APPLICATIONS OF VISION SENSING IN AGRICULTURE

A Dissertation submitted by

Mark T Dunn

B. Info. Tech.

B. Eng (Mechatronic) (Hons. I)

For the award of

Doctor of Philosophy

2007

Abstract

Machine vision systems in agricultural applications are becoming commonplace as technology becomes both affordable and robust. Applications such as fruit and vegetable grading were amongst the earliest applications, but the field has diversified into areas such as yield monitoring, weed identification and spraying, and tractor guidance.

Machine vision systems generally consist of a number of steps that are similar between applications. These steps include image pre-processing, analysis, and post-processing. This leads the way towards a generalisation of the systems to an almost 'colour by number' methodology where the platform may be consistent between many applications, and only algorithms specific to the application differ.

Shape analysis is an important part of many machine vision applications. Many methods exist for determining existence of particular objects, such as Hough Transforms and statistical matching. A method of describing the outline of objects, called s-ψ (s-psi) offers advantages over other methods in that it reduces a two dimensional object to a series of one dimensional numbers. This graph, or chain, of numbers may be directly manipulated to perform such tasks as determining the convex hull, or template matching.

A machine vision system to automate yield monitoring macadamia harvesting is proposed as a partial solution to the labour shortage problems facing researchers undertaking macadamia varietal trials in Australia.

A novel method for objectively measuring citrus texture is to measure the shape of a light terminator as the fruit is spun in front of a video camera. A system to accomplish this task is described.

S-psi template matching is used to identify animals to species level in another case study. The system implemented has the capability to identify animals, record video and also open or shut a gate remotely, allowing control over limited resources.

CERTIFICATION OF DISSERTATION

I certify that the ideas, experimental work, results, analyses, software and conclusions reported in this dissertation are entirely my own effort, except where otherwise acknowledged. I also certify that the work is original and has not been previously submitted for any other award, except where otherwise acknowledged.

_____ _____

Signature of Candidate Date

ENDORSEMENT

_____ _____

Signature of Supervisor Date

Acknowledgements

I would initially like to thank my family, especially my wife Madonna. Without their support, encouragement and forbearance, this undertaking would not have been possible.

Next, I would like to thank my supervisor, Professor John Billingsley, whose ideas and suggestions always intrigued and motivated me.

I would also take this opportunity to thank the staff at the National Centre for Engineering in Agriculture (NCEA), especially Erik Schmidt and Matthew Durack, for their encouragement and assistance.

To Peter Driver and Peter Murray for their editorial advice and assistance, many thanks are owed.

To Neal Finch, David Bell and everyone else involved in the case studies, thanks for your belief, enthusiasm and assistance.

Table of contents

List of figures

List of tables

Acronyms

ANN	Artificial Neural Network
ARM	Advanced RISC Machines
DGPS	Differential Global Positioning System
GPS	Global Positioning System
HSV	Hue/Saturation/Value colour space
IR	Infrared
NCEA	National Centre for Engineering in Agriculture
NIR	Near Infrared
PC	Personal Computer
RFID	Radio Frequency Identification Device
RGB	Red Green Blue (colour space)
RISC	Reduced Instruction Set Computer
ROC	Rugged Outdoor Camera
ROI	Region of Interest
USB	Universal Serial Bus
VB	Microsoft™ Visual Basic
VC	Microsoft™ Visual C/C++
YUV	Luminance/chrominance colour space

1 Introduction

An ongoing technological revolution is changing the face of agriculture in Australia. As farmers and families become exposed to, and comfortable with low cost, powerful computing platforms in their houses, cars and mobile telephones, more attention is turning to the uses of advanced technology to assist in the most resource intensive tasks.

This research has been undertaken in search of new uses for cheap image sensors. The starting point is a base level of user technology that is both cheap and reliable. Webcams such as the items produced by Logitech ™ and Creative ™ are providing increasingly high resolution image capture technology. The mass production of these items means that the price is kept low. Further research and development has introduced a new low cost, rugged platform of 'smart cameras' for many agricultural applications.

Agricultural applications are an ideal target for the implementation of machine vision technologies. The ability to use image sensors remotely, requiring no contact with the subject or intervention by an operator provides a measure of robustness usually difficult to achieve in this area.

This research has identified a need for real time, on-site processing. Due to physical and geophysical constraints, communication and interaction with users may occur rarely. A major advantage of on-site processing is the reduction of data communication and storage requirements. This has led directly to investigation of new methods of shape encoding and processing.

1.1 Aims and objectives

This research commenced with the broadly defined area of machine vision in agriculture. A whole-of-field approach was used to provide a breadth of research, rather than selecting a small arbitrary subfield in which to specialise.

During initial reviews, it became apparent that there are many existing models for image processing applications, each specific to a particular application to a greater or lesser extent.

The National Centre for Engineering in Agriculture (NCEA) is a joint venture between the Queensland Department of Natural Resources and Mines (QNRM) and the University of Southern Queensland (USQ). NCEA specialises in developing collaborative teams of private and public sector research individuals and organisations to provide industry focused solutions. The focus of NCEA's activities is undertaking engineering research relevant to the agribusiness sector and the natural resource base it utilises.

Being a member of the NCEA has presented unique opportunities for in-depth research into diverse applications for machine vision technology and the underlying generic model linking these solutions.

This project attempts to define and demonstrate a single model that can be applied in the majority of machine vision applications relating to agriculture. A toolset of general purpose software and analysis techniques will be described. This toolset allows machine vision solutions to be quickly developed to suit specific applications without the need for extensive training on the part of the end-user.

1.2 Case study overview

Several discrete projects have arisen from the research and an overview of each project is provided below. A more thorough coverage is provided in later chapters.

1.2.1 Macadamia project

Initial work had been undertaken by Professor J Billingsley [1] to consider the most efficient and accurate method of determining the location of macadamia nuts as they are harvested. The result of Professor Billingsley's study was a proposal for a prototype machine vision system.

The project, funded by Horticulture Australia Ltd as project MC003020 was undertaken as part of this thesis.

Macadamia trees have around five years lead time between planting and commercially viable harvest of macadamia nuts. Plant breeders are continually creating and trialling new varieties to improve production from their orchards. The viability of a particular tree is determined by a number of factors, such as yield and disease resistance. As there is such a long lead time, it is vital to determine as early as possible the yield produced by a specific trial cultivar.

The current method of determining yield is to manually rake, collect and count every nut under trees in the varietal trials. As there are currently 1600 trees in industry-funded varietal trials, the manual labour constituted in this work is considerable. It has been estimated that up to 59%, or $109,824 [2, 3] could be saved by the industry each year by automated mechanical harvesting.

There are also proven benefits to using this technology in production harvests. The prime example is the creation of yield coverage maps, which have been proven [4-6] to increase efficiency in many other agricultural cropping industries.

In simple terms, the problem is to determine the presence of macadamia nuts using images from a video stream (Figure 1). This can be achieved easily through the use of colour segmentation. However, the presence of leaves, husks and other trash in the image dictates that we must use circle detection to verify nut location.

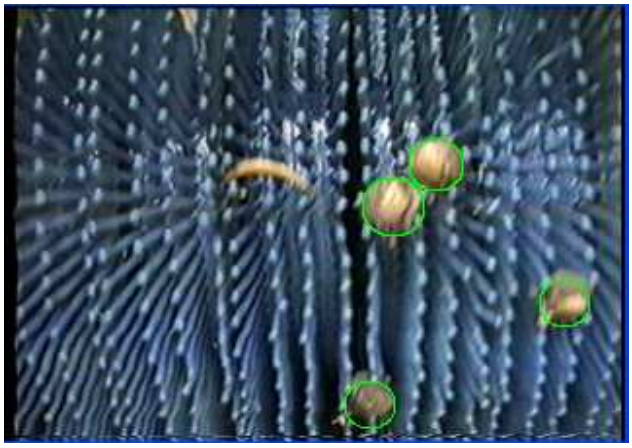


Figure 1 Example of an image from a macadamia harvester

This project uses object location and object tracking. The result of this research is a robust system for identifying and counting macadamia nuts in real time, together with novel location techniques to pinpoint the original position to within 10cm of each identified nut.

1.2.2 Citrus project

In 2003, an enquiry was received by the NCEA to measure the texture of skin of citrus fruit. The current method in varietal trials is for human experts to subjectively measure texture by feeling the fruit and looking at the surface for the degree of indentations [7].

Malcolm Smith from the Department of Primary Industries, Bundaberg, was interested in automated methods. After investigating alternative and expensive methods using stylus measurement instruments (similar to a record player pin reading from the spinning fruit), he contacted NCEA for advice on other methods.

A machine vision solution was proposed and has been completed as part of this thesis.

A citrus fruit has three types of variations in skin texture, macro (form), micro (waviness), and oil glands (roughness) as displayed in Figure 2. Varietal trials are undertaken, similar to those undertaken in the macadamia industry, to identify the

best variety based on several characteristics. The consumer tends to prefer smoother fruit. The waviness texture variations have the most effect on consumer appraisal.

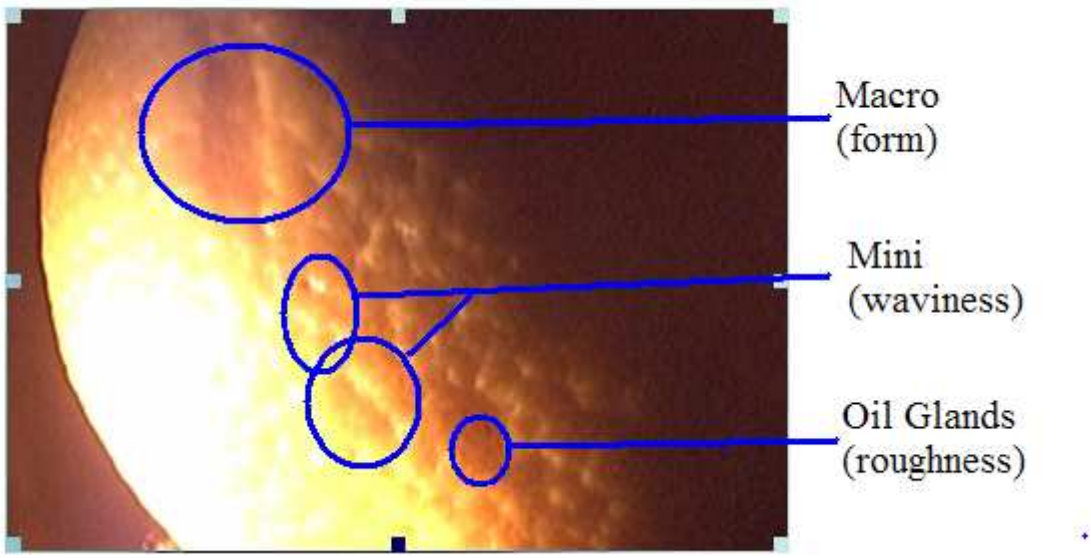


Figure 2 Example of an image of a citrus fruit

From detecting circles, to determine the extent of the fruit, this project moved on to detecting texture. It was determined that the most efficient way of detecting the texture was to use shaped light, and monitor the shape of the fruit at the shadow boundary of a light set at 90 degrees to the image sensor. In this way, minor changes in fruit circumference will be readily visible and measurable.

