University of Southern Queensland
Faculty of Engineering & Surveying

Advance Rate Measurement for Furrow Irrigation

A dissertation submitted by

Cheryl McCarthy

in fulfilment of the requirements of

ENG4112 Research Project

towards the degree of

Bachelor of Mechatronic Engineering

Submitted: October, 2004

Abstract

Furrow irrigation is used for 96% of the irrigated cotton crop area in Australia. The performance of furrow irrigation can be improved by measuring furrow irrigation advance rate, from which optimised values for irrigation parameters can be determined and implemented in subsequent irrigations. Most water detection methods for advance rate measurement use water sensors that are located in the furrow, and which must be tediously collected from the furrow after the irrigation takes place. Design of an alternate system for advance rate measurement has emphasised convenient, immediate recovery of advance data.

Machine vision was investigated as an approach for measuring advance rate for furrow irrigation. A high resolution digital camera at a fixed location at the end of the field monitors changes in field appearance during an irrigation. In the absence of a crop canopy, an unobstructed view of the advancing water front can be obtained from the end of the field. For more mature crops, under certain conditions, change in crop canopy appearance during an irrigation also travels visibly down the field.

The digital camera has been interfaced with a laptop so that images can be captured, downloaded and analysed during the irrigation. Field trials indicate that a single camera can detect objects for distances up to 500 metres with an accuracy of $\pm$5 metres, by use of optical zoom and by increasing the height of the camera. Experiments reveal significant tonal response of dry crop plants to water within twenty minutes of application of water. The image analysis involves two separate approaches for the bare soil and closed canopy conditions.

Temperature sensing of soil and crop canopy temperature was considered in conjunction with the digital camera advance front detection, however due to budgetary constraints thermal response of the field could not be investigated for this project.

University of Southern Queensland

Faculty of Engineering and Surveying

ENG4111/2 *Research Project*

Limitations of Use

The Council of the University of Southern Queensland, its Faculty of Engineering and Surveying, and the staff of the University of Southern Queensland, do not accept any responsibility for the truth, accuracy or completeness of material contained within or associated with this dissertation.

Persons using all or any part of this material do so at their own risk, and not at the risk of the Council of the University of Southern Queensland, its Faculty of Engineering and Surveying or the staff of the University of Southern Queensland.

This dissertation reports an educational exercise and has no purpose or validity beyond this exercise. The sole purpose of the course pair entitled "Research Project" is to contribute to the overall education within the student's chosen degree program. This document, the associated hardware, software, drawings, and other material set out in the associated appendices should not be used for any other purpose: if they are so used, it is entirely at the risk of the user.

Prof G Baker

Dean

Faculty of Engineering and Surveying

Certification of Dissertation

I certify that the ideas, designs and experimental work, results, analyses and conclusions set out in this dissertation are entirely my own effort, except where otherwise indicated and acknowledged.

I further certify that the work is original and has not been previously submitted for assessment in any other course or institution, except where specifically stated.

CHERYL MCCARTHY

00111214729

Signature

28-10-04

Date

Acknowledgments

Thank you to my family for their support during the year and throughout my degree.

The guidance of my project supervisors, Mark Phythian and Joseph Foley, is much appreciated.

Valuable field experience was obtained from the Hopsons in Jondaryan and the Griffiths in Dalby.

CHERYL MCCARTHY

University of Southern Queensland
October 2004

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Furrow irrigation is used for over 90% of Australia's cotton industry and a large proportion of Australia's sugar industry. The National Centre for Engineering in Agriculture (NCEA) is developing tools for the optimisation of furrow irrigation, such as through mathematical simulation using the models Infilt and SIRMOD.

Infilt and SIRMOD require the advance rate for a furrow irrigation event to be known. This project aims to develop instrumentation for the measurement of advance rate which enables simpler collection of advance data than existing processes.

1.1 Broad aims and specific objectives

The project aims to create and test a transportable mechatronic device that will measure irrigation advance rate in adjacent furrows and will enable convenient recovery of measurements immediately after or while irrigation takes place. Emphasis has been placed on saving labour and time requirements for equipment installation and data collection from the field after an irrigation event.

The system is intended to be suitable for use in the place of the 'advancemeters' of the type currently employed by the Irrimate system.

Specific objectives of the project are:

1. Observe the characteristics of furrow irrigation and the role of advance rate measurement.

2. Research existing methods of measuring advance rate in furrows.

3. Investigate methods of detecting water along several furrows simultaneously taking into account variations across farms such as length of furrow and maturity of crop.

4. Design a measurement system to detect the time at which the irrigation water reaches, to within 5 metres, at least five specified positions along a furrow 300-1000 metres long, and capable of being extended to measure advance rate simultaneously across at least eight furrows.

5. Design a supporting structure if necessary and setup procedure that allows transportability of the chosen measuring system.

6. Provide a method by which advance rate measurements can be stored and presented in a format compatible with Irrimate, within a short period after irrigation takes place.

7. Construct a prototype and evaluate its performance under simulated conditions and, where possible, on the field.

1.2 Overview of the dissertation

This dissertation is organised as follows:

Chapter 2 describes the features of furrow irrigation relevant to advance rate measurement, and the purpose of measuring advance rate for furrow irrigation.

Chapter 3 discusses the existing methods of measuring advance rate and theoretical solutions. Literature regarding the use of machine vision in plant stress identification is reviewed.

Chapter 4 develops possible alternative approaches to measuring advance rate for furrow irrigation, particularly by non-contact sensing methods.

Chapter 5 investigates the merit of the conceptual design of using crop canopy appearance as an indicator of water status.

Chapter 6 compares possible features and orientations of an image acquisition device to monitor a furrow-irrigating field. This includes a description of the construction of towers used to gather data on resolution achievable by a single camera.

Chapter 7 describes considerations for the analysis of acquired images, for both the bare soil and crowded canopy situations. Analysis involves identifying the position of the advance front in acquired images and correlating that position with a distance.

Chapter 8 discusses the interfacing achieved between a PC and hardware components (such as the high resolution digital camera) to create an automated advance rate measurement system.

Chapter 9 concludes the dissertation and recommends further research into crop responses and the extension of the system to real-time furrow irrigation control.

Chapter 2

Background

2.1 Furrow irrigation in Australia

Seventy percent of Australia's extracted water usage is attributed to agriculture each year, with 96% of Australia's cotton farms being irrigated by furrow irrigation (Spragge 2002). Other crops that are furrow irrigated in Australia include sorghum, sunflowers and grain crops.

On the Darling Downs efficiencies of furrow irrigated cotton farms have been measured as ranging between 30 and 95% (Spragge 2002). Optimising irrigation management practices can improve the lower water efficiencies by up to 30% (Purcell 2004).

2.2 What is furrow irrigation?

Furrow irrigation is a type of surface irrigation, where surface irrigation is the flow of water on a field aided only by gravity and soil surface characteristics of the field (no pumps). In furrow irrigation the field is sloped and corrugated, and the corrugations are called furrows (Figure 2.1).

Figure 2.1: Cotton farm at Jondaryan, early January 2004

Rogers (1995) describes that on the basis of water losses, furrow length should not exceed 180 metres on sand soils and 400 metres on clay soils, but that on some low intake soils, furrow lengths of 800 metres are suitable.

Figure 2.2: Head ditch of a cotton crop at Jondaryan, early January 2004

In furrow irrigation water is pumped from the farm's reservoir or dam to the head ditch of a field, from which water is siphoned to a group of furrows (Figure 2.2). The number of furrows that are irrigated during a particular irrigation event is called the set size. Inflow rate (or flow rate) to the set is influenced by variables including the number of furrows in the set and the diameter of the siphon pipes.

Run time is the time the water takes to flow from the top of the furrow to the end and is typically between eight hours and 36 hours. Furrow spacing depends on crop type and for cotton is typically one metre.

Figure 2.3: Tail end of a cotton crop at Jondaryan, early January 2004

Tailwater is the water that reaches the lower end of the field (evident in Figure 2.3) and can be returned and reused.

Irrigation performance can be measured in terms of application efficiency, distribution uniformity and requirement efficiency. Application efficiency is defined as the proportion of water finally stored in the crop root zone, over that quantity of water put into the field. Distribution uniformity is a measure of the consistency of the amount of water applied to the length of the field. Requirement efficiency is the proportion of the soil moisture deficit that has been replenished by the irrigation. A furrow irrigation event is optimised based on a selected combination of these irrigation measures.

Figure 2.4 illustrates poor uniformity along the run length. Waterlogging has occurred at the start of the furrow, where water has infiltrated beyond the crop root zone. To achieve better uniformity for this field (Figure 2.5) the flowrate and siphon run time need to be adjusted (Rogers 1995).

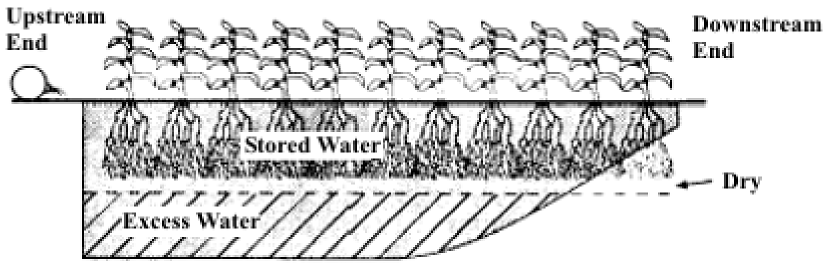


Figure 2.4: Poor uniformity - adjust stream size and set time (Source: Rogers 1995)

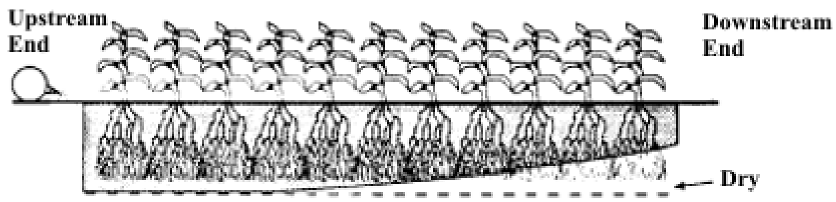


Figure 2.5: Ideal infiltration pattern (Source: Rogers 1995)

2.3 Improving furrow irrigation efficiency with advance rate

The front of the stream of water flowing along the furrow is the advance front (Figure 2.6). The rate at which the advance front progresses downfield is the advance rate. Advance rate for a particular furrow is usually measured as a series of data pairs of advance front position and time. Water depth and soil moisture are not required.

Advance rate, inflow rate, furrow shape and furrow slope are input parameters for a mathematical model of furrow irrigation. Simulation of this model yields current irrigation performance level and allows growers to optimise siphon inflow rate and irrigation time.

Optimisation of the irrigation parameters immediately after one irrigation event enables use of the optimised data in the next irrigation set.

Example software packages that model and simulate furrow irrigation are Infilt and

(a)

(b)

Figure 2.6: Advance stream during a prewatering on a cotton farm at Jondaryan, mid September 2004 (a) front view of advance stream (b) top view of advance stream

SIRMOD II respectively (Purcell 2004). This software has been developed by the NCEA (National Centre for Engineering in Agriculture) for use in the advance rate measurement system, Irrimate (see more in Chapter 3). Infilt yields the infiltration characteristic of a furrow irrigation event, which is then entered into SIRMOD to replicate the current irrigation as an indicator of irrigation performance. Alternate irrigations can then be simulated in SIRMOD to yield optimised flowrate and time to cutoff. The purpose of this project is to develop a measurement system which would be suitable to replace the ‘advancemeters’ of the Irrimate system.

2.4 Mathematical modelling of furrow irrigation

SIRMOD uses the Kostiakov-Lewis set of mathematical equations to model furrow irrigation, which was originally developed in the 1980s at the Utah State University (Purcell 2004). These equations require advance data for at least five locations down-field.

One- and two-point methods of modelling furrow irrigation have been developed, and these models have the advantage of not requiring the measurement and collection of many advance data points. Such methods are described in Shepard et al. (1993) and Elliott and Walker (cited in McClymont and Smith 1996).

Figure 2.7 features sample advance rate curves from the work of Walker and Busman (1990), where measured and modelled advance data have been plotted from a furrow irrigation event of duration seven hours and for furrows of length 300 metres.

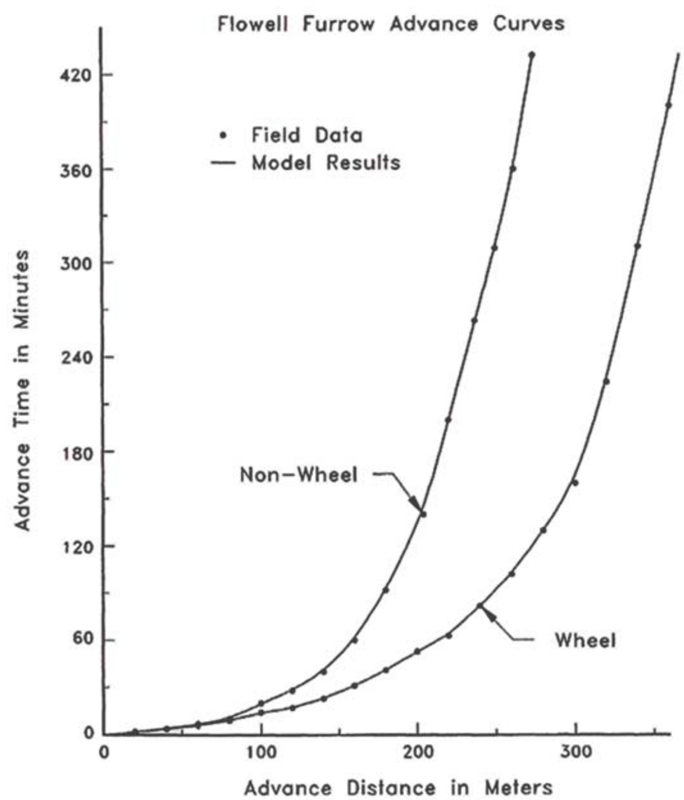


Figure 2.7: Example advance curves for various furrows (Source: Walker & Busman 1990)

Chapter 3

Literature Review

Most water detection methods for advance rate measurement use the principle of water short-circuiting two conductors that are electrically isolated before the irrigation takes place (Turnell et al. 1997).

Other approaches that the NCEA has theorised about include measuring the reflection time of a sound wave sent down the furrow stream, or tracking a beacon travelling with advance front by GPS. These suggested methods are sensitive to obstructions in the furrow such as twigs and leaves.

Determination of water stress is a common application for machine vision. Use of machine vision to measure advance rate is a natural extension.

3.1 Advance rate measurement systems

3.1.1 Irrimate

Purcell (2004) describes Irrimate, the currently implemented method of measuring advance rate in Australia that was developed by the NCEA. This method has been successfully implemented in farms in Queensland and New South Wales to improve irrigation application efficiencies by up to 30%.

The Irrimate water detection method uses an advancemeter that comprises a PIC chip and eight long pairs of open-ended wires that are tipped with gold pins. Each advancemeter is laid out across eight furrows (or four furrows for every other furrow irrigation) such that each pair of wire ends rests within a furrow. When the water stream for a particular furrow reaches the gold pins on the wire ends, a short-circuit is created and the time is recorded on the microcontroller.

To obtain five advance data points, five advancemeters must be installed downfield.

Download of data from the advancemeter occurs via infrared link to a Palmtop computer. Unfortunately this data transfer must occur shortly after irrigation has taken place, so the furrows are still wet when the advancemeters must be located by foot. This aspect makes collection of data a time consuming and messy task.

3.1.2 Ribbon cable sensor

A method of measuring advance rate that does not require laborious collection of data is a system developed by Turnell et al. (1997), which has reported satisfactory success and boasts a low-cost water detection method. The method has been developed for use in poor regions of the world such as Brazil, and for people with minimal technical expertise.

The water sensing element is a modified length of ribbon cable which changes capacitance when submerged in water. Several of these water sensors are placed in a furrow downfield. The sensors are connected by twin-wire line to a laptop at the end of the furrow. The laptop detects the change in capacitance in the ribbon cable.

3.1.3 A two-point advance measurement system

A system that uses conductive elements as the water sensor is described by Latimer and Reddell (1990). This system has been designed for a two-point method of furrow irrigation modelling but is mainly of interest because of the method that is used to communicate the data from the sensors.

Each water sensor is linked to an IR or RF telemetry field station, through which data is transmitted to a base station or control room. The infrared telemetry system is the less expensive transmission method and has a range of 800 metres, but requires line of sight access between field and base stations. The radio frequency telemetry system is limited to 400 metres for the no-permit transmission radio frequency (Federal Communication Commission, cited in Latimer and Reddell 1990).

This method, with only two sets of water sensors, evaluates to be an expensive alternative in relation to other advance rate measurement systems.

Chapter 4

Conceptual Investigations

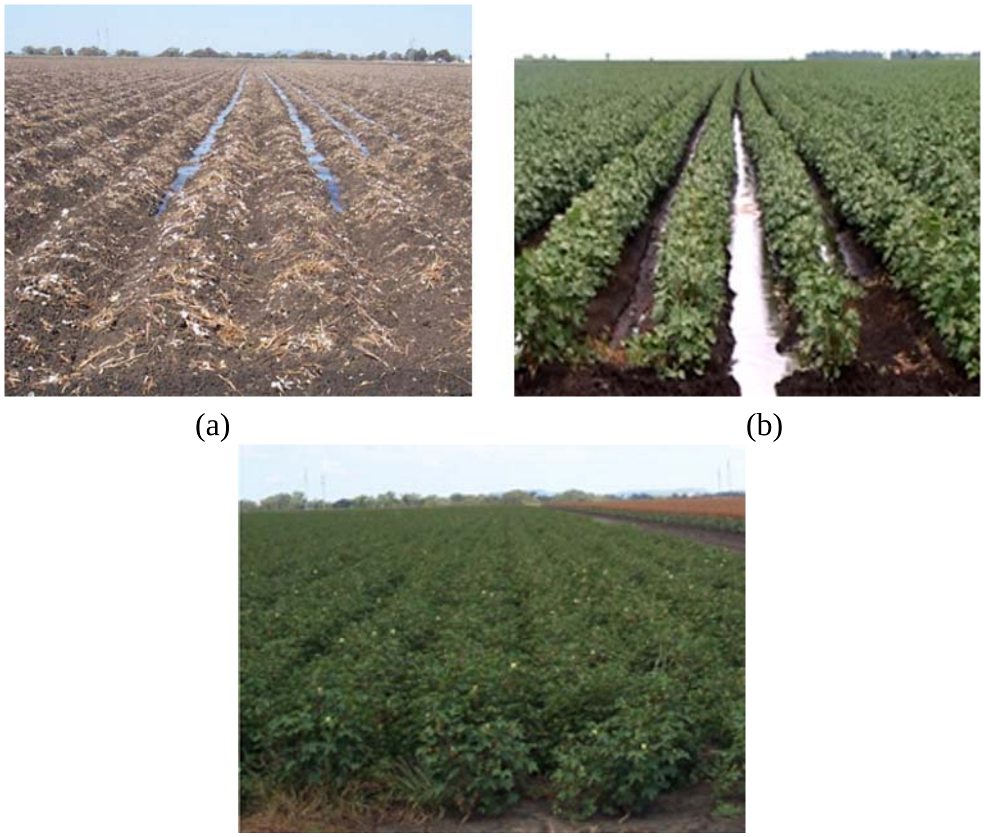
The need for continuing research into advance rate measurement for furrow irrigation arises from the common difficulty of the existing methods, which is the labour requirement involved with physically collecting the advance data, or sensing equipment, from the field once irrigation has taken place.

The desire to create an advance rate measurement system that requires minimal setup in the field heavily influenced the candidate systems considered. Machine vision emerged as a promising approach.

4.1 Machine vision of natural farm conditions

4.1.1 Development of concept

On the second watering of a furrow-irrigated cotton farm in early January, the observation was made that the advancing water could be seen from the end of the field (Figure 4.1(b)). Using image processing techniques, a camera taking snapshots at regular intervals would also be able to 'watch' the advance of water under these conditions. A more complicated solution is required when lighting conditions and maturity of crop are considered. Appropriate artificial lighting is expected to enable machine vision of the advance stream at night, however under all lighting conditions a crop canopy inhibits view of the advance stream.



(a) (b)

(c)

Figure 4.1: Various crop conditions (a) bare soil (b) crop and soil visible (c) closed canopy

On-site interviews with farmers revealed that under certain conditions, the advance front position could be determined by the appearance of the crop canopy, since the transformation in crop canopy during an irrigation is ‘like a wave’ travelling along the row. D Hopson (2004, pers. comm., 29 February), a cotton farmer at Jondaryan, claimed that crop canopy appearance changes visibly within twenty minutes of the application of water during a furrow irrigation event.

Researchers acknowledge that canopy appearance and temperature are indicators of plant water status (Murase 1997, Core 2002), however the focus of such past investigations has been to determine plant stress for the purpose of irrigation scheduling, where physiological characteristics of plants before watering are of principal concern. In direct contrast to previous research, advance rate measurement using physiological responses of plant leaves focuses on the time response of plant leaves after irrigation.

The difference in appearance of the foliage of freshly watered and dry plants is a familiar sight, for example in Figure 4.2, where two distinct regions of green are apparent in the field. This image was captured during an irrigation in January and demonstrates a brighter green in the region of field that has finished irrigating (below the dashed line), compared to the region of field which is commencing irrigating (above the dashed line).

Figure 4.2: Colour difference between irrigated and unirrigated regions of crop

Research of the colour change phenomena will investigate the time frame in which a discernible difference in canopy colour occurs following application of water, for the purpose of determining advance front position during an irrigation.

Soil colour is another characteristic of irrigating fields that changes upon application of water and is detectable by machine vision, however this feature is concealed in the presence of a closed canopy and hence unusable.

4.1.2 Literature review: Thermal imaging for determining water stress

Much literature exists on the use of temperature and thermal imagery as an indicator for water stress. Such research has commonly been focused on improving irrigation scheduling.

Water stress and soil moisture can be detected with remote sensing. Scott et al. (2003) compares various approaches for remote sensing of soil moisture, such as by using the microwave, visible and thermal infrared electromagnetic spectrums.

The use of infrared thermometers to measure canopy temperature for the purpose of irrigation scheduling has been in practice for decades (Alves & Periera 2000). Sadler et al. (2002) and Core (2002) describe methods of measuring canopy temperature using infrared thermometers mounted on a boom that pivots in the horizontal plane to focus on a specific region of the crop. The thermometers take readings from within one metre of the crop canopy.

Measuring advance rate with infrared thermometers follows naturally from water stress measurements. Magnitude of change in temperature for a freshly-watered crop depends on humidity, air temperature, wind speed and sunlight (Core 2002).

4.1.3 Thermal imagery equipment

Thermal cameras are capable of relaying the temperature of an object via an image, where each colour in the image indicates a particular temperature or range of temperatures. A thermal camera would enable investigation of the immediate thermal response of a crop canopy to the application of water from the advance stream.

Thermal cameras are difficult to acquire due to their expense (in excess of $10 000) and scarceness in Australia. An infrared thermometer is an alternative remote temperature sensing device.

A TM-908 infrared thermometer was obtained for conceptual investigations. A stepper motor controlled rig was developed (Appendix B) but the range of the thermometer was limited. The range of the thermometer was found to be improved by reducing the lens aperture size, such as by wrapping cloth tape over part of the lens. The thermometer has been considered throughout this report because the approach is promising yet restricted by budget.

4.1.4 Literature review: Crop canopy as an indicator of water status

Use of machine vision to measure advance rate aims to encapsulate the observations of Murase et al. (1997) who studied response of tomato seedlings to varying moisture conditions and stated that tonal characteristics of plant canopy substantially reflect plant moisture conditions.

Seginer et al. (1992) investigated the leaf wilt of tomato plants as an indicator of plant stress, with results included in Figure 4.3. This graph accentuates the degree of wilting of leaf tips in the days and part-days before a watering, but also exhibits an unquantified yet definite increase (and rate of increase) in leaf tip vertical displacement in response to water, during the periods of artificial lighting.

Continuous machine vision monitoring of the movement of individual crop plant leaves within an entire crop canopy is unfeasible. However Seginer's research indicates that a possible contributor to the change in colour of crop canopy following irrigation is the deflection of plant leaves. The theory that under certain conditions, a discernible difference in crop canopy occurs a short time following irrigation is reinforced by this research.

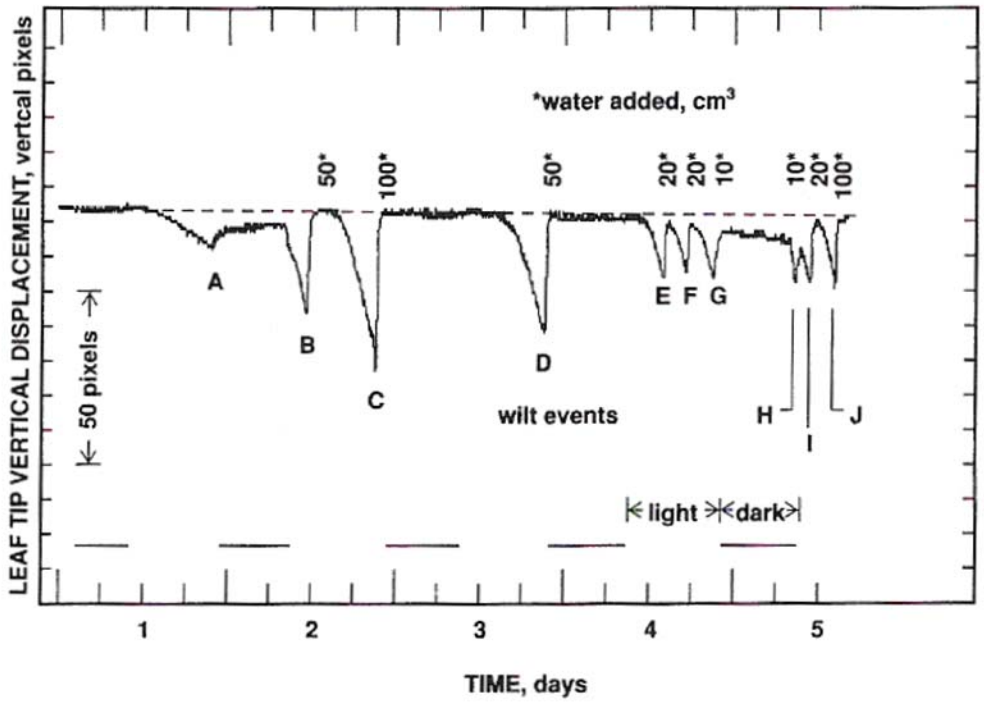


Figure 4.3: Leaf wilt displacement time dependence. Both the number of wilt events and amounts of water added are indicated. Horizontal bars indicate dark (lighting) periods. (Source: Seginer et al. 1992)

4.2 Machine vision of flags in the field

An approach that uses machine vision in conjunction with in-field ‘sensors’ involves mounting a camera at one end of the field and installing highly-visible flags or balloons within the field via water-soluble triggers. When the advance front reaches the balloon release mechanism, the balloon floats away, and this change in scene is detected by the camera. This approach is not dependent on plant physiological conditions (as is the previous method), but requires a hardy flag and flag release mechanism, and involves a laborious setup procedure before each irrigation. For five measurements in each of eight furrows, forty balloons are required to be installed within the field.

The system involves a much simpler intellectual problem for determining advance front position than observing plant response, however the system’s infeasibility prevented further investigation.

4.3 Furrow-trekking robot

The NCEA's theoretical ideas for advance rate measurement for furrow irrigation include a self-propelling device which is released at one end of the furrow, moving ahead of the advance front and recording displacement at regular intervals. The device finally emerges from the other end of the furrow at the end of the irrigation. Practical considerations which may cause the device to fail to reach the end of the furrow are trash, mud and cracking soils in the furrow, hence this method was not pursued.

J Billingsley (2004, pers. comm., 27 February) suggested that by connecting a cable between the end of the furrow and the furrow-trekking robot, the robot is more likely to emerge from the furrow. The cable rewinds to pull the device ahead of the advance front and to the end of the furrow.

Chapter 5

Plant Response to Water Status

The viability of inferring the advance front position from crop canopy response has been investigated through simple experiments, which are recounted in this chapter. The experiments aimed to determine the time frame in which a discernible change in plant foliage occurs after a watering, and the nature of the appearance change that occurs.

Within an entire crop canopy the tonal change is expected to be more pronounced than for the case considering individual crop plants, as in the experiments. The visible light response is considered, however multispectral scanning requiring specialist sensing equipment, is anticipated to be an important avenue for further investigation of plant response to water.

5.1 Plant response in advance rate measurement

Experiments were conducted to identify any significant, immediate change in appearance of stressed plant foliage following watering. The first experiment involved artificially created conditions, whereas the second experiment attempted to more accurately reflect the crop environment.

5.1.1 Plant response experiment using artificially cultivated plants

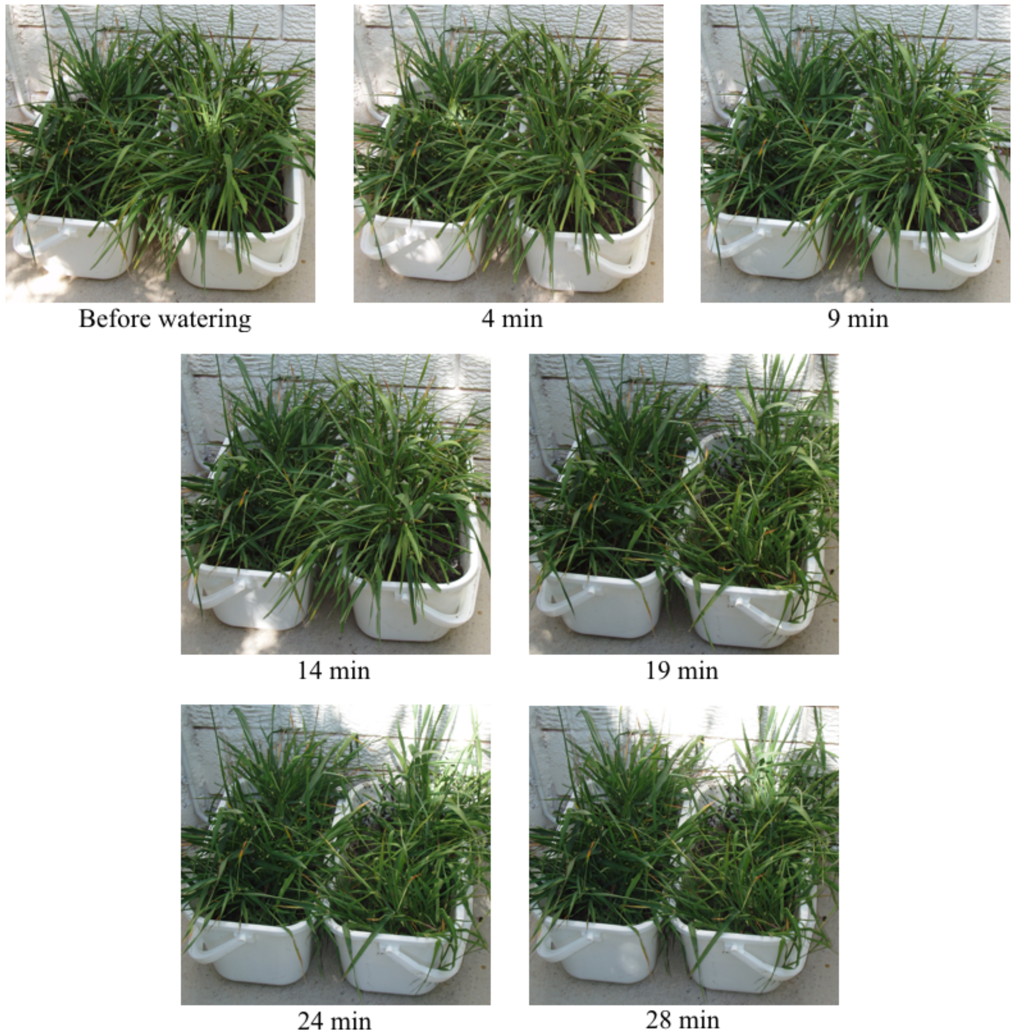
Millet seed was cultivated in two shallow dishes and watering withdrawn when the millet reached a height of approximately 20 cm, at which time the millet was very dense. After seven days without water both specimens of millet were noticeably duller in colour and water was applied to one of the plants. The immediate plant response to the watering was monitored by a camera capturing images of the two dishes at intervals of five minutes and under controlled lighting conditions.

