

From the results presented in the previous section, the Kalman filter successfully smooths noisy measurements and it can produce a development trend that correlates well to the actual growth of the plants. The difference between the smoothed results and the raw measurements can be used as a tool to indicate abnormalities of the plants in the image. For example, the raw height measurements decrease at the time  $t = 3$  and  $t = 6$  in Figure 5.9. The Kalman filter smooths these two unexpected measurements and produces a trend reflecting the fact that plants grow continuously with time. However, at these two points in time, the image data were collected shortly after watering and applying pesticides respectively (see original snapshots in Figure 5.17). These operations are known to cause a short-term stress effect, that changes the plant’s growth habit and make plants appear ‘shorter’. Another cause of the noise in the measurement comes from the way of calculating the average. In this work, a simple averaging operation was applied, which works well if all the components have similar statistical population. However, a piece of leaf that is different from the others could be represented by one component, while a few leaves with similar characteristics might be represented by an-

**Table 5.3:** Height comparisons between the hand measurements and Kalman estimates for the leaves heights at the time  $t = 3$  and  $t = 6$ 

Time	Human at $t = 3$ (cm)	Kalman Estimates at $t = 3$ (cm)	Human at $t = 6$ (cm)	Kalman Estimates at $t = 6$ (cm)
Plant 1	1.7	1.57	1.8	2.47
Plant 2	1.8	1.99	1.9	2.50
Plant 3	1.7	1.32	1.8	1.83
Plant 4	1.8	1.97	1.9	2.52
Avg.	1.75	1.71	1.85	2.33
Std. Dev.	0.0577	0.3254	0.0577	0.3340

other component. The average between the two components is therefore not the average of all leaves. A better approach is the weighted-average method by taking account of the statistical population.

The inconsistency of the human operator contributes to a difference between the hand measurements and the estimates. According to the hand measurements on the leaf height, which are shown in Figure 5.9, the height of the plant did not change much between the time  $t = 3$  and  $t = 6$ . However, the plants certainly grew substantially during these days (see Figure 5.17). Therefore, the tracked estimates appear to represent growth better. Table 5.3 presents the height measurements by hand and the Kalman estimates at the time  $t = 3$  and  $t = 6$ . The human hand measurements should therefore not be considered as the ‘ground truth’ data in Section 5.5, although they are the only available reference in this thesis. The limitations of hand measurement have been highlighted in Chapter 1.

Table 5.4 presents the results of the population measurements around the time  $t = 17$ , while Figure 5.18 presents snapshots. The increase of ‘others’

**Table 5.4:** Comparisons of the population measurements at the time  $t = 16$ ,  $t = 17$  and  $t = 18$ 

Population	$t = 16$	$t = 17$	$t = 18$
Flowers	0%	0%	1.65%
Leaves	95.25%	92.56%	92.72%
Others	4.75%	7.44%	5.63%

**Table 5.5:** RMS performance comparisons between the fitted curve and Kalman estimates for the leaf height.

	Hand Measurements	MGMM Measurements
Curve Fitting	0.5761	0.5402
Kalman Estimates	<b>0.5474</b>	<b>0.5155</b>

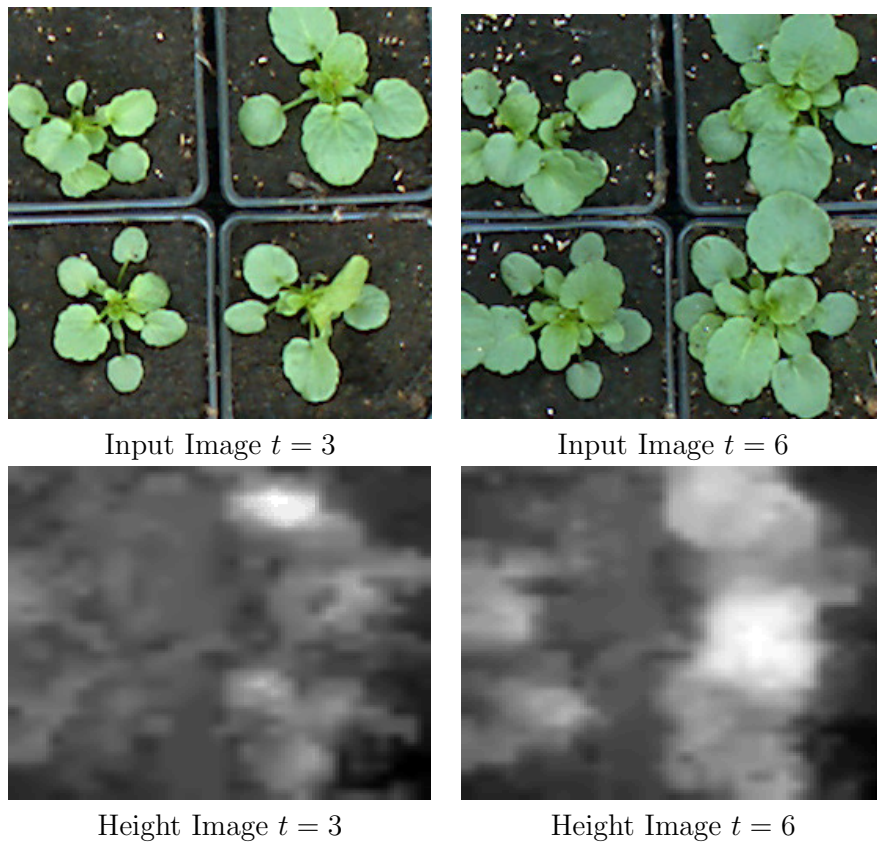
at  $t = 17$  was mainly caused by assuming there is no flower in the image until time  $t = 18$ . The well-developed flowers visible at the time  $t = 17$  are therefore considered as ‘others’.

In this thesis, the noise covariance matrix includes only diagonal terms, which is a common practice, and this leads to the use of a diagonal matrix solely describing the variances of each attributes. However, off-diagonal terms should also be considered, since there may exist some known correlation between the attributes [102]. For example, the increase in height could be linked to the increase of population.

It can be shown in Table 5.5 that the Kalman filter produces estimates that correlate better to hand measurements and MGMM measurements than a curve fitting procedure does. Furthermore, the results using the curve fitting procedure appear to be over-smoothed, and could be inappropriate to represent plant growth in practice.

There were only three measurements in a week, and the total number of states here is 21. This would be likely to increase in practice, since automated monitoring could be performed at least on a daily basis. In that case, the Kalman filter would work better since more data would be processed [24]. Given more data, parameters are tuned after each run, and the filter's reliance on the guessed parameters, such as initial state noise, becomes negligible.

The variations within a crop, for example, which part of the crop grows faster, is another factor of interest in plant monitoring. As discussed above, the lack of fit, in other words, the difference between smooth estimates and raw measurements, can indicate abnormalities. The variances of the fitted data can be obtained from the covariances within each component in the MGMM. Together with the lack-of-fit method, the variations within a crop could be captured. The analysis of colour in this thesis is performed by using the Hue value in the HSV space to quantify 'pure colour' variation over time. This could be improved by using a statistical colour model, which has a probability distribution for modelling colour features such as the histogram counts [61].



**Figure 5.17:** Original images collected at the time  $t = 3$  and  $t = 6$  and corresponding height images. In each image, top row, Plant 1 and Plant 2; bottom row, Plant 3 and Plant 4.



 $t = 16$  $t = 17$  $t = 18$ 

**Figure 5.18:** Input images at the time  $t = 16, 17, 18$ .

## 5.8 Summary

This chapter has presented a Kalman filter approach for tracking a set of attributes over time for monitoring plant growth. The key to tracking attributes over time is the state-space formation. Once in the state space, the Kalman filter predicts and updates the state as more observations are available. The effect of various parameters on the filter's performance has also been shown with an example on synthetic data. It has been demonstrated that the Kalman filter successfully smooths the noisy measurements and can produce a development trend for a crop.

# Chapter 6

## Conclusions

This thesis proposes various approaches for height measurement, plant data modelling, and tracking to achieve automated crop growth monitoring and plant quality assessment in a commercial environment. This final chapter presents a summary of the thesis, and concludes the work. The drawbacks of the proposed approaches are also discussed, together with some suggestions for further research.

### 6.1 Summary

#### 6.1.1 Plant Height Measurement

The height of a growing plant is an important attribute for quantifying plant growth. A stereopsis approach is used to obtain the implied geometry in a stereo pair of images, since the  $x$ - $y$  position data are readily available from each image except the height  $z$ . There are commercial systems already available on the market to measure plant height from side views, but they are

only applicable for plants grown in individual pots. Current active techniques for 3D imaging like laser range scanning tend to be more expensive, slower and more intrusive than a stereopsis approach, and they are therefore not applicable for the horticulture industry.

The proposed stereopsis technique is a passive vision approach, which does not interrupt the normal operation in a commercial glasshouse. The scanning rate is fast, which ensures efficiency. The core hardware is either one or two cameras, which reduces the complexity of the system and enables a flexible mounting anywhere in the glasshouse. Finally, the stereopsis system only requires ‘off-the-shelf’ products as core components. Until further advances in active vision techniques are made, the stereopsis approach is the most suitable method for collecting image data for horticultural applications.