A prototype device was created to spin the fruit while examining it with an image sensor. The problem then reduced to a method of measuring texture in a way that is statistically valid and repeatable.

1.2.3 Animal identification project

December 2003 brought the next machine vision application to NCEA. A University of Queensland Animal Sciences PhD student, Neal Finch, proposed a project to deny or allow access to artificial watering points. The Great Artesian Basin Sustainability Initiative [8] was in the process of removing artificial water drains from the landscape and replacing them with artificial watering points. This scheme was introduced to reduce evaporation and other issues with water wastage.

The removal of widespread access to water from the landscape has the effect of forcing animals to gather at a single location to drink. At this time, with a trap around the water point in which animals entered via an automated gate, a producer could deny access to feral animals, such as pigs, or trap them inside once they had entered. The objective was to identify a way to automatically detect the species of animal entering the watering point.

A machine vision solution was proposed, and was again undertaken as part of this thesis work. This project has been funded in part by the Natural Heritage Trust (Australia) Project 46954. A system was established that can determine the presence of an animal (Figure 3) and take action based on an automatic classification of that animal

Figure 3 Example of an animal image

This application incorporates object detection, location, tracking and classification. The classification of the detected animal is accomplished using evidence gathering techniques, with shape information forming the central focus.

The investigation of the three case studies above illustrates some of the potential uses for machine vision in agricultural applications. The next section describes the structure of the thesis.

1.3 Thesis overview

This thesis consists of nine chapters. There is a CD of software tools included to accompany the physical thesis. These tools may also be downloaded by contacting the author.

Chapter 2 provides an overview of the current position of machine vision applications in real world agricultural fields. This chapter also provides a breakdown of the types of applications, their use and current trends. A theoretical model to generalise machine vision applications is introduced and discussed, with regard to the application and practical implementation of machine vision undertaken as part of this thesis.

Some new algorithms for s-psi techniques are introduced in Chapter 3. S-Psi methods are recommended as viable tools for object detection, location and classification.

Chapter 4 describes the methodology used in this research, in particular the hardware and software regimes used and analysis techniques. A generic hardware platform is described, and a thorough description of the 'Rugged Outdoor Camera' (ROC) vision platform designed and built by the author during this thesis work is also provided.

Chapter 5 continues with the examination of the macadamia nut application. Comprehensive prototype setup is detailed, together with software implementation, new algorithms, and field trial results.

Chapter 6 moves on to the citrus fruit skin texture project, and again details the work undertaken and the results achieved by the completed system.

Chapter 7 is the species identification project. Again, the prototype information and field trial results are gathered together in this chapter.

Chapter 8 generalises all of the case studies by describing the fundamental shape analysis and feature identification techniques that were implemented. A toolkit for

new researchers in the field of machine vision is proposed, accompanied by suggestions and guidelines for approaching real-world machine vision projects. Emerging projects are described in terms of the models and the toolkit application.

Chapter 9 concludes the thesis with a summary of the journey travelled.

Appendix A contains further details of the CD contents. Appendix B consists of a tutorial for programming in DirectShow. The hardware design for the ROC is specified in Appendix C. Appendix D lists some raw data examples from the species identification application.

The software on the enclosed CD includes:

- DirectShow filter templates for VC6/VC7
- DirectShow application templates for VC6/VC7
- DirectShow application sample for VB
- Implementations of DirectShow filters with sample code for
    - Canny edge detection
    - Hough circle detection
    - Sobel edge detection
- Sample media files
- Sample results files for case studies

2 Machine vision in agriculture

Machine vision algorithms in general are only briefly discussed in this chapter. The readers is referred to relevant books such as Davies [9] and Billingsley [10]. As relevant machine vision topics are encountered, comment is made where required.

There has been considerable effort placed into machine vision theory and algorithms over the last 25 years. This is filtering through from research based, high cost applications to more practical uses in almost all fields.

There are currently two main trends in mainstream machine vision research at the moment. The first is in hardware. Machine vision systems are becoming smaller and more suitable to stand alone, embedded systems. Zuech [11] predicts an eventual move back to centralised systems with enough cheap computing power and communications bandwidth to process multiple image capture devices. Most other analysts are predicting a move toward smart cameras [12, 13].

In software research, a considerable amount of effort is spent towards investigation of neural networking. In this field, the objective is to have learning algorithms that are flexible enough to generalise objects that have not previously been presented, into the correct classifications.

2.1 Agricultural applications

Agricultural machine vision applications can be seen to lie in one of the following areas:

Location
Tracking
Classification/recognition
Measurement/guidance

Note that this is a sequential hierarchy. Tracking applications must use location to find the object of interest first. Classification applications may use tracking to follow the object between frames. Guidance applications may use tracking and classification to determine current position and change of position. In this way, we can investigate the links between applications with reference to the level of processing required.

The earliest machine vision applications in agriculture were in fruit and vegetable grading. Practical applications have been around for over 25 years, and early examples include potato grading [14] and carrot grading [15]. However implementations more than 10 years old have become obsolete in the face of current processor size, speed and power consumption. For example, new processors were introduced into the PC market in 1995 with operating speeds of up to 200MHz. This speed is now available on slower embedded (single chip) computers. Since 2005, common desktop PC's have incorporated processors that operate at over 3GHz. Ever since, there has been a growing trend towards dual-core, multicore or multiprocessor systems that are capable of executing more than one application or thread of code simultaneously or in parallel.

Over recent years, there have been many published examples of machine vision being implemented in agriculture. Recent publications have been reviewed. Figure 4 below illustrates where these publications fall in the application areas.

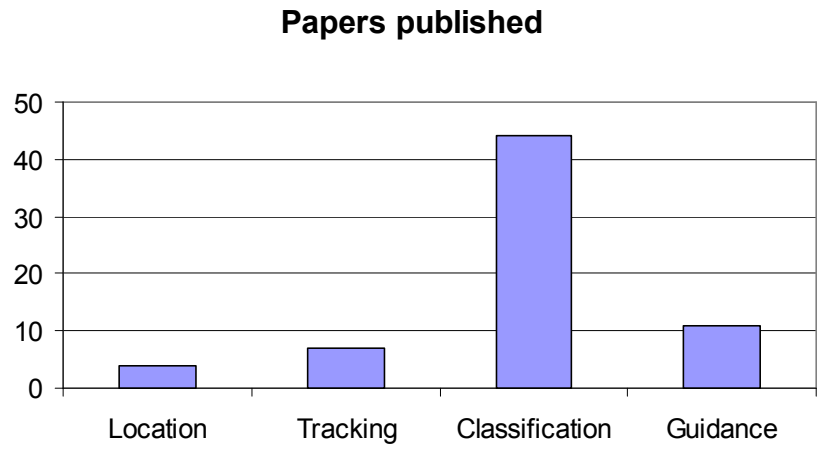


Figure 4 Recently published papers on machine vision in agriculture.

2.2 Location

Location of objects in an image is one of the basic level applications for a machine vision system. Most applications will have algorithms for this process at some level. In some applications, the core algorithm is related solely to the location of objects. In the macadamia project, the location of macadamia nuts in the video stream is the major task to be accomplished.

The simplest method of object location is to simply apply a threshold and segment an image into target and non-target regions. Figueiredo *et al*. [16] describe a poultry behaviour system based on this method. This paper does not handle clusters of multiple birds, which are reported as errors, nor tracking, which would make the system much more robust. The error rates reported are high, ranging from 20-70%

Location can also be achieved by relying heavily on prior knowledge, such as the application described by Phillip and Rath [17], which implemented a system for detecting the stem and calyx region on an apple. This research used shaped light to assist in object detection by reflections generated around the stem area. This paper describes a classification structure to determine the stem from the shape of the reflections.

Both Plebe and Grasso [18] and Stajnko *et al*. [19] describe locating fruit on the tree in orchards, the former oranges and the latter apples. Even though standard image sensors are used for oranges, and a thermal imaging device for apples, both projects proceed to use variants of feature extraction for determining a positive match. Feature extraction is a useful technique for algorithms that detect some portion(s) in an image that can be combined to indicate the existence of an object at a certain spot. Classic examples of these techniques are Hough transforms and their variants. [20]

Image registration techniques encompass a large proportion of location algorithms. In these applications, the requirement is to find the common areas of two images and determine the translation, rotation and scaling between them. Erives & Fitzgerald [21] describe a system for correlating multiple images taken with differing wavelength filters (multispectral imaging), where pixel intensity relates to

wavelength response. Phase correlation registration methods were used in this application, with Fourier transforms made on each image and the resultants correlated. This solution is not as yet real-time, but may become so as embedded processing power increases.

2.3 Tracking

After an object of interest has been located, it is far simpler to track the object in successive frames, rather than locating it again in every frame. Tracking requires the constraint that the images should translate (or indeed, the objects should translate) by less than half the image size to ensure that the object of interest is visible in at least two consecutive frames. Obviously, increasing the frame rate increases the volume of raw image processing required, yet more 'hits' on the object will allow a model (such as a Kalman Filter) to be created, thereby increasing the accuracy of predicted position, and reducing processing time.

Tracking has important implications in animal studies, for example welfare and behavioural investigations. Sergeant *et al*. [22] reported on the use of a machine vision system to track multiple birds in a poultry broiler house. Objects are located in each frame by segmentation, the centroids are determined and motion vectors determined. This is then used in successive images to resolve position. Shape analysis is used in this paper to identify single bird position within a large region of segmented pixels. A cost function based on the parameters area, compactness, line angle, split line length and concavity curvature automatically determines the best split into multiple single objects.

Similar to the poultry application, Zelek and Kanwar [23] describes an application monitoring multiple pigs in a pen with a vision system. At the time of publishing, the authors reported that image segmentation was not feasible in real time, however the increases in technology have somewhat outdated this research. Several other shortcomings, such as the requirement for manual initialisation and orthogonal rectangular windows for detection were identified. This paper uses 'blob' tracking,

where pixels are segmented and the centre of mass for each distinct area is treated as a single object. The chief problem with this method is the lack of any higher level information in generating the list of objects. This can cause false positives from noise, or false negatives from lack of target discrimination.

A mechatronic application has been developed by Frost *et al*. [24], where a specific point on a pig is to be identified as accurately as possible as the input into the robotic measuring device. The image is of a single pig in plan view and uses edge detection to determine the outline. From the outline, four changes of curvature (concavities) are detected and can be used to predict the full position of the animal.

A software application called EthoVision has been developed and is described by Noldus *et al*. [25]. This software can identify objects by threshholding, reference subtraction or colour matching. Designed for tracking insects, it is a generic utility for object tracking given manual setup information.

2.4 Classification / recognition

By far the highest proportion of practical applications fit into the classification area. These systems find use in fruit and vegetable grading, weed detection, and many other applications for produce grading. There is potential for immense labour savings in this field, as well as higher levels of correct classifications when compared to subjective human grading.

The basic concept of most applications is to classify the sampled item into predefined grades, based on some elements or features present and detectable. In most produce industries, differing grades of product receive different prices, so it is important to separate and pack appropriately. There are also different penalties involved in misclassification, which implies a weighted error structure, to ensure the best return for the given error rate. For example, if a single grade ‘B’ item is found in the same batch where the majority of products are classified as being of grade ‘A’ (higher quality), the entire batch of products may be automatically downgraded to the lower

quality grade, grade 'B'. In most product industries, higher grade products attract higher prices. The number of misclassifications and errors should be taken into account so that pricing can more accurately reflect the quality of the entire batch f products rather than being based on just one, or a few, low grade products in the batch.