Unfortunately analysis of the acquired images revealed that since the application of water was by spray to the foliage of the specimen, water droplets on the foliage reflected in the camera flash and caused significant interference in the images.

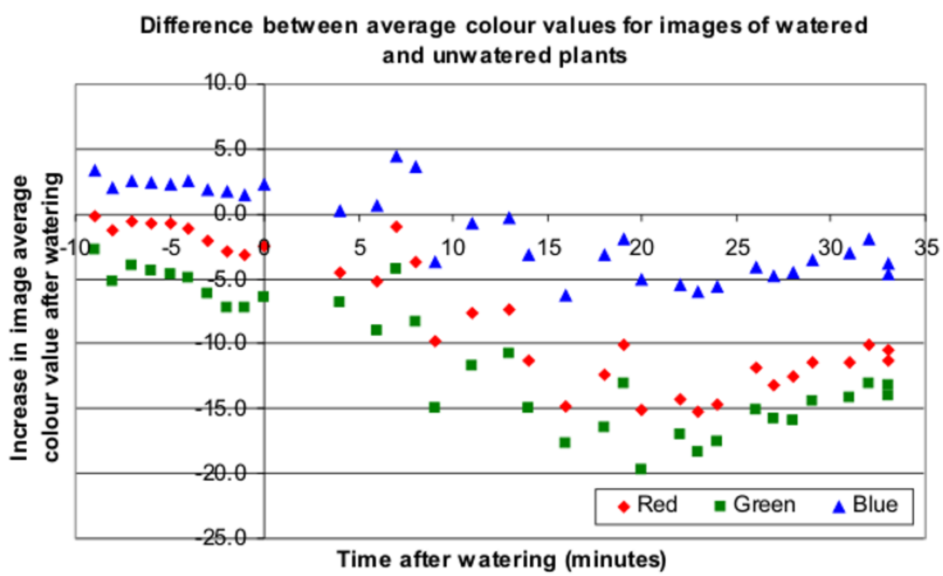
Necessary improvements to the experimental design include experiment authenticity, such as mode of application of water and location of experimental apparatus (the experiment was conducted indoors which is not an accurate indication of plant response in an outdoor enivironment).

5.1.2 Plant response experiment using crop plants

In late August, specimens of wheat plants were obtained from a farm near Dalby for the purpose of investigating the response of the plants to a watering. The plants were kept in a natural outdoor environment for one week before (the base of) one plant was applied with an amount of water equivalent to an irrigation watering of 100 mm. Images were captured of the plants at regular intervals before and after the watering (see Figure 5.1(a)).



(a)



(b)

Figure 5.1: Plant response experiment (a) sample images (plant on the right watered, time indicated is the elapsed time after watering occurred) (b) plot of response

The acquired images were cropped and the average red, green and blue channel values were found for each of the images, using a program written in Delphi. The obtained averages were then entered into a spreadsheet file to allow observation of any difference in average colour value that occurred for the plants following the watering.

Since the two specimens exhibited different colour tones even before the watering, plots of absolute plant colour did not offer suitable grounds for comparison (Appendix D features the plot of absolute plant colour). Similarly, each image exhibits varying amounts of ambient light, so a single area of each image would not depict a change in colour solely dependent on water response.

To overcome these issues, the difference in colour between the two specimens was plotted. Within each image, both plants are subject to the same lighting conditions, so the dry plant serves as a colour reference point. The plot of the response of the plants is included in Figure 5.1(b).

From the plot, the difference between the two specimens is approximately constant. Following the watering, the red and green channels of the specimen image differences exhibit variation in behaviour, whereas the blue channel responds to a lesser extent.

A numerical description of this tonal change, for use as a criterion for determining whether a plant has received water, is:

$$r_d > 5r_{d0} \tag{5.1}$$

$$\text{or} \quad g_d > 3g_{d0} \tag{5.2}$$

where r_{d0}, g_{d0} and b_{d0} are the initial image differences in the red, green and blue channels, and r_d, g_d and b_d are the corresponding differences at some later time.

This experiment indicates that the expected effect is a tonal change as a function of time (that is, over a series of images), as opposed to regions of ‘watered’ and ‘unwatered’ foliage discernible in a single image (as in Figure 4.2).

Repeat experiments, including experiments for different crop plants (such as cotton), are required to verify and compare results but are beyond the scope of this project.

5.1.3 Crop canopy response during an irrigation event

A series of images were captured during irrigation of a wheat crop in Dalby in the early afternoon of a day in June (Figure 5.2 features a sample image). During this irrigation substantial crop foliage was present.

The camera was mounted on a tripod 1.5 metres above the ground and images were taken at five minute intervals for a period of 35 minutes, in which time the advance front moved approximately 15 metres.

The height of the camera constrained the advance stream to twenty pixel rows for the series of images. The average colour of crop foliage neighbouring the advance stream was investigated and features in the test regions of Figure 5.2.

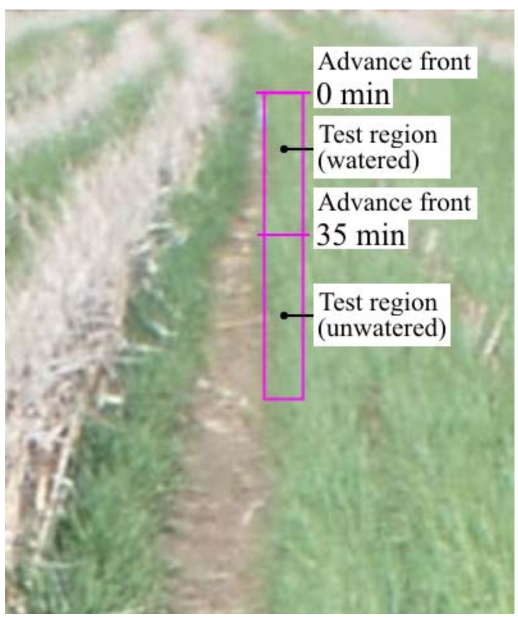


Figure 5.2: Wheat crop foliage during irrigation

Plots drawn from the average colour of the test regions as a function of time did not exhibit an identifiable effect (criteria 5.1 and 5.2 or otherwise) however the findings are by no means conclusive, as the experiments are required to be repeated for a longer duration and using a higher tower (to improve resolution).

5.2 Practical expectations for plant response

The magnitude and speed of plant response to a watering has been briefly investigated for wheat plants. Large variation in response is expected for other crops. The response may be influenced by time of day, for example mid-noon or night-time, and time of year, such as summer or winter. The experiments conducted here took place near midday on days in late winter.

Another consideration for response of crop foliage is indicated in Figure 5.3. In furrow irrigation where every furrow is being irrigated, each row is potentially being influenced by more than one advance stream. For example, in Figure 5.3 (a), the irrigated furrow F2 is adjacent to two rows, R2 and R3.

This is expected to be less of an issue for every other furrow irrigation, where alternate furrows are irrigated (Figure 5.3(b)).

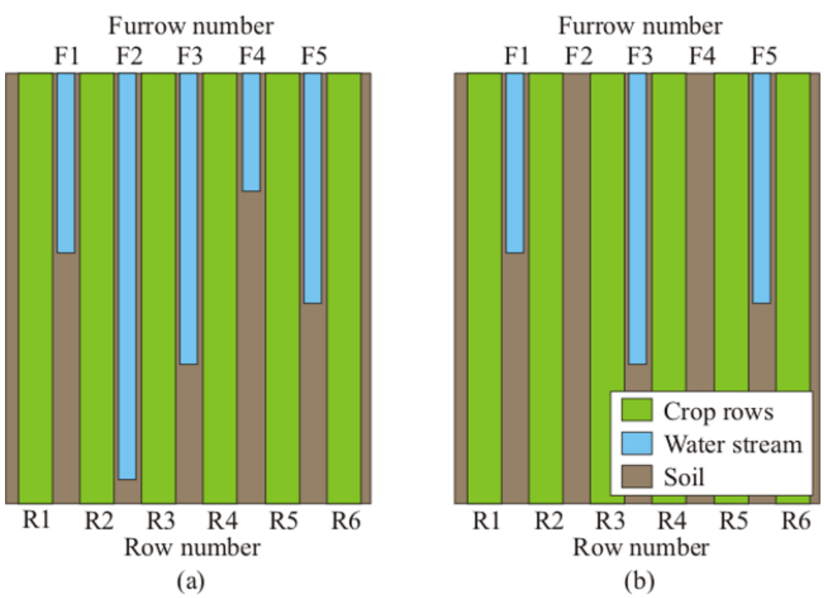


Figure 5.3: Top view of irrigating field (a) furrow irrigation (b) every other furrow irrigation

Chapter 6

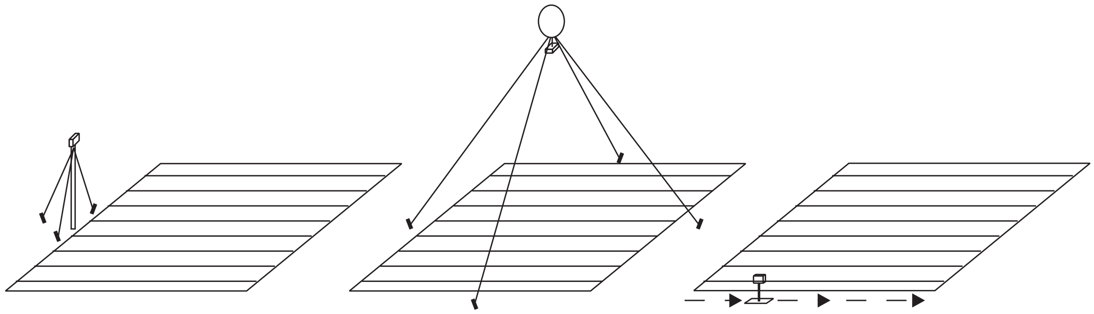
Crop Image Acquisition

A standard camera has been established as capable of capturing a crop plant's immediate response to water. Further considerations for a suitable camera to acquire for the research include resolution, cost, interfacing potential and capture mode (video or still). The requirement for these features is largely determined by the mounting position of the camera.

6.1 Mounting position of the camera

Several alternatives for position of the camera have been identified. These include mounting the camera on a tower at the end or in the middle of the field; on a balloon above the field; or on a motorised vehicle that travels alongside the field (Figure 6.1). Ease of setup, accessibility and quality of images obtained influence the most appropriate mounting position.

Tower at end of field. A tall tower or a high resolution camera must be used to obtain the required precision using this method since images acquired from the tower will exhibit one-point perspective, and the degree to which distant objects appear



(a) Tower at end of field (b) Balloon above field (c) Vehicle alongside field

Figure 6.1: Possible mounting positions of camera, relative to field

to converge will depend on the height of the camera. Variations to this approach include erecting a tower in the middle of the field, effectively halving the distance in any one direction that the camera must monitor, but also invalidating the system's 'remote sensing' tag. Alternatively, halving the required distance to measure can be achieved by installing a camera on both ends of the field.

Balloon above field. A top view of the field is obtained. Acquired images are downloaded wirelessly, however historically the majority of images taken from balloons are unusable due to movement of the balloon (J Foley, 2004, pers. comm., 12 July).

Vehicle alongside field. This approach restricts the areas of crop that can be monitored, since the vehicle must traverse alongside the edge of the field, parallel with the furrows. Compared to the other two approaches, a lower resolution video camera is possible since smaller distances are involved between the camera and the crop. Use of a mobile vehicle raises issues relating to vehicle reliability and accuracy, for example the rate at which the vehicle advances along the field is dependent on the perceived and anticipated position of the advance front.

Mounting the camera in a fixed position at one end of the field is the chosen approach due to the system's relative simplicity. A high resolution camera is required to monitor the long distances, however still images are sufficient. Interfacing of the camera is considered in Chapter 8.

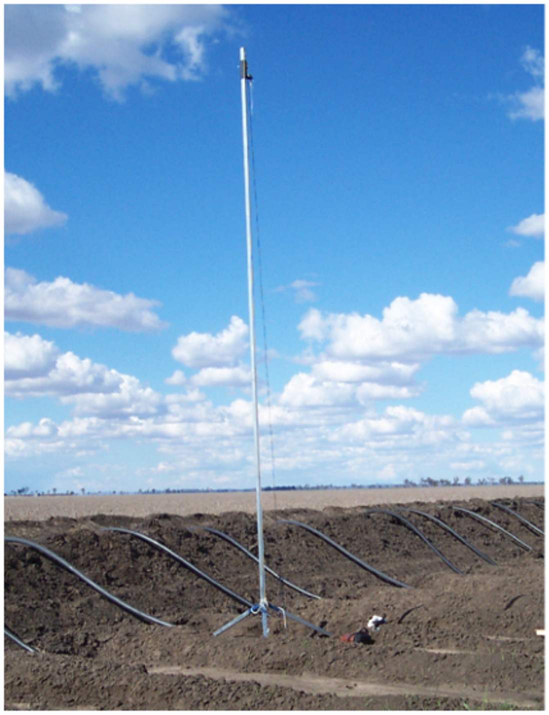
6.2 Construction and assembly of towers

Investigating the resolution of image captured from a single camera at one end of the field entailed constructing and testing a series of towers. For ease of storage and transport, the towers tested were all constructed of 1.2 metre lengths of square-section steel which pushed together onto a tripod base of the same make. The differences between each of the towers were both the height and method of assembly. Figure 6.2 illustrates three of the constructed towers.

Tower (a), 6.7 metres, was assembled on the ground from five lengths of steel, which was then lifted and fitted into the base that was ready in place on the ground. The camera and camera remote control were mounted at the top of the tower.

Tower (b), 14 m #1, consisted of the tower in two parts of length 6.5 and 11 metres, where the longer length was pin-jointed onto the rest of the tower. The tower was assembled by fitting the lengths of steel and tripod base together on the ground, pushing the tower and base upright, then using rope to pull the pin-jointed length of tower above the rest of the tower. The camera was mounted on top of the tower for the duration of the setup procedure. Guy wires were used on this tower, but no pixel distribution data was obtained, since as mentioned in Chapter 8, the camera remote control inexplicably and sporadically stopped operating when mounted on towers higher than seven metres.

Tower (c), 14 m #2, was an improvement on Tower (b) because a winch was used to rotate the pin-jointed length of tower up and down. With the camera mounted at the end of the pin-jointed length, the winch enabled simpler access to the camera. Multiple attempts at this experiment were conducted before the remote control operated successfully. Guy wires were required for this tower, and note the spotlight mounted midway along the tower in Figure 6.2. The spotlight is discussed in Chapter 7.3.5.



(a)

(b)

(c)

Figure 6.2: Sample towers trialled (a) 6.7 m (b) 14 m # 1 (c) 14 m # 2

The ten-metre tower used in later experiments was similar to tower (a) but consisting of extra lengths of steel, and the camera and remote control were mounted on a sleeve that fit around the tower. The sleeve could be raised and lowered along the tower via a pulley and string.

Resolution tests were also conducted using a 3.2 m tower, which was merely the camera mounted on a tripod, placed on top of the natural elevation of a head ditch.

6.3 Performance comparison of towers

6.3.1 Test of resolution

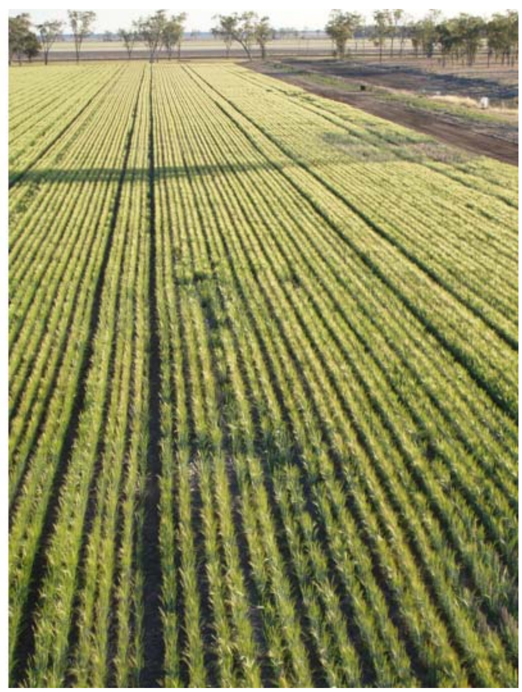
The process used to test image resolution was to capture images of a 1-metre square MDF board at ten-metre intervals along a furrow. From the acquired images the height of the top of the board (in pixels) was noted and plotted. Sample images captured by four towers (3.2 m, 6.7 m, 10 m and 14 m) are included in Figure 6.3. Images (a), (b) and (d) are from a barley crop in Dalby whereas image (c) is from a cotton crop in Jondaryan. The ten-metre tower is considered separately in Appendix E since experiments with this tower were conducted after the conclusion of the height experiments with the other towers.

The plots of Figure 6.4 have been constructed using data from the acquired images. Data for up to 500 metres are represented in the plots.

(a)

(b)

(c)



(d)

Figure 6.3: Sample images from the four towers (a) 3.2 metres (b) 6.7 metres (c) 10 metres (d) 14 metres

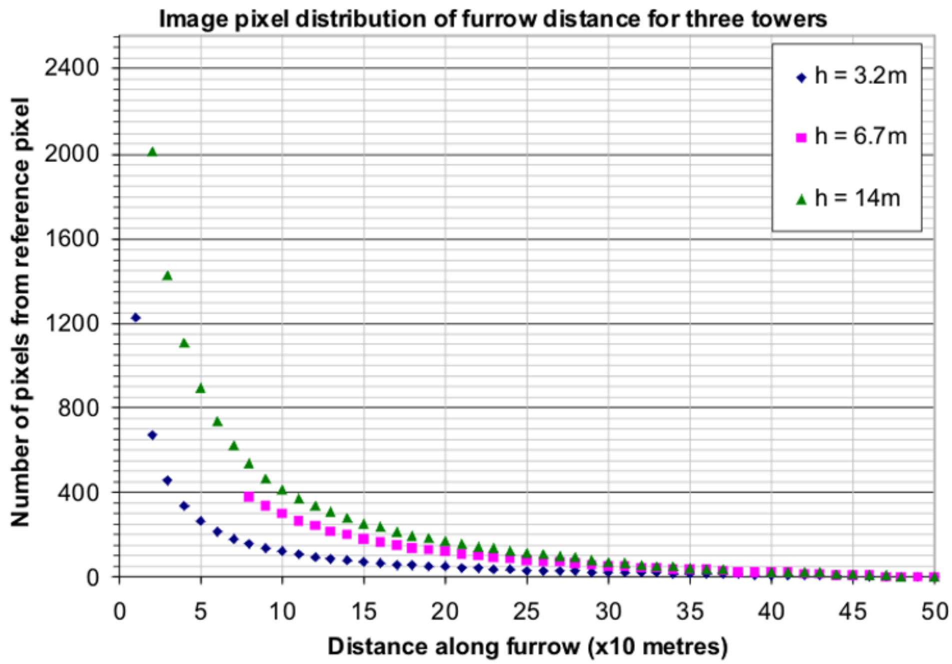
The first series of Figure 6.4, for the 3.2 metre tower, indicates a smooth curve since the day of the test was not windy and the tripod did not move on the head ditch.

The readings from the second tower (6.7 metres) were less smooth due to windy conditions on the day of the tests and subsequent movement of the camera. This movement was alleviated to an extent by observing for each image the pixel position of the board, as well as the pixel position of the horizon at the end of the furrow (a fixed object). The difference between the two pixel positions was plotted. The first eight metres are not included due to setup difficulties during this interval which resulted in substantial reorientation of the camera.

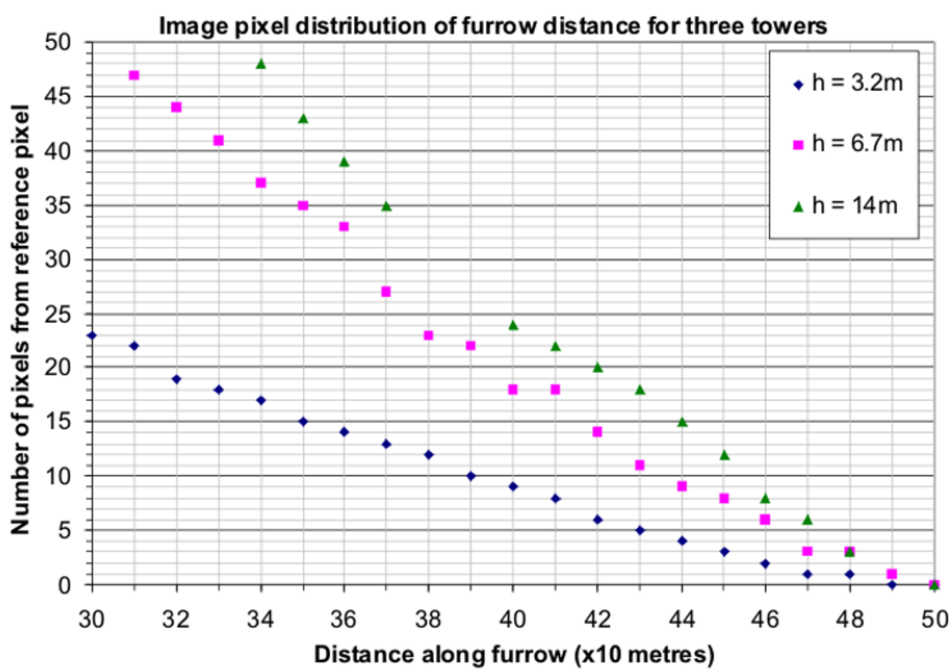
The third tower (14 metres) was supported by guy wires and was tested on a still day so no movement in the camera was observed through the acquired images. However some data points have been omitted due to intermittent excessively bright lighting conditions (the experiment was conducted over a thirty-minute interval in late afternoon) which caused small objects in the acquired images to be indistinguishable. This situation will be alleviated in software by a check of lighting conditions before conducting an image analysis to determine advance front position.

Maximum resolutions for these towers have been estimated by observing the last distance at which one pixel covers ten metres. Where data points are not smooth, an estimate was made of the maximum resolution based on the overall trend of the data points. The resolutions apply to a five megapixel camera and to objects of size 1-metre square (comparable to the advance stream and crop row widths). Further dividing the pixel axes of Figure 6.4 allows estimation of the maximum distance measurable to within ten metres by higher resolution cameras.

The 3.2 metre tower has a maximum resolution of 475 metres ± 5 metres. The 6.7 and 14 metre towers both consist of at least one pixel in the distance range of 490 to 500 metres, so the resolution of these towers is at least 500 metres ± 5 metres. From the plots of Appendix E, the resolution of the 10 metre tower is at least 480 metres $\pm$ 5 metres (since the range 460 to 480 metres is described by four pixels).



(a)



(b)

Figure 6.4: Resolution from three towers (a) for 0 to 500 metres (b) for 300 to 500 metres

6.3.2 Practical considerations for towers

The taller towers yield higher resolution but complexity of system setup is increased. The towers are all collapsible for ease of transportability and storage, but if the towers are not disassembled before relocation (for example, when the tower is mounted on the tray of a truck) caution for powerlines must be heeded.

A more practical solution for elevating the camera to large heights is to mount the camera on a cherry picker or hydraulic tower.

6.4 Methods of increasing camera resolution

In these experiments a five megapixel camera has been used, as set by budgetary constraints. A greater resolution camera will enable more accurate readings for longer distances. At the time of market research, a 12 megapixel camera was the highest existing resolution. A 12 megapixel camera is not expected to yield a marked improvement in resolution, particularly since the pixel distribution for distance is five pixels wide for the last 50 metres of a 500 metre-long furrow, and the increase in the number of pixels along the length of the image is 1.56 (calculated from $4000 \div 2560$, as indicated in Table 6.1).

Table 6.1: Comparison of high resolution digital cameras

Number of pixels ($\times 10^6$)	Resolution	Aspect ratio
5	2560×1920	4:3
12	4000×3000	4:3

Other methods of improving resolution include the following practices:

- The camera was mounted in the portrait orientation, reducing the number of furrows viewable in the image but increasing the number of pixels along the length of the featured furrows.

- The camera was tilted and optical zoom employed so that the acquired image featured the field for a majority of the image (that is, the sky featured in a small proportion of the image). Digital zoom is not recommended since digital zoom involves stretching pixels at the centre of the image and discarding pixels at the edges, reducing image resolution. Optical zoom adjusts the camera's focal length to obtain more pixels at the centre of the image.

- Further improvements in resolution may result by zooming in on regions of the field, or by reducing the length of field featured in a single image. This approach was not investigated since computer control of the camera zoom function has not been achieved, and a fixed zoom of a single camera on one region of the field would result in absence of data for the other regions of the field.

Chapter 7

Image Analysis

Due to the timescale involved (one photo every half hour at a nominal maximum), computational time is not a limiting factor so there is no requirement to condense the information stored in each field image.

As suggested in Chapter 4, there are several states of crop maturity in which an irrigation usually occurs (refer Figure 4.1) that the image analysis algorithm must handle:

1. No crop present
2. Crop and soil visible
3. Closed crop canopy

The situations in which the water stream is visible require different analyses from when the crop canopy is closed. During development of the image analysis software a useful tool was to apply effects to acquired images manually in Corel Photopaint. This enabled visualisation of the effectiveness of candidate image analysis techniques before coding specific procedures in Delphi.

7.1 Identification of furrows in acquired images

To determine advance rate in separate adjacent furrows, the location of the furrows in the image must be known. The installer of the system is required to identify and input the location of the furrows and the number of furrows into the software during setup.

Automating this process is only possible when distinct regions of crop and soil are visible in an image of the crop, or when an irrigation has begun and the water streams are visible. For example, a technique described in *Robotics and Machine Vision* (USQ 2004) identifies furrows by the transition of dark to light to dark, from dark soil to emergent crop and back to dark soil. This condition will only occur for the emergent and second irrigations.

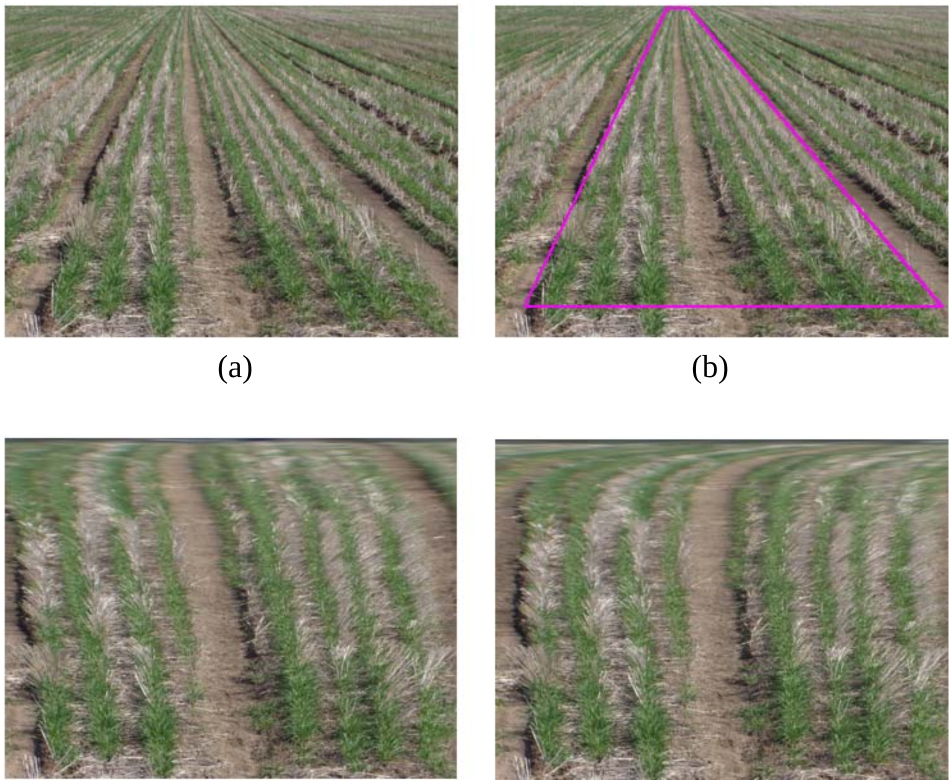
Two approaches have been considered for identifying furrow location, assuming the terminating positions of the furrows are known. The aim is that any pixel in the image can be selected and attributed to a particular furrow. The methods are by horizontal stretch and flagged boundaries.

7.1.1 Horizontal stretch

This approach is based on fitting a rhombus to a group of furrows as they converge towards the horizon. If the top two corners of the rhombus are stretched so that the rhombus is transformed into a rectangle, each furrow occupies a vertical strip of the transformed image and the furrow in which any pixel lies is indicated by the pixel's horizontal position in the image. During setup of the system, the user is required to manually identify the desired rhombus on a sample image.

Analysis of the transformed image would involve scanning through each line of the image, where every n pixels depicts a separate furrow. The method produces extra pixels to analyse without adding new information, but enables direct and linear correlation between pixel and furrow location.

Since a linear stretch is being applied, the camera, start of the central furrow and end of the central furrow must be collinear (or the furrows are symmetrically located about the image centreline), which must be heeded during setup of the camera. Furrows which exhibit curvature cannot meet this condition. Improper camera setup results in unuseful stretched images such as in Figure 7.1.



(a) (b) (c) (d)

Figure 7.1: Horizontal stretch applied to crop images (a) original image (b) original image with rhombus superimposed over a group of furrows (c) image with 0.389 to 0.421 of original image width stretched (d) image with 0.372 to 0.396 of original image width stretched

The ratios of Figure 7.1 have been empirically determined. The warping at the top of the stretched images indicates that the original image is not suitable for the stretch method. In Figure 7.1(c) the ratios have been chosen so that the transformed image gently curves towards the top of the image, whereas the ratios of Figure 7.1(d) produce an image in which the furrows are vertical for a greater proportion of the image before dramatic curvature occurs at the top of the image. In both cases a pixel's horizontal coordinate is not linearly related to actual furrow location, especially at the top of the image.

7.1.2 Flagged boundaries

If a grid is overlaid on the field image and adjusted to fit the furrows, the grid can be merged with the original image as a permanent 'embedded' indicator of furrow location. The grid must be a distinct colour so that when each line of the image is scanned, each instance of that distinct colour indicates that the boundary of a furrow has been located. The process allows less stringent placement of the camera and relies on the system user to manually adjust the grid.

Assuming equal spacing, additional grid lines for furrows that terminate outside the bounds of the image can be extrapolated, allowing advance rate measurements for extra furrows to be taken. With further software development, this method is capable of being extended to curvilinear furrows and non-uniformly spaced furrows.

The process consumes several pixels of data in each row of the image (for example, n+1 pixels for an image row containing n furrows) however the crucial factor of resolution along the length of the furrow is not affected. Figure 7.2 illustrates this approach (the widths of the gridlines are exaggerated for illustrative purposes).

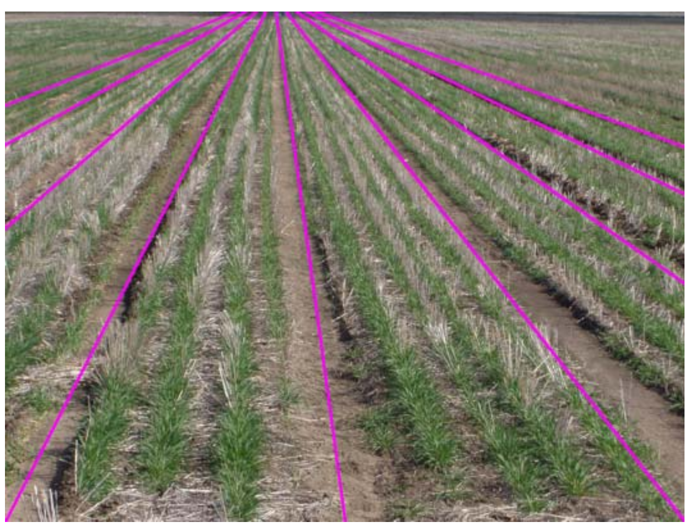


Figure 7.2: Example grid overlaid on crop image

7.1.3 Design of software setup procedure and practical considerations

The software setup process must be easy to understand and execute, and provide ample instructions and feedback for the installer, who in the most cases is anticipated to be a farmer.