An efficient stereo algorithm to estimate plant height has been discussed. The algorithm uses a Gaussian pyramid to represent input images at different resolutions. For each block in each resolution, a weighting method developed for an optimal parent node is applied to prevent the direct parent node from propagating poor results. By using Kalman filter updating at each level together with the weighting method for selecting an optimal parent, results from neighbours and scales are used to correct isolated wrong estimates. It has been demonstrated in Chapter 2 that the proposed stereo algorithm produces smooth depth maps for different plant types using the same parameter settings. Among all the algorithms tested, the proposed method was the most efficient, in terms of the running time, and was the most accurate, when compared with hand measurements.

### 6.1.2 Modelling of Plant Attributes

Given the estimated geometry and colour data from images, two approaches to modelling plant surface attributes have been presented. The plant surface modelling discussed in this thesis is concerned with plant attributes, rather than visual appearance. To this end, a Self-Organising Map (SOM) approach was adopted. As demonstrated in Chapter 3, this neural network approach can preserve the global geometry and colour information in the image, while eliminating incorrect height estimates. However, the fixed structure can cause issues in optimal data representation, and the choice of the final size is ambiguous. Furthermore, the SOM cannot form optimal classification when overlapped distributions are present, since it is not a probabilistic model and is identical to the  $K$ -means algorithm if no neighbourhood update is used.

Although the SOM is particularly useful for mapping the image data, a probabilistic model such as Multiresolution Gaussian Mixture Model (MGMM) can solve many of the problems faced by the SOM. In Chapter 4, it has been demonstrated that arbitrary probability densities can be approximated and evolved to a size which balances data fit and model complexity. Surface attributes, such as the plant height and the amount of flowers, are readily obtainable from the components in the MGMM. Furthermore, this parametric model is formulated in a Bayesian framework, which allows knowledge (that is, model parameters) to be updated from past experience and latest observations. This is a very similar process to what a human operator does, and its potential has been shown for modelling plant attributes for horticulture monitoring and grading purposes.

### 6.1.3 Tracking Plant Attributes Over Time

A Kalman filter can be used to track both the plant model and the associated statistical ‘knowledge’ (for example, noise models), and the key to track the model over time is the state-space formation. The Kalman filter predicts and updates the state as more observations become available. The effect of various parameters upon the filter’s performance has also been demonstrated using an example with synthetic data. It has been demonstrated that the Kalman filter successfully smoothens the noisy measurements and can produce a development trend for plant images.

The Kalman filter is a parametric model, which enables the parameters to be tuned for a particular crop and then re-used. Another feature is that the filter can give not only the general growth trend, but also produces a lack of fit using the difference between the smooth estimate and raw model estimate, which may indicate some unexpected problems in a crop. The covariance within each component in the MGMM can be used to indicate the variance of fitted data. Together with the lack-of-fit approach, the variation within a crop can be captured. This is very appealing, since it can indicate abnormalities of plants in a small area due to conditions such as irregular watering, spraying operations, or pests and diseases.

## 6.2 Limitations and Further Work

Although this thesis has presented modelling and analysis techniques and demonstrated their effectiveness and practicality in horticulture, there are still a number of limitations that could be removed.

### 6.2.1 Stereopsis

The most important drawback of the proposed stereo algorithm in Chapter 2 lies in the simplicity of the occlusion handling, which assumes a smooth surface. This approach works well for an image containing numerous plants, in which each plant can be seen as a smooth object. However, it is limited for images capturing a single plant. Rapid changes of depth occur frequently in this kind of image, due to the complex nature of plant branches and leaves. In this case, the proposed algorithm would produce a smooth change of depth, which does not strictly reflect the actual plant shape.

Since there are only two views of a scene, information is either ‘there’ or ‘not there’. Humans have a remarkable ability to judge depth from a single monocular image. This is done using monocular cues such as texture, occlusion and known object shapes. Similar techniques can be applied for stereo by using segmentation cues for example [114]. Marr [76] maintained that the basis for segmentation must be embedded in the early visual processes as general constraints, together with the geometrical consequence of the fact that surfaces coexist in 3D space. Nakayama *et al.* [82] further discussed the use of segmentation in relation to machine vision.

### 6.2.2 Multiresolution Gaussian Mixture Model

Currently, prior knowledge in the MGMM comes from the parent level, but another three sources of information could be used to aid estimation. The first source of knowledge is the uncertainty associated with each point generated by the stereo algorithm. As discussed in Chapter 4, this uncertainty was not built into a noise model for the MGMM, unlike the learning rate in the SOM, which leads to the assumption that the height data were 100% correct. A notable difference between the MGMM and the SOM is that the MGMM only uses the prior from the parent level, and does not use any neighbours, while the SOM is constructed from both the neighbours and the parents. Finally, knowledge about the model could come from the tracked Kalman estimates over time, which extends the prior knowledge to a time dimension.

Chapter 4 has presented a model selection criterion derived from the Bayesian Information Criterion (BIC), which balances data fit and model complexity. There is plenty of scope for further research in this area. The BIC can be seen as an approximation to the Bayes factors for Bayesian model choice, although exact Bayesian model selection can be much more complicated than BIC [42]. Wilson [120] discussed the use of log-Bayes factor and minimum information criteria. A better selection criterion should take account of preferences in different scenarios, similar to a decision problem [9]. For example, plants with well-developed flowers should be associated with a more complicated model than those that have just been potted.



### 6.2.3 Kalman Filter

The process and observation models presented in Chapter 5 are linear. Attributes such as population and colour, however, exhibit a more sophisticated trend over time. Although the Kalman filter has been shown to work well even with simple linear models, these models can be non-linear, and an extended Kalman filter [24] can then be applied.

The noise covariance matrix only has diagonal terms, which is common practice, but this leads to the use of a diagonal matrix solely describing the variances of each attributes. Off-diagonal terms should also be considered, due to the fact that correlation between the attributes may exist [102, 116].

Another possible improvement is the use of tracked Kalman estimates as prior knowledge for other techniques. This has already been discussed above for the Gaussian mixture model, but the tracked estimates may also be fed into the stereo algorithm [78]. This would enable the stereo algorithm to update the tracked estimates under a Bayesian scheme, given a new pair of images.

### 6.2.4 High-Dimensional Data Analysis

This thesis has presented techniques for extracting a set of plant attributes and tracking them over time. In practice, a quality score is usually defined to represent the plant quality in general. This score that classifies plants into different grades could be derived from all the attributes using a high-dimensional data classifier, such as a support vector machine [30].

In addition to the quality score, the detection of pests and diseases could

be achieved by examining the outliers in the data. Some symptoms of pests and diseases can only be identified by examining a number of attributes. Consider the *Chrysanthemum* stunt viroid for example, which is probably present in all *Chrysanthemum* growing regions [32]. The symptoms of this disease include reduced flower size, flower break or bleaching (that is, a reduction in colour intensity), and a reduction of up to 50% in overall height in mature plants. A plant suffering from these symptoms could be identified as an outlier compared with other healthy plants.

### 6.2.5 Other Sources of Information

Given the technique developed to analyse high-dimensional data, the number of data dimensions could be increased by incorporating information from other sources. In this thesis, all the image data were collected from conventional digital cameras, which give images similar to those seen by human eyes. Thermal imaging could be used to provide temperature information, which is not available from the visible spectrum (see [11] for an application in civilian law enforcement), and more recently applications were developed in plant physiology [25].

Researchers have already treated chlorophyll fluorescence as a tool in plant physiology, and interpretation of fluorescence signals can provide valuable information on stress effects [25]. Following the similar procedure suggested above for thermal imaging, a chlorophyll fluorescence sensor could be added into the existing structure to provide another dimension of data.

## 6.3 Concluding Remarks

This thesis has investigated techniques for modelling and analysing plant image data. There is much work yet to be done; nevertheless, the outcomes of this thesis could lead to a number of potentially important applications in horticulture such as crop growth monitoring. Furthermore, the methodologies of modelling and analysis developed in this thesis for ornamental plants grown in glasshouses could be applied to a wide range of other plants, including field crops and fruit crops, such as potato, sweet corn and strawberry.

# Appendix A

## A Brief Introduction to Bayesian Mixture Modelling

### A.1 Gaussian Mixture Model

Recall that in a Gaussian Mixture Model  $g(\mathbf{x})$  described in Chapter 4, data are modelled by a mixture of probability distributions. The density of  $g(\mathbf{x})$  can be approximated as follows:

$$g(\mathbf{x}) \simeq \hat{g}(\mathbf{x}) = \sum_{j=1}^K \pi_j f_j(\mathbf{x}|\theta_j) \quad (\text{A.1})$$

where the mixture weights  $\pi_j$  satisfy  $\sum_{j=1}^K \pi_j = 1$ . Suppose there are  $K$  components in the input data  $\mathbf{x}_i$ ,  $i = \{1, \dots, N\}$ , and each component is a Gaussian parameterised by its mean vector  $\mu_j$  and covariance matrix  $\Sigma_j$ ,

$j = \{1, \dots, K\}$ . The density of component  $j$  is

$$f_j(\mathbf{x}|\theta_j) = N_v(\mathbf{x}|\mu_j, \Sigma_j) \quad (\text{A.2})$$

$$= \frac{1}{(2\pi)^{D/2}} |\Sigma_j|^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \mu_j)^T \Sigma_j^{-1} (\mathbf{x} - \mu_j) \right\} \quad (\text{A.3})$$

where  $D$  is the dimension of the input data. A sample of input data  $\mathbf{x}_i$ ,  $i = \{1, \dots, N\}$ , can therefore be expressed as

$$\mathbf{x}_i \sim N_v(\mu_j, \Sigma_j) \quad (\text{A.4})$$

## A.2 Missing Data Structure

Robert [93] proposed that the class labels  $z_i, z_i \in \{1, \dots, K\}$ , which identify the component to which each data sample belongs, can be seen as the missing data part of the sample, since it is not observed.