2.4.1 Direct pixel based classification

Many algorithms use characteristics of individual pixels, or areas of pixels, to determine the classification for that area. The areas are then usually combined to determine either local or global classification structures.

Bennedsen *et al*. [26] identifies defects in images of rotating apples using images taken with 740 and 950nm filters. Each successive image is resized, shifted and flat mapped to give multiple views of the same areas of skin. A threshold is applied to each pixel, and the multiple images are summed. Areas with at least 3 out of 6 votes are classified as defects. Given that the rotation of the fruit is known, a far better method, more in keeping with real time systems, would be to track the position of the defect in each successive image. This would eliminate false positives reportedly caused by shadows.

A line segment detector was described by Davies [27], where the line segments indicate the presence of insects among rice grains. The Vectorial Line Segment Detector involves the application of two 7x7 masks to each pixel. The response of this stage is ANDed with an intensity threshold of the original image to provide accurate detection. Results have been reported as no false positives and one false negative in a trial of 60 images containing 150 insects.

Image analysis without a camera is described by Warren [28], where the images of chrysanthemum leaves are acquired by placing the leaf in a standard desktop document scanner. The images generated are then processed by automatic measurement of characteristics. A simple adaptive threshold converts the image to binary levels, reducing the processing required. The measurements are made using heuristics and a-priori information and categorised into 2-9 discrete states. Shape

information by higher level coding would prove beneficial to this project, adding flexibility to the classification rules.

Funck *et al*. [29] implemented a number of pixel based algorithms for detection of wood surface defects. Edge detection, gradient masks, region extraction, entropy threshholding and clustering techniques were tested on 70 images containing 10 each of 7 different defects. The study showed that combining the region growing technique with clustering algorithms provided the best overall accuracy.

Yadav & Jindal [30] investigated the use of grey level imaging to determine the percentage of broken rice kernels after milling, however no results were supplied. The broken kernels could be detected using shape analysis techniques. The paper suggested that the imaging would require manual positioning of the kernels to separate each object. This would not be required using shape analysis as the detection of touching objects is now a standard processing step.

Red, green and infrared wavelengths were used by Marchant *et al*. [31] to classify vegetation from background. A red/near infrared (NIR) ratio threshold was determined to have the lowest proportion of pixels misclassified over four images. The images were manually 'ground truthed', or classified into vegetation and non-vegetation prior to the automatic algorithm processing. Choice of threshold was determined by modality calculations on the full image histograms. This method will not work as effectively when the image is either all or no vegetation.

Philipp & Rath [17] investigated the different colour spaces and transformations and their efficiency at vegetation discrimination. Logarithmic Discrimination Analysis provided excellent results using pixels transformed into i1i2i3 colour space (98% accuracy); however the processing period was 10 minutes per image using a 350MHz processor. The processing time in this case could be cut substantially using dedicated hardware.

Image processing can apply equally to large scale satellite land cover maps, as discussed by Maxwell *et al*. [32]. In these applications, each pixel may represent a

large area, and the channel value for that pixel will represent the optical average of intensities over that area. Threshholding is performed simply based on Mahalanobis distance of each pixel from a known crop type pixel. This work would benefit from the implementation of field shape analysis, which would cut down the errors from scattered pixels of misclassified types.

Alchanatis *et al*. [33] selected two wavelengths (660nm and 800nm) using acousto-optic tuneable filters. These filters may cost up to $US160,000 making this technology unfeasible for widespread use at the current time. Fixed filters passing these wavelengths, however, are readily available. A threshold for classifying weed and cotton leaves was determined using a training set of 78 images. Around 15% of both false negatives and false positives were reported.

CIE-Luv colour space is used by Tantaswadi [34] to provide a Euclidean distance measure between colours to discriminate defects in cotton samples. This isodiscrimination contour is generated into a Look Up Table (LUT) for each RGB value. Each pixel is then classified into target or non-target based on the RGB intensities. Again scattered misclassified pixels could be removed by judicious use of region based techniques.

Noordam *et al*. [35] describes potato grading, an area which has been investigated by many researchers. This application uses mirrors to provide a full 3D view of the object. Linear Discriminant Analysis is used with Mahalanobis distance to separate pixels into defect or non defect areas. Classification with 90% accuracy is achieved for pixel based region checking.

Sena Jr *et al*. [36] reported 94.7% accuracy over 720 images of damaged and non damaged maize plants. The threshold was determined by searching the excess green histogram for the optimal threshold between foreground and background.

2.4.2 Models

Jeyamkondan *et al*. [37] used the bimodal distribution in the red channel to distinguish meat from background. A Fuzzy c-means process was then applied to

separate each image into two areas of least cost (meat and fat areas), determining an adaptive threshold. A classification into distinct grades was then performed with several linear regression models, giving equivalence with manual grading results.

Both Granitto *et al*. [38] and Marchant & Onyango [39] compared the use of Baysian classifiers to Artificial Neural Networks (ANNs), and both concluded that the Bayesian methods are more efficient as well as more accurate.

Aleixos *et al*. [40] provides details of a system that processes five citrus fruit per second. Each fruit object is located by simple threshholding as the lighting is well controlled. The edge is chain coded and touching fruit are identified by large changes in the boundary tangent. Geometrical features are extracted as well as defect information from the colour channels. The end result is a classification based on all features into one of three grades using a Bayesian discriminant model, with cross category accuracy of over 90%.

Lee *et al*. [41] reports a machine vision system for detection of weed targets. The images were segmented into a binary image by a Bayesian LUT. The images were further enhanced by the application of morphological operations including shrinking and swelling. Leaf shape features such as area; centroid and area to length ration were then determined and classified with a Bayesian classifier. The accuracy reported in this study was as low as 30% in some outdoor trials.

Fuzzy Logic

Fuzzy c-means and Fuzzy Gustafson–Kessel methods were used by Meyer *et al*. [42] to segment images into a fixed number of classifications (plant/residue/soil) in this research. These methods are unsupervised classifications, given the number of clusters.

Multi dimensional image processing was introduced to the lumber industry by Kline *et al*. [43], with an RGB image sensor, laser distance ranging system and X-ray scanner, providing five different streams of information. Histogram techniques were used to detect modality in each of the channels and extract regions. Fuzzy

membership functions are used at a higher level to determine the defect class. Accuracy up to 89% was reported.

2.4.3 Artificial neural networks

Many other projects have used Artificial Neural Networks (ANN) for classifying fruit. For example Kondo *et al*. [44] describes a system where the number of layers in the neural network was varied manually until the best result was achieved.

Guyer & Yang [45] reported 73% classification accuracy using an artificial neural network with genetic algorithms for forward propagation of weight learning. This research investigated defects in cherries in a pixel-wise manner. The pixel values from multi spectral images were passed to the ANN for classification. Training images required 36 seconds; each process thereafter required 2.8 seconds. Around 70-75% accuracy was achieved, with most of the errors being misclassification between types of defects.

Goel *et al*. [46] compared ANNs and Decision Trees on airborne multispectral images (72 wavelengths) of corn crops to determine nitrogen application rate and weed infestations. Decision tree algorithms delivered between 27% and 43% misclassification rates when both weed and nitrogen effects were considered. Even split, misclassification errors were between 14% and 30%. An ANN was implemented with 2 hidden layers on the same data giving 30-40% misclassifications. In later work, Uno et al. [47] attained greater prediction accuracy (about 20% validation RMS error) with an ANN model than with either of the three conventional empirical models based on normalised difference vegetation index, simple ratio, or photochemical reflectance index.

Moshou *et al*. [48] also used multispectral information at 543, 630m 750 and 861nm wavelengths to identify yellow rust disease in wheat, using a spray boom mounted spectrograph. This research reported up to 99% accuracy using a multilayer ANN with 10 hidden neurons.

Patel *et al*.[49] used histograms of each of the RGB channels as direct inputs to a neural network with 384 input nodes and 24 hidden nodes. The accuracy reported was between 85% and 93% for various defect types.

Siddaiah *et al*. [50] demonstrates a combination of ANN and fuzzy rules, where the membership rules are updated by training a Neural Network with the correct classifications for defect types in sample images.

2.5 Guidance and measurement

Automatic guidance on tractors has been available since 1913 [51]. The first system was a mechanical furrow following device with a centring wheel mechanically driving the steering mechanism. Improving on mechanical methods, GPS systems have recently revolutionised guidance systems on tractors, however high accuracy GPS systems are still so expensive that there is a large market for cheaper machine vision solutions. Machine vision solutions will also identify and track the (vegetative) target, rather than a preset line. This will allow deviations by the tractor where required to ensure that crop integrity is retained.

Work by Billingsley and Schoenfisch [52-54] described one of the first automated vision guidance systems implemented on a tractor. This system used trapezoidal Region of Interest (ROI) windows to lock onto crop rows. A line of best fit calculation was performed to generate cross track errors as the visual target moves within the ROI windows.

The windrower described by Fitzpatrick *et al*. [55] is another early example of vision guidance on tractors. This system detects the cut line from the last swath and automatically steers the tractor. The discrimination of the cut line is performed with a 'best fit' step function. This system also displays some data fusion, adding GPS, obstacle avoidance and manual control to the decision making process.

Tillett *et al*. [56] used a Kalman Filter model of a series of crop rows to perform predictive control of a steerage hoe. This application used a band pass filter to

extract the row information from the image. This filter however has as its basis a sinusoidal function where the frequency – which relates to the distance between rows – is known. On loss of signal from the row tracking algorithms, the guidance state machine retains a straight heading, on the assumption that the rows are more likely to be in straight lines.

Slaughter *et al*. [57] also used a trapezoidal ROI window to determine crop row position. This study used a median measure for estimating the location of the crop row within the ROI. The accuracy was reported as 4.2mm RMS under ideal conditions to 11.9mm under heavy weed loads.

Hague *et al*.[58] implemented an Extended Kalman Filter updated with row position detected with a modified Hough Transform for line detection. This methodology uses prior knowledge of the row type and row width to assist in detection and tracking.

Sogaard & Olsen [59] reported a system which does not use segmentation to detect crop rows, instead a grey level histogram is used directly in a least squares regression fit. The calculations determine the most likely position of the known crop row spacing in each segment of the image. This technique is susceptible to errors introduced by inter row weeds; however a quality of fit value is reported for these circumstances.

A unique guidance system that is not crop dependent was described by Vaughan *et al*. [60]. This application guides a robotic sheepdog gathering a flock of ducks. In this instance, the camera is mounted not on the robot, but at the side of the arena discriminating between the robot and the flock of ducks. This system describes a model based on varying levels of attractive and repulsive potential fields.

Measurement

As well as guidance platforms, machine vision is becoming a standard in measurement applications. The main advantage of machine vision solutions is non-contact sensing. This provides robustness advantages over mechanical methods.