During the developmental stages of the software, the location of the furrows was initialised by clicking four points on the image which encompassed a group of furrows and inputting the number of furrows in the selection. This process was deemed unsuitable for general use due to susceptibility to user input error, particularly in identifying and inputting the correct terminating position of the furrows. The process was changed to include a grid overlaid on the crop image, which could be adjusted and verified by sight using separate input buttons.

7.2 Determining distance from acquired images

Mounting an image acquisition device at one end of the field to measure advance rate has the advantages of user convenience and possible extension to real-time furrow irrigation control, but introduces the need to determine advance front position from a one-point perspective image. The perspective image features distorted distances such that objects converge to a single point on the horizon, thus the perceived position of the advance front in the image will have a non-linear correlation to the real advance front position.

7.2.1 Technical drawing model

In technical drawing a perspective image is constructed using the layout of Figure 7.3. This model approximates a real-life perspective image from a plan view of a scene and parameters such as height of the camera and distance between the camera and 'picture plane'. Use of this model to determine distance may involve 'reverse engineering' the perspective image to yield a plan view of the scene. This approach was not investigated in detail due to the complexity of the solution.

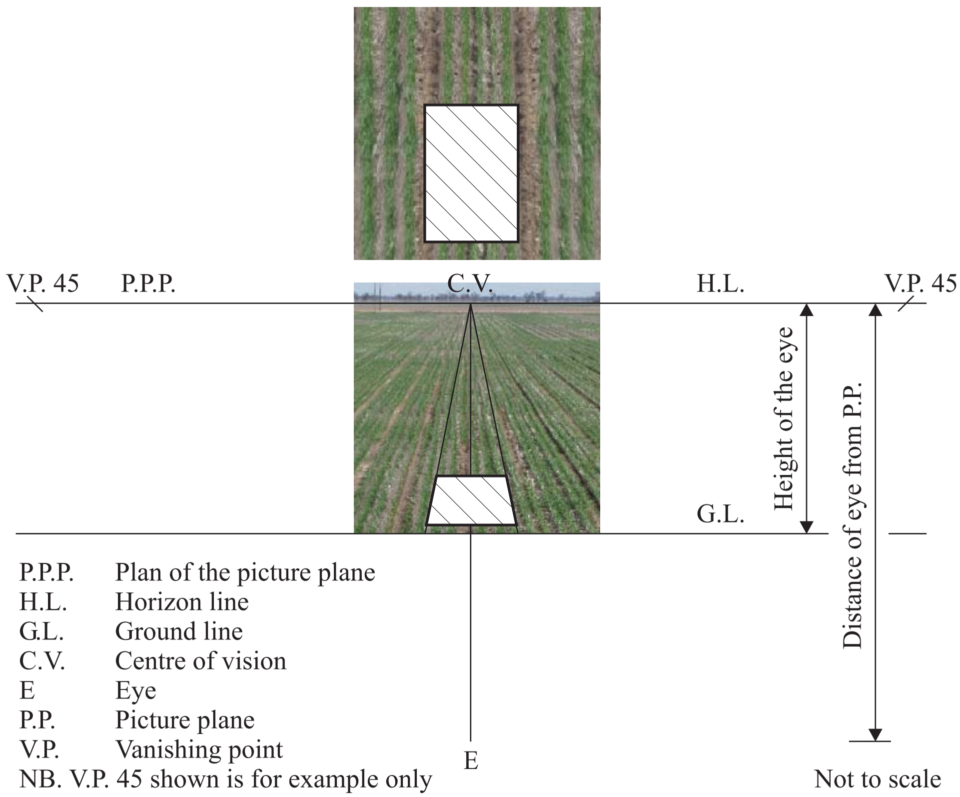


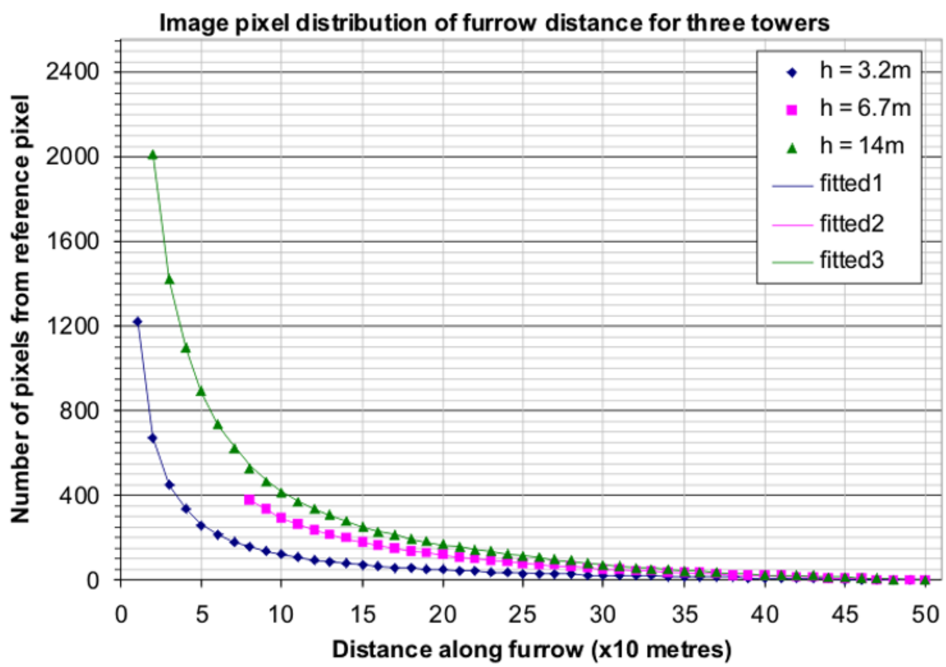
Figure 7.3: Perspective image: the plan method

7.2.2 Pixel position using experimental data

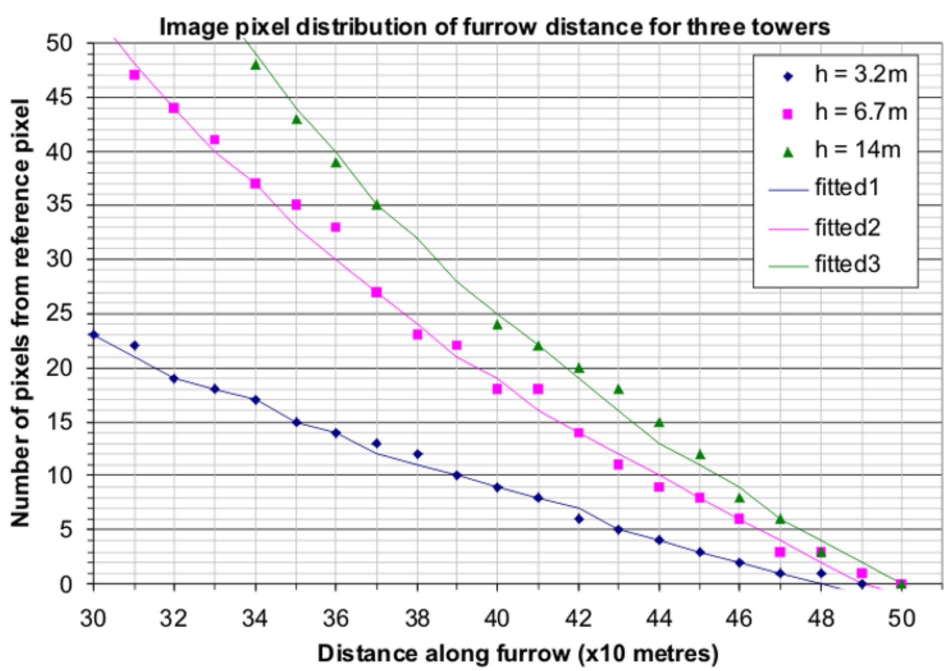
Advance front position can be determined from acquired images using pixel distribution data obtained by the experimentation described in Section 6.3.

The procedure to obtain these graphs was tedious however the curves exhibit similarity in shape, so further experimentation should yield a family of curves based on the height of the camera (and perhaps on the degree of tilting or zoom employed by the camera). Normalising the curves by considering pixel position as a percentage of the height of the image as opposed to absolute pixel position enables relationships for higher-resolution cameras to be extrapolated.

The curves may be more useful if their shape was described by a formula. By plotting the points in spreadsheeting software, a trendline was fitted to the points to yield possible formulae. Using this process, the curves of pixel position were found to be described very accurately by quartic polynomials in the natural logarithm of distance (see Figure 7.4).



(a)



(b)

Figure 7.4: Quartic polynomial curves fitted to pixel distribution data (a) for 0 to 500 metres (b) for 300 to 500 metres

The polynomials are (where y is the pixel number and x is the natural logarithm of the distance in decametres):

$$
\begin{aligned}
h = 3.2\text{m}: &\qquad y = 2.387x^4 - 46.34x^3 + 321.3x^2 - 1005x + 1226 \\
h = 6.7\text{m}: &\qquad y = 7.527x^4 - 114.3x^3 + 702.0x^2 - 2126x + 2653 \\
h = 14\text{m}: &\qquad y = 3.333x^4 - 82.06x^3 + 678.2x^2 - 2475x + 3431
\end{aligned}
$$

7.2.3 Pixel position using system geometry

With a camera mounted at a fixed height and capturing a known length of field, the system can be represented as displayed in Figure 7.5. Grade of the land has been considered negligible and the tower is assumed to be perpendicular to the ground.

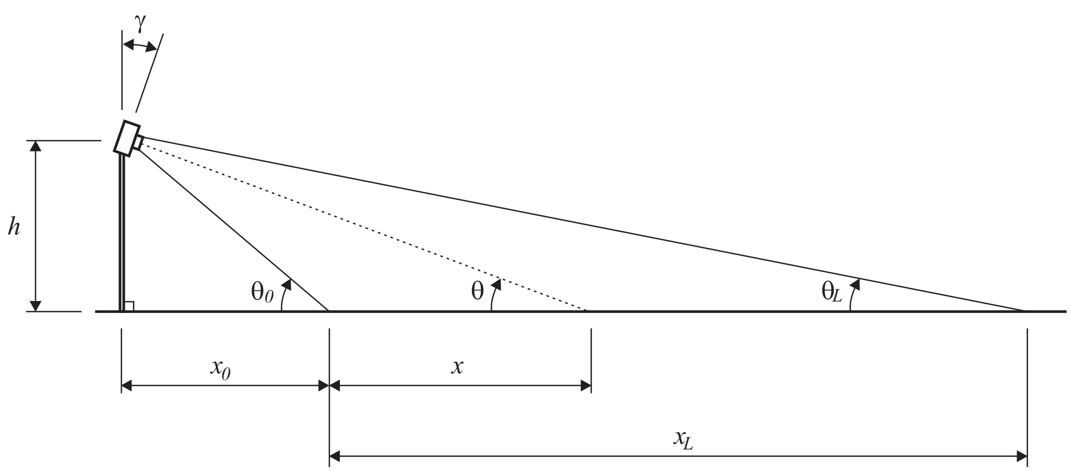


Figure 7.5: Geometry of camera and field

In Figure 7.5, the variables are as follows:

- x_L = distance captured in the image (metres)
- h = height of camera (metres)
- x_0 = distance between base of tower and start of image (metres)
- x = distance of interest in image (metres)

- θ = angle of elevation to camera, from distance x on the ground (radians). Similarly θ_0 and θ_L correspond to x_0 and x_L respectively.

- γ = tilt angle of camera (radians)

Neglecting the tilt angle γ, the relationship between θ and x is:

$$\theta = \arctan\left(\frac{h}{x_0 + x}\right) \tag{7.1}$$

To relate the angle of elevation θ to a pixel number px (where px_{MAX} is the number of pixels in the image that corresponds to the field):

$$px = \mathrm{ROUND}\left(\frac{\theta - \theta_L}{\theta_0 - \theta_L}\right) \times px_{MAX} \tag{7.2}$$

A comparison of inverse tan plots with the experimental data is included in Figure 7.6. Values of x_0, the distance between the base of the tower and the start of the image that was measured from the field trips, are 7.2 m, 17 m and 19 m for the 3.2 m, 6.7 m and 14 m towers respectively.

The curves reflect the data points and have a maximum error of 10 metres for the 3.2 m and 6.7 m towers. The fit is poorest for the 14 m tower. Sources of error include the simplifying assumptions that the tower is perpendicular to the ground and that the tilt angle of the camera is negligible. The required forward tilt angle of the camera increases as the height of the camera increases, which may explain why the inverse tan graph for the 14 metre tower is a worse fit than for the shorter two towers.

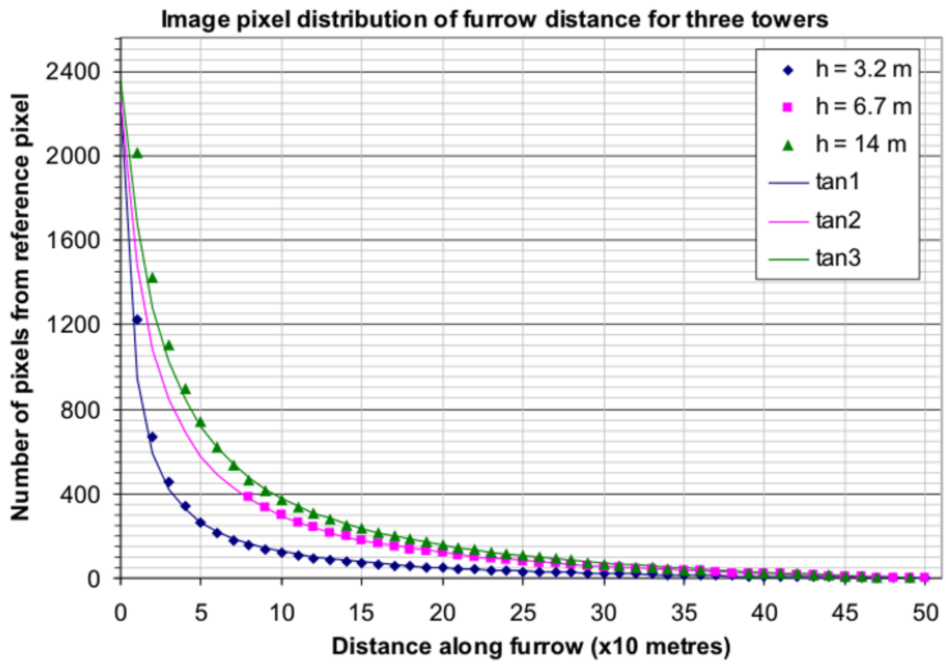
Rearranging Equations 7.1 and 7.2 yields

$$\theta = \frac{px}{px_{MAX}}(\theta_0 - \theta_L) + \theta_L \tag{7.3}$$

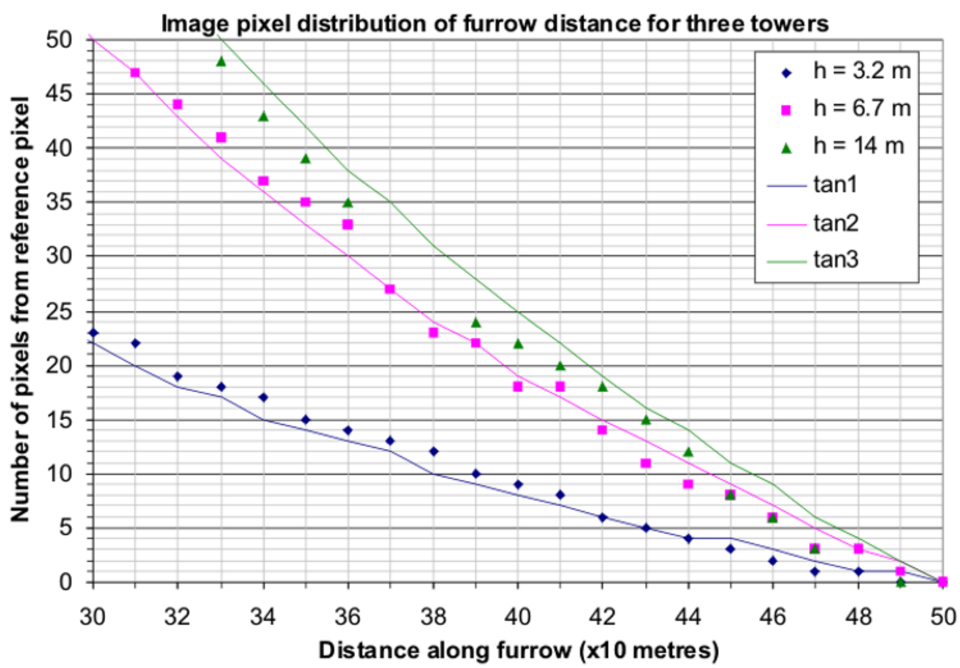
$$x = \frac{h}{\tan\theta} - x_0 \tag{7.4}$$

so that distance can be determined from a pixel position.

Knowing the length of field x_L that is captured in the image can be achieved by ensuring that the bounds of the captured length are defined by the tail drain and head ditch, or that the far end of the field is identified by the user during system setup (which is already incorporated into the furrow location process of Section 7.1.2). If this is not possible, a marker (such as a flag or post) could be installed at a known location in the field as a reference distance in acquired images.



(a)



(b)

Figure 7.6: Inverse tan curves fitted to pixel distribution data (a) for 0 to 500 metres (b) for 300 to 500 metres

7.3 Algorithm to identify colour change in crop and soil

7.3.1 Algorithm for bare soil case

Since water is highly reflective and contrasts strongly with bare soil, the location of water on a bare soil field can be determined from a single image, such as those in Figure 7.7. These images were captured during the prewatering and third irrigations of cotton and cereal farms, respectively. The irrigation events were conducted on fine days so the effect of cloud cover on the water stream could not be investigated. However the effect of early morning sunshine and early afternoon sunshine on the water streams can be observed in Figures 7.7(a) and (b).

Figure 7.7: Sample images from furrow irrigation events (a) prewatering in early morning (b) prewatering in early afternoon (c) third irrigation in early afternoon *Inset:* Magnified section of original images

These images were initially analysed in Corel Photo-Paint to determine which colour models and channels were most affected by the presence of the water stream. The colour models investigated were RGB, HSI and CMYK and the results are plotted in Figures 7.8 and 7.9 (the K channel of CMYK is not included). Conversions from the RGB colour model to the HSI and CMYK colour models are included in Appendix F. Once a colour channel has been determined in which the water streams are extenuated strongly, the image can be thresholded to discard the pixels that are not part of the water stream.

Figure 7.8, corresponding to Figure 7.7(a), features images from the early morning irrigation event, where the scene is relatively bright and the water streams appear to be white in colour. From this figure, the water streams appear most visible by visual inspection in all colour channels except the hue channel of the HSI colour model. Since the acquired images are RGB format by default, the colour model eventually used to discern the water stream for early morning images was a grayscale image from the RGB colour model (red, green and blue channels averaged, as in Figure 7.10(a)).

Images captured between mid morning and mid afternoon more closely resemble the image featured in Figures 7.7(b) and (c), where the colour of the water streams appears blue. Figure 7.7(b) has been separated into channels of the RGB, HSI and CMYK colour models in Figure 7.9. By visual inspection, the hue channel of the HSI colour model extenuates the water stream to the greatest effect. The contrast between the water stream and surrounding soil is not as marked for the other colour channels, particularly at the far end of the furrow.

A colour channel study of an irrigation featuring a crop canopy (Figure 7.7(c)) has been conducted and included in Appendix F. The image was captured in the early afternoon of an irrigation and the colour channel study yielded similar results from Figure 7.7(b), regardless of the crop canopy.

The chosen approach for analysis in the bare soil case is to convert images acquired in the early morning to RGB grayscale, and images from the middle of the day (for example 11 am to 4 pm) to the HSI colour model. Unfortunately late afternoon images have not been investigated however the results from such images are anticipated to be similar to the results from early morning images, since in both cases the water stream is expected to be bright relative to the rest of the field.

Timing for the images is as follows. Five advance points are required, and since the position of the advance front can be determined from a single image, only five images are required. These five images are captured at intervals specified in software by the user of the advance rate measurement system. The intervals are expected to be distributed throughout the irrigation.

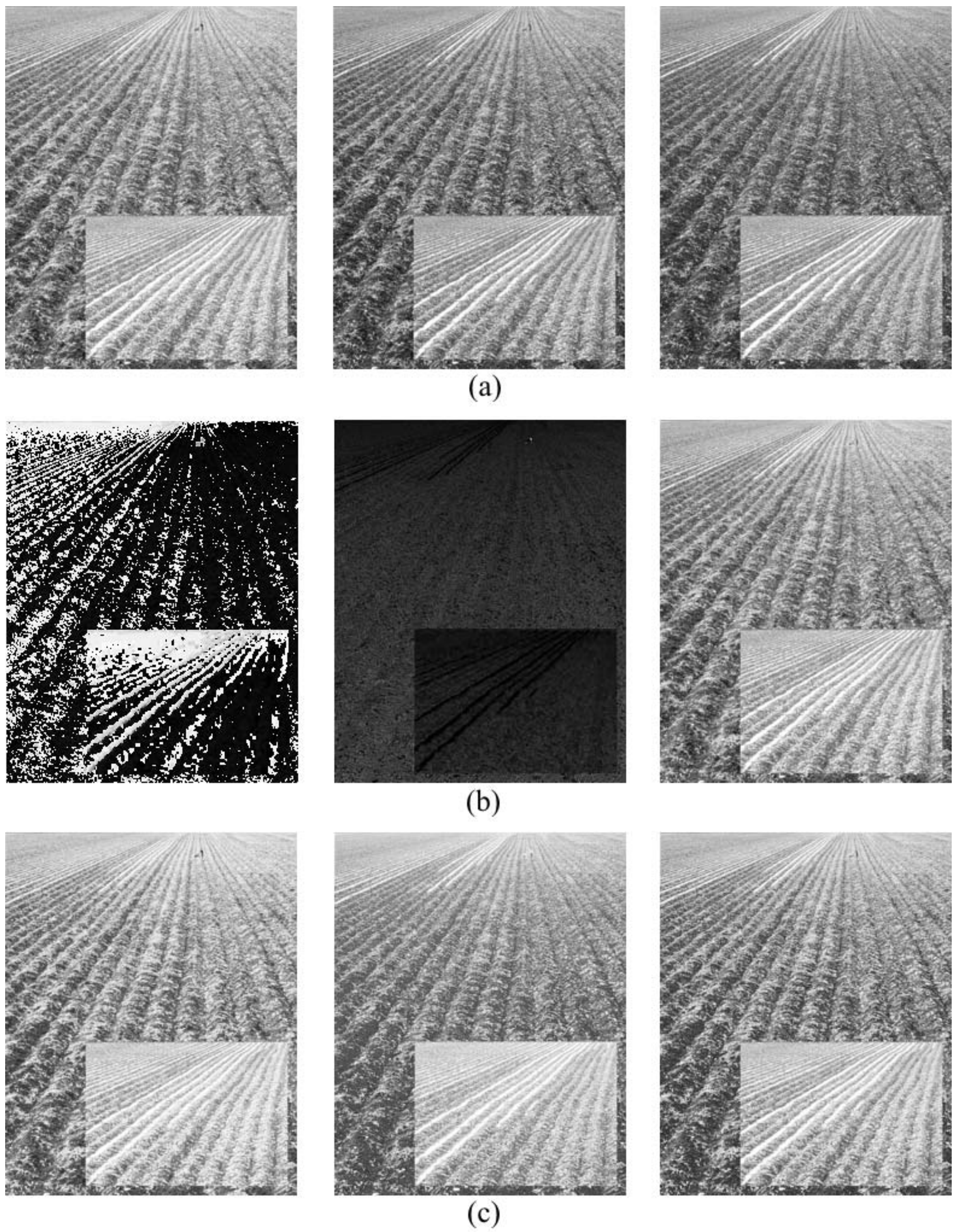


Figure 7.8: Sample images from furrow irrigation event in early morning separated into channels of RGB, HSI and CMYK colour models (a) red, green and blue channels (b) hue, saturation and intensity channels (c) cyan, magenta and yellow channels

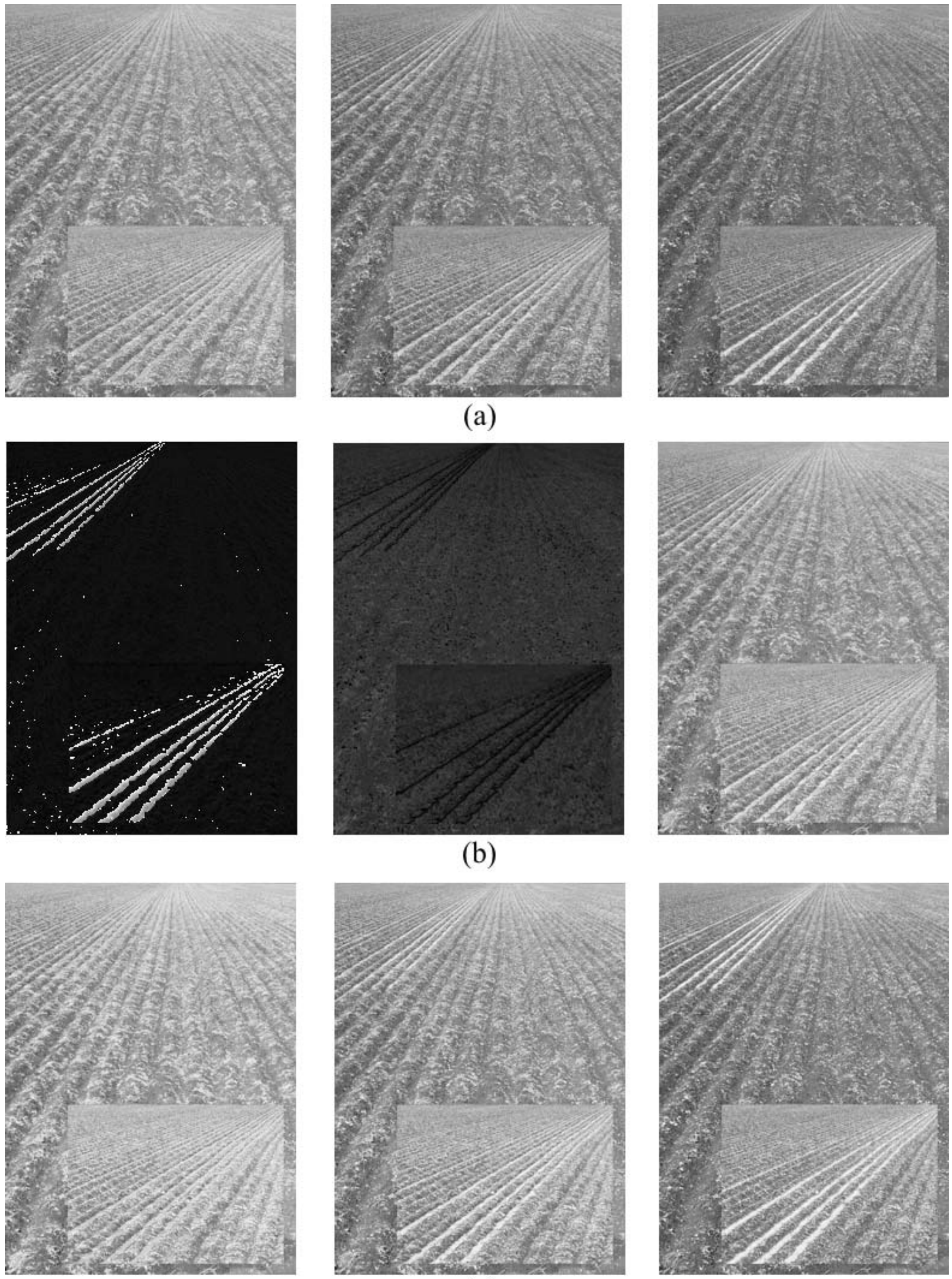
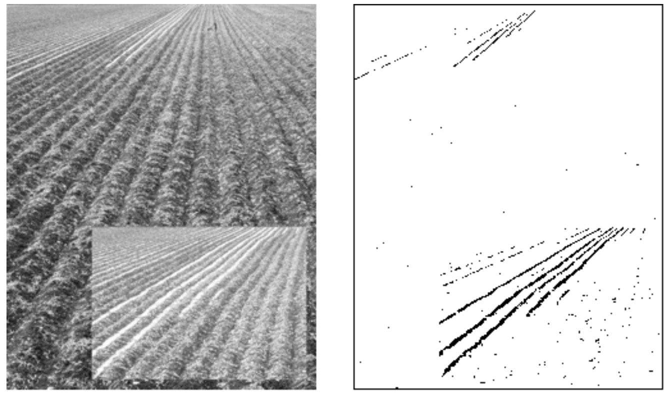

(a)
(b)
(c)

Figure 7.9: Sample images from furrow irrigation event in early afternoon separated into channels of RGB, HSI and CMYK colour models (a) red, green and blue channels (b) hue, saturation and intensity channels (c) cyan, magenta and yellow channels

Figure 7.10(b) exhibits a threshold value of 235 applied to the early morning irrigation image. From the thresholded image, the water streams are visible on a black



(a) (b)

Figure 7.10: Sample image from irrigation in early morning (a) converted to grayscale (b) thresholded at a value of 235 and inverted

background, where the water stream is represented by a pixel of value 1, and the soil is represented by a pixel of value 0. Some noise is apparent caused by reflections from trash in the field.

A median filter usually removes image noise but was not used since the process may degrade the edges of the water streams, particularly the tip or advance front. A median filter assigns a new value to each pixel of an image based on the average value of adjoining pixels.

Instead an approach was devised based on the water stream being a solid block of pixels of value 1, while the noise features pixels of value 1 amongst pixels of value 0. The plots of Figure 7.12 demonstrate the described effect. In these plots, the furrow numbers refer to the designations of Figure 7.11, which have been implemented using the flagged boundaries approach described in Section 7.1.2. The plots have been constructed by summing the number of pixels of value 1 for each pixel row of the furrow, where the x-axis is of the most interest. Along the x-axis, the end of the advance stream is indicated by the concentrated distribution of pixel rows that sum to zero following many consecutive pixel rows that sum to non-zero.

To detect this changeover position in software, the algorithm used was:

If there exists at least three pixel rows that sum to zero in any block of ten

Figure 7.11: Sample image from irrigation in early morning featuring superimposed grid, and with furrows numbered

pixel rows, the position of the advance front is given by the first pixel row that sums to zero in that block of ten pixel rows.

This algorithm will not detect advance streams encompassed by less than four pixels.

Sample results that compare detected advance front position with visually determined advance front position are displayed in Table 7.1. The distances obtained were calculated using the discussion of Section 7.2.3 in conjunction with the ten-metre tower data in Appendix E. Of these furrows, furrow numbers 2, 4-9 and 10 were being irrigated. Furrows 1, 3 and 11 did not contain water, while furrow 7 featured water that had seeped through from neighbouring furrows. Using the algorithm, the error in identifying advance front for irrigating furrows is small. The exception is furrow 10, which was incorrectly identified as containing no advance stream, as expected since the advance stream was three pixels high. The complete set of plots for all 11 furrows of Figure 7.11 can be found in Appendix F.

The early afternoon images, which are thresholded based on the hue channel of the HSI colour model, yielded similar graphs but exhibited more noise. Therefore the criterion for determining the advance front from the hue channel is:

If there exists at least ten pixel rows that sum to zero in any block of twenty pixel rows, the position of the advance front is given by the first pixel row that sums to zero in that block of twenty pixel rows.

Table 7.1: Advance front position measured by visual inspection and by image analysis

	Advance front position				
Furrow number	Visual inspection (pixels)	Distance (metres)	Image analysis (pixels)	Distance (metres)	Error (metres)
1	No water	-	No water	-	-
2	232	147	227	149	-2
3	No water	-	No water	-	-
4	359	104	355	105	-1
5	329	112	331	111	1
6	197	165	198	165	1
7	22	417	17	436	-18
8	16	440	17	436	4
9	3	497	No water	-	497
10	13	452	14	448	4
11	No water	-	No water	-	-

A moving average approach was considered for detecting the end of the advance front. Acceptable results were produced for substantial advance streams, however short advance streams failed to be detected by the method. Edge detection was not investigated.