In a Bayesian framework, given an appropriate prior density  $p(\theta_j)$  for the parameter  $\theta_j$ , the posterior density can be expressed as follows:

$$p(\theta_j|\mathbf{x}_i) \propto \mathbf{L}(\theta_j)p(\theta_j) \quad (\text{A.5})$$

$$= C^{-1} \sum_i \mathbf{L}(\theta_j)p(z_i|\theta_j)p(\theta_j) \quad (\text{A.6})$$

where  $\mathbf{L}(\theta_j)$  denotes the likelihood formed on the data  $\mathbf{x}_i$ , and  $p(z_i|\theta_j)$  denotes the conditional density of  $z_i$  given  $\theta_j$ . The normalising constant  $C$  is obtained by:

$$C = \int \sum_i \mathbf{L}(\theta_j)p(z_i|\theta_j)p(\theta_j) d\theta_j \quad (\text{A.7})$$

If ‘conjugate prior’ is specified for a missing data structure  $z_i$ , simulation conditional on  $z_i$  is then possible.

### A.3 Conjugate Priors

In Bayesian methods, prior is usually chosen so that the posterior belongs to the same form of distribution as the prior. Prior and posterior chosen in this way are said to be conjugate. For example, given a Gaussian likelihood,

$$p(\mathbf{x}|\theta_1) = N_v(\mu_1, \Sigma_1) \quad (\text{A.8})$$

and choosing a Gaussian prior,

$$p(\theta_1) = N_v(\mu_2, \Sigma_2) \quad (\text{A.9})$$

the posterior is still Gaussian as shown below,

$$p(\theta_1|\mathbf{x}) = N_v(\Sigma_1 \Sigma^{-1} \mu_1 + \Sigma_2 \Sigma^{-1} \mu_2, \Sigma^{-1}) \quad (\text{A.10})$$

where  $\Sigma$  is the sum of  $\Sigma_1$  and  $\Sigma_2$ :

$$\Sigma = \Sigma_1 + \Sigma_2 \quad (\text{A.11})$$

For the Gaussian mixture, this implies using the following conjugate densities: Dirichlet for the mixture weights, Gaussian for the means and inverse

Wishart for the covariances.

$$\pi_j \sim D(\alpha_1, \dots, \alpha_K) \quad (\text{A.12})$$

$$\mu_j \sim N_v(\epsilon_j, \tau_j \Sigma_j) \quad (\text{A.13})$$

$$\Sigma_j^{-1} \sim W_v(\omega_j, \lambda_j) \quad (\text{A.14})$$

The Wishart distribution  $W_v(\omega_j, \lambda_j)$  is defined as follows:

$$W_v(\omega_j, \lambda_j) = \frac{|\mathbf{U}|^{(\lambda_j - v - 1)/2} \exp\{-tr(\omega_j^{-1} \mathbf{U}/2)\}}{2^{v\lambda_j/2} |\omega_j|^{\lambda_j/2} \Gamma_v(\lambda_j/2)} \quad (\text{A.15})$$

where  $\lambda_j$  represents degree of freedom.  $\mathbf{U}$  is a  $v \times v$  positive-definite matrix of random variables, and  $\Gamma_v$  is a multivariate gamma function defined as follows:

$$\Gamma_v(\lambda_j/2) = \pi^{v(v-1)/4} \prod_{j=1}^v \Gamma(\lambda_j/2 + (1-j)/2) \quad (\text{A.16})$$

## A.4 MCMC: Posterior Simulation

The posterior distribution can be approximated by *Markov Chain Monte Carlo* (MCMC) sampling methods, which construct a *Markov chain* with stationary distribution equal to the posterior distribution. By sampling, *Monte Carlo* estimates can be obtained for the expectations of a probability distribution. Let  $\theta$  be a set of random variables,  $\theta = \{\theta_1, \dots, \theta_K\}$ , which characterise a process described by a probability distribution function  $f(\theta)$ . The expectation of the distribution,  $E(f(\theta_{ij})), j = \{1, \dots, K\}$ , can be ap-

proximated by samples from  $f(\theta)$  shown below:

$$E(f(\theta_{ij})) \approx \frac{1}{N} \sum_{i=1}^N f(\theta_{ij}) \quad (\text{A.17})$$

where  $\theta_{ij}$  is the  $i$ -th sample, and  $N$  is the number of samples. It can be shown from equation (A.17) that a sufficiently large number of sampling iterations ensures accurate approximation. However, a set of suitable starting values reduces the number of iterations, which is often applied in practice.

Gibbs sampling is an established MCMC sampling method [93]. As shown in Chapter 4, the use of Gibbs sampling leads to a missing-data Markov chain and a model parameter Markov chain, which iteratively simulating from the conditional distribution of a vector of parameters while fixing all the others.



# Appendix B

## Publication

- Y. Song, R. Wilson, R. Edmondson, and N. Parsons. Surface modelling of plants from stereo images. In *Proceedings of the 6th International Conference on 3-D Digital Imaging and Modeling*, Montreal, Canada, 2007.

# Surface Modelling of Plants from Stereo Images

Yu Song

Warwick HRI & Computer Science,  
University of Warwick, UK

yu@dcs.warwick.ac.uk

Rodney Edmondson

Warwick HRI,  
University of Warwick, UK

rodney.edmondson@warwick.ac.uk

Roland Wilson

Department of Computer Science,  
University of Warwick, UK

rgw@dcs.warwick.ac.uk

Nick Parsons

Department of Statistics,  
University of Warwick, UK

n.parsons.1@warwick.ac.uk

## Abstract

*Plants are characterised by a range of complex and variable attributes, and measuring these attributes accurately and reliably is a major challenge for the industry. In this paper, we investigate creating a surface model of plant from images taken by a stereo pair of cameras. The proposed modelling architecture comprises a fast stereo algorithm to estimate depths in the scene and a model of the scene based on visual appearance and 3D geometry measurements. Our stereo algorithm employs a coarse-fine strategy for disparity estimation. We develop a weighting method and use Kalman filter to refine estimations across scales. A self-organising map is applied to reconstruct a surface from these sample points created by the stereo algorithm. We compare and evaluate our stereo results against other popular stereo algorithms, and also demonstrate that the proposed surface model can be used to extract useful plant features that can be of importance in plant management and assessing quality for marketing.*

## 1. Introduction

This paper introduces a novel multiresolution surface model to represent plant characteristics from a pair of stereo images. Plants such as *pansy* and *poinsettia* can be characterised by a number of quality attributes that affect customer acceptance. The measurement of those plant characteristics that are most relevant to a grower is a challenging problem, which has previously been tackled almost exclusively by a combination of manual measurement and visual inspection [1]. This is a highly labour-intensive process, given that tens of thousands of plants have to be assessed, and one which is inevitably limited by the subjective assessments of individual observers. The purpose of our work is to automate this process to reduce labour requirements, and

to remove subjective factors from the assessment. Our approach is to use a stereo pair of cameras to capture images of plants and then to model the chief visual characteristics of the plants, including height, leaf and flower areas and colour.

Currently, there is no method available for automated monitoring of plant growth remotely, and there are limited systems for monitoring plants under controlled conditions at marketing. Although the idea of automated monitoring has been suggested by many authors, there are currently no methods for monitoring crops in the manner we are suggesting. Ehret *et al.* [1] gave a general review of automated monitoring of greenhouse crops, and discussed applications of digital imaging plants in greenhouses, but did not present any practical results. In this paper, we present a modelling system that can be used either remotely or under controlled conditions, and results for a selection of ornamental plants growing in packs. The hardware part of our modelling system is a stereo pair of ‘off-the-shelf’ digital cameras that can be mounted in a glasshouse for remote monitoring or on a production grading line. This is readily accomplished, cheap and causes minimal disruption to the normal operations of growers. Moreover, the use of conventional cameras allows capture of colour information, which is useful in assessing plant quality. The proposed modelling architecture is divided into two parts: *a*) a high-speed stereo algorithm to estimate depths in the scene; *b*) a model of the scene based on visual appearance combined with geometry information.

Stereopsis has been extensively researched in the vision community during the last two decades, and recently, energy minimisation using methods like graph cuts [2] and belief propagation [3] have become popular as a basis for many of the best stereo algorithms in the Middlebury list [4]. Figure 1(c)-(j) show a selection of stereo images of ornamental plants, and the complex structure of small,

thin branches and green leaves leads to substantial occlusions in the image. Handling occlusions is still an active research area, and some papers have described energy minimisation techniques [2, 3, 4] to regularise the depth results. Instead of trying to refine estimated depths for individual leaves and branches, our approach is to model the global geometry of the plant. To this end, we develop our stereo algorithm using a multiresolution pyramid structure by assuming the plant surface is smooth. The use of the pyramid structure also gives a significant speed advantage (see [5]). We present our results against other popular stereo algorithms (Section 3), and prove our method is most efficient and reliable for plant images.

Another challenge of plant images is large colour variation within each part of the object (e.g. flower, leaf, background and etc.). We can see from the *pansies* shown in Figure 1(c) that parts of some yellow flowers are black and there is white residue visible on the green leaves due to a fungicidal spray. Such large colour variation in the image causes problems in discriminating each class of the objects. In this paper, we propose a surface model that can effectively simplify the colour data.