Regression models can be used to provide a classification based on some given values. Schofield *et al*. [61] used multiple measurements of the visible surface area of a pig, viewed from above, to estimate weight and weight gain of pigs over multiple days. The mean weight of a group of pigs could be estimated to within 5% of their actual weight. The area was based solely on threshholding pixel intensities, yet with enough repeats, inaccurate measurements can be filtered out. White *et al*. [62] reported extension of this work into detection of pig type using a radial basis function neural network. Accuracies between 65% and 81% were reported for classification of pig type. At the same research centre, Wu *et al*.[63] measured pig volume with three pairs of stereo cameras. The pigs are viewed on a blue background for contrast. A 3D map is constructed from a combination of each of the three range images, in turn built from the two stereo images.

Peacock & Boyce [64] used stereo vision to estimate distance to cow teats for an automated milking system. Combined with novel actuators using air pressure differentials, this system is a complete measurement and guidance application.

Thus there is great potential for the use of measurement and guidance systems for further agricultural applications including those studied in this thesis.

2.6 Conclusions

The material in this chapter displays just how ad-hoc most machine vision applications are. There are as yet no grand unifying theories of optimal system design. Many scientists working in the ANN field would argue that these systems are approaching the point where true intelligence will emerge, yet there is still substantial work to be done. On the other hand, workers dedicated to problem solving in a more direct method cannot yet agree on even the most effective colour space to use for any given application, or the means to decide such. Therefore, this research provides insight into a methodology that can be used generically for machine vision application development.

3 S-Psi coding

One major area of focus in this research has been the development of new algorithms using s-psi methodologies. Much of the available advanced theoretical work has concerned intensive techniques such as Hough transform. For applications in the field, it is desirable to use more pragmatic methods, which are not as computationally expensive. Significant among these is the use of an edge tracing routine, yielding a Freeman chain, which becomes an s-psi profile. Although it is quite feasible to encompass intensive computation on recent PC's, there is a need to embed the algorithms on a much simpler chip that can be used for a robust, low cost platform.

S-psi coding is similar to Freeman's chain [65] in that it codes the direction of steps taken around the boundary of a shape, but it also codes the distance of the step. Freeman's chain assumes unit step size, but using the exact inter-pixel distance offers s-psi methods some advantages which can be used in subsequent processing steps.

Freeman's chain has been used extensively in handwriting recognition techniques [66] where chain codes are used in a multi classifier architecture to label each handwritten character. This type of chain coding has also been used in fingerprint analysis [67], and many other areas where shapes can be inferred as or from lines and line segments.

S-psi has also been used in a variety of ways. It is called Tangent Space by Latecki and Lakamper [68], who use curve evolution to determine minimum cost for transformation of curves. This provides a shape similarity measure for two objects.

S-psi has also been called Turning Function by Arkin *et al*. [69] who also used it as a shape similarity measure based on the L/sub 2/ distance. Davies [9] also discussed this method for use as a general shape encoding method.

This method has the potential to be an effective tool for a general form of shape analysis. The following sections explore in detail some of the algorithms of this method.

3.1 Basic algorithm

Given an arbitrary shape in an image, there is a function F(x, y) that determines each pixel's membership to the object. The function is not relevant in our discussions, but is the subject of a complete field of research in its own right. Segmentation, texture analysis and cluster analysis are but some of the methodologies used to determine this membership function.

The membership function is used by an object tracing routine to determine the edges of the shape. Standard edge detectors such as Canny [70] and derivatives only provide a binary image with edge pixels indicated. This provides absolutely no information about object shape. The tracing method, however, will indicate edge pixels **as well as** encoding the entire object.

The edge of the shape is traversed in a consistent direction. For the purposes of standardisation, retain the object on the right of the boundary. As each step on a standard grid is taken, record the direction (ψ) and the step length (ds). This chain pair contains all the information to completely and uniquely recreate the object.

Chain codes are based on directional steps. The most common use either 4 or 8 angles (Figure 5).

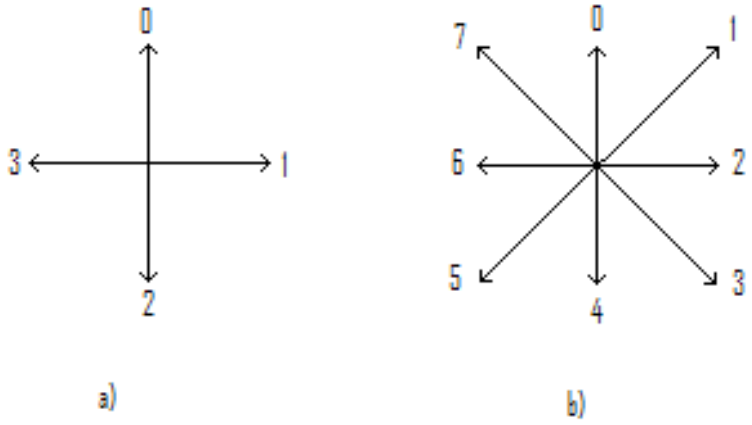


Figure 5 Chain encoding directions for steps in outline tracing

3.1.1 Edge tracing routine.

The sequence of tracing the edge of an object is given below:

Search image course grid for a target pixel.
When found, step directly up (direction 0) until non-target pixel is found. Step down one pixel. This is the inside edge of the object and the start point.

The edge is then traced by a combination of checking and stepping.
For each edge pixel found:
Check neighbouring pixels for a non-target pixel. Commence in the last stepped direction, rotating the search anti-clockwise until a non-target pixel is found. Rotate clockwise one and step. This is the next pixel in the chain.
Continue checking and stepping until the start point is reached again.

Figure 6 below is displayed as a general example. Starting at the pixel 'a', the chain code generated is:

2343445446666000000210

This method uses the centre of each pixel as the coding point. The midpoint of the line joining two boundary pixel centres could also be used.

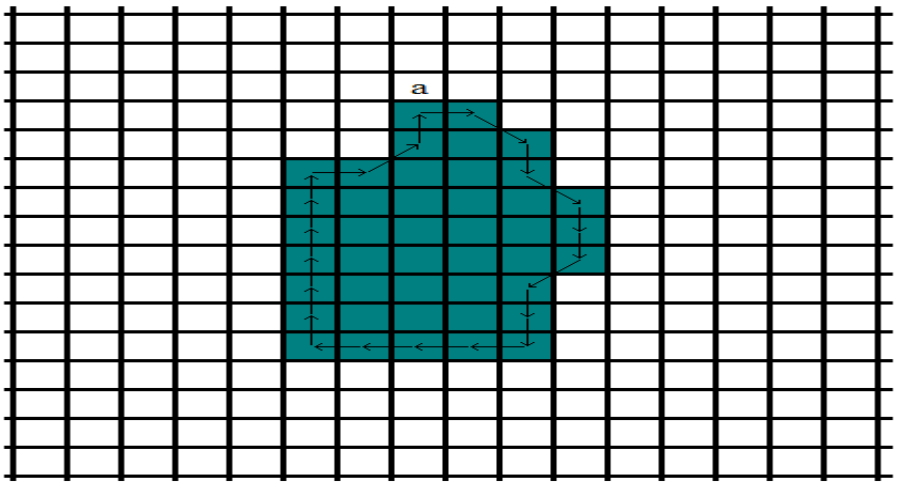


Figure 6 Edge encoding for a sample object

The s-psi graph related to this object and generated chain is displayed in Figure 7.

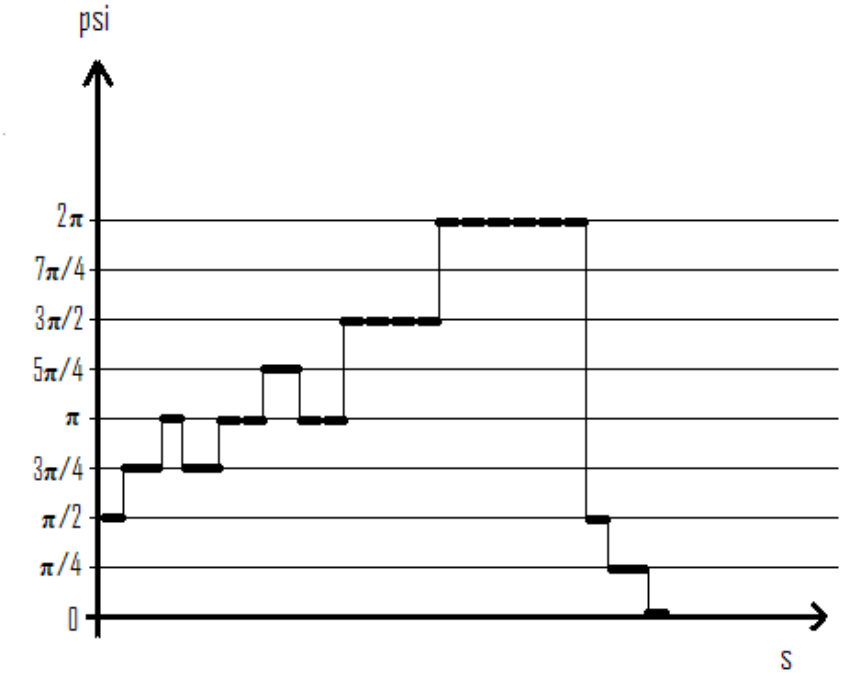


Figure 7 S-psi graph for a sample object

Figure 8 displays some basic shapes and the corresponding s-psi curves. Note the cyclical properties of the graph means that any direction is equivalent to itself plus

2π. The presence of straight edges is easily recognisable in the s-psi curve as horizontal areas. Similarly, the presence of curved edges in the object can be identified by straight oblique lines in the s-psi curve.