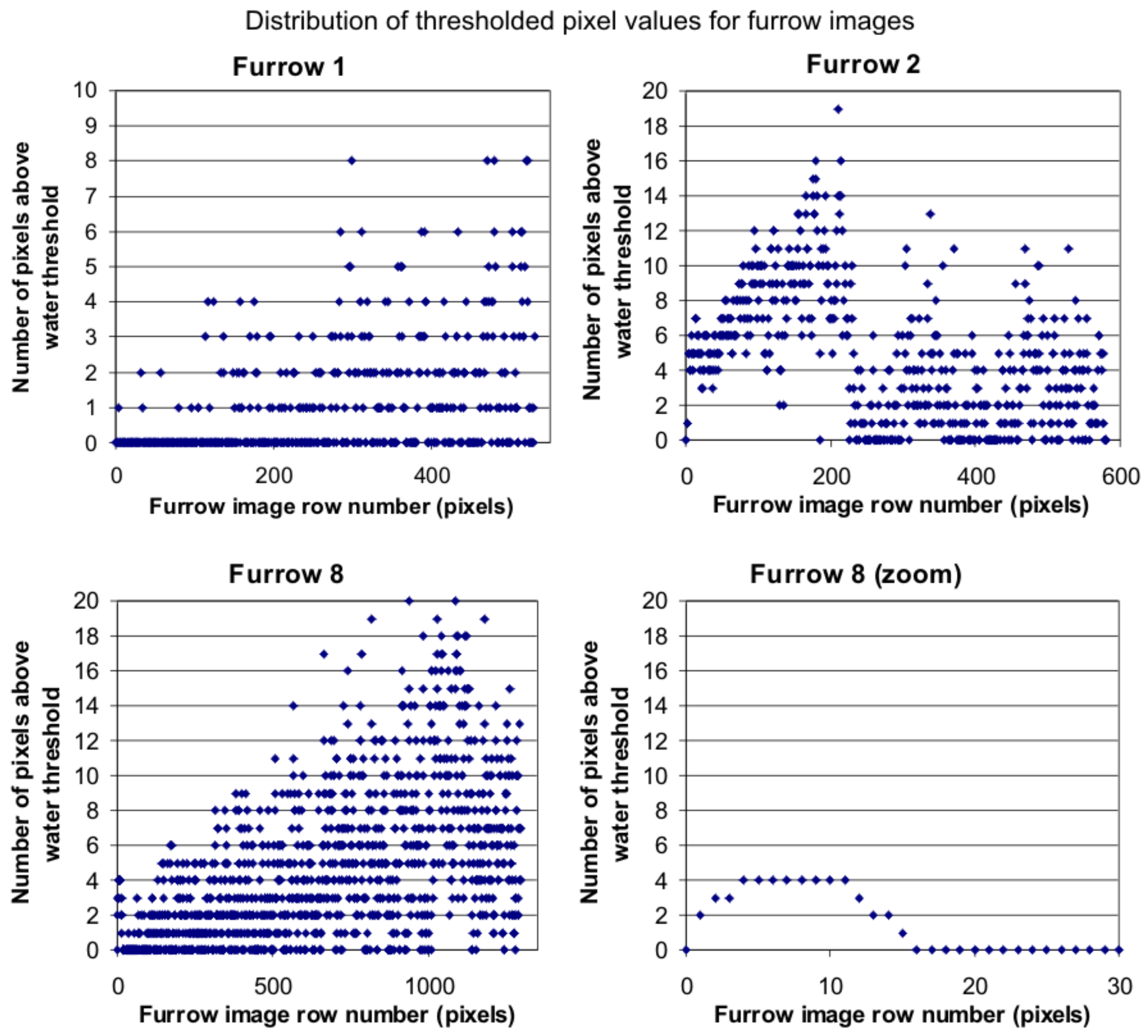


Figure 7.12: Graphs indicating distribution of thresholded pixels for furrow images, from irrigation in early morning sample image

The derived bare soil algorithm was tested on images captured from an irrigation (Figure 7.13), for which actual advance position was known. A grayscale analysis was conducted for images (a) and (b) whereas a hue analysis was conducted for images (c) and (d). Furrow 6 (defined in Figure 7.11) is considered. The ten-metre tower was used.

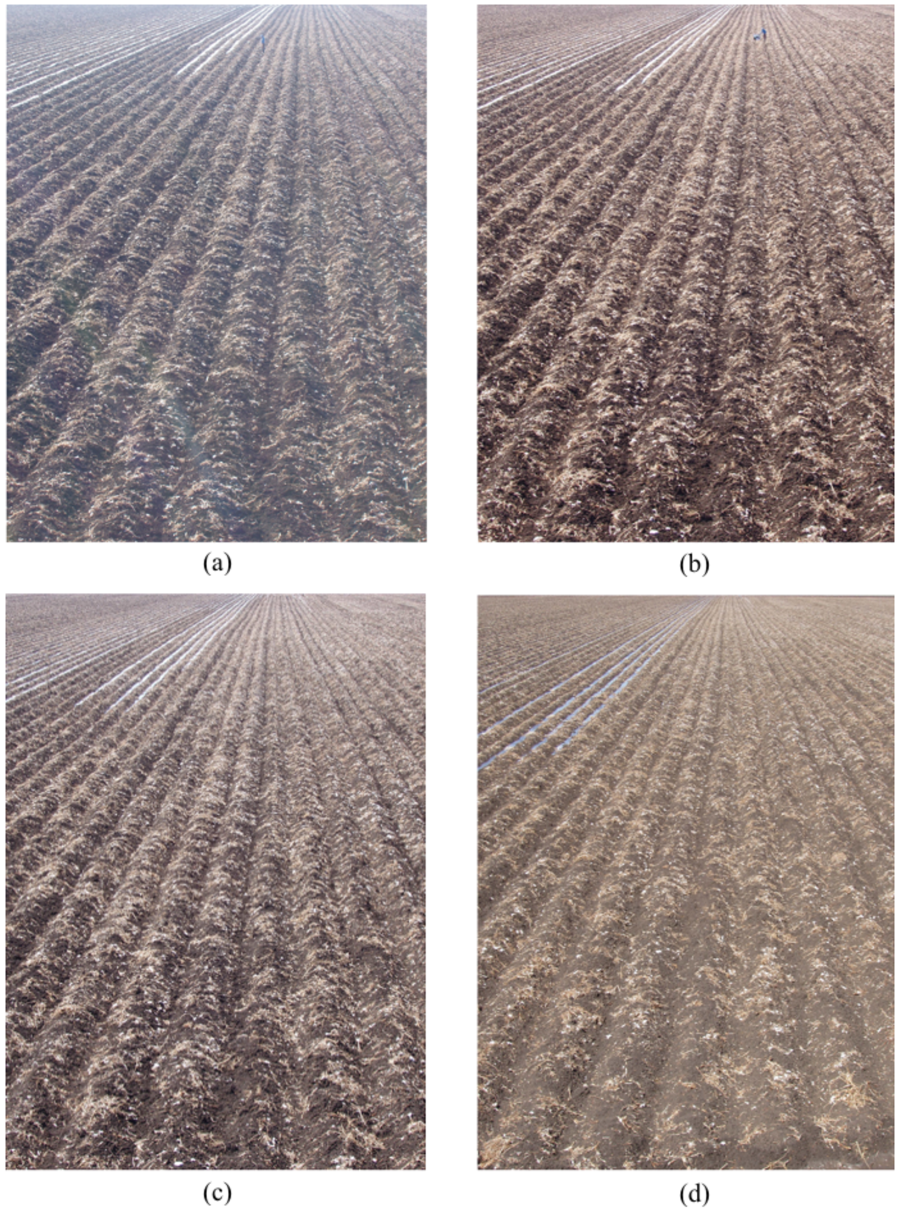


Figure 7.13: Series of images illustrating advance of water stream during irrigation (irrigation start time 5.30 am) (a) 7.39 am (b) 9.08 am (c) 11.32 am (d) 1.41 pm

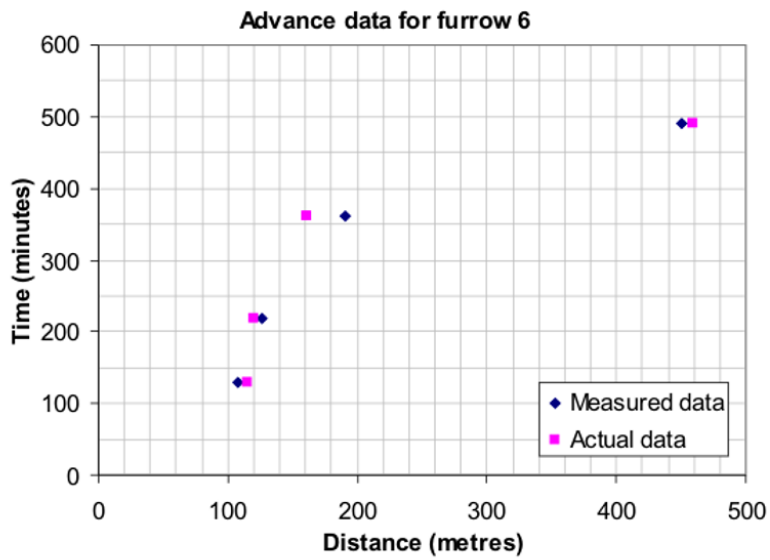


Figure 7.14: Advance curve for Furrow 6, measured and actual data

The error is within ten metres for Figures 7.14(a), (b) and (d). However the reading from Figure 7.14(c), an image captured at 11.32 am, exhibits an error greater than thirty metres. A possible software factor is that the colour of the advance stream was not detected properly since in the adavnce stream is not extremely bright, nor extenuating blueness. This issue can be overcome by measuring more advance points at a different time of day, or with further software development by conducting an alternate analysis for a different colour component of the advance stream.

These data points are not distributed evenly throughout the length of the field, so are not optimal as a complete set of advance data points. However as individual data points in the range 100 and 450 metres, they are within the ten metre margin as required (except Figure 7.14(c)). The major determinant of accuracy and resolution at other (longer) distances is the height of the tower, as discussed in Chapter 6.

The observation was made that in Figure 7.14(c) and (d), the last twenty metres of the field features low contrast between the water and surrounding soil, and the hue channel was ineffective at discerning the water. This is another resolution issue that otherwise can be avoided by restricting advance measurements to other regions of the field.

7.3.2 Algorithm for closed canopy case

Experiments from Chapter 5 identified that the tonal response of wheat plants to water occurred in the red and green channels of the RGB colour model, and that the change could be detected by comparing successive images of the wheat plants. Comparison of the images involved averaging the colour of the test region of crop, minusing the average colour of a reference region, and comparing this difference with the corresponding difference of a prior image. This same approach can be adopted for an image of a crop as is illustrated in Figure 7.15(a), and may be an alternate approach for the bare soil case, as illustrated in Figure 7.15(b).

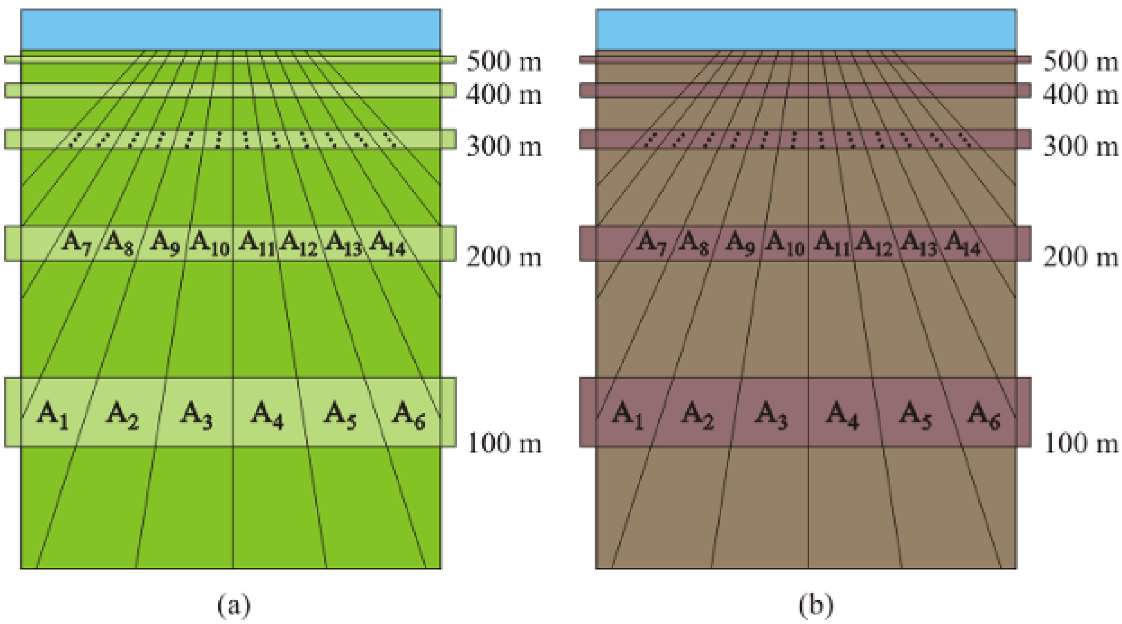


Figure 7.15: Possible approach for image analysis, with regions of field labelled A_1 to A_{14} (a) rows in closed canopy irrigation (b) furrows in bare soil irrigation

Each band of Figure 7.15 represents (for example) five metres of field, at intervals along the length of the field. For each crop row in each distance band (Figure 7.15(a)), an average colour value for the red and green channels can be calculated and then stored in an array (A_1 to A_{14}). The array then accumulates values for each region of the field that indicate the average colour value for that region of the field as a function of time. In accordance with the findings of Chapter 5, when the elements of the array change in magnitude by 300%, the change in colour may be attributed to the arrival of water at a furrow adjacent to that particular row in the last twenty minutes, or some other custom time delay.

To obtain five advance data points, five (sparcely distributed) bands or regions of each crop row can be monitored. Images are captured at twenty-minute intervals so that the tonal response of the specified regions is detected promptly, since if the advance front moves 10 metres in 60 minutes, then an error of 60 minutes equates to 10 metres, whereas an error of twenty minutes equates to only 3.3 metres on average. The number or frequency of images required can be reduced by increasing the number of regions of the field that are being monitored, for example the entire length of field could be divided into regions that are monitored.

The reference point used is required to be subjected to the same lighting conditions as the test region, while being in a position that is not affected by the presence of water. This is so a measure of crop colour can be compared from one image to another independent of ambient light.

7.3.3 Effect of uneven lighting on acquired images

Investigation has focused on daylight conditions where the main source of lighting inconsistencies is cloud cover. Figure 7.16 indicates a selection of encountered situations. In these images, ‘partial cloud cover’ indicates a single region of cloud, ‘intermittent cloud cover’ indicates several regions of cloud, ‘uniform cloud cover’ indicates a field not affected by clouds, ‘light cloud cover’ indicates light shadows cast onto the field, and ‘heavy cloud cover’ indicates dark shadows cast onto the field. These images were acquired during testing of the 6.7 metre tower, conducted in early afternoon.

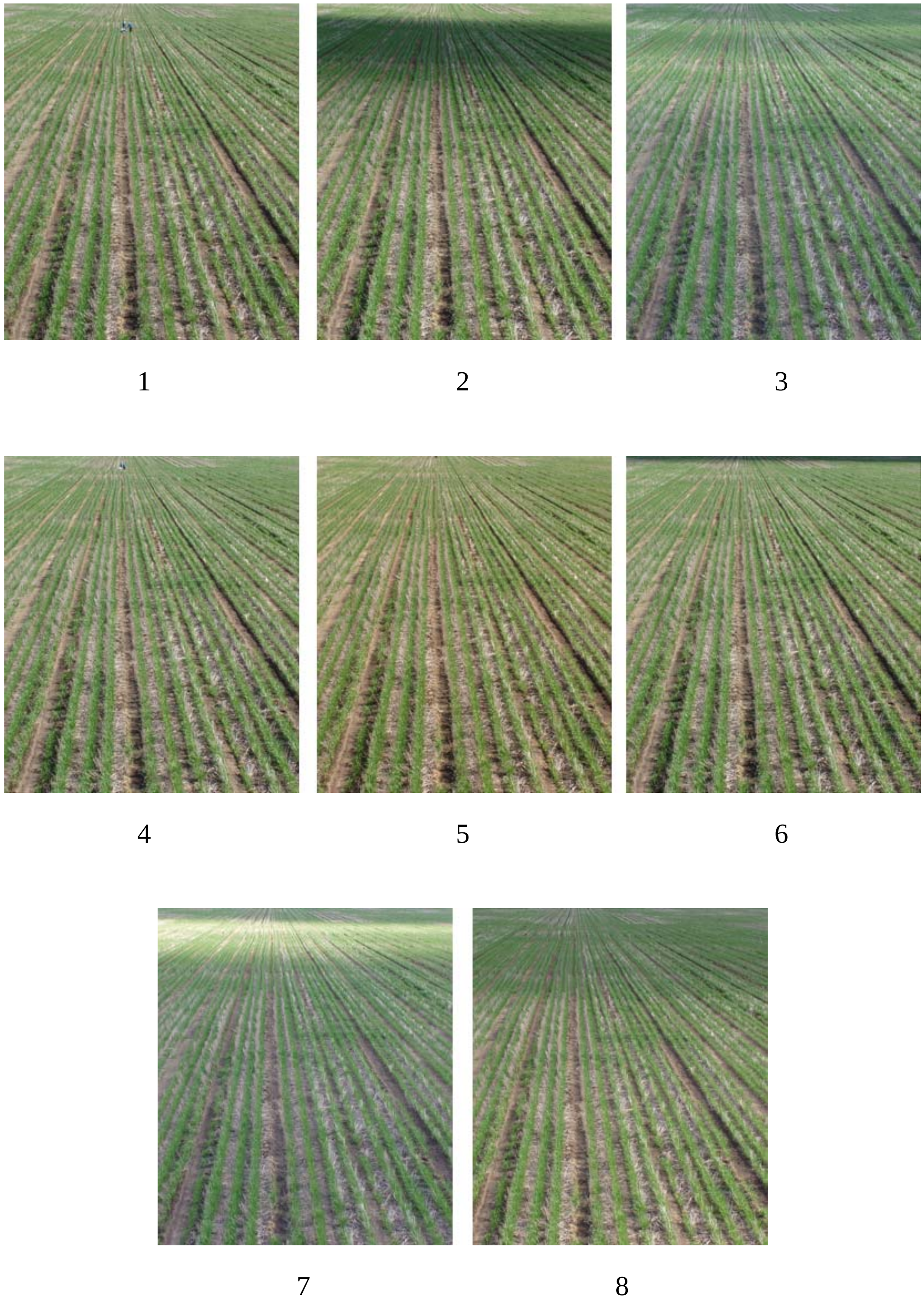


Figure 7.16: Sample images exhibiting cloud cover variation

1 Uniform cloud cover; 2 Partial heavy cloud cover; 3 Intermittent light cloud cover; 4 Uniform cloud cover; 5 Uniform cloud cover; 6 Partial heavy cloud cover; 7 Intermittent heavy cloud cover; 8 Partial light cloud cover.

Modelling outdoor lighting for machine vision is a substantial research area in which identifying the real colour of a surface subject to outdoor lighting is often the item of interest, such as in Buluswar and Draper (2002). Observing plant response does not require ‘real’ colour of the observed objects to be known, but the *change* in crop canopy or bare soil colour that occurs. This simplification enables regions of image to be compared providing they are subject to the same lighting conditions, or if this is not the case, then the variation in lighting conditions between the regions of comparison must be identifiable.

Variation in cloud cover is not a limiting factor for analysis in the bare soil situation since the water stream is highly visible regardless of variation in ambient light. The effect of cloud is accounted for by comparing change in dry furrow colour with change in wet furrow colour for adjacent regions of the image.

The situation in which a closed canopy occurs is more complex since the change in colour is more subtle and occurs over the time domain, a domain in which cloud cover also varies. While adjacent regions of crop may be assumed to be subject to the same cloud cover (as in the bare soil case), using an adjacent region of crop for reference is not valid since the adjacent crop rows may also be affected by the stream of water.

The presence of heavy clouds such as Figure 7.16 (2), (6) and (7) may be detected in an algorithm since a much larger change in colour is incurred from one image to the next than for a response effected only by the presence of water. Similarly, cloud action may be screened if the cloud (and resulting colour change) moves significantly faster than the expected advance rate.

In situations such as Figure 7.16 (3) and (8), the cloud cover produces a smaller change in crop tone which may confound the colour change as a result of the presence of water. The inclusion of cloud induced colour change in a final plot will result in a general trend caused by the advancing water front, plus noise caused by detection of unidentified cloud cover. The general trend may be filtered after a significant number of data points have been taken.

To eliminate the possibility of confounding by the presence of clouds an initial check for cloud cover could be conducted. This enables analysis on images to take place with the knowledge that any colour change that occurs can be attributed to the presence of water, as opposed to analysing an image and making a decision about whether obtained results are the result of clouds or the advance front. The presence of uneven cloud cover in acquired images can be detected using pixel histograms.

7.3.4 Pixel intensity histograms to identify cloud cover

Since image brightness is affected by the presence of clouds, a histogram can be drawn of pixel intensities. Histograms of the images in Figure 7.16 have been drawn for red, green and blue channels (see Figures 7.17 and 7.18). Since the blue channel is not affected as much as red and green by the presence of water, the effect of cloud cover on the blue channel has been further investigated to include the first differences, or slope of the frequency distribution. The histograms have underwent a smoothing routine based on averaging counts.

Observation of these histograms reveals that the images exhibiting uniform cloud cover (Figure 7.16 (1), (4) and (5)) yield the same shaped distibution, but which is displaced to the right on the red channel for the brighter of the three images, image (4). This finding is confirmed by Buluswar and Draper (2002), who say that bright images appear ‘redder’ and dull images extentuate ‘blueness’.

For all channels, the intermittent cloud cover and partial light cloud cover distributions generally peak to the left of the uniform cloud cover curves, with the partial light cloud cover distributions least affected. The heavy cloud cover distributions (2) and (6) yield an extra peak on the red and green channels, and on the blue channel exhibit a ‘kink’ on the left of the distribution.

By inspection, the non-uniform cloud cover distributions on the blue channel have the common quality of regions of higher slope than their uniform cloud cover counterparts. This feature is demonstrated in the plot of first differences of Figure 7.18. From this plot, a threshold value of 1000 can be used for images of this size to determine whether the image is suitable for analysis, based on the presence of clouds.

In mathematical notation, an image which exhibits non-uniform cloud cover meets the following condition:

$$\Delta I_b \quad > \quad 1000 \tag{7.5}$$

where I_b is the intensity of the blue channel of the image.

These histograms have been drawn for the absolute pixel counts. Normalising these histograms by dividing by the total number of pixels in the image provides no commonality for the threshold value of 1000 (smaller images yielded a more ‘squashed’ distribution, with lower peaks). Alternatives to using the hard-wired threshold value are investigating curvature of the plot, relative minimum and maximum slopes for uniform and non-uniform cloud cover (since on the blue channel the non-uniform cloud cover distributions generally exhibit higher absolute values of maxima than minima), and observing the magnitude of the first few counts (since on the blue channel some non-uniform cloud cover distributions exhibit a larger proportion of zero counts). The usefulness of these approaches is influenced by the smoothing routine, which may degrade data significantly for smaller images.

Small foreign objects of the same intensity as the field are not detected by the method (note the people present in Figure 7.16 (1) and (4)). Therefore the method assumes that no foreign objects (such as birds and plastic bags) are present in the scene, but in their presence they will be read as noise. When a bare soil field is being irrigated, the stream of water may disrupt the distribution, however identification of cloud cover is not required in this situation, as discussed in Section 7.3.1.

Large shadows cast upon the field by neighbouring objects such as trees are also expected to be detected by the histogram distribution. This can be alleviated by measuring advance rate for a sample of furrows that is inset from the edge of the field, which is desirable anyway to improve the likelihood of the furrows representing the majority of the field and not the furrows at the end of the field. More extensive software is expected to be capable of identifying slow-moving shadows thereby increasing the number of usable images however this is a special case that can be addressed with further research.

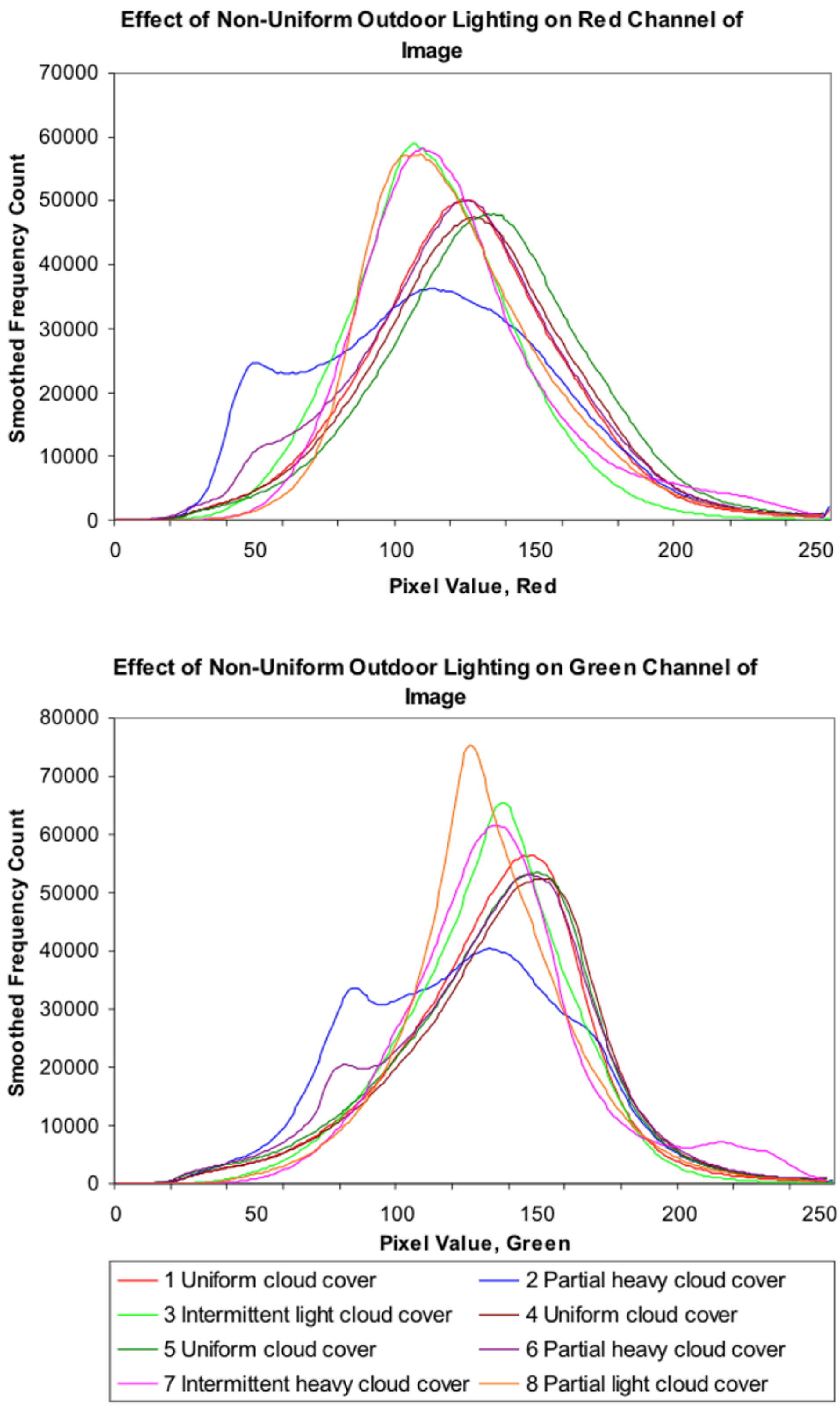


Figure 7.17: Effect of varying light conditions on red and green channel distributions in crop image

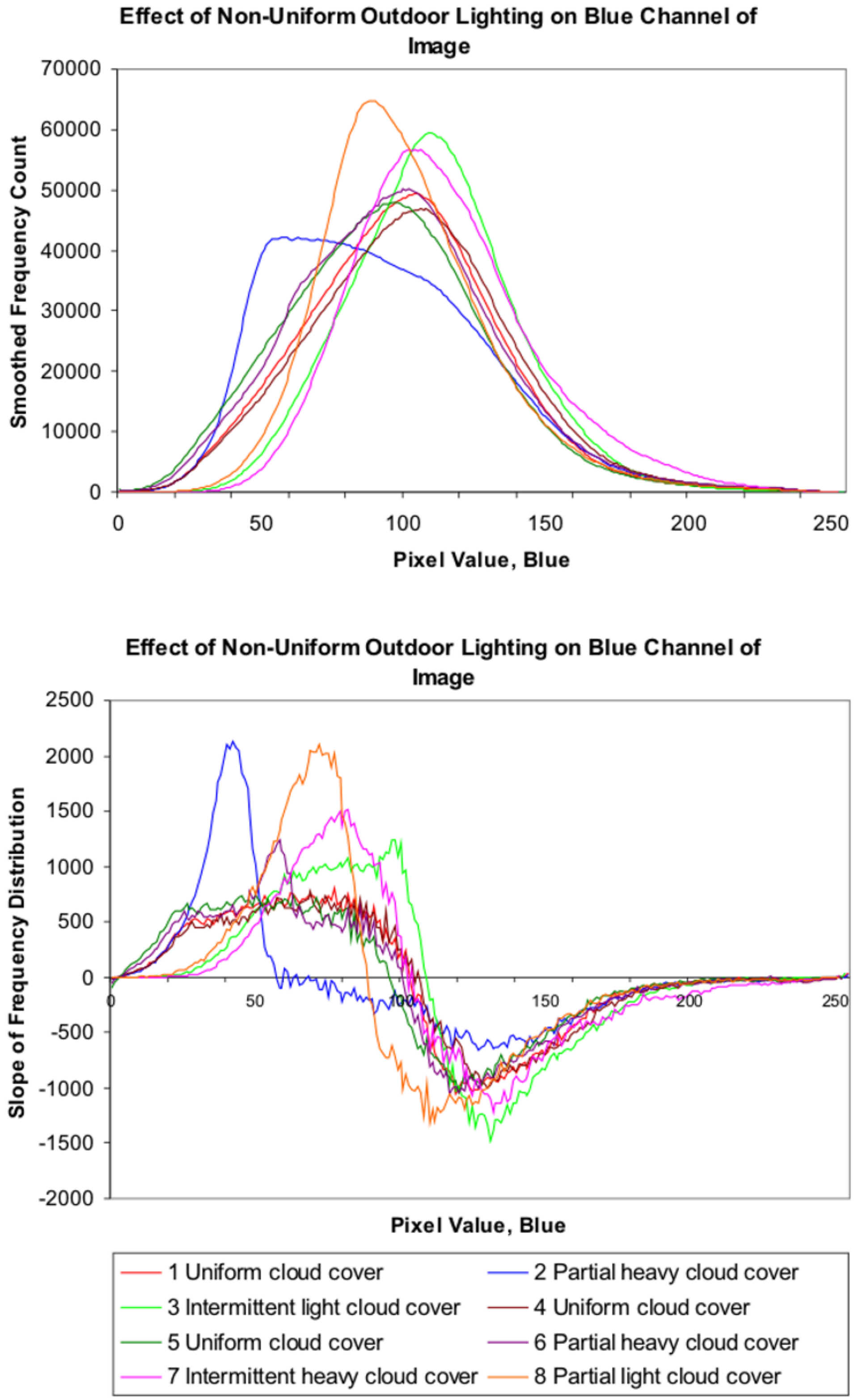


Figure 7.18: Effect of varying light conditions on distribution and distribution slope of blue channel in crop image

7.3.5 Night time irrigations

Artificial lighting for irrigations that occur at night time is provided by a 100 W adjustable parabolic beam spotlight. The spotlight was found to be effective at illuminating the length of a 700-metre field during a night time field trip to a cereal farm in Dalby.

Illumination of the field was evaluated with the spotlight mounted on the camera tower at one end of the field, at elevations of midway along the tower and crop height. Illumination of the length of the field occurred when the spotlight was mounted at crop height, with higher mounting positions causing the illumination range of the spotlight to be severely limited.

The anticipated effect of the artificial lighting on a water stream during an irrigation event is that the water stream will appear white in acquired images, as does the water stream in images captured in early morning. Therefore the same image analysis approach is proposed for the night time and early morning irrigations.

The effectiveness of the spotlight at detecting advance front from a crop canopy at night time is dependent on the physiology of the crop (as discussed in Chapter 5) and is an area that requires specialised research.