This rest of the paper is organised as follows: Section 2 describes details of methods we used to model plants. The results of modelling are shown in Section 3, and Section 4 discusses the results. Section 5 concludes the paper and outlines plans for the further work.

## 2. Methods

### 2.1. Overview

Our proposed modelling architecture comprises two parts: *a)* a high-speed stereo algorithm to estimate depths in the scene; *b)* a model of the scene based on visual appearance combined with geometry information.

The stereo algorithm we develop uses a multiresolution Gaussian pyramid [5]. The input images are first divided into a number of blocks overlapped by 50% in each level. And for each block in each resolution, a weighting method for an optimal parent node is applied. Within a small search window around the optimal parent node, an adopted matching cost function by Birchfield and Tomasi [6] is applied to find the corresponding blocks. After matching the corresponding blocks, a Kalman filter [7] is used to update the matching results based on estimations in the current and previous levels, and the disparity results are then interpolated to the next level. Finally, we convert the disparity results into depths by triangulation.<sup>1</sup> We discuss these steps in details in Section 2.2.

Having estimated the surface geometry, we model the

data to represent the plant in the scene. The surface data has a number of resolutions, and each resolution includes (R,G,B) colour information and (X,Y,Z) geometry information obtained from the stereo algorithm. Since the depths estimated contain errors due to many occlusions in the scene, the surface model must minimise the effects of the outliers. We can see from Figure 1 that plants have complex structures as well as ambiguous visual appearances. Moreover, growers require summary statistics for plant growth management and marketing purposes. Consequently, a method to simplify the surface data is required. We investigated the use of a Self-Organising Map (SOM) [9] to reconstruct a surface for each resolution of the data, and more details are shown in Section 2.3.

### 2.2. The Multiresolution Stereo Algorithm

**2.2.1. Matching.** An important cue to depth perception - disparities between stereo images - can be measured by finding matching pixels [10]. The task of matching is also known as the correspondence problem. The common matching cost functions are squared or absolute differencing. It is possible to efficiently compute correlation in the frequency domain using the fast Fourier transform producing similar results to the squared differencing method. Normalised cross-correlation in the spatial domain [11], which is invariant to local changes of intensity and illumination, can be applied. Phase-matching [12] is another alternative. For a more detailed review on this subject, see Scharstein and Szeliski's paper [4]. The matching cost function we used for each colour channel to compute dissimilarity  $\{D_R, D_G, D_B\}$  is identical to the one proposed by Birchfield and Tomasi [6]. The maximum cost of  $\{D_R, D_G, D_B\}$  is set to be 30, and this enables us to normalise the cost of matching to an error score  $e$  ranging between 0 and 1. The RGB pixel dissimilarities  $\{D_R, D_G, D_B\}$  are then summed for the overall cost  $D_a$ .

$$D_a = D_R + D_G + D_B \quad (1)$$

For the ideal parallel camera setup or the calibrated setup [8], the corresponding points in both images lie on the same horizontal scanline. Therefore, the displacement of a pixel in the left image  $L_{i,j}$  and the corresponding one in the right image  $R_{i,j}$  can be expressed as:

$$R_{i,j} = L_{i,j-d} + V_{i,j} \quad (2)$$

where  $d$  is the disparity measured between two pixels, and  $V_{i,j}$  is a noise factor depending on the error score  $e_{i,j}$ . However, it is not unusual for multiple points in the same scanline to have identical pixel values. One way to improve this is to match corresponding regions instead of individual pixels. Assuming pixels in a small region have the same disparity value, we can divide the image into a number of blocks overlapped by 50% and then match these blocks.

<sup>1</sup>After calibrating the camera setup [8], it is straightforward to convert the disparity results into depths by triangulation.

**2.2.2. Multiresolution Representation.** The main problem with many matching methods is to deal with local minima in the search window. To effectively limit the size of the search window, one can follow a multiresolution pyramid approach [5]. The corresponding pixels between stereo images are first found in the highest level of the pyramid, and then refined by adding innovations further down the pyramid. This is expressed using a state-space approach:

$$d(n) = I(n|n+1) \times d(n+1) + w(n) \quad (3)$$

where  $d(n)$  is the disparity image data at level  $n$  of the pyramid,  $I(n|n+1)$  is an interpolator between level  $n$  and level  $n+1$ , and  $w(n)$  represents innovations or details to be added at level  $n$ . Using the pyramid structure, the chances of falling into local minima are reduced. The size of the search window is effectively limited, and it remains constant throughout different levels. Another appealing feature of the multiresolution approach is efficiency. The computational costs are reduced greatly compared with using the original images directly. In this work, we use a Gaussian pyramid [5], and a bilinear interpolator for propagating disparity results to the next level.

**2.2.3. The Optimal Parent.** As the search window is small for every level in the pyramid, poor results could be propagated by direct parent nodes through levels, leading to false matches between images. To prevent this, a method to weight a set of parent nodes to form an optimum estimate is applied. The method we propose takes account of inputs from neighbouring parent nodes as well as the direct parent node. The method initialises the weights  $w_{i,j,n|n+1}$  for block  $(i, j)$  at level  $n$  by selecting ‘good’ parents from 9 nodes around the direct parent (including the direct parent itself). This selection can be done by median thresholding on the errors  $e_{x,y,n+1}$  associated with each parent node. After that, the weights are calculated as follows:

$$w_{x,y,n|n+1} = e_{max} - e_{x,y,n|n+1} \quad (4)$$

$$w_{x,y,n|n+1} = \frac{w_{x,y,n|n+1}}{\sum_{x,y} w_{x,y,n|n+1}} \quad (5)$$

where  $x = \{i-1, i, i+1\}$ ,  $y = \{j-1, j, j+1\}$  for the child block  $(i, j)$ ,  $e_{x,y,n|n+1}$  is the error interpolated from level  $n+1$  to  $n$  for one of the selected good parents, and  $e_{max}$  is the maximum error in these good parent nodes. The optimal parent node can then be inferred as follows:

$$d_{i,j,n|n+1} = \sum_{x,y} d_{x,y,n|n+1} \times w_{x,y,n|n+1} \quad (6)$$

$$e_{i,j,n|n+1} = \sum_{x,y} e_{x,y,n|n+1} \times w_{x,y,n|n+1} \quad (7)$$

where  $d_{i,j,n|n+1}$  and  $e_{i,j,n|n+1}$  are disparities and errors respectively in the optimal parent node for the child block  $(i, j)$ .

**2.2.4. The Kalman Filter.** Given disparities and errors associated from the coarser resolution, we can combine the results previously estimated and the results from the current resolution to form optimal results using a Kalman filter [7]. The Kalman gain  $G_{i,j,n}$  is calculated using:

$$G_{i,j,n} = \frac{e_{i,j,n|n+1}}{e_{i,j,n|n+1} + e_{i,j,n}} \quad (8)$$

Denoting  $d_{i,j,n}$  and  $e_{i,j,n}$  as the disparities and errors respectively for block  $(i, j)$  at level  $n$ , the Kalman updating process works as follows:

$$d_{i,j,n} = d_{i,j,n|n+1} + G_{i,j,n} \times \{d_{i,j,n} - d_{i,j,n|n+1}\} \quad (9)$$

$$e_{i,j,n} = e_{i,j,n|n+1} + G_{i,j,n} \times \{e_{i,j,n} - e_{i,j,n|n+1}\} \quad (10)$$

In this work, we assume the Kalman gain  $G_{i,j,n}$  to be a constant for each block  $(i, j)$  at level  $n$ .

### 2.3. Surface Modelling

The basic configuration of the SOM we used is a two-dimensional grid of cells denoted by  $\{C_1, C_2, \dots, C_m\}$ . Each cell contains a model vector, which has six dimensions (R,G,B,X,Y,Z) identical to the input data. The SOM evolves by learning random-selected samples of input data. For each sample  $x(t)$ , the SOM chooses the best cell  $C_b(t)$  to match the data. This can be formulated as:

$$\|C_b(t) - x(t)\| = \min_i \|C_i(t) - x(t)\| \quad (11)$$

Then, the ‘winner’ cell  $C_b(t)$  and its neighbours  $C_i(t)$  update depending on the stage of the learning process. This learning process is shown below:

$$C_i(t+1) = C_i(t) + h_{C_b(t)} \{x(t) - C_i(t)\} \quad (12)$$

where  $h_{C_b(t)}$  is the neighbourhood function, and it is similar to a smoothing kernel that is time-variable and its location depends on the position of the ‘winner’ cell  $C_b(t)$ . This learning process repeats until each dimension of the surface model has a sufficiently accurate representation for the data. At the end of the learning process, the SOM surface model preserves the global geometry as well as colour information of the object featured in the image. Using this approach, incorrect depths estimated by the stereo algorithm can be effectively eliminated and the complex visual appearance is also simplified. In this work, an incremental-learning SOM implementation was used [13].

## 3. Results

This section presents some of the results obtained from our stereo and surface modelling methods discussed in this paper. Figure 1(c)-(j) show images capturing four packs of

different plants, *large pansies*, *small pansies*, *dianthus* and *viola*. Also, an object with known geometry, *an upended flowerpot*, is displayed in Figure 1(a)-(b). The size of input images is  $512 \times 384$ , and all these images were rectified using calibration procedures in [8].

We compared our multiresolution stereo algorithm with the Pixel-to-Pixel algorithm (P2P) by Birchfield and Tomasi [14]. In addition, the stereo algorithm with graph-cut optimisation (GC) discussed in Scharstein and Szeliski's paper [4] was also evaluated. Input images for these two methods are grey-level, while our algorithm takes colour images as input.