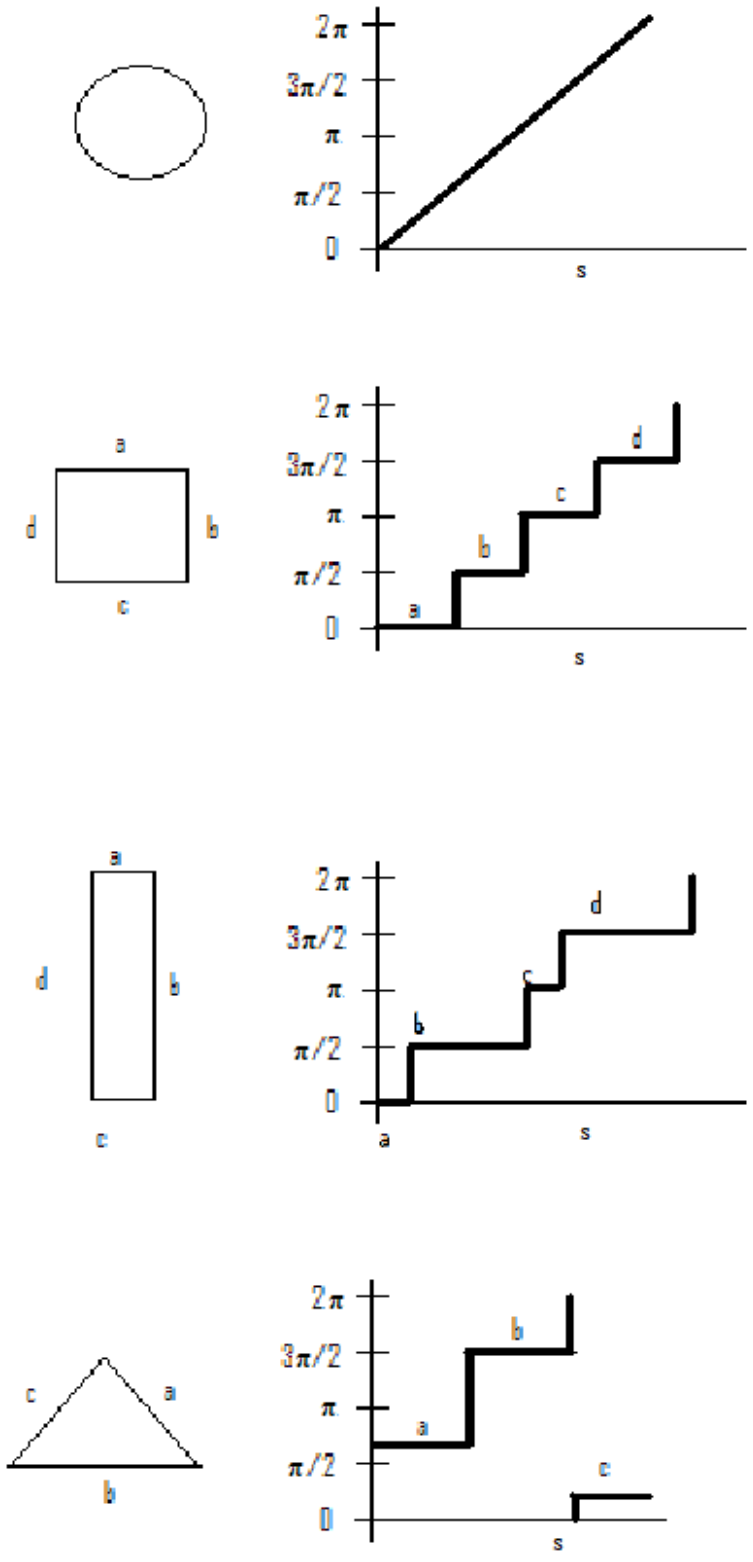


Figure 8 Sample shapes and s-psi graphs

3.2 New methodologies

The following algorithms are extensions to the current s-psi methodologies. The assumption is made that the curve corresponds to a closed shape, and that the shape is compact.

Psi = ψ

ψ is coded on 0 to 2π scale.

The s-ψ curve is "wrap around", so ψ_{-1} refers also to ψ_n

$Link_x$ refers to the x^{th} s-ψ doublet in the chain

3.2.1 General

Some simple conclusions can be drawn from an s-ψ chain with n items of perimeter increment (s_i, ψ_i).

Shape perimeter $S = \sum_{i=0}^{n} s_i$. (3.1)

Maximum possible area for the shape $A_{\max} = \frac{S^2}{4\pi}$ (circle) (3.2)

For a closed shape $\int \sin(\psi).ds = 0$ *and* $\int \cos(\psi).ds = 0$ (3.3)

Combination (addition) of 2 links to form a single combined link (3.4)

where $(s_2\psi_2)$ *replaces* $(s_0\psi_0)+(s_1\psi_1)$

then

$$s_2 = sqrt\left(s_0^{\,2} + s_1^{\,2} - 2s_0 s_1 \cos(\psi_1 - \psi_0)\right)$$

$$\psi_2 = \psi_0 + \sin^{-1}\left(\frac{s_1 \sin(\psi_1 - \psi_0)}{s_2}\right)$$

Cross Product

The cross product is the vector perpendicular to two links in the chain. The cross product is useful for determining the area formed by the completion of the parallelogram between the two links.

$Link_o \times Link_1 = s_0 s_1 \sin(\psi_1 - \psi_0)$ (3.5)

Dot Product

The dot product is the projection of one vector onto another vector, or the distance of one vector in the direction of another vector.

$$Link_{o} \bullet Link_{1} = s_0 s_1 \cos(\psi_1 - \psi_0) \quad (3.6)$$

Figure 9, Figure 11, Figure 12 and Figure 12 are examples that will be used for demonstration purposes through the following algorithm descriptions.

Figure 9 An example of an input frame

Figure 10 The binary foreground/background example image

Figure 11 The traced edge of the input frame

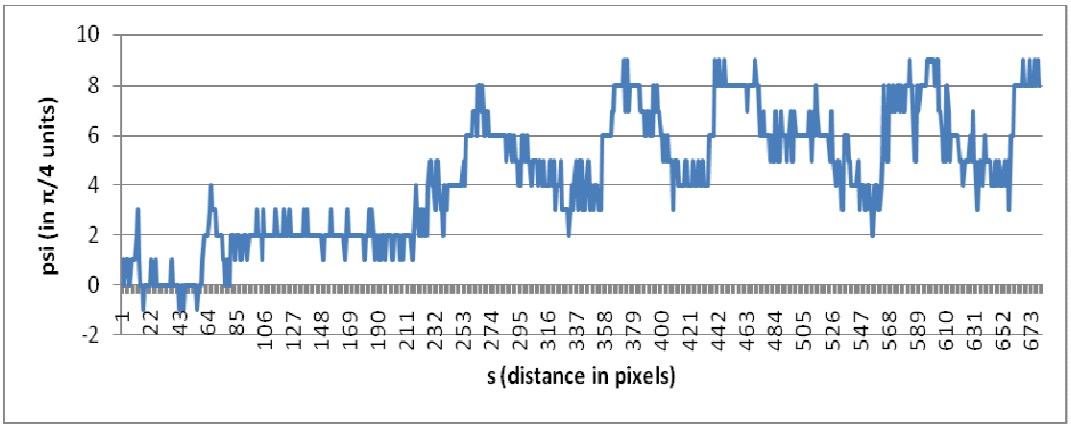


Figure 12 The S-Psi graph of the input frame.

3.2.2 Compression

For a given chain of n items, there exists a compressed chain of maximum n and minimum 3 items that can also uniquely describe the object. All links with non-zero dψ can be merged.

Demonstration:

Consider a chain segment of a vertical straight line with ψ: 0,0,0,0 and s: 1,1,1,1. This can be compressed to a single link ψ: 0 and s: 4.

In this case, the minimum chain code length $= n_v$, as all straight lines may be compressed to a single step.

The worst case scenario for this compression is that every step has a non-zero dψ. As an example, consider a circle. In this case, the compressed coding is identical to the original.

Algorithm:

j=0

For i=0 to n

If $\psi_{i+1} - \psi_i <> 0$ then

Store ψ_j, s_j

j=j+1

$$\psi_j = \psi_{i+1}, s_j = 0$$

else

$$s_j = s_j + s_i$$

$$n_v = j$$

This technique is lossless, as the original outline may be exactly reconstructed. There are also methods such as described by Schuster and Katsaggelos [71], which reduce the bitcount for encoding the chain by using predictive models. These techniques can reduce the size of the chain by up to 50%, but can also lose edge information.

3.2.3 Edge approximation

One valuable tool of this methodology is the ability to approximate edges, but retain exact substantial corner information. This method replaces curved or rough edges with straight lines.

To accomplish this, a change threshold may be selected manually or automatically. This threshold will be the amount of deviation (in radians) that will be approximated by a straight line. Obviously, a threshold of 0 will result in no change to the original chain.

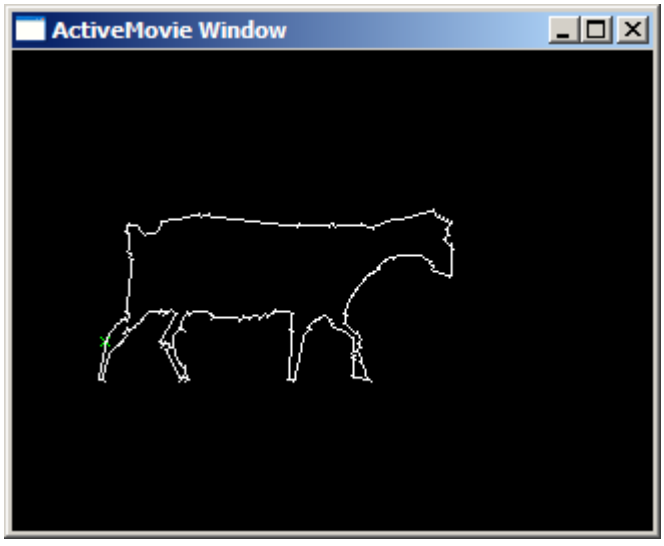


Figure 13 Example of edge approximation, using 1.0 radians

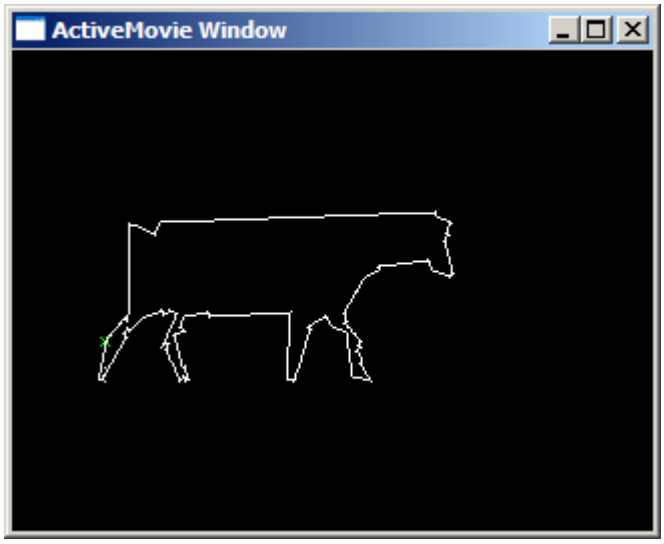


Figure 14 Example of edge approximation, using 1.3 radians

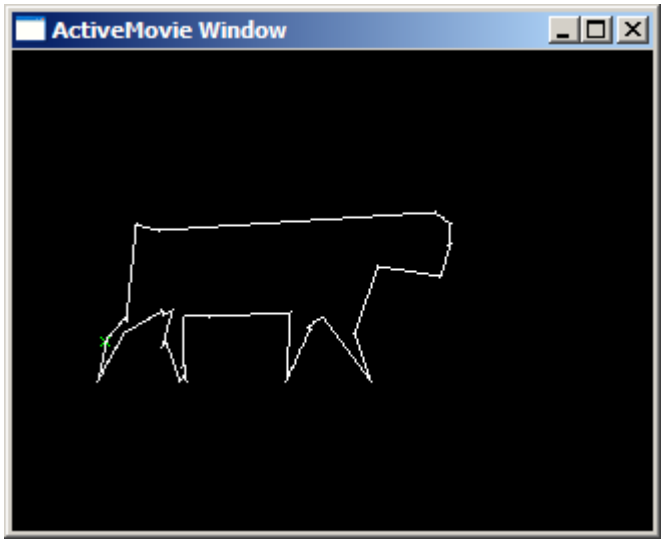


Figure 15 Example of edge approximation, using 2.0 radians

Algorithm:

j=0

For i=0 to n

If $d\psi_{j-i}$>threshold

j++

$link_j=link_i$

Else

$link_j$+= $link_i$

3.2.4 Area

The area of a shape is the number of pixels contained within the boundary of the shape. Depending on coding methodology, this can either include or exclude pixels comprising the boundary.

General:

$$Area = \sum p_{x,y} \tag{3.7}$$

Where $p_{x,y}$ is the set of all member pixels.