Testing of the spotlight during a night time irrigation event was not conducted because the participating field testing sites did not engage in night time irrigations.

7.4 Presentation format of advance data

In compliance with the project specification, the advance rate data must be presented in a format compatible with the existing advance rate measurement equipment, Irrimate.

The protocol for Irrimate 2002 advancemeter communications, and the format of the input file for the Irrimate furrow irrigation modelling software Infilt, is included in Appendix G.

Implications of the protocol are that the obtained advance data should be stored in a data file in a format such as the following, for each of the eight furrows:

Furrow #
inflow rate m^3/min
cross-sectional area of flow, m^2
sigma y
number of advance points
distance 1, time 1
distance 2, time 2
distance 3, time 3
⋮
distance *n*, time *n*

Time is the number of minutes between initiation of the advance rate measurement system and the arrival of the advance front, for a particular furrow and distance. This format has been coded into the advance rate measurement software (see Appendix C).

7.5 Overview of software

The image analysis software consists of the following stages:

1. Initialisation of software, including placement of the grid that flags the location of the rows or furrows (more detail is provided in Appendix H).

2. Identification of location of advance front. For the bare soil case, this is based on the highly contrasting advance stream compared with the rest of the field.

When the advance stream appears white, the advance stream is discerned based on RGB grayscale value, and when the advance stream appears blue, the advance stream is discerned based on hue value. In the case of a closed canopy, the field is divided into regions in which average colour values are maintained as a function of time. Appendix H features flowcharts for these procedures.

3. When the location of the advance front is determined, a pixel number is converted into a distance. This is achieved using the inverse tan relationship between pixel number and distance.

4. The distance and time data pairs are stored in a *.dat file for each furrow or row.

Chapter 8

Integrating the Hardware

For the system to be automated in these infant developmental stages, personal computer (PC) control of components can be achieved through a PC parallel port. The components that require integration to achieve basic input/output commands with the control software include the infrared thermometer, the stepper motor rig and the digital camera. A spotlight is included for extension of the system to night time irrigation events. Figure 8.1 features an interfaced tower. This tower was two metres in height and was not tested for image resolution.

8.1 Interfacing via the PC parallel port

All PCs contain a parallel port which is used for interfacing. The software address of the port can be accessed in the Delphi software development environment.

The output of the parallel port is TTL compatible. To amplify current to a level capable of driving relays or stepper motors, a Darlington array integrated circuit can be used. Relays are required for the camera and spotlight interfacing, and stepper motors used for the thermometer probe rig.

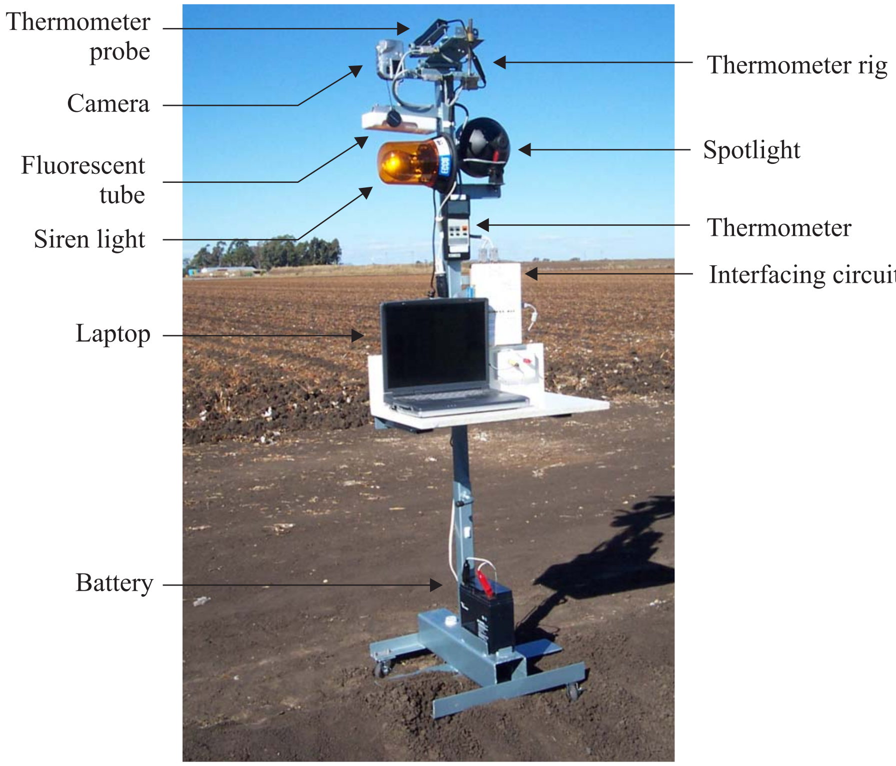


Figure 8.1: Tower featuring laptop interfaced with measuring equipment

A block diagram of the circuit used for interfacing equipment is included in Figure 8.2 (a schematic diagram is included in Appendix I). This circuit is explained in the following sections.

8.2 PC interfacing of the high resolution digital camera

The OLYMPUS C-5000ZOOM consists of several attractive secondary features amenable to the automation of the advance rate measurement system, by way of PC interfacing. The camera features a push-button remote control that activates image capturing, and a PC interface (via USB) consisting of a removable drive on which acquired images are stored.

The PC interface of the digital camera is a powerful capability for the advance rate

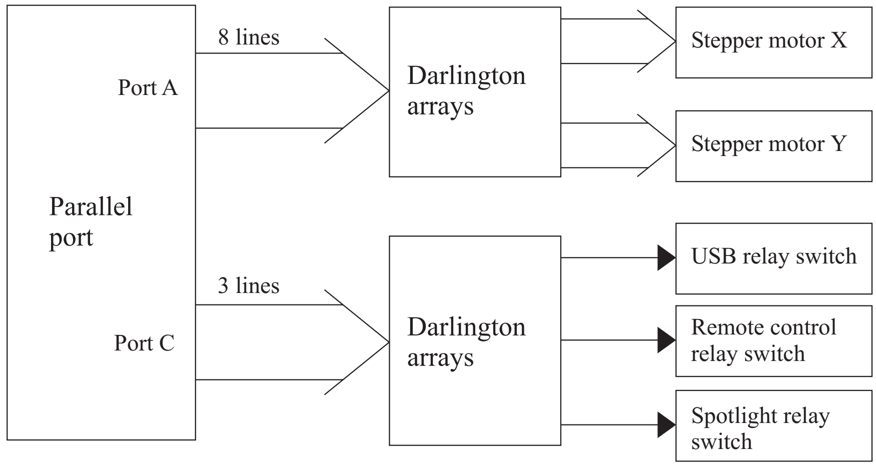


Figure 8.2: Block diagram of interfacing circuit

measurement system since images can be captured and downloaded for analysis at the discretion of the software. This is especially useful when an image is captured but determined to be unsuitable for analysis, or produces inconclusive results, and a replacement image is required to be captured and analysed.

8.2.1 Activation of image capturing

The remote control action for image capturing can be incorporated into PC control by connecting wires in parallel with the remote control switch, and contacting them via a relay (single pole single throw is sufficient). This involves disassembling the remote control and soldering wires to the switch terminals. Physically, the remote control must be mounted so that when the remote control is activated, the camera receives the remote control signal. To meet this condition a remote control mount has been built onto the camera mount for each of the tested towers. The remote control was affixed to the remote control mount with hook and loop tape, and later with an adjustable bracket (Figure 8.3).

The relay activation of the remote control and subsequent image capturing was successful when the camera and remote control were mounted on the 3.2- and 6.7-metre towers. Unfortunately during field testing of taller towers (heights of 10 and

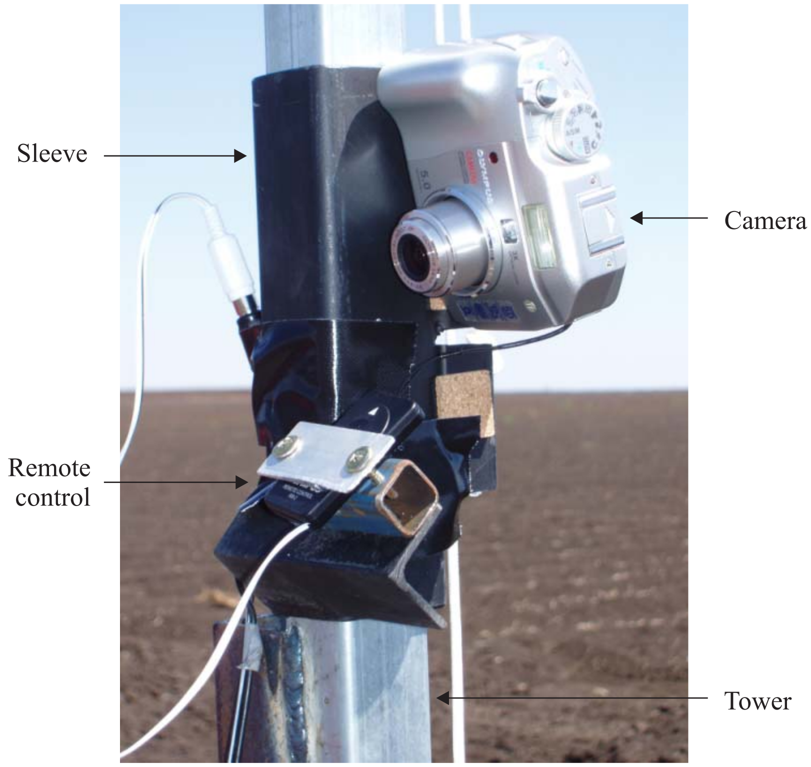


Figure 8.3: Remote control for camera image capturing

14 metres), the remote control was established as being very unreliable, on several occasions. The reason for the remote control failure is unknown, however low battery power and misalignment of the remote control were eliminated as possible causes.

Substantial disruption to field testing caused by the remote control resulted in a late redesign of the image capturing activation process which did not involve the remote control. This approach involved a radio-controlled actuator physically depressing the camera's image capturing button. The design has been implemented and tested using a control handset (Figure 8.4) but a computer interface has not been developed.

Research of other high resolution digital cameras may reveal commercial software applications for PC image previewing and capturing.

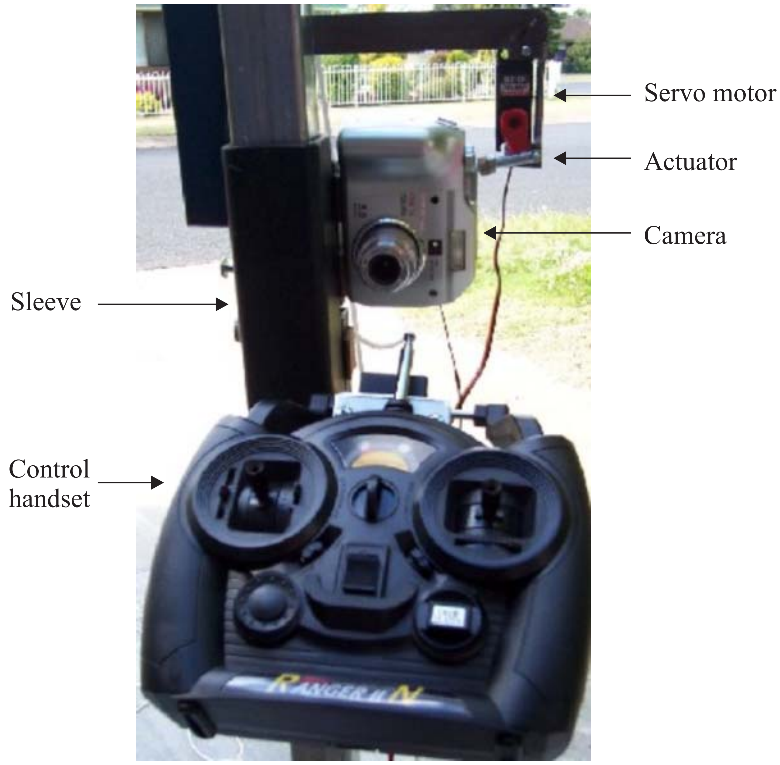


Figure 8.4: Radio control for camera image capturing

8.2.2 Image download via USB connection

Testing of the Olympus C-5000ZOOM revealed that a conflict existed between image capturing and PC connection, so to capture an image, the camera was required to be disconnected from the PC. Once the image is captured the camera must be reconnected to the PC for image download. Connecting and disconnecting of the USB cable can be emulated by activating and deactivating a 4-pole relay (USB cables consist of four wires: common, ground, data in and data out). Disconnecting of a USB device was trialled using a single-pole relay on the supply line of the USB cable, however this approach was not successful and caused the laptop to not recognise the USB device.

Use of one double throw relay to switch between remote control activation and USB connection is not possible since the camera requires a few seconds to store the acquired image after the remote control is activated and before the camera is reconnected to the computer. This timing sequence is only possible using separate relays for the remote control and the USB connection.

If emulation of camera connection and disconnection from the PC was not possible, a suitably-sized picture memory card would have the capacity to store images from the entire irrigation, in which case a software analysis of all the images must be conducted after the irrigation event. The user of the system would be required to manually download the images at the end of the irrigation and run a software application to determine the advance rate measurements. The largest capacity card for the Olympus C-5000ZOOM is 512MB, capable of storing approximately 416 high resolution images (1920 $\times$ 2560 pixels). This is equivalent to twelve images per hour for 36 hours, an ample quantity of images for determining five advance positions.

8.2.3 Possibility for image previewing before image capture

Potential advantages of a camera that enables real-time image previewing before image capturing include remote adjustments to the camera mounting rig, and a pre-check of field conditions (for example, cloud cover). The currently adopted method for checking camera orientation suitability using the Olympus is to capture and visually inspect an image, then (if required) lower, adjust and restore the camera mount. If the digital camera is mounted on a software-controlled rig, and adjustments to the rig orientation (and therefore digital camera viewing window) can be monitored through the camera image-previewing application, the camera orientation can be adjusted without lowering the camera down the tower.

The effect of previewing overall field conditions without capturing a high-resolution image could be achieved to an extent by mounting a webcam, fixed in the same viewing window as the digital camera, and monitoring the webcam image in real-time. Similarly the viewing window of the camera may be accurately adjusted remotely if the webcam and digital camera were mounted on a rig such that the webcam viewing window corresponds to the digital camera viewing window, and the rig position could be remotely adjusted.

8.3 PC interfacing of the spotlight

The spotlight globe is rated at 100W 12V, so the current drawn is

$$I = \frac{P}{V} = \frac{100}{12} = 8.3\text{A}$$

where P and V are maximum ratings for power and voltage respectively. Therefore a relay with a current rating of 10 A has been used for the spotlight.

The spotlight is required to be activated when the outdoor lighting conditions become dark. The system's decision to activate the spotlight is timer-based, where the times at which the spotlight is activated follows the non-daylight hours and are set by the user.

An alternative approach for deciding whether to activate the spotlight is based on sensed lighting conditions, which requires a circuit for a light-activated switch. This approach was not pursued due to the relative complexity of the solution compared to the timer-based approach.

8.4 PC interfacing of the infrared thermometer

The infrared thermometer has a serial PC interface such that temperature readings can be accessed in software (and stored in memory). Interfacing of the infrared thermometer is therefore possible.

8.5 Power consumption

The system is currently being powered (not for extended periods) by a 17 Ah battery. The battery charges an inverter, to which the laptop power supply is connected. The laptop battery can operate for 1.5 hours without charging but then requires 2 hours' charging time. The 17 Ah battery also supplies the spotlight and interfacing circuit.

The digital camera is powered by a Li-ion battery which can operate without recharging for one week if the camera viewfinder is off, or 1.5 hours if the camera viewfinder is on.

To estimate the capacity of battery required to power the system for 36 hours, the following current measurements were taken using an AC/DC digital clamp meter.

Table 8.1: Current readings for advance rate measurement equipment

Component	Current (A)	Cycle time (hr)	Number of cycles in 36 hr	Ampere hours (Ah)
Laptop				
Powered by battery	1.6	1.5	11	26.4
Charging via inverter	3.7	2	11	81.4
Spotlight	8.9	2.2×10^{-3}	36	0.7
			Total	**108.5**

From these calculations, a 12 V 120 Ah battery (which is a standard car battery size) is sufficient to power the advance rate measurement equipment for 36 hours.

The required capacity is reduced if solar panels are used. Solar panels have been connected to the system but unfortunately remain untested.

Chapter 9

Conclusions and Further Work

9.1 System block diagram

The final designed advance rate measurement system is represented as a block diagram in Figure 9.1.

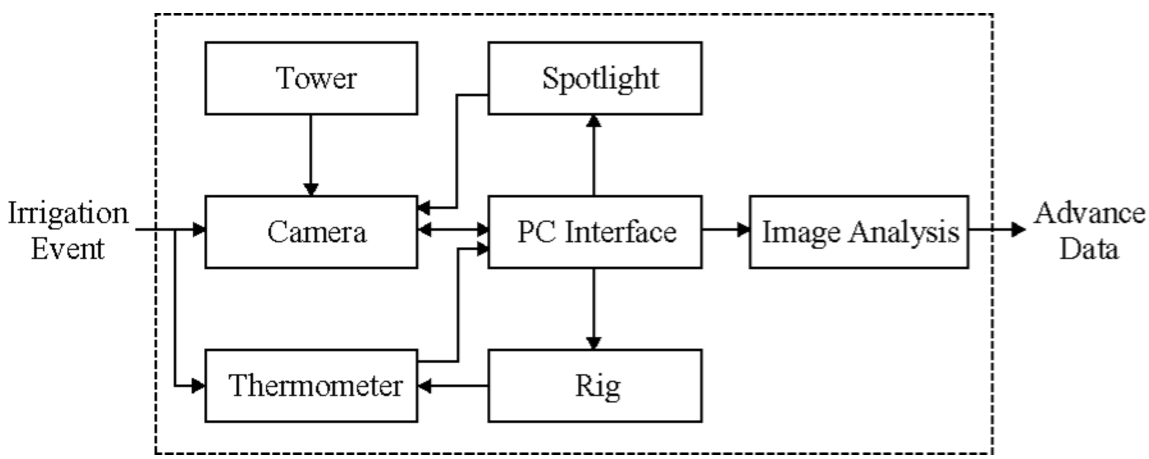


Figure 9.1: Overview of system

9.2 Achievement of project objectives

The following objectives have been met:

Observation of furrow irrigation and the role of advance rate measurement. Chapter 2 provided an overview of furrow irrigation and methods used for the optimisation of furrow irrigation, such as through mathematical modelling which requires advance rate.

Research of existing advance rate measurement methods. There are a small number of approaches developed for advance rate measurement, all involving water sensors in the field, as discussed in Chapter 3.

Investigation of possible approaches for advance rate measurement. Robots and machine vision were considered as candidate advance rate measurement systems. Machine vision was chosen after preliminary investigations mentioned in Chapter 4 revealed a gap in the literature relating to tonal responses of plants to water.

Design of an advance rate measurement system. Chapters 6 to 8 investigate the hardware and software required to monitor advance rate using a high resolution digital camera as part of an automated system. The required accuracy is obtained by mounting the camera on top of a tower.

Design of a supporting structure and setup procedure. Several camera tower structures were constructed (Chapter 6), and a software setup procedure has been developed (Appendix H).

Prompt retrieval of Irrimate-compatible advance data. The developed advance rate measurement software enables advance data to be downloaded at the user's discretion during an irrigation, in a format that can be directly entered into the Irrimate software package Infilt (Chapter 7).

Construction and evaluation of prototype. Each of the prototype camera towers, including the interfaced tower with the laptop (Chapter 8) were assembled and tested on farm sites. Image analysis software was tested at a later date, using images from a separate irrigation event.

Field test of prototype during irrigation. A prototype camera tower was used to monitor advance rate during an irrigation event. Images were later downloaded and analysed (results in Chapter 7).

9.3 Further work

This project has raised several potential future research areas which are listed below.

9.3.1 Plant response

The largest avenue of possible study arising from this project relates to the immediate response of plants to the application of water, through observation of tonal response or some other characteristic of the plant. Such investigation will involve multispectral scanning, including thermal infrared, which requires a large budget for specialist equipment.

9.3.2 Image analysis

Analysis of digital images of crop fields, obtained during irrigation or otherwise, is required to be conducted for images captured during a variety of seasonal and outdoor lighting conditions. A resume of image analysis methods can be developed so that advance front position can be determined using one particular approach and confirmed with another approach. One such approach is to compare successive images, which is the recommended approach for the closed canopy situation but is a secondary approach for the bare soil analysis.

The software developed in this project implemented an analysis procedure based on the time of day, as determined by the user. More sophisticated software is anticipated to be capable of observing the colour composition of captured images and executing an analysis procedure based on that composition. Similarly, criterion for non-uniform cloud cover using pixel intensity histograms has been developed only from images captured in early afternoon. Applicability of the criterion, and alternate criteria, are required to be investigated for other daylight conditions. A literature review of outdoor machine vision techniques and colour models is recommended.

Automated row and furrow identfication (such as in tractor guidance systems) could be pursued for the field condition where both soil and crop are visible. The grid and flagged bounds approach developed in this project for determining furrow or row location is necessary in the case of a closed canopy, when the lines of the rows are indiscernible. However the grid system is limited to perfectly straight and evenly spaced rows and furrows, such that if this condition is not met, the grid provides a poor fit, especially at longer distances where the furrows and rows consume only a few pixels. An adjustable grid that caters for curvillinear and non-equally spaced furrows or rows will be a useful tool in the advance rate measurement software.

9.3.3 Image pixel distribution

Further investigation is required to determine the relationship between a pixel of a one-point perspective image and the actual distance represented by that pixel. An inverse tan relationship has been established however the tilt angle of the camera, which was initially considered negligible, may influence the pixel-distance distribution for the image.

Using the inverse tan relationship, the error caused by irregular height in the field can be quantified by superimposing plots for tower height $h \pm 0.3$ m (for example) and observing the offset in reading at individual data points.

9.3.4 Camera interfacing

Market research for a high resolution digital camera with a real-time image previewing feature and image capture option in software is considered worthwhile. Other areas of development for the camera includes a remotely adjustable rig to alter the camera orientation without requiring the camera to be lowered and adjusted manually.

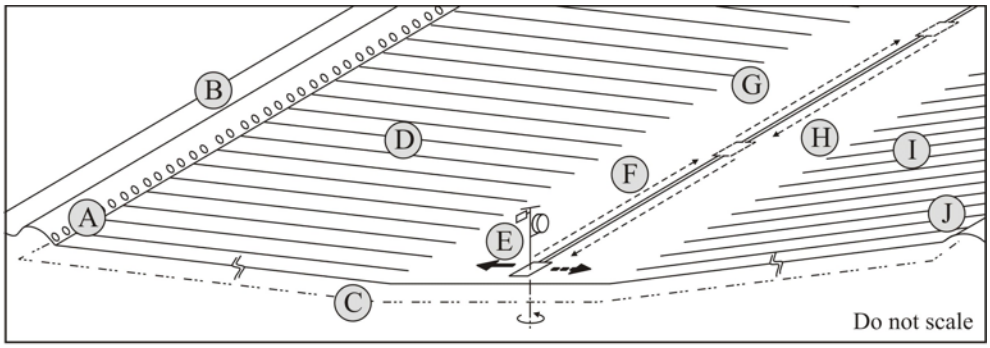
9.3.5 End-of-irrigation event alarm

When the advance rate measurement system detects that the advance front has progressed to a predefined position in the field, a signal from the control software can initiate a flashing light (Figure 8.1) or telephone call (as used currently on large mobile irrigation machines) to notify the irrigation manager of the impending conclusion of the irrigation run.

9.3.6 Extension to real-time furrow irrigation control

The potential exists for the advance rate measurement system for furrow irrigation to be extended to real-time control of furrow irrigation, which entails the unassisted and optimised irrigation of an entire field following an initial command from an irrigation manager.

The system will integrate the technologies of machine vision advance rate monitoring, irrigation parameter optimisation and automatic inflow rate initialisation and control. An example irrigation system is depicted in Figure 9.2, which was formulated as a research proposal for assessment in the course ENG4110 Engineering Research Methodology.



A Inflow rate control mechanism to furrows in field #1
B Head ditch of field #1
C Electrical/radio signal to inflow rate control mechanism from software that calculates optimised irrigation parameters
D Furrows in field #1
E Advance rate monitoring equipment. Equipment moves along path (F) and rotates 180 degrees to face either field #1 or field #2
F Path for advance rate monitoring equipment
G Tailend of field #1
H Tailend of field #2
I Furrows in field #2
J Head ditch of field #2

Figure 9.2: Advance rate measurement in real-time furrow irrigation control

Existing automatic feedback irrigation systems, such as in Latimer and Reddell (1990), reduce labour and tailwater but provide no benefit of improved infiltration uniformity. Integrating automatic inflow rate control with advance rate monitoring will create an automatic furrow irrigation system which features all of these advantages.

9.4 Other project outcomes

The project was presented at a joint MUGs/TAPs meeting on 5 July 2004, as well as at the annual Water Panel Research Seminars Evening organised by Engineers, Australia - Queensland Division on 20 October 2004 (see poster presentation for this event in Appendix J).

References

Alves, I. & Periera, L. (2000), 'Non-water-stressed baselines for irrigation scheduling with infrared thermometers: A new approach', *Irrigation Science* **19**, 101–106.

Buluswar, S. & Draper, B. (2002), 'Color models for outdoor machine vision', *Computer Vision and Image Understanding* **85**, 71–79. Viewed 23 May 2004, Elsevier Science.

Core, J. (2002), 'Hot cotton: New system senses water stress', *The Australian Cottongrower* **23**(4), 36.

Crane, R. (1997), *A simplified approach to image processing: classical and modern techniques in C*, Prentice Hall of Australia Pty. Limited, Sydney.

King, L. & Amend, J. (1983), 'Radio- or clock-activated semi-automatic irrigation gate controller', *Agricultural Electronics – 1983 and Beyond, Proceedings of the ASAE National Conference on Agricultural Electronic Applications, American Society of Agricultural Engineers, Chicago Illinois* **1**, 172–181.

Latimer, E. & Reddell, D. (1990), 'Components for an advance rate feedback irrigation system (ARFIS)', *Transactions of the ASAE* **33**(4), 1162–1170.

McClymont, D. & Smith, R. (1996), 'Infiltration parameters from optimisation on furrow irrigation advance data', *Irrigation Science* **17**, 15–22.

Murase, H., Nishiura, Y. & Mitani, K. (1997), 'Environmental control strategies based on plant responses using intelligent machine vision technique', *Computers and Electronics in Agriculture* **18**(2–3), 137–148.

Purcell, J. (2004), 'Doing better with what we have: The art of surface irrigation', *The Australian Cottongrower* **27**(7), 36–38.

Rogers, D. (1995), *Managing Furrow Irrigation Systems*, Kansas State University, Kansas. `http://www.oznet.ksu.edu/library/ageng2/L913.PDF` viewed 21 January 2004.

Sadler, E., Camp, C., Evans, D. & Millen, J. (2002), 'Corn canopy temperatures measured with a moving infrared thermometer array', *Transactions of the ASAE* **45**(3), 581–591.

Scott, C., Bastiaanssen, W. & Ahmad, M. (2003), 'Mapping root zone soil moisture using remotely sensed optical imagery', *Journal of Irrigation and Drainage Engineering* **129**(5), 326–335.

Seginer, I., Elster, R., Goodrum, J. & Rieger, M. (1992), 'Plant wilt detection by computer-vision tracking of leaf tips', *Transactions of the ASAE* **35**(5), 1563–1567.

Shepard, J., Wallender, W. & Hopmans, J. (1993), 'One-point method for estimating furrow irrigation', *Transactions of the ASAE* **36**(2), 395–404.

Spragge, A. (2002), 'Evaluating cotton irrigation efficiency', *The Australian Cottongrower* **23**(4), 40.

Turnell, D., Singh Deep, G. & Freire, R. (1997), 'A low-cost water detection method for furrow irrigation control', *IEEE Transactions on Instrumentation and Measurement* **46**(4), 986–990.

University of Southern Queensland (USQ) (2004), *ENG4406 Robotics and Machine Vision Study Book*, University of Southern Queensland, Toowoomba.

Walker, W. & Busman, J. (1990), 'Real-time estimation of furrow infiltration', *Journal of Irrigation and Drainage Engineering* **116**(3), 299–317.

Appendix A

Project Specification

University of Southern Queensland
Faculty of Engineering and Surveying

ENG 4111/2 Research Project
Project Specification

FOR: **Cheryl McCARTHY**

TOPIC: Advance rate management for furrow irrigation

SUPERVISORS: Mark Phythian
Joseph Foley

PROJECT AIM: The project aims to create and test a model transportable mechatronic device that will measure irrigation advance rate in adjacent furrows and will enable convenient recovery of measurements immediately after irrigation takes place.

PROGRAMME: **Issue B, 9 August 2004**

1. Observe the characteristics of furrow irrigation and the role of advance rate measurement.

2. Research existing methods of measuring advance rate in furrows.

3. Investigate methods of detecting water along several furrows simultaneously taking into account variations across farms such as length of furrow and maturity of crop.

4. Design a measurement system to detect the time at which the irrigation water reaches, to within ±5 metres, at least five specified positions along a furrow 300-1000 metres long, and capable of being extended to measure advance rate simultaneously across at least eight furrows.

5. Design a supporting structure if necessary and setup procedure that allows transportability of the chosen measuring system.

6. Provide a method by which advance rate measurements can be stored and presented in a format compatible with Irrimate, within a short period after irrigation takes place.

7. Construct a prototype and evaluate its performance under simulated conditions.

As time permits:

8. Test the prototype in the field, during irrigation.

Agreed:

__________ (Student) __________ , __________ (Supervisors)

9/8/2004 9/8/2004 9/8/2004

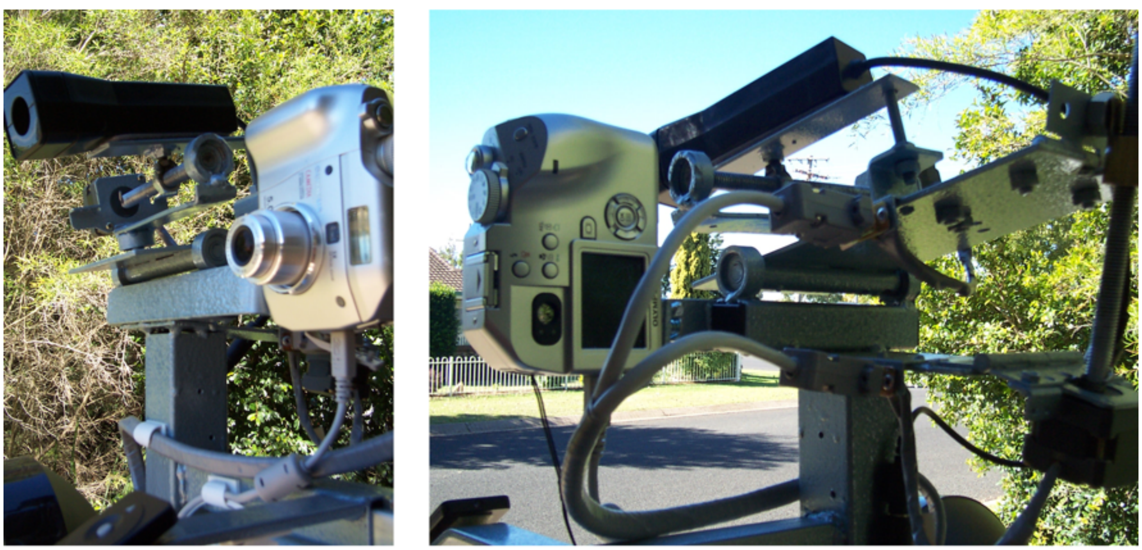
Appendix B

Infrared Thermometer Development

An infrared thermometer was obtained to investigate as a possible method of measuring advance rate.