### 3.1. Stereo Results of a Known-Geometry Object

Firstly, we tested and evaluated our stereo algorithm by processing images featuring a known-geometry object (Figure 1(a)-(b)). The expected disparity result should clearly show a near-cylinder-like object slightly tapering to the top. Our stereo algorithm used a 4-level Gaussian pyramid with blocks sized  $16 \times 16$  overlapped by 50%, as well as techniques discussed in this paper. The search range used in our method was 9 as the maximum disparity was 70. For the P2P algorithm, default parameters were used. The graph-cut method used Birchfield-Tomasi's matching cost function [6] and a smoothness weight of 50, without any penalty cost. The gradient-dependent smoothness cost for the graph-cut method was 8 in all experiments. Disparity results are presented in Figure 2. A summary of the running time and height measured by a human operator and measurements derived from the stereo results in Figure 2 is presented in Table 1.

	Running Time (in seconds)	Height (in cm)
Human	*	29.2
P2P	1.9	30.0
GC	1164.9	29.2
MR	0.8	29.2

**Table 1. Comparisons of the running time and height measured from Figure 2**

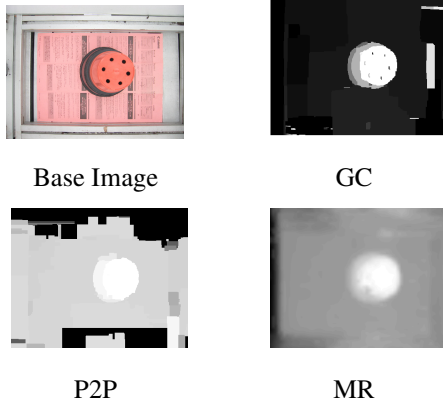
The height is measured from the bottom to the top of the flowerpot.

### 3.2. Stereo Results for Plants

For the plant images (Figure 1(c)-(j)), the settings for our stereo algorithm were identical to those discussed in Section 3.1. However, even after testing various parameters, the P2P stereo algorithm could not produce any reasonable solution, and therefore was excluded. We present two re-



**Figure 1. 5 pairs of stereo images: An upended flowerpot - (a), (b); Large pansies - (c), (d); Small pansies - (e), (f); Dianthus - (g), (h); Viola - (i), (j)**



**Figure 2. The stereo results of an upended flow-erpot with known geometry**

Top-Left: The base image; Top-Right: Results produced by the graph-cut method; Bottom-Left: Results produced by the P2P algorithm; Bottom-Right: Results produced by our proposed multiresolution algorithm.

sults from the graph-cut algorithm proposed by Scharstein & Szeliski with different smoothness (20 and 50) and penalty costs (4 and 2). The graph-cut method also used the matching cost function proposed by Birchfield and Tomasi.

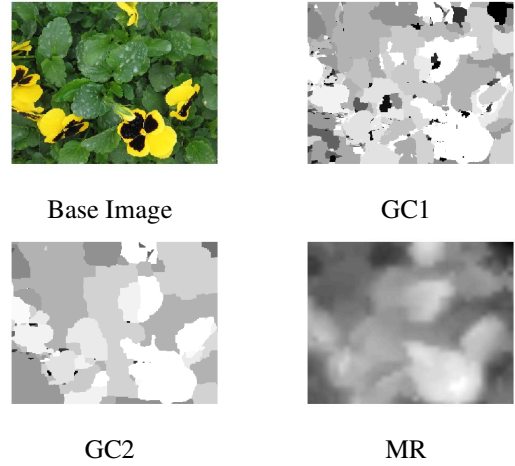
As the boundary information of the pack holding the plant was available (i.e. the position and size of the pack), background was cropped outside the boundary region. Results for the *large pansies* in Figure 1(c)-(d) is shown in Figure 3, and the *small pansies* in Figure 1(e)-(f) is in Figure 4. Results for the *dianthus* (Figure 1(g)-(h)) and the *viola* (Figure 1(i)-(j)) are presented in Figure 5 and Figure 6 respectively.

### 3.3. Surface Modelling Results of Plants

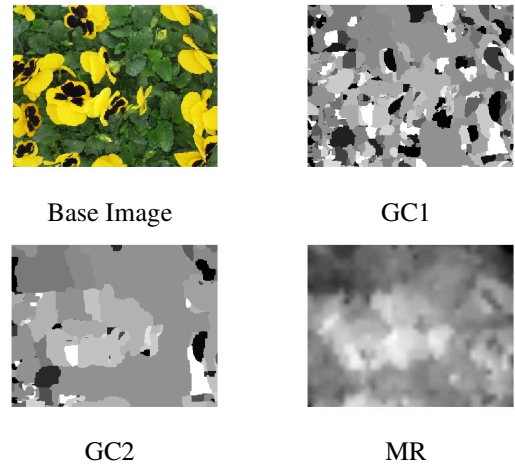
Using our stereo results obtained in Section 3.2, we model the plant surface by using a Self-Organising Map (SOM). The SOM we used had a  $32 \times 32$  hexagonal grid of neurons and a Gaussian neighbourhood function [13]. Figure 7 presents the surface models for *large pansies*, *small pansies*, *dianthus* and *viola* shown in Figure 1.

### 3.4. Running Time

Our stereo algorithm is written in C, and the Linux machine running these experiments has one *Intel Pentium 4* 3.0G HZ CPU with 1GB RAM. The running time of our stereo algorithm for each pair of stereo images in Figure 1 was less than 1 second, including reading and writing images. The C code for the P2P algorithm took around 2 seconds, while the C++ code for the graph-cut method used about 20-25 minutes for each parameter setting. Table 1

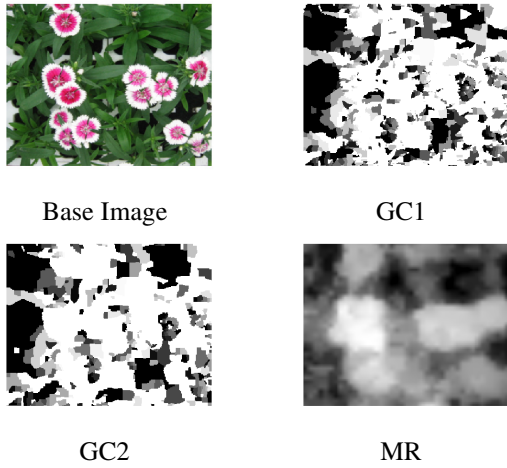


**Figure 3. The stereo results of the large pansies**  
Top-Left: The base image; Top-Right: Results produced by the graph-cut method with a smoothness cost of 20 and a penalty cost of 4; Bottom-Left: Results produced by the graph-cut method with a smoothness cost of 50 and a penalty cost of 2; Bottom-Right: Results produced by our proposed multiresolution algorithm.



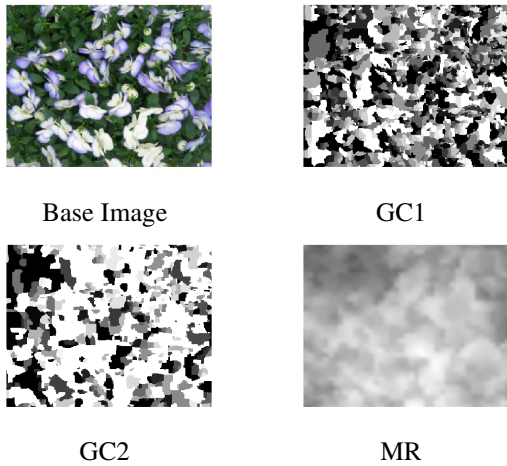
**Figure 4. The stereo results of the small pansies**  
Top-Left: The base image; Top-Right: Results produced by the graph-cut method with a smoothness cost of 20 and a penalty cost of 4; Bottom-Left: Results produced by the graph-cut method with a smoothness cost of 50 and a penalty cost of 2; Bottom-Right: Results produced by our proposed multiresolution algorithm.





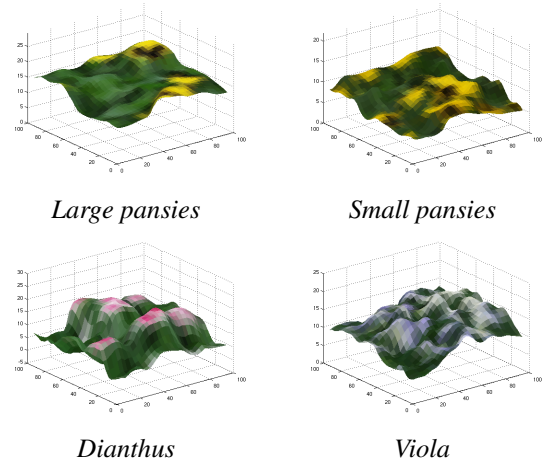
**Figure 5. The stereo results of the *dianthus***

Top-Left: The base image; Top-Right: Results produced by the graph-cut method with a smoothness cost of 20 and a penalty cost of 4; Bottom-Left: Results produced by the graph-cut method with a smoothness cost of 50 and a penalty cost of 2; Bottom-Right: Results produced by our proposed multiresolution algorithm.



**Figure 6. The stereo results of the *viola***

Top-Left: The base image; Top-Right: Results produced by the graph-cut method with a smoothness cost of 20 and a penalty cost of 4; Bottom-Left: Results produced by the graph-cut method with a smoothness cost of 50 and a penalty cost of 2; Bottom-Right: Results produced by our proposed multiresolution algorithm.