Consider that the s-psi chain is a set of vectors in 2D space, and the vector addition of any two successive links completes the triangle. In this case, we can build the object from a set of triangles, with one vertex fixed and the other two incrementing along the chain of vertices. The sign of the area made by the triangle is given by the middle internal angle.

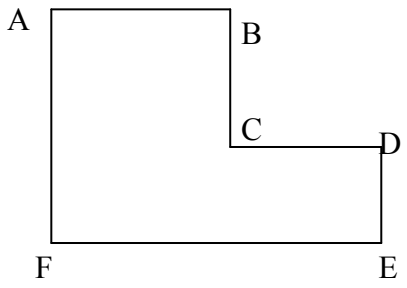
Note that a compressed s-ψ chain will result in many less calculations.

A demonstration of this is given below:

Given an arbitrary shape with corners ABCDEF,

Triangles can be produced from ABC, ACD, ADE and AEF.

In the example figure below, Area = ABC – ACD +ADE +AEF



Algorithm:

$$v_0 = s_0 \angle \psi_0$$
$$v_1 = s_1 \angle \psi_1$$
$$v_t = v_0 + v_1$$
$$Area = \frac{|v_1 \times v_0|}{2}$$

For i=2 to n (using equation 4 for addition and equation 5 for cross product)

$$v_t = v_t + v_i$$
$$Area = Area + \frac{|v_t \times v_{t-1}|}{2} \qquad (3.8)$$

Note that the addition is signed, giving the area exact to within the precision of the pixilation error.

The area thus generated will always be less than the true 'pixel count' area, but a correction factor can be generated directly from the psi chain links. The following equation is used for each ψ_i, ψi_{+1} pair.

$$d\psi = \psi_{i+1} - \psi_i$$
$$if\ d\psi < 0\ then\ d\psi = d\psi + 8\ (retain\ positive\ delta)$$
$$if\ d\psi < 4 \quad then$$
$$Area = Area + \frac{4 + d\psi}{8} \qquad (3.9)$$
$$else$$
$$Area = Area + \frac{d\psi - 4}{8}$$

The equation above follows from the fact that the pivotal pixel for the two links may be treated as a 1 by 1 square. The area calculations may then be envisaged as two lines intersecting at the centre of the square. Thus, for a zero direction change link, the true area will be ½ pixel greater than the calculated area. Similarly, a pixel with a direction change of 2 (for example links AB, BC above) will have added ¼ pixel to the calculated area, so ¾ pixel needs to be added for the true area.

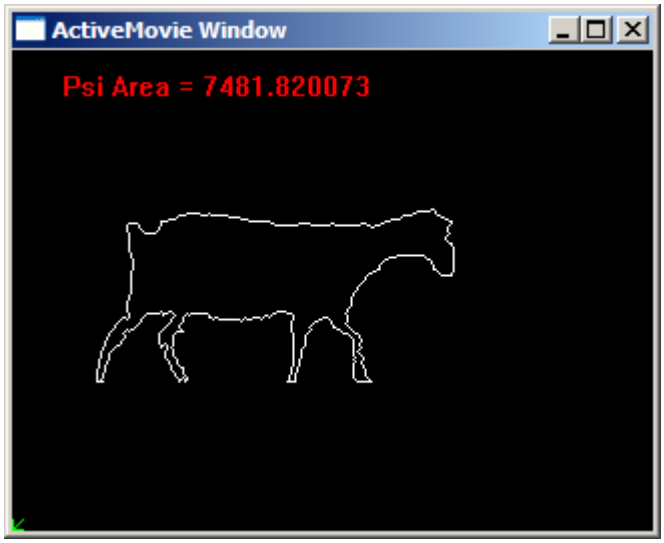


Figure 16 Example calculation of S-Psi area.

Real Area: 7449 pixels, calculated area: 7481.8 pixels (error<0.5%)

3.2.5 Point / member

Given an arbitrary shape, a starting coordinate x_0, y_0 and an arbitrary point x_p, y_p to check, it is possible to determine whether the point is inside the shape, outside the shape, or on the edge with a maximum of n_v calculations. In this algorithm, the sum of the angles made from the point to each of the corners of the object will sum to either 2π or 0. Figure 15 displays example internal and external points for a shape. The internal point angles sum to 2π.

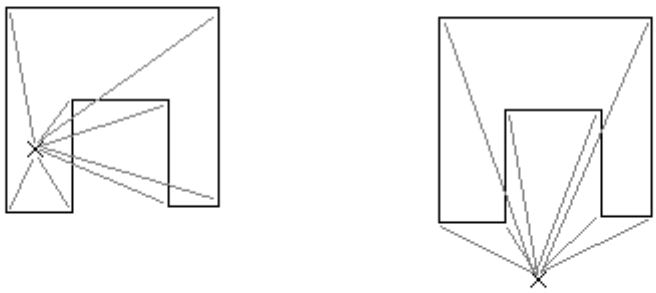


Figure 17 Example internal and external points.

Algorithm: (3.10)

$$\vec{v}_1 = (x_0 - x_p, y_0 - y_p)$$

For i=1 to n_v

$$v_{i+1} = v_i + s_i \angle \psi_i$$
$$if \quad \angle v_{i+1} - \angle v_i = \pi \rightarrow pixel\ is\ on\ edge$$
$$\psi_{sum} = \psi_{sum} + (\angle v_{i+1} - \angle v_i)$$

If $\psi_{sum} = 2\pi$, then the point is inside the shape described by the chain, otherwise the point is outside.

3.2.6 Convex hull

The convex hull of a shape is the outline of the object with no concavities. Concavities may be defined as a pair of links with negative delta ψ. This feature is useful for coarse shape matching, for instance, where the macro features (maximum length, bounding rectangle etc) are more important than the detail.

The convex hull is found by processing the chain in sequence. Each pair of links with a negative delta-ψ is merged. If a pair of links is merged, the new link is checked with the reverse direction also. This process is continued until there are no pairs of links with negative delta-ψ.

Algorithm: (3.11)

j=0
$link_j = link_0$
For i=1 to n
If $\psi(link_i) < \psi(link_j)$
    $Link_j = link_j + link_i$
    While $\psi(link_j) < \psi(link_{j-1})$ and j>0
        $link_{j-1} = link_{j-1} + link_j$
        j=j-1
    Else

j=j+1

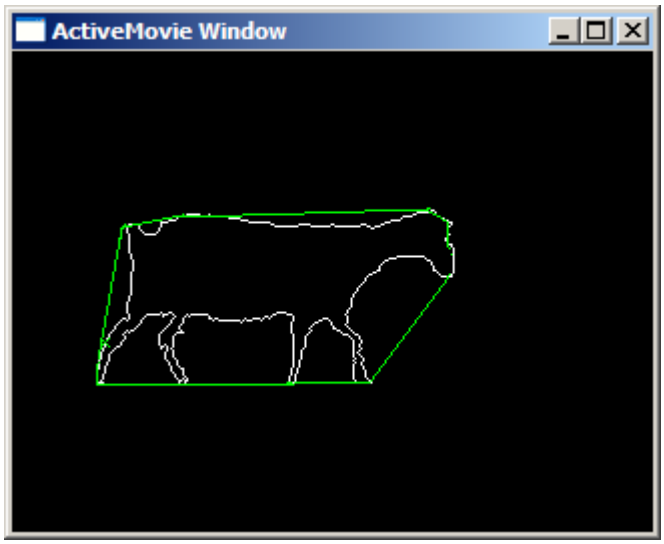


Figure 18 An example of a convex hull

3.2.7 Mathematical smoothing

Smoothing can be done mathematically given unit size steps

For i=0 to n

$$\psi_i = (\psi_{i-1} + 2 * \psi_i + \psi_{i+1}) / 4 \qquad (3.12)$$

Note that this may need a correction factor and will not necessarily be a lossless translation.

The correction can be calculated by summing the new components and determining the error. That error can be spread evenly between each of the steps. This will ensure that the shape is closed again at the completion of the algorithm thus:

$$\text{Error} = \frac{\sum dx}{\sum dy} \qquad (3.13)$$

For i=0 to n_v

$$\psi_i += \frac{\text{Error}}{n_v}$$

Any convolution smoothing regime may be used, such as application of Gaussian masks [1 2 1]/4, [1 2 5 2 1]/10 etc. Median filtering, however, will not work due to the loss of accurate directional information implied.

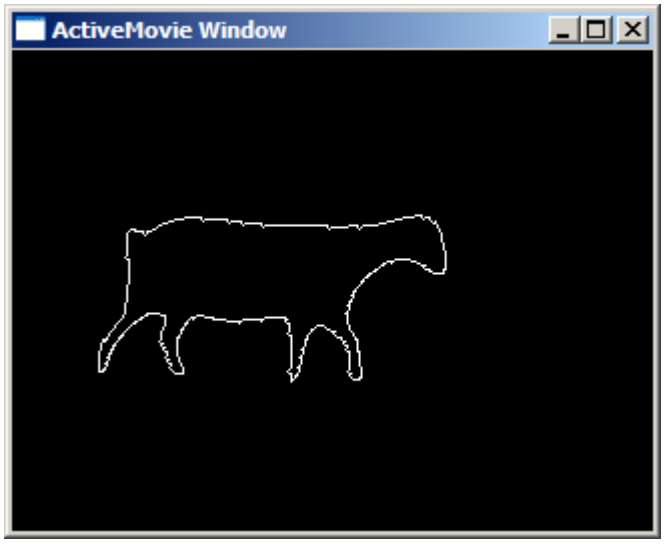


Figure 19 Smoothed example image

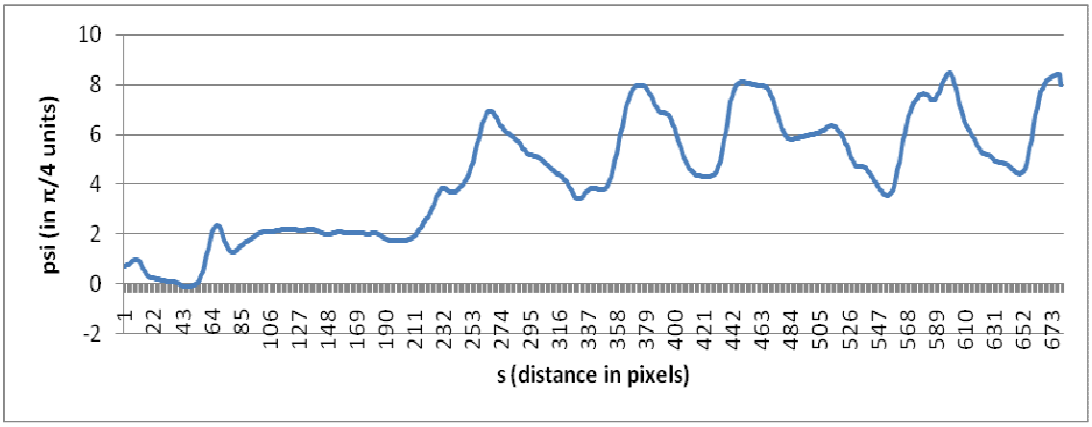


Figure 20 The smoothed s-psi graph

Figure 20 above displays the end result of the smoothing process on the s-ψ graph. As the outline is traced clockwise, turns to the right (such as the tail and the nose areas at s=64 and s=273, respectively) are areas of increasing psi in the graph, and turns to the left are decreasing. Transitions between turn directions are local minima and maxima. Straight lines in the outline (at any orientation) are horizontal sections of constant ψ in the graph.