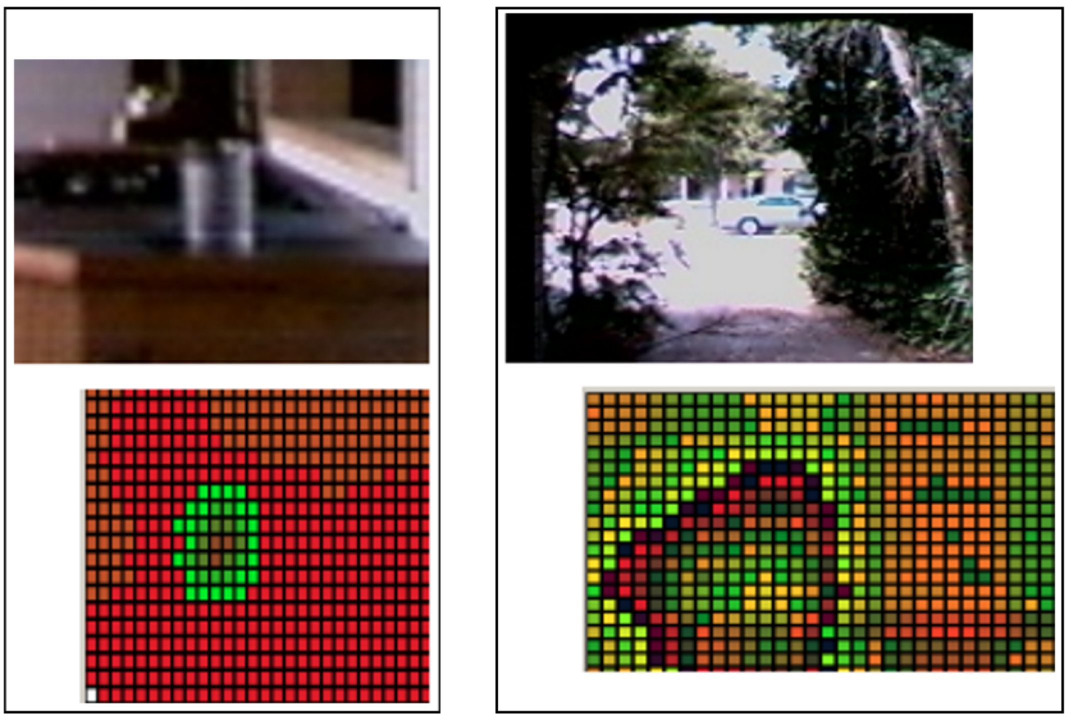
Through the software control of a pair of stepper motors, an adjustable rig was developed for the infrared thermometer probe. Figure B.1 illustrates the infrared thermometer probe mounted on an adjustable rig. The high resolution digital camera also features in this image.

The thermometer has a RS232C PC interface which enables sensed temperatures to be accessed in software. A computer program was developed which sequentially adjusted the position of the infrared thermometer probe (via stepper motors connected to a PC parallel port), and a unique colour was plotted for each temperature the thermometer sensed. Figure B.2 features sample images acquired using this process.



(a) (b)

Figure B.1: Views of infrared thermometer probe on stepper motor controlled rig



(a) (b)

Figure B.2: Image obtained using infrared thermometer, stepper motor rig and PC interface (a) Glass of water, 2 metres (b) View from driveway, 50 metres

Appendix C

Source Code

The following listing is for the main form of the advance rate measurement software. The programming environment used was Delphi.

This code implements the discussion of Chapter 7. Further code is required to be written for the closed canopy 'banded' region analysis; image window adjustment (rotate and cropping); append data file, for use when collating a data file from two cameras and two laptops (at either end of the field); and selection of a reference point on the image for the closed canopy analysis.

The screen capture of this program is included in Appendix H.

```
unit UnitAdvanceRate;

interface

uses
  Windows, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Buttons, ComCtrls, ExtCtrls, Spin, StdCtrls, PortIO,
  JPEG, Math;

const
 // Port and folder locations for laptop
 PCPort='COM3';
 OlympusFolder='F:\Dcim\100olymp\';
 PortC = $37A;

 // Addresses of pins on interfacing circuit (via parallel port)
 LampPin=$02;
 RemotePin=$04;
 CameraPin=$08;

 // Empirically determined delay factors
 LampDelay=5;
 RemoteDelay=1;
 ProcessDelay=45;
 CameraDelay=50;

 // Storage location for captured images
 StoredPhotos='C:\StoredPhotos\';

 // Image size parameters, as set by Olympus C5000-ZOOM
 PhotoWidth = 1920;
 PhotoHeight = 2560;

 // Default number of furrows to monitor
 furrows = 11;

 // Times at which water stream appears blue
 DullStart = 1100;
 DullStop =  1630;

type
  // Used by PortIO routines
  TBase = (bDecimal, bHex);
  TData = (dtaByte, dtaWord, dtaDWord);

  // Variable type used for adjustable grid
  coord = (coTL, coTR, coBL, coBR);

  // Variable type to discern whether analysis is for bare soil
```

```
// or closed canopy
AnalysisType = (atBareSoil, atClosedCanopy);

// Required to access RGB components of image
TRGBArray = ARRAY[0..32767] of TRGBTriple;
pRGBArray = ^TRGBArray;

TfrmAdvanceRate = class(TForm)
  gbxAnalysis: TGroupBox;
  labRef: TLabel;
  cmdChangeRef: TButton;
  labRefBox: TLabel;
  gbxInfilt: TGroupBox;
  labInflow: TLabel;
  edInflow: TEdit;
  labAreaFlow: TLabel;
  edAreaFlow: TEdit;
  labSigmaY: TLabel;
  edSigmaY: TEdit;
  labNoPoints: TLabel;
  speNoPoints: TSpinEdit;
  gbxTiming: TGroupBox;
  labStart: TLabel;
  dtpStart: TDateTimePicker;
  labRuntime: TLabel;
  edRunTime: TEdit;
  labIntervals: TLabel;
  edIntervals: TEdit;
  labDaylight: TLabel;
  labAM: TLabel;
  labResponse: TLabel;
  speDelay: TSpinEdit;
  gbxTower: TGroupBox;
  labHeight: TLabel;
  speHeight: TSpinEdit;
  labDistance: TLabel;
  speDistance: TSpinEdit;
  speLength: TSpinEdit;
  labLength: TLabel;
  gbxImage: TGroupBox;
  img1: TImage;
  gbxWindow: TGroupBox;
  gbxGrid: TGroupBox;
  gbxResults: TGroupBox;
  lsvResults: TListView;
  cmdCreateFiles: TButton;
  cmdAppendFiles: TButton;
  cmdCapture: TButton;
  bitRoLeft: TBitBtn;
  bitRoRight: TBitBtn;
  bitCrop: TBitBtn;
  cmdAcceptWindow: TButton;
  cmdCancelWindow: TButton;
  gbxStatus: TGroupBox;
  labStatus: TLabel;
  gbxTimer: TGroupBox;
  cmdStartTimer: TButton;
  cmdStopTimer: TButton;
  cmdExit: TButton;
  cmdGTLeft: TButton;
  cmdGTRight: TButton;
  cmdGTUp: TButton;
  cmdGTDown: TButton;
  cmdGTIn: TButton;
  cmdGTOut: TButton;
```

```
    cmdGBUp: TButton;
    cmdGBLeft: TButton;
    cmdGBRight: TButton;
    cmdGBDown: TButton;
    cmdG2Left: TButton;
    cmdG2Right: TButton;
    labNoFurrows: TLabel;
    speNoFurrows: TSpinEdit;
    DLPortIO: TDLPortIO;
    Timer1: TTimer;
    labStepSize: TLabel;
    speStepSize: TSpinEdit;
    cmdAccept: TButton;
    panOptions2: TPanel;
    radHead: TRadioButton;
    radTail: TRadioButton;
    panOptions1: TPanel;
    radClosed: TRadioButton;
    radBare: TRadioButton;
    edDay1: TEdit;
    labPM: TLabel;
    edDay2: TEdit;
    cmdAnalyseNow: TButton;
    opd1: TOpenDialog;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    procedure FormCreate(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure cmdGTLeftClick(Sender: TObject);
    procedure cmdGTRightClick(Sender: TObject);
    procedure cmdGTUpClick(Sender: TObject);
    procedure cmdGTDownClick(Sender: TObject);
    procedure cmdGTOutClick(Sender: TObject);
    procedure cmdGTInClick(Sender: TObject);
    procedure cmdGBUpClick(Sender: TObject);
    procedure cmdGBLeftClick(Sender: TObject);
    procedure cmdGBRightClick(Sender: TObject);
    procedure cmdGBDownClick(Sender: TObject);
    procedure speStepSizeChange(Sender: TObject);
    procedure speNoFurrowsChange(Sender: TObject);
    procedure cmdExitClick(Sender: TObject);
    procedure cmdAcceptClick(Sender: TObject);
    procedure cmdStartTimerClick(Sender: TObject);
    procedure cmdCaptureClick(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
    procedure cmdStopTimerClick(Sender: TObject);
    procedure cmdAnalyseNowClick(Sender: TObject);
    procedure cmdCreateFilesClick(Sender: TObject);
  private
    { Private declarations }
    FBaseMode : TBase;                // used by PortIO routines
    FDataType : TData;

    procedure DrawFurrows;
    procedure InitialiseArray;
    procedure UpgradeGrid;

    procedure BareSoilAnalysis;
    procedure GrayscaleAnalysis;
    procedure HueAnalysis;

    procedure ClosedCanopyAnalysis;
    function CloudCoverCheck:boolean;
    procedure HistConstruct(channel:string);
```

```
    function MaxIndex(ArrayInt:array of integer):integer;

    function LookBetweenLines
                (Furrow,ZeroCount,Space:integer):integer;
    function AssignDistance(px:integer):integer;
    procedure AddResultItem(f,Dist,Time:integer);

    procedure TakePhoto;
    procedure Delay(Seconds:integer);
    procedure ConnectCamera(CameraStatus:boolean);
    procedure LampOn(LampStatus:boolean);

    procedure DownloadLatestPhoto;
    procedure CaptureImage;
    procedure Rotate90(Direction:boolean);
  public
    { Public declarations }
  end;

var
  jpg1:TJPEGImage;          // Image downloaded from camera
  bmp1:TBitmap;             // Image used for analysis
  strNewName:string;        // Filename of image from camera
  CWord:integer;

  coXY:coOrd;               // Used for adjustable grid
  c1,c2,c3,c4,p1,p2:TPoint;
  stepsize:integer;

  // Stores grid as coordinates
  FurrowStart:array [1..20, 1..2560] of integer;

  // Stores colour histogram values
  HistCount:array [0..255] of integer;
  // Slope of colour histogram
  FirstDiff:array[0..254] of integer;

  DarkTimeStart,DarkTimeEnd:integer;  // Non-daylight hours (to
                    // determine whether spotlight is required)
  Analysis:AnalysisType;
  IntervalCounter:integer;       // Timing for bare soil images
  pxChange:integer;              // Advance front position, pixels
  Distance:integer;              // Advance front position, metres
  CurrentTime,StartTime:integer; // Stores time of image capture
  CurrentMins,StartMins:integer;

  frmAdvanceRate: TfrmAdvanceRate;

implementation

uses MessagesUnit;

{$R *.dfm}

//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
// Code for adjustable grid
//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

//////////////////////////////////////////////////////////////////
// Draws the adjustable grid on the image canvas.
//////////////////////////////////////////////////////////////////
procedure TfrmAdvanceRate.DrawFurrows;
var
 n,d,k:integer;
begin
 img1.Canvas.Pen.Color:=clFuchsia;
 img1.Canvas.Pen.Width:=1;
 try
```

```
   n:=speNoFurrows.Value;   // Number of furrows in foreground
   d:=(furrows+1-n)div 2;

   // (Extrapolates grid to furrows outside foreground of image,
   //  assuming equal spacing of furrows)
   for k:=0-d to n+d do
   begin
    p1:=point(round(c1.X+k/n*(c2.X-c1.X)),c1.Y);
    p2:=point(round(c4.X+k/n*(c3.X-c4.X)),c4.Y);
    img1.Canvas.MoveTo(p1.X,p1.Y);
    img1.Canvas.LineTo(p2.X,p2.Y);
   end;
   img1.Canvas.MoveTo(0,c4.Y);
   img1.Canvas.LineTo(img1.Width,c3.Y);
 except        // For when spin edit box does not contain integer
  messagedlg('Number of furrows must be an integer.',mtWarning,
           [mbOK],0);
 end;
end;

/////////////////////////////////////////////////////////////////
// Tests whether the pixel px is fuchsia in colour (and therefore
// part of grid).
/////////////////////////////////////////////////////////////////
function pinktest(px:tagRGBTriple):boolean;
begin
 if (px.rgbtRed=255) and
    (px.rgbtGreen=0) and
    (px.rgbtBlue=255) then
  pinktest:=true
 else
  pinktest:=false;
end;

/////////////////////////////////////////////////////////////////
// Steps through the bitmap on which the adjustable grid is drawn
// in order to construct an array containing the grid pixel
// positions for the entire image. The array FurrowStart can
// contain data for a maximum of 19 furrows and a grid 2560
// pixels in height.  For each furrow, the array FurrowStart
// contains the bounding pixel positions for the furrow.
/////////////////////////////////////////////////////////////////
procedure TfrmAdvanceRate.InitialiseArray;
var
 sli1:pRGBArray;
 k,f,i,j,furrowsoffedge:integer;
 fx,fy:integer;
begin
 // Initialises array in which grid data is stored
 for fx:=1 to 20 do
  for fy:=1 to 2560 do FurrowStart[fx,fy]:=-1;

 j:=0;               // Image pixel row counter
 f:=0;               // Furrow counter follower; lags fx by 1
 k:=0;               // Image pixel column counter

 // Number of furrows at the top of the image that have now run
 //  outside the image; required to maintain correct furrow count
 //  for furrows that are still in the image
 furrowsoffedge:=0;
 fy:=-1;             // Row counter, equivalent to j

 while j<bmp1.Height do
 begin
  fy:=fy+1;
  fx:=0;             // Furrow counter
  sli1:=bmp1.Scanline[j];
```

```
  // If the first pixel of the image row is part of the grid,
  // then the furrow at the edge of the image runs outside the
  // image in the next image row
  if pinktest(sli1[k])=true then
  begin
   furrowsoffedge:=furrowsoffedge+1;
   // Set the rest
   for i:=j+1 to 2560 do
     FurrowStart[furrowsoffedge,i]:=-1;
  end;

  // Step through image pixel row
  while k<bmp1.Width-1 do
  begin
   // Step until a grid pixel is found or end of pixel row is
   // reached
   while (k<bmp1.Width-1) and (pinktest(sli1[k])=false) do k:=k+1;

   // Update furrow counter
   fx:=fx+1;

   // If grid pixel was found then save pixel position k in array
   if ((pinktest(sli1[k])=true) and (k=bmp1.Width-1)) or
     (k<bmp1.Width-1)
    then FurrowStart[fx+furrowsoffedge,fy]:=k;

   // Step over grid on original image (if grid was greater than
   // one pixel wide, brought about by the diagonal grid lines)
   while (k<bmp1.Width-1) and (pinktest(sli1[k])=true) do k:=k+1;

   // Tests whether the last grid pixel found is at the end of
   // the image row
   if k<bmp1.Width-1 then
   begin
    // Still within image, increment furrow counter follower
    f:=f+1;
    k:=k+1;
   end
   else
    // Ensure that loop shall be terminated in the next pass,
    // since the end of the image row has been found
    k:=bmp1.Width-1;
  end;

  // Prepare counters for a new row of the image
  j:=j+1;
  f:=furrowsoffedge;
  k:=0;
 end;
end;

/////////////////////////////////////////////////////////////////
// Redraws grid and bitmap onto image canvas.
/////////////////////////////////////////////////////////////////
procedure TfrmAdvanceRate.UpgradeGrid;
begin
 img1.Canvas.StretchDraw(rect(0,0,img1.Width,img1.Height),bmp1);
 DrawFurrows;
end;

/////////////////////////////////////////////////////////////////
// Moves grid centre of vision to the left.
/////////////////////////////////////////////////////////////////
procedure TfrmAdvanceRate.cmdGTLeftClick(Sender: TObject);
begin
 dec(c4.X,stepsize);
 dec(c3.X,stepsize);
 UpgradeGrid;
```

```
end;

/////////////////////////////////////////////////////////////////
// Moves grid centre of vision to the right.
/////////////////////////////////////////////////////////////////
procedure TfrmAdvanceRate.cmdGTRightClick(Sender: TObject);
begin
 inc(c4.X,stepsize);
 inc(c3.X,stepsize);
 UpgradeGrid;
end;

/////////////////////////////////////////////////////////////////
// Moves grid centre of vision upwards.
/////////////////////////////////////////////////////////////////
procedure TfrmAdvanceRate.cmdGTUpClick(Sender: TObject);
begin
 dec(c4.Y,stepsize);
 dec(c3.Y,stepsize);
 UpgradeGrid;
end;

/////////////////////////////////////////////////////////////////
// Moves grid centre of vision downwards.
/////////////////////////////////////////////////////////////////
procedure TfrmAdvanceRate.cmdGTDownClick(Sender: TObject);
begin
 inc(c4.Y,stepsize);
 inc(c3.Y,stepsize);
 UpgradeGrid;
end;

/////////////////////////////////////////////////////////////////
// Stretches grid centre of vision outwards.
/////////////////////////////////////////////////////////////////
procedure TfrmAdvanceRate.cmdGTOutClick(Sender: TObject);
begin
 dec(c4.X,stepsize);
 inc(c3.X,stepsize);
 UpgradeGrid;
end;

/////////////////////////////////////////////////////////////////
// Stretches grid centre of vision inwards.
/////////////////////////////////////////////////////////////////
procedure TfrmAdvanceRate.cmdGTInClick(Sender: TObject);
begin
 inc(c4.X,stepsize);
 dec(c3.X,stepsize);
 UpgradeGrid;
end;

/////////////////////////////////////////////////////////////////
// Moves bottom of grid upwards.
/////////////////////////////////////////////////////////////////
procedure TfrmAdvanceRate.cmdGBUpClick(Sender: TObject);
begin
 dec(c1.Y,stepsize);
 dec(c2.Y,stepsize);
 UpgradeGrid;
end;

/////////////////////////////////////////////////////////////////
// Moves bottom of grid downwards.
/////////////////////////////////////////////////////////////////
procedure TfrmAdvanceRate.cmdGBDownClick(Sender: TObject);
begin
```

```
 inc(c1.Y,stepsize);
 inc(c2.Y,stepsize);
 UpgradeGrid;
end;

/////////////////////////////////////////////////////////////////
// Moves bottom of grid to the left.
/////////////////////////////////////////////////////////////////
procedure TfrmAdvanceRate.cmdGBLeftClick(Sender: TObject);
begin
 dec(c1.X,stepsize);
 dec(c2.X,stepsize);
 UpgradeGrid;
end;

/////////////////////////////////////////////////////////////////
// Moves bottom of grid to the right.
/////////////////////////////////////////////////////////////////
procedure TfrmAdvanceRate.cmdGBRightClick(Sender: TObject);
begin
 inc(c1.X,stepsize);
 inc(c2.X,stepsize);
 UpgradeGrid;
end;

/////////////////////////////////////////////////////////////////
// Changes the increment of grid adjustments.
/////////////////////////////////////////////////////////////////
procedure TfrmAdvanceRate.speStepSizeChange(Sender: TObject);
begin
 stepsize:=speStepSize.Value;
end;

/////////////////////////////////////////////////////////////////
// Changes the number of furrows in the grid foreground.
/////////////////////////////////////////////////////////////////
procedure TfrmAdvanceRate.speNoFurrowsChange(Sender: TObject);
begin
 UpgradeGrid;
end;

/////////////////////////////////////////////////////////////////
// Draws the adjustable grid on the bitmap canvas.
/////////////////////////////////////////////////////////////////
procedure TfrmAdvanceRate.cmdAcceptClick(Sender: TObject);
var
 n,k,d:integer;
 b1,b2,b3,b4:TPoint;
begin
 b4:=point(round(c4.X/img1.Width*bmp1.Width),
           round(c4.Y/img1.Height*bmp1.Height));  //top left
 b3:=point(round(c3.X/img1.Width*bmp1.Width),
           round(c3.Y/img1.Height*bmp1.Height));  //top right
 b1:=point(round(c1.X/img1.Width*bmp1.Width),
           round(c1.Y/img1.Height*bmp1.Height));  //bottom left
 b2:=point(round(c2.X/img1.Width*bmp1.Width),
           round(c2.Y/img1.Height*bmp1.Height));  //bottom right
 bmp1.Canvas.Pen.Color:=clFuchsia;
 bmp1.Canvas.Pen.Width:=1;
 try
  n:=speNoFurrows.Value;   // Number of furrows in foreground
  d:=(furrows+1-n)div 2;
   // (Extrapolates grid to furrows outside foreground of image,
   // assuming equal spacing of furrows)
   for k:=0-d to n+d do
   begin
    p1:=point(round(b1.X+k/n*(b2.X-b1.X)),b1.Y);
```

```
    p2:=point(round(b4.X+k/n*(b3.X-b4.X)),b4.Y);
    bmp1.Canvas.MoveTo(p1.X,p1.Y);
    bmp1.Canvas.LineTo(p2.X,p2.Y);
   end;     // For when spin edit box does not contain integer
   img1.Canvas.StretchDraw(rect(0,0,img1.Width,img1.Height),bmp1);
   InitialiseArray;  // just call this once, during setup
 except
  messagedlg('Number of furrows must be an integer.',mtWarning,
             [mbOK],0);
 end;
end;

//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
// Code for image capture
//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

/////////////////////////////////////////////////////////////////
// Delay.  Amount of delay is approximate.
/////////////////////////////////////////////////////////////////
procedure TfrmAdvanceRate.Delay(Seconds:integer);
var
 i,j,k:integer;
begin
 i:=0;
 j:=0;
 k:=0;
 for i:=0 to Seconds*200 do
 begin
  for j:=0 to 1000 do
  begin
   for k:=0 to 1000 do
   begin
   end;
  end;
 end;
end;

/////////////////////////////////////////////////////////////////
// Send pulse to activate remote control.
/////////////////////////////////////////////////////////////////
procedure TfrmAdvanceRate.TakePhoto;
begin
 try
  CWord:=CWord or RemotePin;
  DLPortIO.Port[PortC]:=CWord;
  Delay(RemoteDelay);
 finally
  CWord:=CWord and not(RemotePin);
  DLPortIO.Port[PortC]:=CWord;
 end;
 Delay(ProcessDelay);
end;

/////////////////////////////////////////////////////////////////
// Wrapper for Windows API CopyFile.
// From http://homepages.borland.com/efg2lab/Library/Delphi/
//           IO/Files.htm
// Viewed 17 April 2004.
/////////////////////////////////////////////////////////////////
FUNCTION CopyFile(CONST SourceFile     :  TFilename;
                  CONST DestinationFile:  TFilename):  BOOLEAN;
BEGIN
  RESULT := Windows.CopyFile(pChar(SourceFile),
                             pChar(DestinationFile),
                             FALSE {Fail if Exists});
END {CopyFile};
```

```
////////////////////////////////////////////////////////////////
// Connect/disconnect camera from USB port.
////////////////////////////////////////////////////////////////
procedure TfrmAdvanceRate.ConnectCamera(CameraStatus:boolean);
begin
 if CameraStatus=True then
  CWord:=CWord or CameraPin         // Connect camera
 else
  CWord:=CWord and not(CameraPin);  // Disconnect camera

 DLPortIO.Port[PortC]:=CWord;       // Send command
 Delay(CameraDelay);                // Wait a bit
end;

////////////////////////////////////////////////////////////////
// Turn spotlight on/off.
////////////////////////////////////////////////////////////////
procedure TfrmAdvanceRate.LampOn(LampStatus:boolean);
begin
 if LampStatus=True then
  CWord:=CWord or LampPin           // Turn lamp on
 else
  CWord:=CWord and not(LampPin);    // Turn lamp off

 DLPortIO.Port[PortC]:=CWord;       // Send command
 Delay(LampDelay);                  // Wait a bit
end;

////////////////////////////////////////////////////////////////
// Move image from camera to local drive.
// Camera filename format is hardwired (this is inflexible and
// needs improvement).
////////////////////////////////////////////////////////////////
procedure TfrmAdvanceRate.DownloadLatestPhoto;
var
 strCurrent:string;
 intMonth:integer;
 strDay:string[2];
 OlympMonths:string[12];
begin
 OlympMonths:='123456789ABC';
 intMonth:=strtoint(formatdatetime('m',now));
 strDay:=formatdatetime('dd',now);
 strCurrent:=OlympusFolder+'P'+OlympMonths[intMonth]+
             strDay+'0001.jpg';
 CopyFile(strCurrent, StoredPhotos+strNewName);
 DeleteFile(strCurrent);
end;

////////////////////////////////////////////////////////////////
// Series of steps to capture image, includes disconnect camera,
// turn on spotlight if necessary, activate remote, turn off
// spotlight, reconnect camera, move image file to local drive
// and load image on screen.
////////////////////////////////////////////////////////////////
procedure TfrmAdvanceRate.CaptureImage;
var
 Dark:boolean;
begin
 labStatus.Caption:='Capturing image';
 Repaint;

 CurrentTime:=strtoint(formatdatetime('hhnn',now));
 CurrentMins:=60*strtoint(formatdatetime('hh',now))+
                 strtoint(formatdatetime('nn',now));

 ConnectCamera(False);           // Disconnect camera

 if (CurrentTime>DarkTimeStart) or (CurrentTime<DarkTimeEnd) then
```

```
  Dark:=True else Dark:=False;   // Check time, if dark want lamp
 if Dark=True then LampOn(True); // If dark, turn lamp on
 strNewName:=formatdatetime('yymmddhhnn',now)+'.jpg';
 TakePhoto;                      // Take photo
 if Dark=True then LampOn(False);
 ConnectCamera(True);            // Connect camera
 DownloadLatestPhoto;
 try
  jpg1.LoadFromFile(StoredPhotos+strNewName);
  bmp1.Assign(jpg1);
  Rotate90(True);
  img1.Canvas.StretchDraw(rect(0,0,img1.Width,img1.Height),bmp1);
 except
  showmessage('No photo - '+StoredPhotos+strNewName);
 end;
 labStatus.Caption:='Ready';
 Repaint;
end;
/////////////////////////////////////////////////////////////////
// Rotates the public bitmap bmp1 by +/- 90 degrees.
// Derived from Crane (1997).
/////////////////////////////////////////////////////////////////
procedure TfrmAdvanceRate.Rotate90(Direction:boolean);
var
 x,y:integer;
 bmp2:TBitmap;
 sli1,sli2:pRGBArray;
begin
 // Rotate bitmap
 bmp2:=TBitMap.Create;
 bmp2.PixelFormat:=pf24bit;
 try
  Bmp2.Width:=Bmp1.Height;
  Bmp2.Height:=Bmp1.Width;
  // Switch bmp1 width with bmp2 height
  for y:=0 to Bmp1.Height-1 do
  begin
   sli1:=Bmp1.ScanLine[y];
   for x:=0 to Bmp1.Width-1 do
   begin
    sli2:=Bmp2.ScanLine[x];
    if direction=True then
     sli2[y]:=sli1[(Bmp1.Width-1)-x]  // rotate left
    else
     sli2[(Bmp1.Height-1)-y] := sli1[x]; // rotate right
   end;
  end;
  Bmp1.Width:=Bmp2.Width;
  Bmp1.Height:=Bmp2.Height;
  bmp1.Canvas.Draw(0,0,bmp2);
 finally
  bmp2.Free;
 end;
end;
/////////////////////////////////////////////////////////////////
// Uses system geometry to calculate distance from a pixel number.
// Requires pixel number.
// Returns distance in metres.
/////////////////////////////////////////////////////////////////
function TfrmAdvanceRate.AssignDistance(px:integer):integer;
var
 towerHeight,towerDist,fieldLength:real;
```

```
 theta0,thetaL,theta:real;
 pxmax:integer;
begin
 towerHeight:=speHeight.Value/10;   // Height of tower
 towerDist:=speDistance.Value/10;   // Distance between base of
                                    //   tower and start of field
 fieldLength:=speLength.Value;      // Length of field

 pxmax:=bmp1.Height;                // Number of pixels in image

 // Variables required for analysis
 theta0:=arctan(towerHeight/towerDist);
 thetaL:=arctan(towerHeight/(towerDist+fieldLength));

 // Calculate distance
 theta:=px/pxmax*(theta0-thetaL)+thetaL;

 if radTail.Checked=True then
  Assigndistance:=500-round(towerHeight/tan(theta)-towerDist)
 else
  Assigndistance:=round(towerHeight/tan(theta)-towerDist);
end;
/////////////////////////////////////////////////////////////////
// Determines advance front position for bare soil case.
// Requires a furrow number and the criteria for determining
// whether water stream has terminated (ie. number of pixels of
// value zero (ZeroCount) within a specified number of rows
// (Space)). Returns a pixel value for when the criteria are met,
// or returns -1 when the criteria are not met (so there is no
// water present in the furrow).
/////////////////////////////////////////////////////////////////
function TfrmAdvanceRate.LookBetweenLines(Furrow,ZeroCount,
                         Space:integer):integer;
var
 k,chk,c:integer;
 av:real;
 sli1:pRGBarray;
 zeroarray,newarray:array[1..2560] of integer;
begin
 for k:=1 to 2560 do
 begin
  zeroarray[k]:=0;
  newarray[k]:=0;
 end;
 for k:=1 to 2560 do
 begin
  // uses array of grid pixel locations
  if FurrowStart[Furrow,k]<>-1 then
  begin
   av:=0;
   sli1:=bmp1.ScanLine[k];
   for c:=FurrowStart[Furrow,k]+1 to FurrowStart[Furrow+1,k]-1 do
   begin
    av:=av+sli1[c].rgbtBlue;   // any colour channel is suitable,
                               // bmp1 is now black and white
   end;
   if av=0 then zeroarray[k]:=1;
  end;
 end;
 k:=1;
 while (k<2560-10) do
 begin
  chk:=0;
  if zeroarray[k]=1 then
  begin
   for c:=k to k+Space do chk:=chk+zeroarray[c];
```

```
   if (chk>ZeroCount) then newarray[k]:=0 else newarray[k]:=1;
  end
  else
   newarray[k]:=1;
  k:=k+1;
 end;
 k:=1;
 while (newarray[k]=1) and (k<2560) do k:=k+1;
 if k<>1 then LookBetweenLines:=k else LookBetweenLines:=-1;
end;

procedure TfrmAdvanceRate.AddResultItem(f,Dist,Time:integer);
begin
 with lsvResults do
 begin
  Items.Add;
  Items.Item[Items.Count-1].SubItems.Add(inttostr(Dist));
  Items.Item[Items.Count-1].SubItems.Add(inttostr(Time));
  Items.Item[Items.Count-1].Caption:=inttostr(f);
 end;
end;

//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
// Code for bare soil analysis
//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

//////////////////////////////////////////////////////////////////
// Bare soil analysis has two cases - when the water stream
// appears white and when the water stream appears blue.  The
// distinction is based on time of day, and this procedure
// directs the analysis accordingly.
//////////////////////////////////////////////////////////////////
procedure TfrmAdvanceRate.BareSoilAnalysis;
var
 f:integer;
 Bright:boolean;
begin
 StartTime:=strtoint(formatdatetime('hhmm',dtpStart.Time));
 StartMins:=60*strtoint(formatdatetime('hh',dtpStart.Time))+
               strtoint(formatdatetime('nn',dtpStart.Time));
 CurrentMins:=60*strtoint(formatdatetime('hh',now))+
                 strtoint(formatdatetime('nn',now));

 Bright:=True;
 if (CurrentTime>DullStart) and (CurrentTime<DullStop) then
  Bright:=False;

 If Bright=True then
  HueAnalysis          // blue water stream
 else
  GrayscaleAnalysis;   // white water stream

 for f:=1 to furrows do   // Furrow number
 begin
  pxChange:=-1;
  if Bright=True then
   pxChange:=LookBetweenLines(f,2,10)
  else
   pxChange:=LookBetweenLines(f,10,20);
  showmessage(inttostr(pxchange));
  if pxChange<>-1 then
  begin
   Distance:=AssignDistance(pxChange);
   AddResultItem(f,Distance,CurrentMins-StartMins);
  end;
 end;
```

```
end;
///////////////////////////////////////////////////////////////////
// Grayscale Analysis.
// Alters public variable bmp1.
///////////////////////////////////////////////////////////////////
procedure TfrmAdvanceRate.GrayscaleAnalysis;
var
 k,i,gray:integer;
 r,g,b,h:real;
 sli1:pRGBarray;
begin
 for k:=0 to bmp1.Height-1 do
 begin
  sli1:=bmp1.ScanLine[k];
  for i:=0 to bmp1.Width-1 do
  begin
   // calculate rgb grayscale
   gray:=(sli1[i].rgbtRed+sli1[i].rgbtGreen+
           sli1[i].rgbtBlue) div 3;
   if gray>235 then  // threshold image at a value of 235
   begin
    sli1[i].rgbtRed:=255;
    sli1[i].rgbtGreen:=255;
    sli1[i].rgbtBlue:=255;
   end
   else
   begin
    sli1[i].rgbtRed:=0;
    sli1[i].rgbtGreen:=0;
    sli1[i].rgbtBlue:=0;
   end;
  end;
 end;
end;
///////////////////////////////////////////////////////////////////
// Hue Analysis.
// Alters public variable bmp1.
///////////////////////////////////////////////////////////////////
procedure TfrmAdvanceRate.HueAnalysis;
var
 k,i,hue:integer;
 r,g,b,h:real;
 sli1:pRGBarray;
begin
 StartTime:=strtoint(formatdatetime('hhmm',dtpStart.Time));
 for k:=0 to bmp1.Height-1 do
 begin
  sli1:=bmp1.ScanLine[k];
  for i:=0 to bmp1.Width-1 do
  begin
   if (sli1[i].rgbtRed=sli1[i].rgbtGreen) and
      (sli1[i].rgbtGreen=sli1[i].rgbtBlue) then
    h:=0   // h is undefined if r=g or g=b
   else
   begin
    // calculate normalised r, g, b values
    r:=sli1[i].rgbtRed/(sli1[i].rgbtRed+
       sli1[i].rgbtGreen+sli1[i].rgbtBlue);
    g:=sli1[i].rgbtGreen/(sli1[i].rgbtRed+
       sli1[i].rgbtGreen+sli1[i].rgbtBlue);
    b:=sli1[i].rgbtBlue/(sli1[i].rgbtRed+
       sli1[i].rgbtGreen+sli1[i].rgbtBlue);
    h:=arccos(((r-g)+(r-b))/2/(sqrt((r-g)*(r-g)+(r-b)*(g-b))));
   end;
```

```
   if b>g then h:=2*pi-h;
   hue:=round(h*256/(2*pi))-1; // convert h to the range 0 to 255
   if hue>55 then              // threshold hue at a value of 55
   begin
    sli1[i].rgbtRed:=255;
    sli1[i].rgbtGreen:=255;
    sli1[i].rgbtBlue:=255;
   end
   else
   begin
    sli1[i].rgbtRed:=0;
    sli1[i].rgbtGreen:=0;
    sli1[i].rgbtBlue:=0;
   end;
  end;
 end;
end;

//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
// Code for closed canopy analysis
//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

/////////////////////////////////////////////////////////////////
// Estimates advance front position from crop canopy.
/////////////////////////////////////////////////////////////////
procedure TfrmAdvanceRate.ClosedCanopyAnalysis;
begin
 StartTime:=strtoint(formatdatetime('hhmm',dtpStart.Time))-
            speDelay.Value;
 StartMins:=60*strtoint(formatdatetime('hh',dtpStart.Time))+
                 strtoint(formatdatetime('nn',dtpStart.Time))-
                 speDelay.Value;
 CurrentMins:=60*strtoint(formatdatetime('hh',now))+
                 strtoint(formatdatetime('nn',now));

 if CloudCoverCheck=True then
 begin
   // Insert closed canopy code here.
 end
 else
  Timer1Timer(Self);  // Take another image
end;

/////////////////////////////////////////////////////////////////
// Uses the slope of the pixel intensity histogram for the image
// to determine whether cloud cover is uniform.
// Uses the public variables altered by procedure HistConstruct.
// Returns true for uniform cloud cover and false for non-uniform
// cloud cover.
/////////////////////////////////////////////////////////////////
function TfrmAdvanceRate.CloudCoverCheck:boolean;
var
 i:integer;
begin
 HistConstruct('rgbtBlue');
 for i:=0 to 254 do FirstDiff[i]:=0;
 for i:=0 to 251 do FirstDiff[i]:=HistCount[i+1]-HistCount[i];

 // Find the array index corresponding to maximum histogram slope
 i:=MaxIndex(FirstDiff);

 // If slope of histogram is repeatedly larger than 1000, cloud
 // cover exists in image so the image is not suitable for
 // analysis
 if (FirstDiff[i]>1000) and ((FirstDiff[i-2]>1000) or
                             (FirstDiff[i+2]>1000))
```

```
 then CloudCoverCheck:=False else CloudCoverCheck:=True;
end;

/////////////////////////////////////////////////////////////////
// Finds the index of the maximum array value.
// Requires an array of integers.
// Returns the index of the maximum array value.
/////////////////////////////////////////////////////////////////
function TfrmAdvanceRate.MaxIndex(ArrayInt:array of integer):
                         integer;
var
 x,max,index:integer;
begin
 max:=ArrayInt[0];
 index:=0;
 for x:=1 to 254 do
 begin
  if ArrayInt[x]>max then
  begin
   max:=ArrayInt[x];
   index:=x;
  end;
 end;
 MaxIndex:=index;
end;

/////////////////////////////////////////////////////////////////
// Constructs a pixel intensity histogram.
// Uses public bitmap image bmp1 but requires the colour channel
// (as a string) for which the histogram is to be constructed.
// Stores an array containing histogram counts in the public
// variable HistCount.
/////////////////////////////////////////////////////////////////
procedure TfrmAdvanceRate.HistConstruct(channel:string);
var
 i,x,y:integer;
 sli1:pRGBarray;
begin
 for i:=0 to 255 do HistCount[i]:=0;
 for y:=0 to bmp1.Height-1 do
 begin
  sli1:=bmp1.ScanLine[y];
  if channel='rgbtRed' then for x:=0 to bmp1.Width-1 do
    inc(HistCount[sli1[x].rgbtRed])
  else if channel='rgbtGreen' then for x:=0 to bmp1.Width-1 do
    inc(HistCount[sli1[x].rgbtGreen])
  else if channel='rgbtBlue' then for x:=0 to bmp1.Width-1 do
    inc(HistCount[sli1[x].rgbtBlue])
 end;
end;

//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
// Code for intialising and closing form
//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

procedure TfrmAdvanceRate.FormCreate(Sender: TObject);
begin
 // Initialise bitmap variable
 bmp1:=TBitmap.Create;
 bmp1.Width:=PhotoWidth;
 bmp1.Height:=PhotoHeight;
 bmp1.PixelFormat:=pf24bit;

 // Initialise jpeg variable
 jpg1:=TJPEGImage.Create;
```

```
 // Outermost corners of grid, initial settings
 c4:=point(round(img1.Width/3),0);              //top left
 c3:=point(round(2*img1.Width/3),0);            //top right
 c1:=point(0,img1.Height);                      //bottom left
 c2:=point(img1.Width,img1.Height);             //bottom right

 // Draw default grid on image canvas
 stepsize:=speStepSize.Value;
 DrawFurrows;

 FBaseMode:=bHex;       // Hex mode as default
 FDataType:=dtaByte;    // Word mode as default

 // Driver is in the same directory as the demo.exe file!
 DLPortIO.DriverPath := ExtractFileDir(ParamStr(0));

 // Open the DriverLINX driver
 DLPortIO.OpenDriver();
 if (not DLPortIO.ActiveHW) then
  MessageDlg('Could not open the DriverLINX driver.', mtError,
             [mbOK], 0);

 CWord:=$00 or CameraPin;  // Ensure camera connection
 DLPortIO.Port[PortC]:=CWord;

 strNewName:='';        // Photo filename
 IntervalCounter:=0;    // Counter for bare soil analysis timer
 pxChange:=-1;          // Advance front position in pixels
 Distance:=0;           // Advance front position in metres
 CurrentTime:=0;

 // If StoredPhotos folder doesn't exist then create it
 ForceDirectories(StoredPhotos);
end;

procedure TfrmAdvanceRate.cmdExitClick(Sender: TObject);
begin
 Timer1.Enabled:=False;
 Close;
 frmStartUp.Close;
end;

procedure TfrmAdvanceRate.FormClose(Sender: TObject;
  var Action: TCloseAction);
begin
 // Free memory used by image variables
 bmp1.Free;
 jpg1.Free;
end;
/////////////////////////////////////////////////////////////////
// Enables timers and determines image capturing intervals in
// minutes.
// Timer interval is in milliseconds.
/////////////////////////////////////////////////////////////////
procedure TfrmAdvanceRate.cmdStartTimerClick(Sender: TObject);
begin
 // Initialise non-daylight hours
 DarkTimeStart:=strtoint(edDay2.Text);
 DarkTimeEnd:=strtoint(edDay1.Text)+12;

 if radBare.Checked=True then
 begin
  Analysis:=atBareSoil;
 // Update timer interval
  IntervalCounter:=succ(IntervalCounter);
  try
   Timer1.Interval:=strtoint(edIntervals.Text
     [1+2*(IntervalCounter-1)])*60*60*1000;
```

```
  except
   // If interval is not specified, use 2 hours
   Timer1.Interval:=1000*60*60*2;
 end;
 end
 else
 begin
  Analysis:=atClosedCanopy;
  // Take photos every twenty minutes
  Timer1.Interval:=1000*60*20;
 end;
 Timer1.Enabled:=True;
 labStatus.Caption:='Next photo at ' +
                    inttostr(round(Timer1.Interval/1000/60)) +
                    ' minutes after ' +
                    formatdatetime('h:mm ampm',now);
end;

/////////////////////////////////////////////////////////////////
// Procedure the user runs which enables adjustment of the
// viewing window (that is, the amount of cropping and rotation
// each captured image undergoes before analysis).
/////////////////////////////////////////////////////////////////
procedure TfrmAdvanceRate.cmdCaptureClick(Sender: TObject);
begin
 // Captures image from camera
 CaptureImage;

 // Draws grid on image on screen
 DrawFurrows;
end;

/////////////////////////////////////////////////////////////////
// Conduct analysis.
/////////////////////////////////////////////////////////////////
procedure TfrmAdvanceRate.Timer1Timer(Sender: TObject);
begin
 CaptureImage;
 labStatus.Caption:='Conducting analysis';
 if Analysis=atBareSoil then
  BareSoilAnalysis
 else
  ClosedCanopyAnalysis;
 labStatus.Caption:='Next photo at ' +
                    inttostr(round(Timer1.Interval/1000/60)) +
                    ' minutes after ' +
                    formatdatetime('h:mm ampm',now);
end;

/////////////////////////////////////////////////////////////////
// Suspends advance rate measurement software.
/////////////////////////////////////////////////////////////////
procedure TfrmAdvanceRate.cmdStopTimerClick(Sender: TObject);
begin
 Timer1.Enabled:=False;
 labStatus.Caption:='Ready';
end;

/////////////////////////////////////////////////////////////////
// Analyses the current image independent of the timer status
// (for testing purposes).
/////////////////////////////////////////////////////////////////
procedure TfrmAdvanceRate.cmdAnalyseNowClick(Sender: TObject);
begin
 if radBare.Checked=True then
  Analysis:=atBareSoil
 else
  Analysis:=atClosedCanopy;
```

```
 labStatus.Caption:='Conducting analysis';
 CurrentTime:=60*strtoint(formatdatetime('hh',now))+
              strtoint(formatdatetime('nn',now));
 if Analysis=atBareSoil then
  BareSoilAnalysis
 else
  ClosedCanopyAnalysis;
 labStatus.Caption:='Ready';
end;
/////////////////////////////////////////////////////////////////
// Creates datafiles compatible with Irrimate.
/////////////////////////////////////////////////////////////////
procedure TfrmAdvanceRate.cmdCreateFilesClick(Sender: TObject);
var
 i,j:integer;
 Folder,Filename:string;
 fi:textfile;
 DataExists,blnFirst:boolean;
begin
 if opd1.Execute then
 begin
  Folder:=ExtractFilePath(opd1.FileName);
  for i:=1 to furrows do
  begin
   DataExists:=False;
   // Before creating a file, check that data is in the listview
   //  for this furrow number
   for j:=0 to lsvResults.Items.Count-1 do
    if lsvResults.Items[j].Caption=inttostr(i) then
      DataExists:=True;
   // Data for this furrow exists in the listview
   if DataExists=True then
   begin
    Filename:=Folder + '\Plot[' + inttostr(i) + '].dat';
    AssignFile(fi, FileName);
    Rewrite(fi);

    Write(fi, strtofloat(edInflow.Text):6:4);   // inflow rate
    Write(fi, '"inflow rate m^3/min"');
    WriteLn(fi);

    // cross-sectional area of flow
    Write(fi, strtofloat(edAreaFlow.Text):6:4);
    Write(fi, '"cross-sectional area of flow, m^2"');
    WriteLn(fi);

    Write(fi, strtofloat(edSigmaY.Text):6:4);   // sigma y
    Write(fi, '"sigma y"');
    WriteLn(fi);

    Write(fi, speNoPoints.Value);   // advance points
    Write(fi, '"number of advance points"');
    WriteLn(fi);

    blnFirst:=True;
    // Look through listview for all the entries for this furrow
    //  number
    for j:=0 to lsvResults.Items.Count-1 do
     //
     if lsvResults.Items[j].Caption=inttostr(i) then
     begin
      Write(fi, lsvResults.Items[j].SubItems[0]);
      Write(fi, ',');
      Write(fi, lsvResults.Items[j].SubItems[1]);
      if blnFirst=True then
          // Only puts the following line for the first advance
          // point
```

```
          Write(fi, '"advance points :distance (m),time(m)"');
      WriteLn(fi);
      blnFirst:=False;
     end;
    Closefile(fi);
   end;
  end;
 end;
end;

end.
```