**Figure 7. The plant surface models**

The geometry data is obtained from our stereo results in Section 3.2.

shows the running time using the three methods to process images in Figure 1(a)-(b). The C code for the SOM [13] spent about 50 seconds on the surface modelling. We are developing a multiresolution SOM and the time required would be comparable to our stereo algorithm.

## 4. Discussion

For the stereo images featuring an object with known geometry (Figure 2), our stereo algorithm produced the expected smooth disparity map and was able to give a height estimation with no difference from the one measured by a human operator (Table 1). The P2P algorithm by Birchfield and Tomasi [14] gave better estimates for the sloping side, where occluded regions lie. The best result was obtained by the graph-cut method discussed in Scharstein and Szeliski's paper [4]. For real plant images, however, the P2P method produced very noisy results, and therefore was excluded. The graph-cut method produced either under-smoothed results (GC1) with noisy estimates clearly visible, or over-smoothed results (GC2) leading to fusing different depth layers together. Figure 8 illustrates reconstructed views of the *large pansies* from the stereo results shown in Figure 3. We can see from the over-smoothed results (GC2) in Figure 8 that the shapes of yellow *pansy* flowers and leaves are flat planes without any height variation. In comparison, our stereo algorithm gave better estimates that are much like the real plants captured. Table 2 presents the average height of the whole plant computed from our stereo results and results by the graph-cut method, and measurements made by a human operator are shown in Table 3. Furthermore, when using the graph-cut method, the choice

of parameters was difficult and the running time was in the order of minutes. Our proposed multiresolution stereo algorithm produced disparity results within a second, and it only required to change at most three typical parameters (search range, window size and pyramid levels). These parameters can be easily configured once for a particular camera setup to process different types of images, e.g. *flowerpot* images in Section 3.1 and plant images in Section 3.2.

The expected results of our stereo algorithm are obtained by applying all the following methods: *a)* Matching cost function for colour images is used. *b)* Multiresolution representation is used. The coarse level can deliver a reasonable estimate to the next level, as well as a global effect of the smooth results. *c)* The optimal parent node is used to prevent the direct parent node from propagating poor results. By using this approach together with the Kalman filter updating at each level, results from neighbours as well as inter-scanlines can be used to correct isolated wrong estimates.

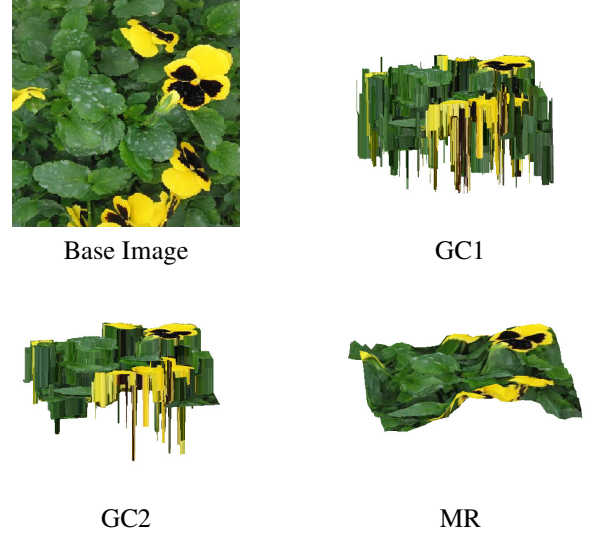
It is clear from Figure 7 that the plant surface model we proposed can reduce the complexity of colour information. Therefore, a supervised multi-thresholding image segmentation method [10] can be used to distinguish between flower and leaf. Many features of interests can be measured from the surface models presented in Figure 7. For example, the average flower and leaf height of the *pansies*, almost exclusively measured by human operators in industry, can be easily extracted (see Table 3).

The average height measurements are presented mainly to validate our approach, and our surface model can quantify a lot visual information (e.g. colour, shape, uniformity and etc.) besides geometry measurements. Table 4 shows the flower area measured in percentage from our models shown in Figure 7, and the measurement of this feature over time could be used for plant growth management and crop scheduling. Besides features from the finest resolution of the model, we can also use attributes from other resolutions. For example, Figure 9 shows a plane, the coarsest resolution of the model, representing the height variation of the plant. We can see the evidence of a sloping surface of the pack, a feature which is important in grading and assessing packs for marketing.

## 5. Conclusions

In this paper, a novel system to model a plant surface from a pair of stereo images has been presented. The proposed surface modelling architecture comprises a high-speed stereo algorithm to infer geometry information in the scene and a model of the scene based on the visual appearance combined with the geometry information measured.

Compared with the Pixel-to-Pixel stereo algorithm [14] and the graph-cut method [4], we have demonstrated that



**Figure 8. Reconstructed views of the *large pansies***

The geometry data is obtained from the stereo results in Figure 3.

	Average height (in cm) from stereo methods		
	GC1	GC2	MR
<i>Large pansies</i>	13.6	14.2	11.5
<i>Small pansies</i>	8.5	8.9	6.3
<i>Dianthus</i>	13.2	13.4	8.6
<i>Viola</i>	12.3	12.3	11.6

**Table 2. Comparisons of the average height of plants measured from our stereo results and the graph-cut method**

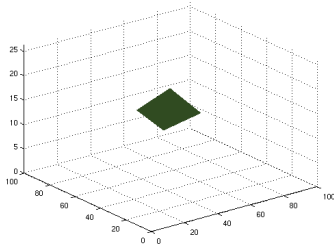
	Our Model Measurements (in cm)		Human Measurements (in cm)	
	Flower	Leaf	Flower	Leaf
<i>Large pansies</i>	14.8	11.0	15	13
<i>Small pansies</i>	7.1	5.9	7	5
<i>Dianthus</i>	13.2	7.4	*	*
<i>Viola</i>	13.2	10.6	*	*

**Table 3. Comparisons of the average height of plants measured by our surface models against a human operator**



Large <i>pansies</i>	Small <i>pansies</i>	<i>Dianthus</i>	<i>Viola</i>
12.7%	31.2%	21.2%	40.5%

**Table 4. Comparisons of the flower area of plants measured in percentage by our surface models**



**Figure 9. The coarsest resolution of the surface model for the large pansies**

The geometry data is obtained from the coarsest resolution of the stereo results in Figure 3.

our stereo algorithm produces smooth depth maps as expected for four examples of different plant types. Among the three algorithms tested, our method was also the most efficient one in terms of the running time.

Our proposed SOM surface model preserves the global geometry of the plant, as well as visual colour information. We have demonstrated that by using the proposed surface model, we can extract many useful plant features from different resolutions, which could be used for a large number of important industry applications in many areas such as plant growth management, crop scheduling and assessing packs for marketing.

The error  $e$  we used in the stereo algorithm only considers the cost of matching. In some cases, false matches could yield small cost. Therefore, some penalty methods similar to the graph-cut method [4] should apply. In this paper, the effects of lighting and shading are not discussed, which could be a major issue for matching. We are analysing images captured under extreme lighting conditions to test the robustness of the stereo algorithm. As the time required to run the SOM is long compared with the stereo algorithm, we are investigating a multiresolution approach to construct the SOM. We are also looking at extracting more important plant features from the surface model.

## Acknowledgements

This work was funded by the Horticultural Development Council, UK (contract number: CP 37 & PC 200). The authors are grateful to members of the Signal and Image Pro-

cessing Group at Warwick for their discussions and helps.

## References

- [1] D. L. Ehret, A. Lau, S. Bittman, W. Lin, and T. Shelford. Automated monitoring of greenhouse crops. *Agronomie*, 21:403–414, 2001.
- [2] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions via graph cuts. In *International Conference on Computer Vision*, pages 508–515, 2001.
- [3] J. Sun, N.-N. Zheng, and H.-Y. Shum. Stereo matching using belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):787–800, 2003.
- [4] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3):7–42, 2002.
- [5] E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt, and J. M. Ogden. Pyramid methods in image processing. *RCA Engineer*, 29(6), 1984.
- [6] S. Birchfield and C. Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(4):401–406, 1998.
- [7] G. Welch and G. Bishop. An introduction to the kalman filter. Technical report, Chapel Hill, NC, USA, 1995.
- [8] J.-Y. Bouguet, Camera calibration toolbox for Matlab.
- [9] T. Kohonen. *Self-Organizing Maps*. Springer, 2nd edition, 1997.
- [10] D. Vernon. *Machine vision: automated visual inspection and robot vision*, pages 49–51, 86–89. Prentice Hall, 1991.
- [11] J. P. Lewis. Fast template matching, 1995.
- [12] C. D. Kuglin and D. C. Hines. The phase correlation image alignment method. *IEEE Conference on Cybernetics and Society*, pages 163–165, 1975.
- [13] T. Kohonen, J. Hynninen, J. Kangas, and J. Laaksonen. SOM PAK: The Self-Organizing Map program package, Report A31, Helsinki University of Technology, Laboratory of Computer and Information Science, 1996.
- [14] S. Birchfield and C. Tomasi. Depth discontinuities by pixel-to-pixel stereo. *International Journal of Computer Vision*, 35(3):269–293, 1999.