3.2.8 Shape analysis using template matching

Shape analysis using template matching is generally performed with the application of a minimum distance calculation. Euclidean distance can be used with caution because any occlusions or obstructions will cause shifting of parts of the curve due to

changes in the perimeter length. This problem was overcome by Latecki and Lakamper [68], using minimum deformation energy calculations to determine the closest matching template. In this circumstance, corresponding maximal/minimal arcs were identified and matched. The energy was calculated as the percentage of the original length by the cost to deform matching segments.

3.2.9 Histogram matching applied to s-psi

Histogram matching can offer an indication of 'likeness' or similarity between shapes. In this algorithm, the frequency histogram of psi steps is generated and matched against templates. Examples include Iivarinen and Visa [72] where each letter of the alphabet is classified based on the histogram generated. It should be noted that the histograms indicate a level of similarity only, the shape cannot be reconstructed.

3.2.10 Feature identification

A major advantage of s/psi coding is the ability to search for two dimensional features in 1d space. Consider an arbitrary shape where one section or part of the edge has a known radius of curvature (ρ). Finding the location of this feature on the edge becomes a matter of a search of the 1 dimensional s-ψ chain where dpsi = ρ.

This becomes an important practical tool for template matching and evidence gathering. Any arbitrary object can be described by a searchable feature set completely describing the outline. A machine vision application may then implement a search routine on the generated s-ψ curve from for the features of each template.

As an example, considering the graph displayed in Figure 20 above, a tail can be considered an area in the graph with a sharp rise and a maxima between psi=0 (tracing directly up) and psi=$2\pi/4$ (along the back of the animal).

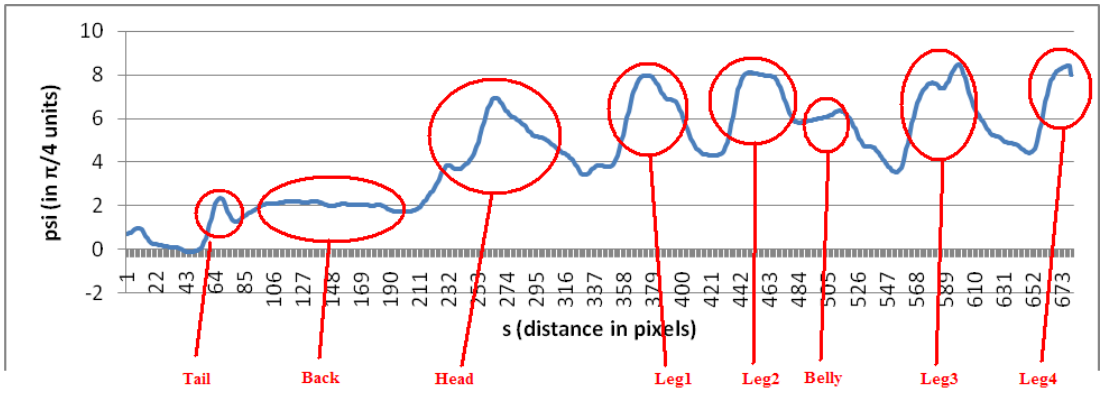


Figure 21 Feature extraction from s-spi graph

Similarly, other features as displayed in Figure 21 can be described by non variant properties and with reference to the position of nearby features.

3.2.11 δs-δψ coding

In this methodology, the object is recorded by links of delta psi and delta s. This provides three distinct types of links in the chain:

$\delta\psi=0$, $\delta s \neq 0$. This represents a straight line segment: there is no change in angle between the start and end of the line segment, length δs.

$\delta\psi\neq0$, $\delta s = 0$. This represents a corner. There is a discontinuity in the δψ graph with 0 distance travelled.

$\delta\psi\neq0$, $\delta s \neq 0$. This represents a curved line. Arc length = δs and δψ represents the change in tangent of the arc. The magnitude of the curvature of the segment is then $\frac{\delta\psi}{\delta s}$.

This coding is completely rotation invariant, which one of the major advantages to using this technique. This coding however is not invariant to starting point or direction. To relate back to a physical object, the chain will need the start point and starting direction recorded.

The formulation of start point invariance may be accomplished by a simple standard procedure, for example maximum ds*dψ. In this way, the chain may be shifted to

always commence at the same point for an arbitrary shape, regardless of real orientation. In this fashion, a circle will start at any point, a square at any corner, a rectangle at a long-short side transition corner and so on. Examples of this are given below:

Circle

A circle of radius S/2π can be coded by a single link:

δs	δψ
S	2π

This single link completely describes the shape. This link indicates that there is a single edge segment, of length S. The tangent to the arc changes in direction by 2π evenly around the segment.

Square

A square with side s will be coded by:

δs	δψ	
s/4	0	(first straight edge, length s/4)
0,	π/2	(90 degree corner)
s/4,	0	(second straight edge, length s/4)
0,	π/2	(90 degree corner)
s/4,	0	(third straight edge, length s/4)
0,	π/2	(90 degree corner)
s/4,	0	(fourth straight edge, length s/4)
0,	π/2	(90 degree corner)

Semicircle

A semicircle with radius r will be coded by:

δs	δψ	
π*r	π	(semicircle arc)
0,	π/2	(90 degree corner)
2*r,	0	(straight edge)
0,	pi/2	(90 degree corner)

This technique adds rotation invariance to any search algorithms, which may afford a substantial reduction in processing timer required. This coding method also allows highly complex shapes to be described by a few chain links, enabling a high degree of compression to the storage and/or transmission of shape data.

3.3 Practical applications

There are many circumstances where the application of S-Psi methodologies may be more efficient than the alternatives previously discussed.

Consider a circle detection/ arc detection routine where the radius of the arc is unknown. Standard operating practice would generally be to implement an edge detection filter and then use a Hough transform to search parameter space for a peak meeting a certain threshold. Using s-ψ, this can be accomplished directly, with edge detection *per se*. As the edge is tracked, dψ can be monitored for constant values, indicating an arc.

3.4 Conclusions

This section has investigated the formal methodology for using s-ψ coding. These algorithms have been generally discounted as a viable option in the field until now, mainly due to limited work in the area.

This chapter has presented the background and basics of s-ψ coding, and extended the algorithms into generic shape analysis and coding. These algorithms have great potential in many practical applications due to their efficiency and flexibility. These algorithms also use minimal CPU cycles when compared to parameter searching methods, a great advantage in embedded systems.

4 Methodology

Machine vision can be broadly described as the practical application of computer vision. In most applications, the mere capture and storage of images is not sufficient as there must be some knowledge or information extracted from the image or data stream.

4.1 Application model

Image processing can be seen as a number of discrete steps that are of a similar nature, whether the application be agricultural or otherwise.

4.1.1 Image acquisition

In early machine vision systems, dedicated specialist hardware was required to capture meaningful optical data. With the rapid growth in personal desktop computers, webcams have entered the marketplace as a cheap method of face to face, personal communications. This has the advantage of user 'burn-in', where the brands that survive are the most reliable and provide the highest quality to price ratio.

As always, there is a trade off between quality and communications bandwidth. For applications requiring extremely high quality images, dedicated hardware is still necessary, not for the image capture itself, but for the means to transfer the raw pixels to an area of memory where further processing can be done. For instance, a 640*480 video stream with 24bits per pixel at 30fps will require 27MB/sec transfer speed. Communication at this speed was not possible for an external device until the advent of FireWire ™ and USB 2.0 ™.

Currently the popularity of mobile phones with built in digital cameras is fuelling an explosion of technology development and distribution of miniaturised image sensors. Devices such as Omnicam™ OV9620 are providing mega-pixel images from a 5mm

by 5mm footprint device. System-on-chip manufacturers are participating with chips designed for video processing, such as Atmel's new AVR32 processors.

4.1.2 Pre-processing tasks

The pre-process stage of a machine vision system is more applicable to some solutions than others. This can consist of many algorithms to remove noise and excess information. In some applications, this may mean converting to a different colour space, or sub-sampling. At the end of this stage, the data should be in a format most conducive to the processing of the next stage. Noise and artefacts should be removed where possible, and the data trimmed to include only the region of interest.

4.1.3 Image analysis

The image analysis stage is the heart of the application. In this process, the raw data is transformed into information. The different methods for achieving this are too numerous to list or discuss here. Fortunately, there are some similar methods that almost all machine vision programmers use to extract information from images, such as edge detection (by comparing the intensity differences between neighbouring pixels), colour detection and comparing the outlines of shapes to standard "template" shapes to find common features that may identify a particular shape to a high degree of confidence. At this stage, there is no general, unifying theory of analysis. There are, however, some basic tools that can be utilised to quickly build a specific solution.

These tool families may be generalised as:

Feature Extraction

Line detection, Hough transforms, Parameter estimation, Shape descriptors.

Segmentation

Texture, Colour.

Geometry

Chain codes, Shape, Curvature scale space (CSS).

Statistical Methods

Histograms, Probabilities.

Model based

Contours, prior knowledge.

Artificial Neural network techniques are a separate category of tools. These algorithms attempt to build a generic classifier based on some learning methods, however there is a fine line between overtraining – where the network is not generic enough to make generalisations, and under training – where the correct classification rate is lower than optimal.

4.1.4 Post processing tasks

After the main processing, the system has acquired some knowledge of the image, or added to the set of knowledge built up over multiple frames. All practical applications must then utilize this information in some way. In general, post processing tasks may be summed into one of three categories. Each application may use one or more of these.

Reporting: The information can be stored for reporting directly in any format from raw images appended with a date/time stamp to lists of interesting events.
Summarisation: The information can be summarised or archived and stored for presentation or offline process at a later time.
Output interfacing: Most practical applications will have some method of interfacing to the physical world. Produce grading machines usually have automatic sorting facilities so signals from the image process can determine where each item terminates. Also common are switches, alarms, alerts, and digital messages.

4.1.5 Offline processing

Some systems may not have the requirement to process all data either in real time or locally. In these circumstances, it may be appropriate to store the data for offline processing on a different platform. Offline processing is also used for testing purposes for validating algorithms.

4.1.6 Additional inputs

The inputs to a practical system are not only from the visual spectrum of light, but can also be from other information sources such as:

- Infrared(IR)/Near Infrared(NIR),
- Hyperspectral/multispectral imaging devices,
- Electronic sensors (Speedometers, Radar, GPS, temperature sensors, Date/Time Clock),
- User input.

4.2 Personal computer platform

One of the main aims of this project was to demonstrate the use of technology accessible to the mainstream user. A perfect example is the use of webcams. These devices have been developed for the mass market of computer users to communicate easily via the internet. As such, these devices have had huge market success and are now mass-produced in ever cheaper packages. The quality to cost ratio is also improving, and devices that initially cost thousands of dollars can now be bought almost as groceries off the shelf for a few dollars.

The introduction of Universal Serial Bus (USB) was enabling for this technology, as one of the main limitations is communications bandwidth. USB 1.1 has the ability for asynchronous communication rated at up to 12Mb/s, whereas the newer USB2.0 will allows transfers at 480Mb/s.