Appendix D

Plant Graphs

The following plot indicates the absolute red, green and blue channel values for the images of the watered and unwatered plants, before and after watering.

The absolute colour values do not exhibit any identifiable pattern. The difference between the dry and wet plant colour values, which did yield a pattern, is plotted in the body of the report (Figure 5.1).

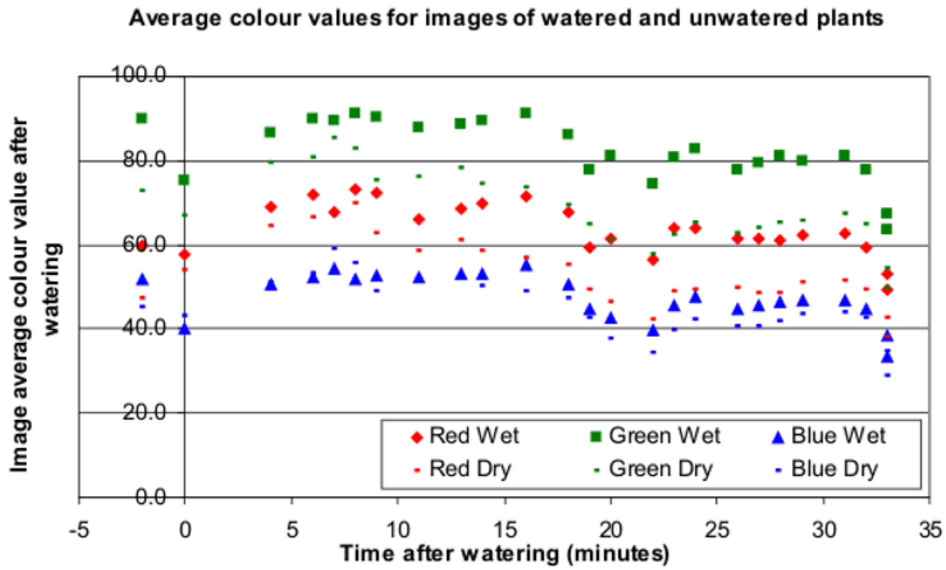
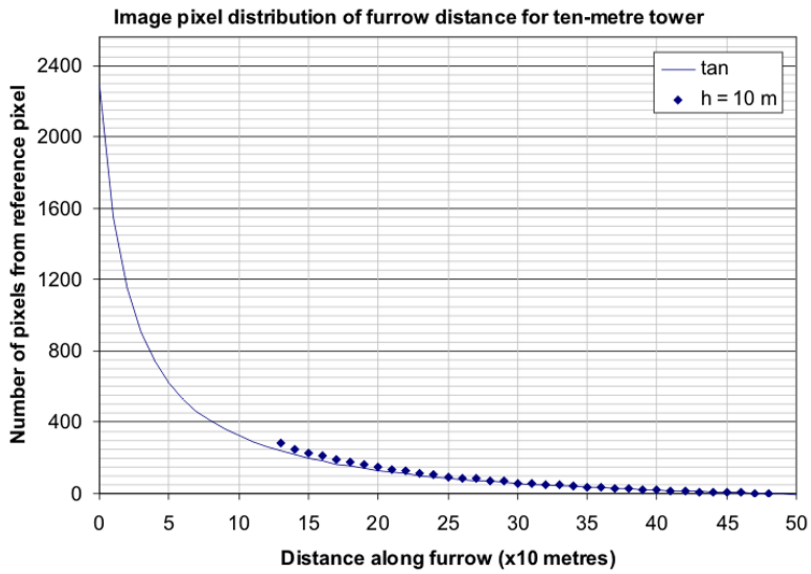


Figure D.1: Absolute plant image colour values following watering

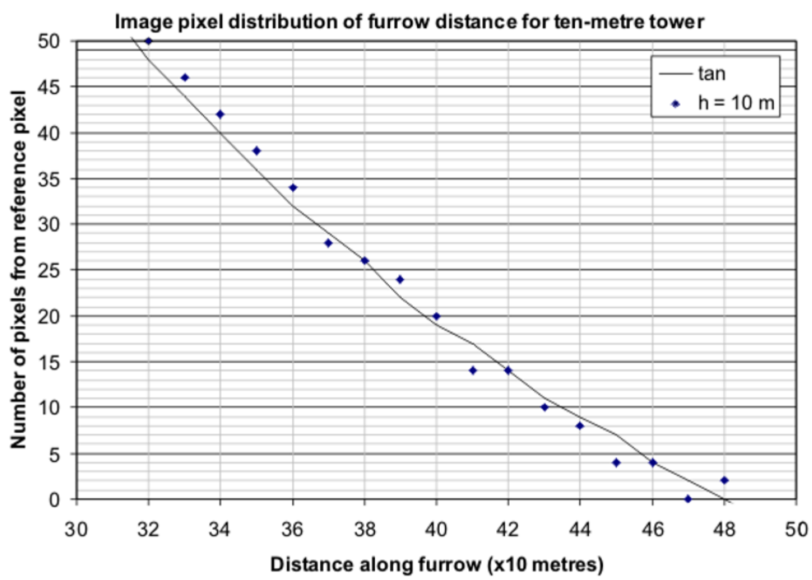
Appendix E

Ten-metre Tower Readings

Figure E.1 contains the pixel distribution for furrow distance for the ten-metre tower, for both experimental data and mathematical model. The parameters used in the mathematical model were distance between tower and start of field x_0 = 7.4 metres, height of tower h = 10 metres and length of field 480 metres.



(a)



(b)

Figure E.1: Resolution from ten-metre tower (a) for 0 to 500 metres (b) for 300 to 500 metres

The original distribution is not a smooth curve but for the distance range of 300 to 480 metres the modelled distribution is generally within 10 metres of the original distribution.

Appendix F

Further Image Processing Results

F.1 Colour model conversions

The following conversions have been taken from Crane (1997). Of these conversions, the hue conversion of the HSI colour model has been coded in the advance rate measurement software.

F.1.1 RGB to CMYK

1. Convert from RGB to CMY (where RGB values are normalised between 0 and 255):

$$\begin{aligned} C &= 1.0 - R \\ M &= 1.0 - G \\ Y &= 1.0 - B \end{aligned}$$

2. Convert from CMY to CMYK:

$$\begin{aligned} K &= \min(C, M, Y) \\ C &= C - K \\ M &= M - K \\ Y &= Y - K \end{aligned}$$

F.1.2 RGB to HSI (double cone model)

R, G and B must be normalised, that is, each colour component must be expressed as a fraction of the sum of all colour components. The returned variables are H, S and I, where H is an angle between 0 and 360°, and S and I are values between 0 and 1. H, S and I can be scaled to the range 0 to 255 to be viewed in grayscale. H is undefined when S or I equals zero.

$$\begin{aligned} I &= \frac{1}{3}(R+G+B) \\ S &= 1 - \frac{3}{R+G+B}\left[\min(R,G,B)\right] \\ H &= \cos^{-1}\left[\frac{\frac{1}{2}\left[(R-G)+(R-B)\right]}{\sqrt{(R-G)^2+(R-B)(G-B)}}\right] \end{aligned}$$

If $B > G$, $H = 360° - H$.

F.2 Colour model and channel study for irrigation featuring crop canopy

Figure F.1 features an image of a wheat crop irrigation in early afternoon separated into the colour channels of the RGB, HSI and CMYK colour models. From these images, the hue channel of the HSI colour model extenuates the water stream from the soil and surrounding crop foliage to the greatest effect.

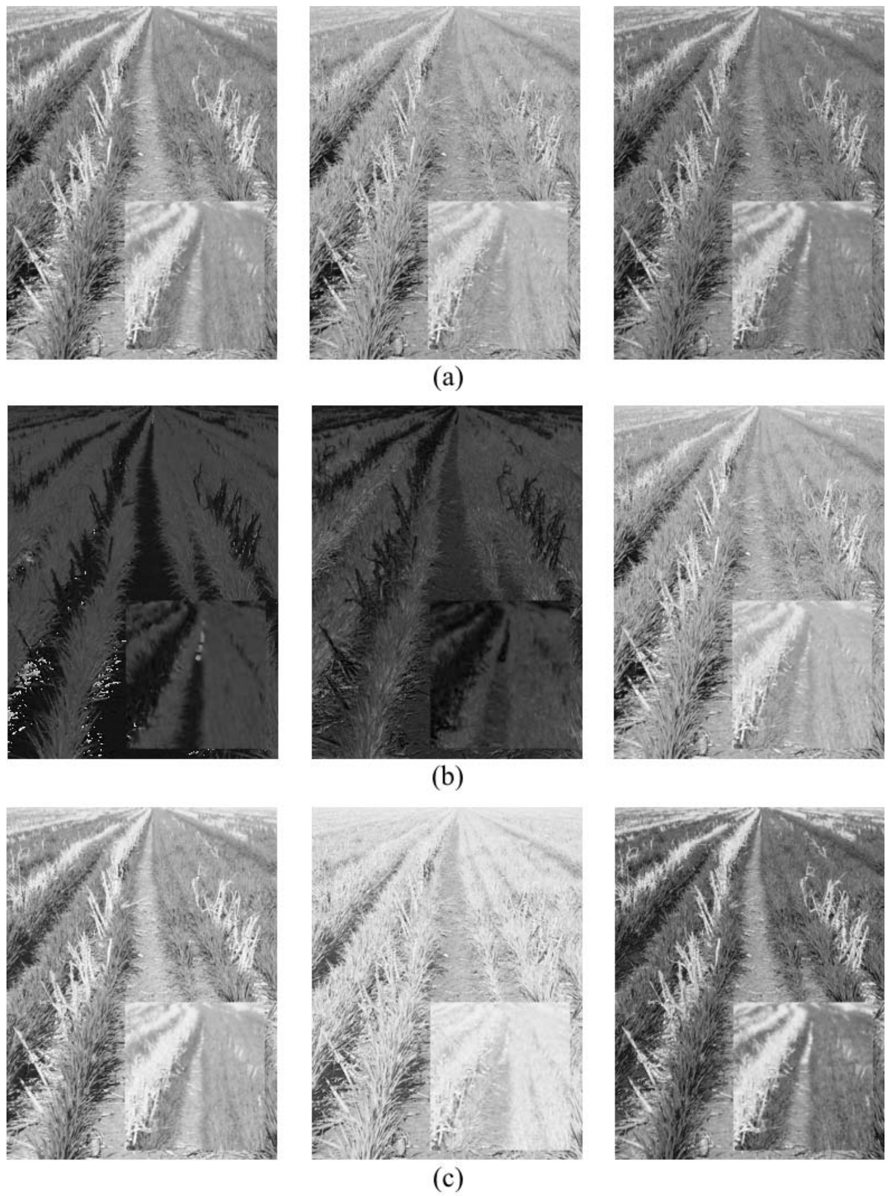


Figure F.1: Sample images from furrow irrigation event in early afternoon (with crop canopy) separated into channels of RGB, HSI and CMYK colour models (a) red, green and blue channels (b) hue, saturation and intensity channels (c) cyan, magenta and yellow channels

F.3 Determination of advance front position from thresholded image

The following series of images are the plots of pixel values along each furrow in an image thresholded to extenuate the water stream, for the early morning irrigation. From these plots, the location of the advance front occurs at the pixel row number where the pixel values begins prevalently equalling zero.

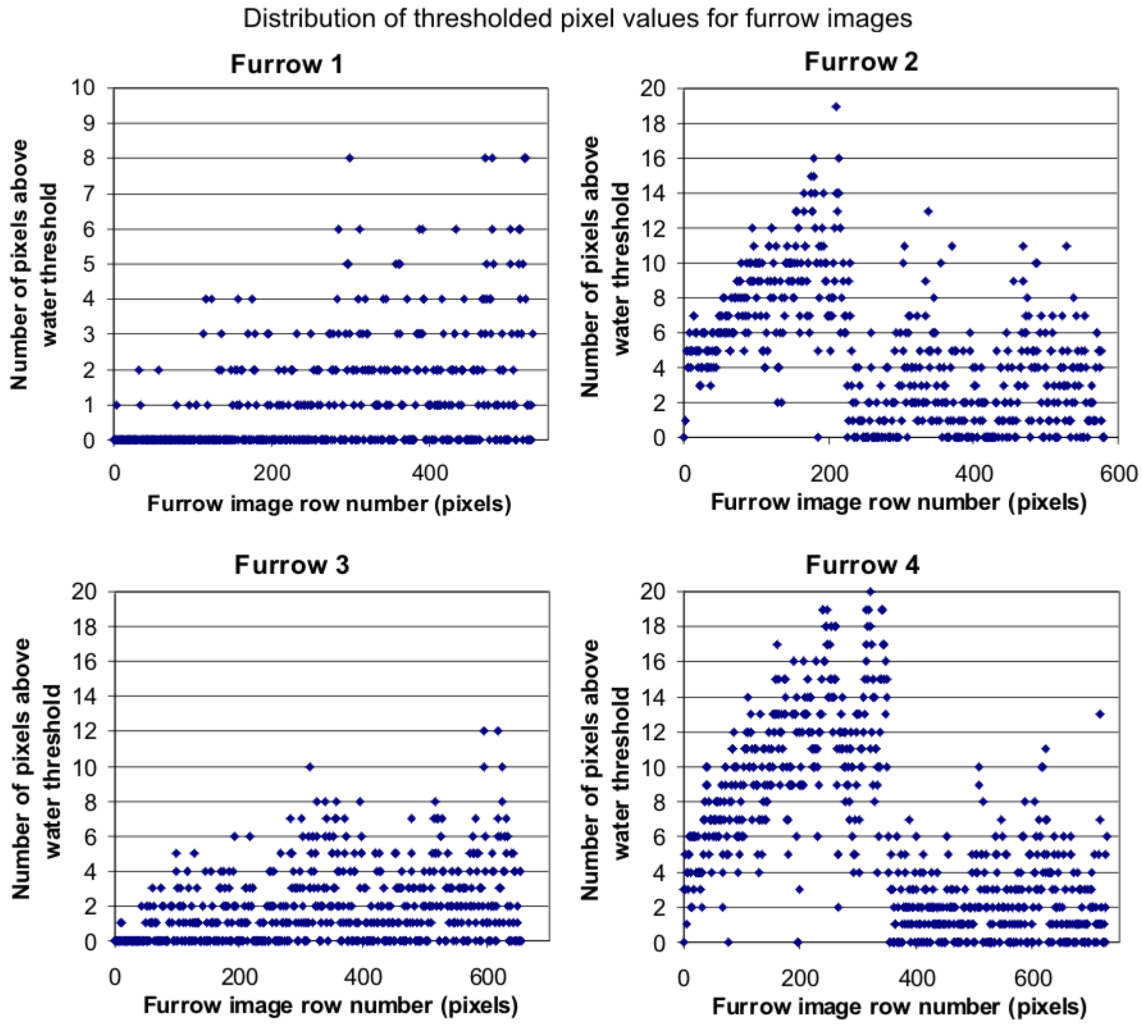


Figure F.2: Graphs indicating distribution of thresholded pixels for furrow images, from irrigation in early morning sample image Part 1

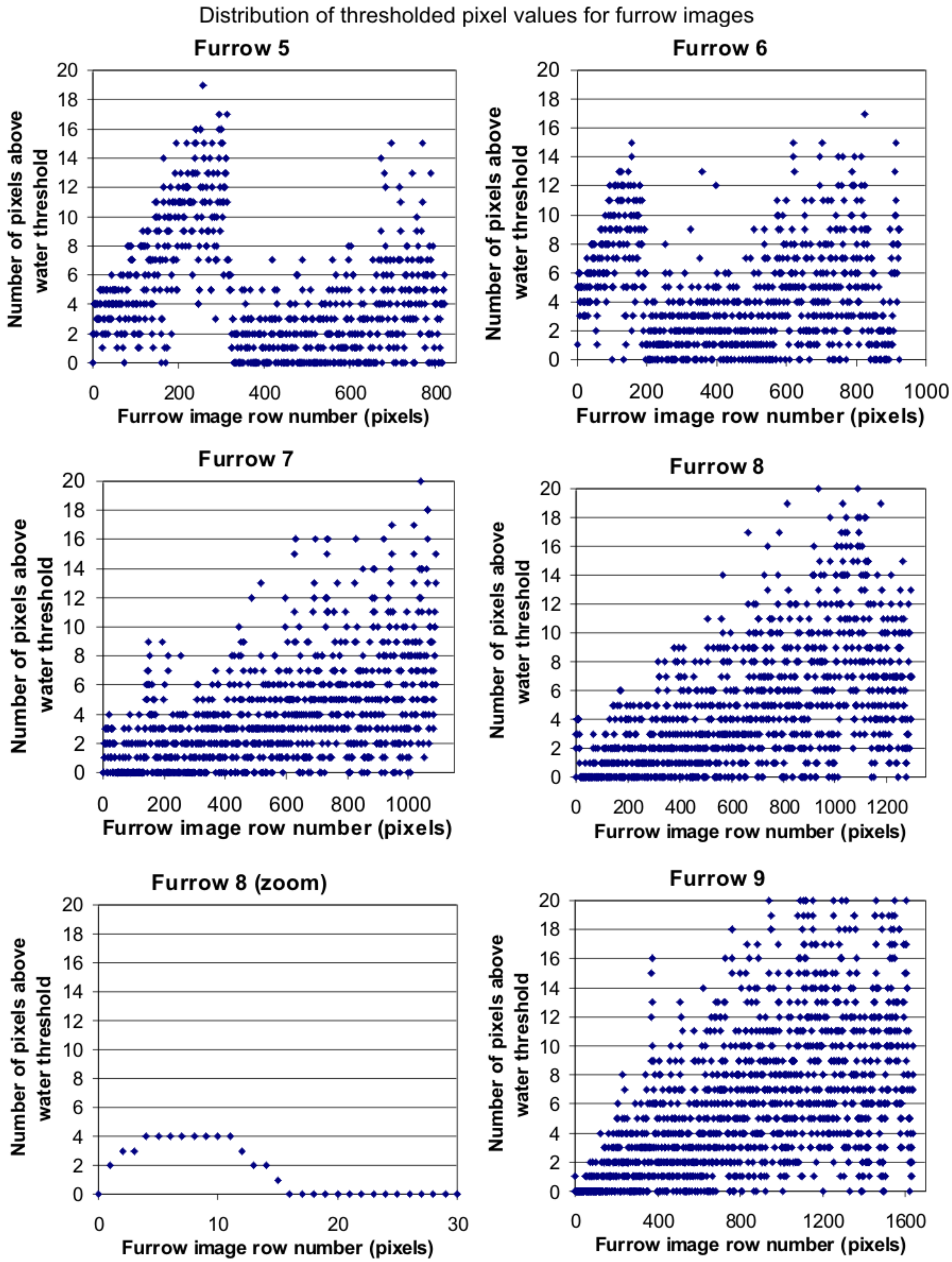


Figure F.3: Graphs indicating distribution of thresholded pixels for furrow images, from irrigation in early morning sample image Part 2