# Bibliography

- [1] E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt, and J. M. Ogden. Pyramid methods in image processing. *RCA Engineer*, 29(6), 1984.
- [2] S. C. Ahalt, A. K. Krishnamurthy, P. Chen, and D. E. Melton. Competitive learning algorithms for vector quantization. *Neural Networks*, 3(3):277–290, 1990.
- [3] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.
- [4] L. B. Alberti. De pictura. In *Laterza*, 1980.
- [5] F. Anastacio, M. C. Sousa, F. Samavati, and J. A. Jorge. Modeling plant structures using concept sketches. In *Proceedings of the 4th international symposium on Non-photorealistic Animation and Rendering*, pages 105–113, 2006.
- [6] D. Ballard and C. Brown. *Computer Vision*. Prentice Hall, 1982.
- [7] J. Batlle, E. Mouaddib, and J. Salvi. Recent progress in coded structured light as a technique to solve the correspondence problem: a survey. *Pattern Recognition*, 31(7):963 – 982, 1998.
- [8] J.-A. Beraldin, M. Picard, S. F. El-Hakim, G. Godin, V. Valzano, A. Bandiera, and D. Latouche. Virtualizing a Byzantine crypt by combining high-resolution textures with laser scanner 3D data. In *Proceedings of the International Conference on Virtual Systems and Multimedia*, pages 25–27, 2002.
- [9] J. O. Berger. *Statistical decision theory and Bayesian analysis*. Springer, 2nd edition, 1985.
- [10] S. Birchfield and C. Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(4):401–406, 1998.

- [11] T. L. Bisbee and D. A. Pritchard. Today's thermal imaging systems: background and applications for civilian law enforcement and military force protection. *Proceedings of the IEEE International Carnahan Conference on Security Technology*, pages 202–208, 1997.
- [12] F. Blais, J. Taylor, L. Cournoyer, M. Picard, L. Borgeat, G. Godin, J.-A. Beraldin, M. Rioux, and C. Lahanier. Ultra high-resolution 3D laser color imaging of paintings: The mona lisa by leonardo da vinci. In *Proceedings of the 7th International Conference on Lasers in the Conservation of Artworks*, 2007.
- [13] J.-Y. Bouguet. Camera calibration toolbox for Matlab.
- [14] A. Bowen. *Multiresolution image modelling and estimation*. PhD thesis, University of Warwick, UK, 2008.
- [15] A. M. Bowen, A. Mullins, N. M. Rajpoot, and R. G. Wilson. Photo-consistency and multiresolution methods for light field disparity estimation. In *Proceedings IEE International Conference on Visual Information Engineering*, 2005.
- [16] G. E. P. Box and G. Jenkins. *Time Series Analysis, Forecasting and Control*. Holden-Day, Incorporated, 1990.
- [17] P. S. Bradley and U. M. Fayyad. Refining initial points for k-means clustering. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 91–99, 1998.
- [18] R. G. Brown and P. Y. C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering*. John Wiley & Sons, 3rd edition, 1997.
- [19] J. Bruske and G. Sommer. Dynamic cell structure learns perfectly topology preserving map. *Neural Computation*, 7(4):845–865, 1995.
- [20] C. R. Bull, R. Zwiggelaar, and J. V. Stafford. Imaging as a technique for assessment and control in the field. *Aspects of Applied Biology*, 43:197–204, 1995.
- [21] K. P. Burnham and D. R. Anderson. *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*. Springer, New York, NY, 2002.
- [22] P. J. Burt and E. H. Adelson. The Laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, COM-31,4:532–540, 1983.
- [23] A. D. Calway, H. Knutsson, and R. Wilson. Multiresolution estimation of 2-d disparity using a frequency domain approach. In *British Machine Vision Conference*, pages 227–236, 1992.

- [24] J. V. Candy. *Signal Processing: The model-based approach*. McGraw-Hill, 1986.
- [25] L. Chaerle, D. Hagenbeek, E. de Bruyne, R. Valcke, and D. van der Straeten. Thermal and chlorophyll-fluorescence imaging distinguish plant-pathogen interactions at an early stage. *Plant and Cell Physiology*, 45(7):887–896, 2004.
- [26] K. C. Chou and A. S. Willsky. A multi-resolution, probabilistic approach to 2D inverse conductivity problems. Technical report, Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, MA, USA, 1988.
- [27] K.C. Chou, A.S. Willsky, and A. Benveniste. Multiscale recursive estimation, data fusion, and regularization. *IEEE Transactions on Automatic Control*, 39(3):464–478, 1994.
- [28] S. Clippingdale. *Multiresolution image modelling and estimation*. PhD thesis, University of Warwick, UK, 1988.
- [29] A. Criminisi, I. D. Reid, and A. Zisserman. Single view metrology. *International Journal of Computer Vision*, 40(2):123–148, 2000.
- [30] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines: And Other Kernel-based Learning Methods*. Cambridge University Press, 1999.
- [31] J.C. Davey, G.W. Horgan, and M. Talbot. Image analysis: a tool for assessing plant uniformity and variety matching. *Journal of Applied Genetics*, 38:120–135, 1997.
- [32] DEFRA. Chrysanthemum stunt viroid. Department for Environment, Food and Rural Affairs, UK, 2000.
- [33] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.
- [34] F. W. Depiero and M. M. Trivedi. 3-d computer vision using structured light. In *Advances in Computers, Design, Calibration, and Implementation Issues*, pages 243–278. Academic Press, 1996.
- [35] F. W. DePiero and M. M. Trivedi. 3-D computer vision using structured light: Design, calibration, and implementation issues. *Advances in Computers*, 43:243–278, 1996.
- [36] J. Diebolt and C. P. Robert. Estimation of finite mixture distributions through bayesian sampling. *Journal of the Royal Statistical Society. Series B (Methodological)*, 56(2):363–375, 1994.

- [37] J. Durbin and S. J. Koopman. *Time Series Analysis by State Space Methods*. Oxford University Press, 2001.
- [38] R.N. Edmondson, N. Parsons, S. Adams, and Y. Song. Putting plant growth in the picture - image analysis for computer assisted grading and crop tracking. *HDC News*, 128:16–18, 2006.
- [39] R.N. Edmondson, N. Parsons, S. Adams, and Y. Song. The measurement and improvement of robust bedding plant quality and the use of digital imaging for quality assessment. *HDC News*, 139:20–21, 2007.
- [40] D. L. Ehret, A. Lau, S. Bittman, W. Lin, and T. Shelford. Automated monitoring of greenhouse crops. *Agronomie*, 21:403–414, 2001.
- [41] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. *International Journal of Computer Vision*, 70(1):41–54, 2006.
- [42] A. E. Gelfand and D. K. Dey. Bayesian model choice: Asymptotics and exact calculations. *Journal of the Royal Statistical Society. Series B (Methodological)*, 56(3):501–514, 1994.
- [43] A. E. Gelfand and A. F. M. Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85(410):398–409, 1990.
- [44] C. A. Glasbey and G. W. Horgan. *Image analysis for the biological sciences*. John Wiley & Sons, Inc., New York, NY, USA, 1995.
- [45] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Prentice Hall, 3rd edition, 2008.
- [46] M. S. Grewal, L. R. Weill, and A. P. Andrews. *Global Positioning Systems, Inertial Navigation, and Integration*. John Wiley & Sons, 2nd edition, 2007.
- [47] W. E. L. Grimson. Computational experiments with a feature-based stereo algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(1):17–34, 1985.
- [48] Z. Gu, X. Su, Y. Liu, and Q. Zhang. Local stereo matching with adaptive support-weight, rank transform and disparity calibration. *Pattern Recognition Letters*, 29(9):1230 – 1235, 2008.
- [49] R. M. Haralick and L. G. Shapiro. *Computer and Robot Vision*, volume 2, pages 316–317. Addison-Wesley, 1992.
- [50] A. C. Harvey. *Forecasting, structural time series models and the Kalman filter*. Cambridge University Press, 1989.

- [51] T.D. Harwood and P. Hadley. Graphical tracking systems revisited: A practical approach to computer scheduling in horticulture. *Acta Horticulturae*, 654:179–186, 2004.
- [52] K. A. Heller and Z. Ghahramani. Bayesian hierarchical clustering. In *In International Conference on Machine Learning*, pages 297–304, 2005.
- [53] H. Hirschmuller and D. Scharstein. Evaluation of cost functions for stereo matching. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007.
- [54] H. Hoppe. Progressive meshes. *SIGGRAPH*, pages 99–108, 1996.
- [55] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Mesh optimization. *SIGGRAPH*, pages 19–26, 1993.
- [56] G. W. Horgan. The statistical analysis of plant part appearance – a review. *Computers and Electronics in Agriculture*, 31:169–190, 2001.
- [57] A. K. Jain, R.P.W. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.
- [58] A. K. Jain and D. Maltoni. *Handbook of Fingerprint Recognition*. Springer-Verlag New York, 2003.
- [59] A. H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, 1970.
- [60] A. D. Jepson and M. R. M. Jenkin. The fast computation of disparity from phase differences. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 398–403, 1989.
- [61] M. J. Jones and J. M. Rehg. Statistical color models with application to skin detection. *International Journal of Computer Vision*, 46(1):81–96, 2002.
- [62] J. Joyce. Bayes’ theorem. In E. N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Stanford University, 2008.
- [63] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.
- [64] S. Kaski and T. Kohonen. Winner-take-all networks for physiological models of competitive learning. *Neural Networks*, 7(6-7):973–984, 1994.
- [65] Y. Keller, A. Averbuch, and O. Miller. Robust phase correlation. In *17th International Conference on Pattern Recognition*, pages 740–743, 2004.