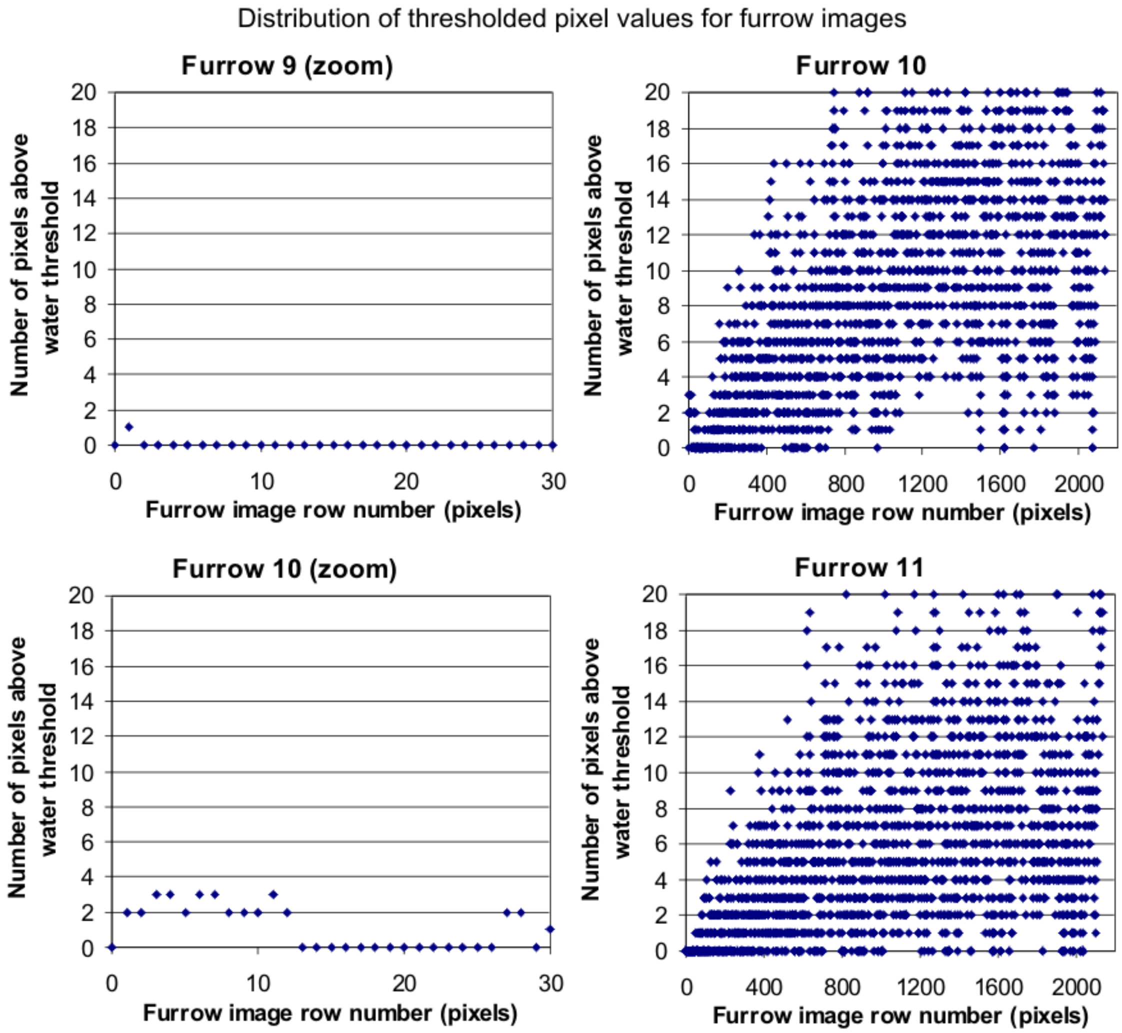


Figure F.4: Graphs indicating distribution of thresholded pixels for furrow images, from irrigation in early morning sample image Part 3

Appendix G

Irrimate Data Formats

G.1 Irrimate 2002 advancemeter communications protocol

Advance readings from the Irrimate advancemeters are downloaded via an infrared link onto a Palmtop computer. The communications protocol for the Irrimate 2002 Advancemeters is:

1200baud, no parity, 8 data bits, 1 stop bit
IRDA standard (ie. $\frac{3}{16}$ bit encoding)

Data packet 20 bytes (each line 1 byte):
AA (hex startbyte)
box number (1 - 6)
furrow1 time low byte
furrow1 time high byte
furrow2 time low byte
furrow2 time high byte
furrow3 time low byte
furrow3 time high byte
furrow4 time low byte
furrow4 time high byte
furrow5 time low byte
furrow5 time high byte
furrow6 time low byte
furrow6 time high byte
furrow7 time low byte
furrow7 time high byte
furrow8 time low byte
furrow8 time high byte
checksum
55 (hex stopbyte)

The 16 bit time is the number of minutes between the box being reset and the water reaching the corresponding furrow sensors. Checksum byte is the sum of all packet bytes excluding start, stop and checksum bytes.

G.2 Infilt input data file

Infilt yields the infiltration characteristic for furrows in the furrow irrigation event and requires as input a data file that is entered manually from the Palmtop computer data. The Infilt input data file (*.dat) features the following fields. A separate file is required for each furrow.

inflow rate m^3/min
cross-sectional area of flow, m^2
sigma y
number of advance points
distance 1, time 1
distance 2, time 2
distance 3, time 3
⋮
distance n, time n

Distance and time are the advance point data pairs. Distance is measured in metres and time in minutes.

Appendix H

Image Analysis Software and Operation

H.1 Software setup procedure

The setup procedure for the advance rate measurement software is as follows. This procedure is required to be conducted after the tower is erected.

1. Initialise software.
   (a) Load software.
   (b) Turn on hardware as indicated in startup screen (Figure H.1).
2. Setup camera viewing window.
   (a) Capture and download image.
   (b) View image. If image can be improved by adjusting camera orientation, lower and adjust camera accordingly. Return to Step 2a and repeat.
   (c) Specify the required rotation and cropping of the image such that the image features a level horizon, and the bounds of the image are defined by the start and end of the field.
   (d) Overlay a grid on the rows or furrows.
3. Input options relevant to image analysis.
   (a) Soil visible for the entire field or closed canopy. If closed canopy:
      i. Delay between arrival of water and response in canopy (minutes).
      ii. Reference position on image.
   (b) Start time of irrigation.
   (c) Estimated run time or finish time (to the nearest hour).
   (d) (Optional) Times at which to capture images. Default is equal intervals between start and finish time.
   (e) Camera at tail drain or head ditch.
   (f) Estimated daylight hours.
   (g) Folder for *.dat files containing advance data.
4. Input mathematical parameters.
   (a) Height of tower, distance between tower and start of field, and length of field.
   (b) Other parameters for Infilt input data file: inflow rate, cross-sectional area of flow, sigma y and number of advance points.
5. Start timer.

At the end of the irrigation event, close software application, power down components and dismantle apparatus. Copy the datafiles as required.

Figure H.2 is a screen capture of the advance rate measurement software, featuring options mentioned in the preceding steps. Aspects of this software which are not complete include the closed canopy analysis procedure (excepting the cloud cover check), the image window adjustment options, the append data file command and the selection of a reference point on the displayed image (required for closed canopy analysis).

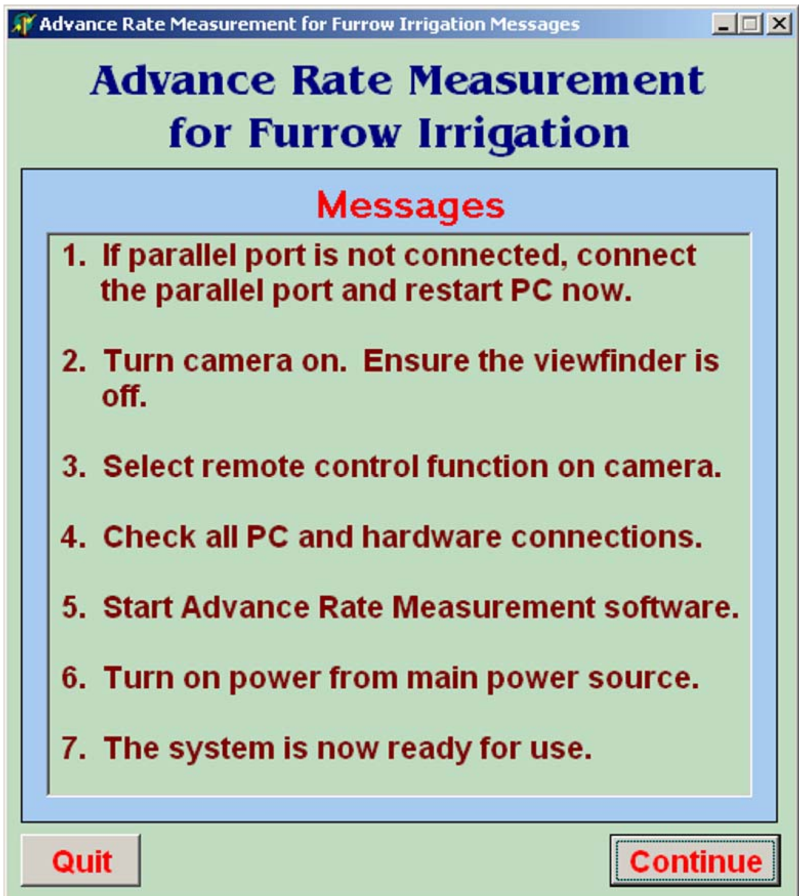


Figure H.1: Hardware setup instructions from advance rate measurement software

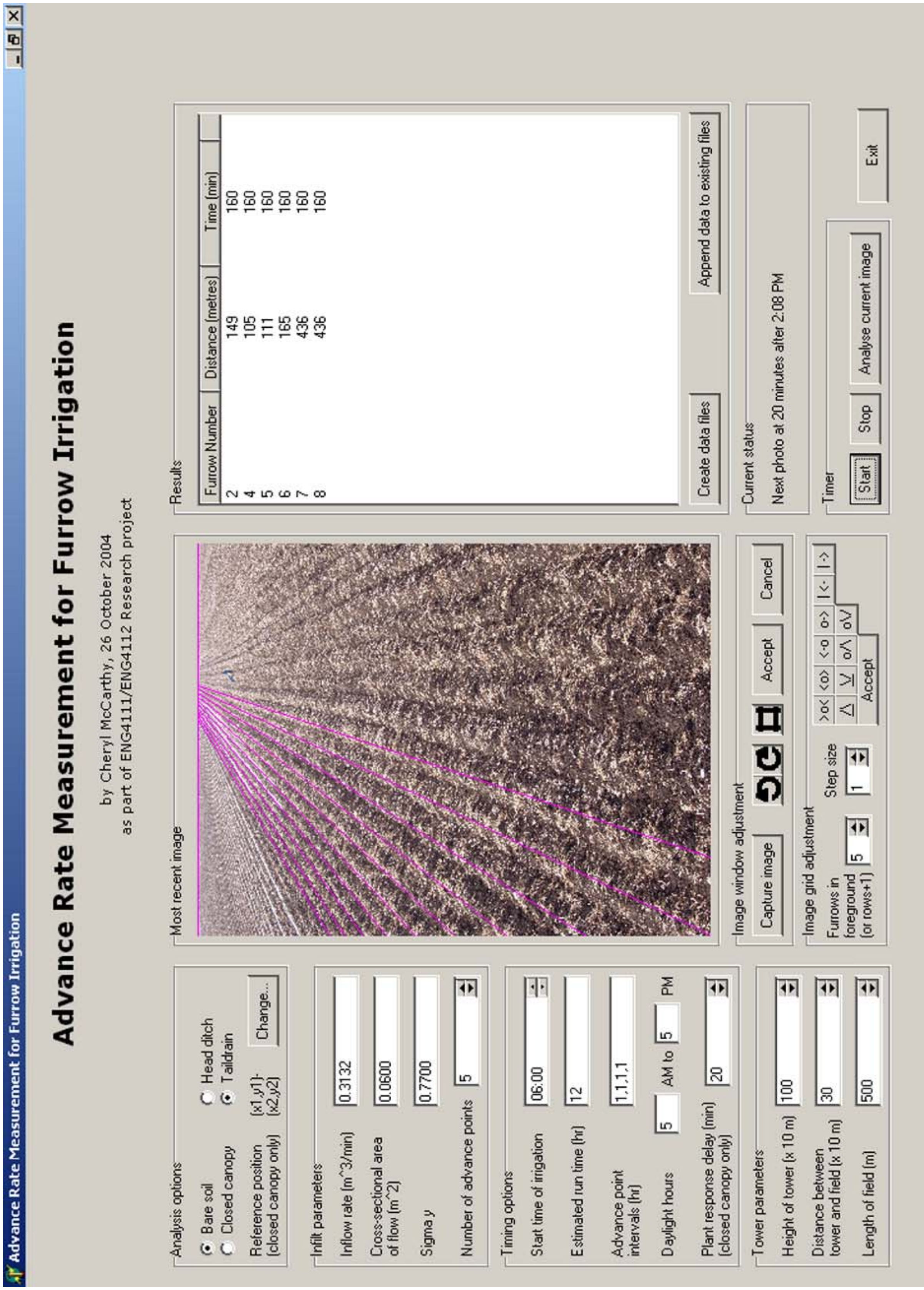


Figure H.2: Advance rate measurement software screen

H.2 Software image analysis flowcharts

H.2.1 Bare soil case

The following image analysis process is conducted for irrigations conducted when bare soil is visible. This process occurs at times specified by by the user.

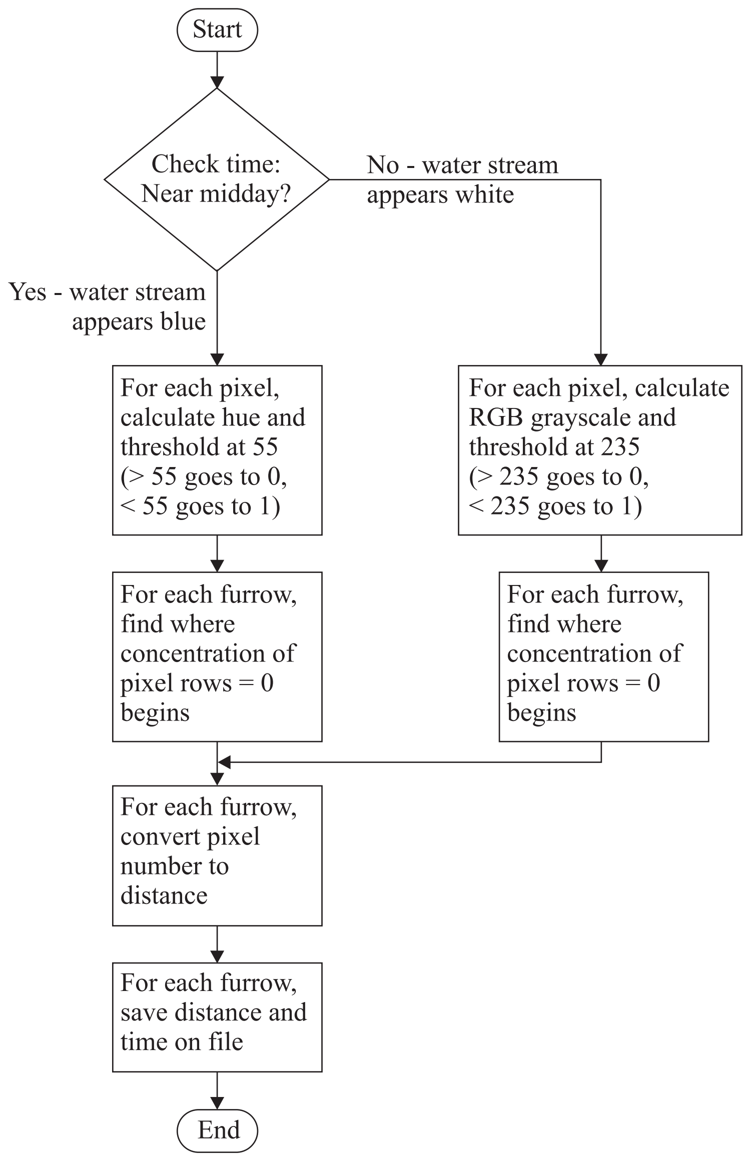


Figure H.3: Flow chart for image analysis of bare soil irrigations

H.2.2 Closed canopy case

For irrigations conducted when the crop canopy is closed, the following image analyis procedure is implemented. This process occurs at intervals of twenty minutes throughout the irrigation event.

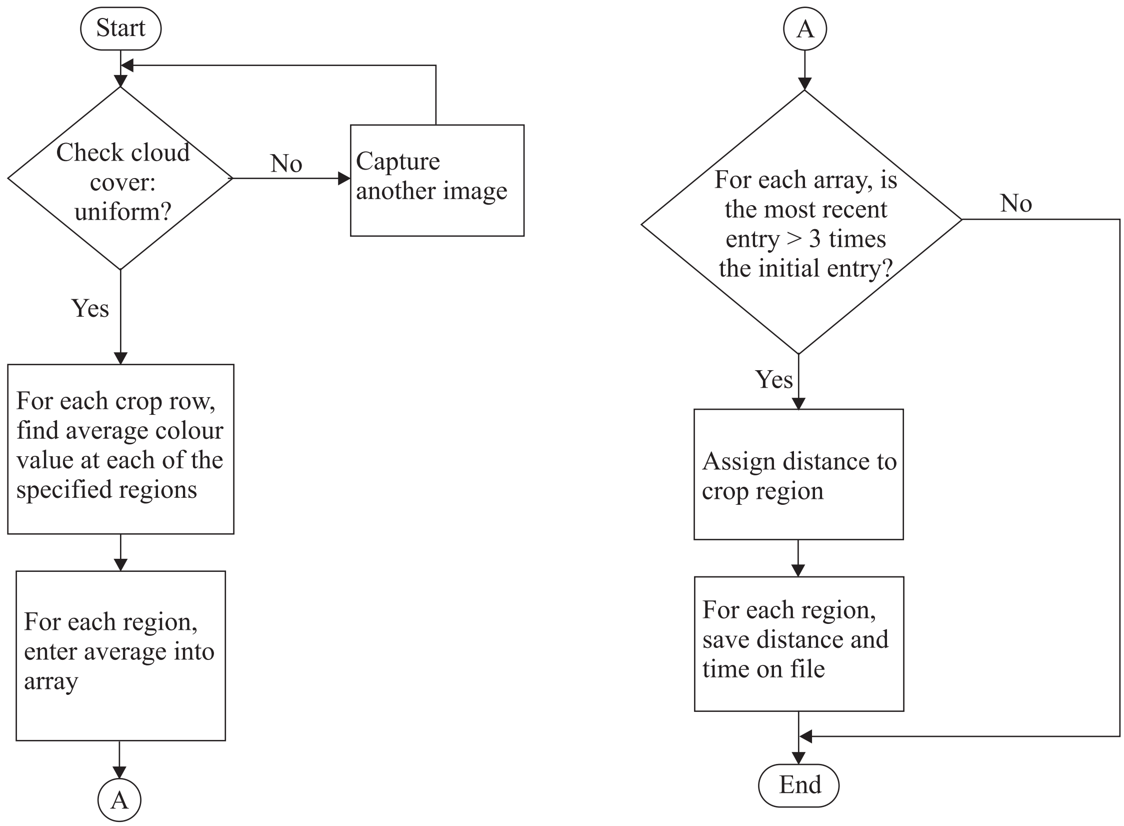


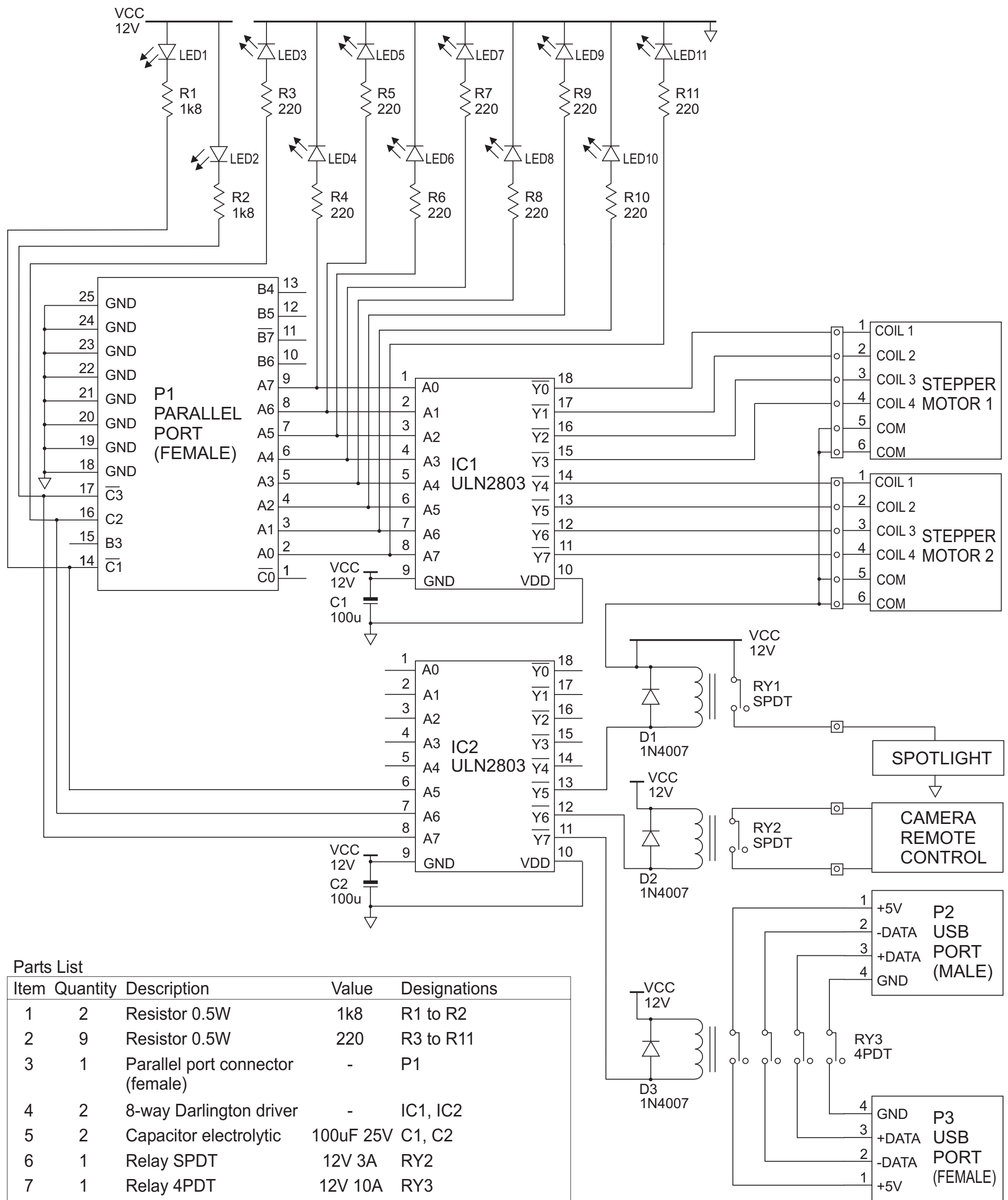
Figure H.4: Flow chart for image analysis of closed canopy irrigations

Appendix I

Interfacing Circuit and Software Flowchart

I.1 Interfacing circuit schematic diagram

The following circuit was designed and built to interface the spotlight, stepper motors and camera via a PC parallel port.



Parts List

Item	Quantity	Description	Value	Designations
1	2	Resistor 0.5W	1k8	R1 to R2
2	9	Resistor 0.5W	220	R3 to R11
3	1	Parallel port connector (female)	-	P1
4	2	8-way Darlington driver	-	IC1, IC2
5	2	Capacitor electrolytic	100uF 25V	C1, C2
6	1	Relay SPDT	12V 3A	RY2
7	1	Relay 4PDT	12V 10A	RY3
8	1	Relay SPDT	30V 10A	RY1
9	4	LED Yellow 5mm	-	LED4 to LED7
10	5	LED Orange 5mm	-	LED1, LED8 to LED11
11	1	LED Green 5mm	-	LED2
12	1	LED Red 5mm	-	LED3
13	3	Diode 1N4007	1000V 1A	D1 to D3
14	1	USB connector (male)	-	P2
15	1	USB connector (female)	-	P3

Figure I.1: Schematic diagram and parts list for interfacing circuit

I.2 Flowchart for image capturing using interfaced components

The flowchart included below indicates the sequence of events used for software control of the interfacing circuit, for the process of image capture and download.

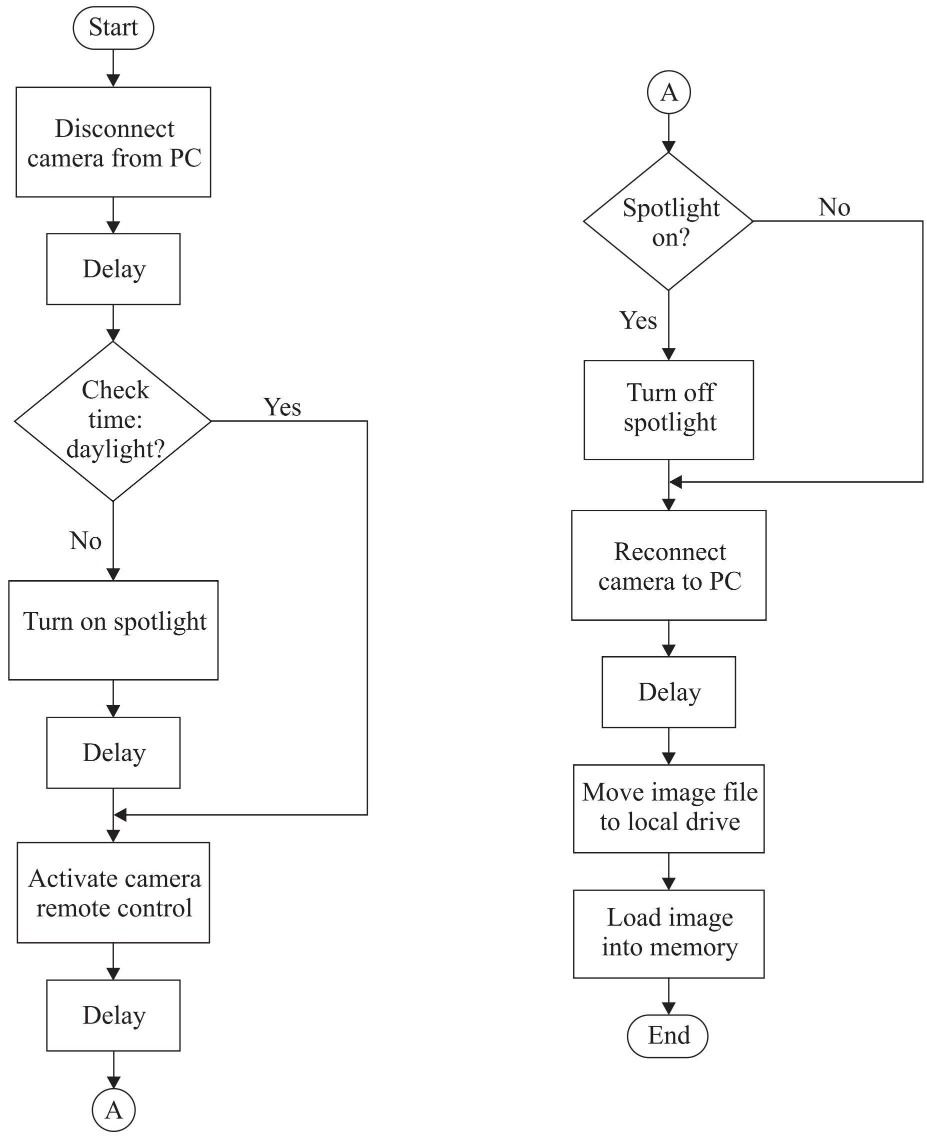


Figure I.2: Flow chart for image capturing software procedure

The interfacing circuit caters for output signals only, so no feedback regarding status of the camera can be obtained (nor are any such feedback signals available for this camera). The time delays around each camera process are therefore set empirically.

Appendix J

Poster Presentation for Water Panel Research Seminars Evening 2004

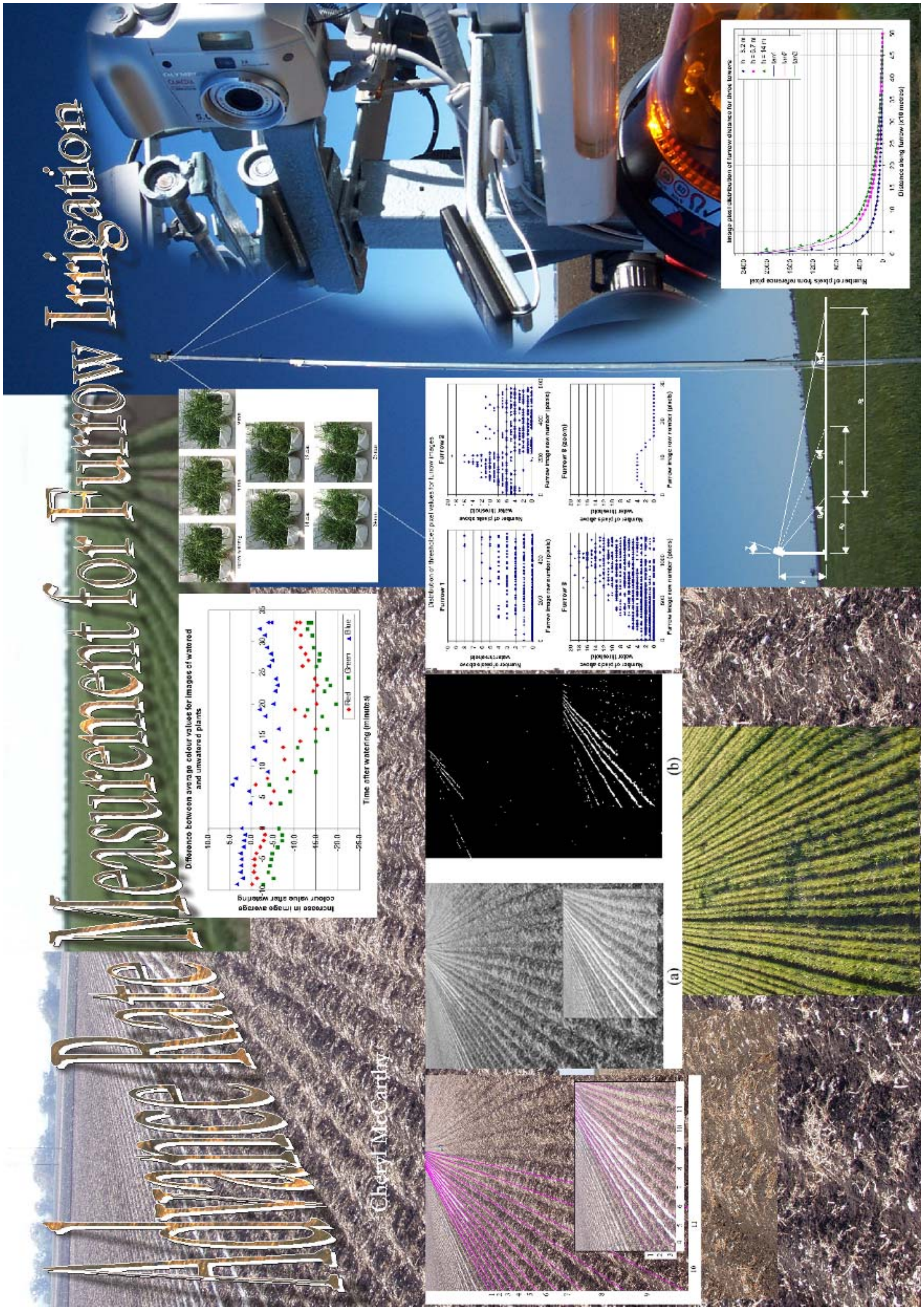


Figure J.1: Poster presentation ‘Advance Rate Measurement for Furrow Irrigation’