- [66] L. Kobbelt and M. Botsch. A survey of point-based techniques in computer graphics. *Computers & Graphics*, 28(6):801–814, 2004.
- [67] T. Kohonen. *Self-organizing maps*. Springer-Verlag, 3rd edition, 1997.
- [68] P. Koikkalainen and E. Oja. Self-organizing hierarchical feature maps. *International Joint Conference on Neural Networks*, 2:279–284, 1990.
- [69] V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. In *Proceedings of the 7th European Conference on Computer Vision*, pages 82–96, London, UK, 2002. Springer-Verlag.
- [70] D. Kong and H. Tao. Stereo matching via learning multiple experts behaviors. *British Machine Vision Conference*, pages 97–106, 2006.
- [71] T. Kostiainen and J. Lampinen. Self-organizing map as a probability density model. *International Joint Conference on Neural Networks*, 1:394–399, 2001.
- [72] C. D. Kuglin and D. C. Hines. The phase correlation image alignment method. *IEEE Conference on Cybernetics and Society*, pages 163–165, 1975.
- [73] S. Kukkonen, H. Kaelviaeinen, and J. P. Parkkinen. Color features for quality control in ceramic tile industry. *Optical Engineering*, 40:170–177, 2001.
- [74] S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.
- [75] M. Lhuillier and L. Quan. A quasi-dense approach to surface reconstruction from uncalibrated images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):418–433, 2005.
- [76] D. Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. Henry Holt and Co., Inc., New York, NY, USA, 1982.
- [77] D. Marr and T. Poggio. Cooperative computation of stereo disparity. *Science*, 194(4262):283–287, 1976.
- [78] L. Matthies, T. Kanade, and R. Szeliski. Kalman filter-based algorithms for estimating depth from image sequences. *International Journal of Computer Vision*, 3(3):209–238, 1989.
- [79] G. J. McLachlan and D. Peel. *Finite Mixture Models*. John Wiley & Sons, 2000.
- [80] R. Moe, N. Glomsrud, I. Bratberg, and S. Valso. Control of plant height in poinsettia by temperature drop and graphical tracking. *Acta Horticulturae*, 327:41–48, 1992.

- [81] A. Mullins, A. Bowen, R. Wilson, and N. Rajpoot. Bayesian surface estimation from multiple cameras using a prior based on the visual hull and its application to image based rendering. *British Machine Vision Conference*, 2007.
- [82] K. Nakayama, S. Shimojo, and G. H. Silverman. Stereoscopic depth: its relation to image segmentation, grouping, and the recognition of occluded objects. *Perception*, 18(1):55–68, 1989.
- [83] M-E. Nilsback and A. Zisserman. A visual vocabulary for flower classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1447–1454, 2006.
- [84] M-E. Nilsback and A. Zisserman. Delving into the whorl of flower segmentation. In *Proceedings of the British Machine Vision Conference*, volume 1, pages 570–579, 2007.
- [85] J.C. Noordam, J. Hemming, C. van Heerde, F. Golbach, R. van Soest, and E. Wekking. Automated rose cutting in greenhouses with 3D vision and robotics: Analysis of 3D vision techniques for stem detection. *Acta Horticulturae*, 691:885–892, 2005.
- [86] Assistant Secretary of Defense for Public Affairs. The pentagon: Facts & figures. <http://pentagon.afis.osd.mil/facts-area.html>.
- [87] Y. Ohta and T. Kanade. Stereo by intra- and inter-scanline search using dynamic programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(2):139–154, 1985.
- [88] C. M. Onyango, J. A. Marchant, and R. Zwiggelaar. Modelling uncertainty in agricultural image analysis. *Computers and Electronics in Agriculture*, 17(3):295–305, 1997.
- [89] A. V. Oppenheim, R. W. Schaffer, and J. R. Buck. *Discrete-Time Signal Processing*. Prentice Hall, 2nd edition, 1999.
- [90] W. H. Press. *Numerical recipes in C*, pages 545–546. Cambridge University Press, 1992.
- [91] L. Quan, P. Tan, G. Zeng, L. Yuan, J. Wang, and S. B. Kang. Image-based plant modeling. In *ACM SIGGRAPH 2006*, pages 599–604, 2006.
- [92] C. E. Rasmussen. The infinite Gaussian mixture model. *Advances in Neural Information Processing Systems*, 12:554–560, 2000.
- [93] C. P. Robert. Mixtures of distributions: Inference and estimation. In W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, editors, *Markov Chain Monte Carlo in Practice*. Chapman & Hall, 1996.



- [94] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, 2004.
- [95] D.B. Rowe. *Multivariate Bayesian Statistics: Models for Source Separation and Signal Unmixing*. Chapman & Hall, New York, 2003.
- [96] C. P. Sanders, P. A. DeBitetto, E. Feron, H. F. Vuong, and N. Leveson. Hierarchical control of small autonomous helicopters. *Proceedings of the 37th IEEE Conference on Decision and Control*, 4:3629–3634, 1998.
- [97] R.R. Schaller. Moore’s law: past, present and future. *IEEE Spectrum*, 34(6):52–59, 1997.
- [98] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3):7–42, 2002.
- [99] G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978.
- [100] W. Seidel, K. Mosler, and M. Alker. A cautionary note on likelihood ratio tests in mixture models. *Annals of the Institute of Statistical Mathematics*, 52(3):481–487, 2000.
- [101] M. C. Shin, K. I. Chang, and L. V. Tsap. Does colorspace transformation make any difference on skin detection? In *Proceedings of the Sixth IEEE Workshop on Applications of Computer Vision*, page 275, 2002.
- [102] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In *Autonomous robot vehicles*, pages 167–193. Springer-Verlag, New York, NY, USA, 1990.
- [103] T. Song and J. Speyer. A stochastic analysis of a modified gain extended kalman filter with applications to estimation with bearings only measurements. *IEEE Transactions on Automatic Control*, 30(10):940–949, 1985.
- [104] Y. Song, R. Wilson, R. Edmondson, and N. Parsons. Surface modelling of plants from stereo images. In *Proceedings of the 6th International Conference on 3-D Digital Imaging and Modeling*, Montreal, Canada, 2007.
- [105] W. Stallings. *Wireless Communications & Networks*. Prentice Hall, 2nd edition, 2005.
- [106] C. Sun. Fast stereo matching using rectangular subregioning and 3D maximum-surface techniques. *International Journal of Computer Vision*, 47(1-3):99–117, 2002.

- [107] P. Tan, G. Zeng, J. Wang, S. B. Kang, and L. Quan. Image-based tree modeling. In *ACM SIGGRAPH 2007*, page 87, 2007.
- [108] M. F. Tappen and W. T. Freeman. Comparison of graph cuts with belief propagation for stereo, using identical mrf parameters. *Proceedings of the 9th IEEE International Conference on Computer Vision*, 2:900–906, 2003.
- [109] K.K. Truong and R.M. Mersereau. Structural image codebooks and the self-organizing feature map algorithm. *International Conference on Acoustics, Speech, and Signal Processing*, 4:2289–2292, Apr 1990.
- [110] M. Valera and S.A. Velastin. Intelligent distributed surveillance systems: a review. *IEE Proceedings on Vision, Image and Signal Processing*, 152(2):192–204, 2005.
- [111] E. J. van Henten, J. Hemming, B. A. J. van Tuijl, J. G. Kornet, J. Meuleman, J. Bontsema, and E. A. van Os. An autonomous robot for harvesting cucumbers in greenhouses. *Autonomous Robots*, 13(3):241–258, 2002.
- [112] U. van Meeteren. Quality models in horticulture need product quality: A rare but challenging field of exploration. *Acta Horticulturae*, 456:175–188, 1998.
- [113] D. Vernon. *Machine vision: automated visual inspection and robot vision*. Prentice Hall, 1991.
- [114] J. Y. A. Wang and E. H. Adelson. Representing moving images with layers. *Image Processing, IEEE Transactions on*, 3(5):625–638, 1994.
- [115] G. Welch and G. Bishop. An introduction to the Kalman filter. Technical report, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, 1995.
- [116] G. F. Welch. *SCAAT: Incremental Tracking with Incomplete Information*. PhD thesis, University of North Carolina, Chapel Hill, 1996.
- [117] C. Wells. *The Kalman filter in finance*. Kluwer Academic, 1996.
- [118] N. Wiener. *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. The MIT Press, 1964.
- [119] C. Williams. A mcmc approach to hierarchical mixture modelling. *Advances in Neural Information Processing Systems*, 12:680–686, 2000.
- [120] R. Wilson. MGMM: Multiresolution Gaussian Mixture Models for computer vision. In *IEEE International Conference on Pattern Recognition*, pages 1212–1215, 2000.

- 
- [121] R. Wilson and H. Knutsson. A multiresolution stereopsis algorithm based on the gabor representation. In *International Conference on Image Processing and its Applications*, pages 19–22, 1989.
  - [122] R. Wilson and C.-T. Li. A class of discrete multiresolution random fields and its application to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(1):42–56, 2003.
  - [123] H. Xu, N. Gossett, and B. Chen. Knowledge and heuristic-based modeling of laser-scanned trees. *ACM Transactions on Graphics*, 26(4):19, 2007.
  - [124] Q. Yang, L. Wang, R. Yang, H. Stewenius, and D. Nister. Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(3):492–504, 2009.
  - [125] H. Yin and N. M. Allinson. Towards the optimal bayes classifier using an extended selforganising map. In *International Conference on Artificial Neural Networks*, pages 45–49, 1995.
  - [126] Y. Z. Yu. Surface reconstruction from unorganized points using self-organizing neural networks. In *IEEE Visualization 99 Conference Proceedings*, pages 61–64, 1999.