Taking a standard, moderate resolution of 320 by 240 (although this is becoming low resolution in the current resolution technology expansion), and using 24 bits per pixel results in 1,843,200 bits per frame with no compression, or 55 Mb/s. Clearly, compression is currently used in these devices to reduce communication problems, but for machine vision applications, the less compression the better as lossy compression can lose valuable information.

This project commenced investigation into machine vision applications using an Omnivision SmartCam Deluxe (OV511), which runs at 320*240 pixels and streams via USB at 25 fps.

The receiver end of this type of communications stream must have the processing capacity to read the incoming data, uncompress it if required, process each frame, and produce output. Any desktop personal computer purchased after about 2001 meets these criteria. Older machines have been used successfully with frame grabber cards, but these technologies are disappearing as newer digital communications appear.

Microsoft DirectShow ™ technologies have been used as the cornerstone of the PC based machine vision applications developed in this thesis. DirectShow is a member of the DirectX (Hardware Abstraction Layer software) suite of tools. This suite allows developers to program in a way that suits the application without knowledge of the end user hardware.

DirectShow tools started with Windows 3.0 in 1990 and are currently at version 9.0c. DirectShow is a modular based architecture, with registered and documented standard input and output interfaces between modules. The DirectShow Software Development Kit (SDK) can be downloaded as part of the Microsoft Platform SDK from www.microsoft.com/downloads.

4.2.1 DirectShow

DirectShow was specifically devised for handling video and audio streams. Each step of the chain from input, processing and output are represented as separate modules (called Filters). The filters have standard input or output interfaces (called Pins) that stream data between filters. Standard control interfaces can be accessed to change the behaviour of the filter. The streaming procedure is logical only, the data is retained in memory at a certain location and it is a pointer to the data that is passed between processes.

The filters are connected in a chain (Filter Graph) that has one or more input devices (for example, video capture devices or files) and one or more output devices (video renderers to display on the default screen, file writers, or null renderers consume frames).

As a frame of video (or audio) is received by each filter, it performs specific processing, and then forwards the frame over to the next filter in line. Notifications to the host application can be initiated by any filter.

One of the tools provided with the SDK is GraphEdit – a graphical user interface that allows the creation and modification of filter graphs. Figure 22 demonstrates an example screenshot.

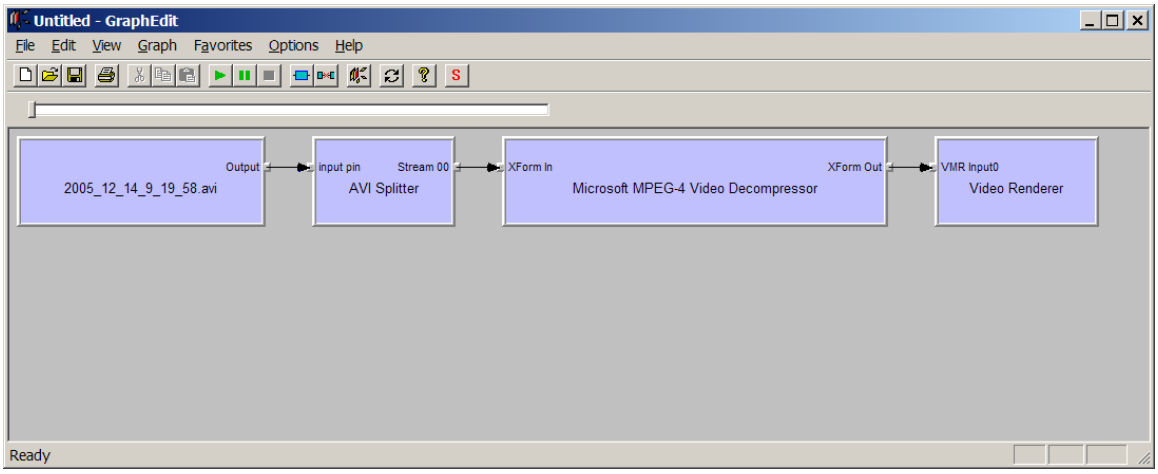


Figure 22 An example of graphedit

In the example in Figure 22, there are four modules.
The first is a file reader, which is reading from the file "2005_12_14_8_19_58.avi". The data read from the file is passed to an AVI (Audio Video Interleave) splitter filter, which decodes the raw data stream into frames and multiple streams, if they exist on the file. The next filter decompresses the frames, as the file was stored using MPEG-4 (Moving Picture Experts Group, Version 4) compressor. The last filter renders the stream to screen.

A set of standard filters are available for use with GraphEdit and these are:

File Source (Async.) – This filter reads data from any file asynchronously, i.e., the processor is not blocked waiting for the read to complete. There are other filters that read from internet addresses. Standard interfaces can be used to set the current read address, bytes required and frame frequency.

File Writer – This filter dumps raw data to a file. The data must already be pre-coded into container format such as AVI or WMV (Windows Media Video) if a standard media player is to be used to replay the file later.

Video Renderer – this filter will render the most appropriate size and colour space to match the current screen capabilities. The output window can be moved, stretched and resized just as a normal window can.

Infinite pin tee –This filter is used to split a single stream of data into multiple streams. Each stream of data is distinct and separate and can have different processes performed. This type of filter can be used, for example, to both process the data and record to a file at the same time.

Compression/Decompression (codecs) – Microsoft Windows contains several codecs pre-installed and others are installed with various software packages. Standard codecs such as MPEG-4 and Microsoft Video-1 are most likely to work with any other machine loaded with Microsoft Windows.

Capture device filters – Almost every webcam and other video capture device installs a DirectShow filter as part of the install process. Inserting one of these filters into GraphEdit then allows for direct, real time streaming video from the device. Other user filters can then be added to the filter graph for machine vision processing.

4.2.2 Programming

In addition to using GraphEdit to create filter graphs, this can be done programmatically. Microsoft DirectShow can be programmed in Visual Basic (VB)

and Visual C (VC), however reduced functionality is available in VB. The user can create and run graphs from VB, but cannot access all the control interfaces. Projects for this thesis have been written in Microsoft Visual Studio 6 and Microsoft Visual Studio.Net.

Overviews of DirectShow programming can be found in books such as Pesce's *Programming Microsoft DirectShow for Digital Video and Television* [73], or on the Microsoft MSDN website at http://msdn.microsoft.com/library/

Appendix B provides a detailed programming tutorial, including code snippets and templates for immediate use. This tutorial provides a starting point for the development and deployment of a machine vision solution in hours.

4.3 Embedded platform

The PC/webcam hardware platform is suitable for many applications where power and stability are provided. There are many applications however, where the processing should be performed in a mobile device, preferably using limited power. At the time of commencing these projects, there were no devices available on the market for a reasonable price that enabled embedded processing at the camera.

The author has designed and manufactured an embedded PC and camera combination based on an ARM (Advanced Reduced instruction set computer (RISC) Machine) processor with an Omnivision image sensor module. This device has been labelled the Rugged Outdoor Camera (ROC). One of the main benefits of this platform is very low power consumption. The processor runs full speed using less than 100mA at 6V, which means a single torch battery (4Ah) will power the system almost 2 days. A rechargeable battery with a small solar panel (6V/1W) will power the system indefinitely. This makes the system ideal for both mobile applications and remote sites without mains power.
The basic processor is an Atmel™ ARM9 AT91RM9200, which is a low powered, full function PC. Full specifications may be found at:
http://www.atmel.com/dyn/products/product_card.asp?part_id=2983

The processor contains, in summary:
200MHz CPU
4 serial ports
USB device
USB host
SD/MMC interface
SPI
TWI
16kB internal SRAM
General Purpose Input/Output pins (GPIO)

As a Reduced Instruction Set Computer (RISC), there are fewer instructions, and the processing is streamlined. The board design also includes 16MB external SDRAM, 4MB serial Dataflash, Secure Digital Card interface, 4 channel Analogue to Digital Converter (ADC), and an LCD interface.

The camera interface is tied to unused GPIO pins. This has the advantage of providing flexibility, as any image sensor module can be used, but it is not as fast as the Direct Memory Access (DMA) transfer techniques implemented on newer Intel XScale processors. A schematic diagram of the high level hardware design is given in Figure 23 and brief descriptions of the embedded hardware are given in Section 4.3.1.

4.3.1 Hardware design

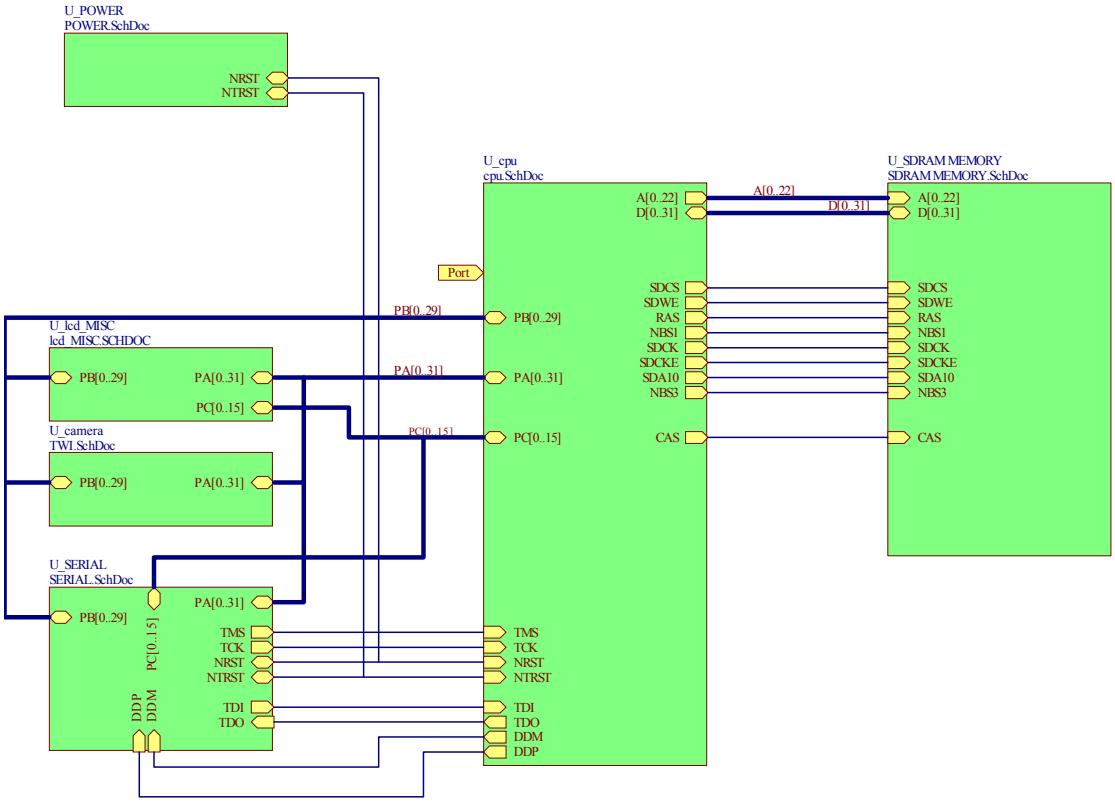


Figure 23 The design for an embedded processor module design

The hardware design is separated into seven logical blocks: Power, CPU, Memory, Serial Devices, Camera, and I/O devices.

Appendix C provides detailed schematic diagrams, PCB layout diagrams and a parts list for the embedded processor.

4.3.2 Software architecture

There are several operating systems that could be used on this hardware platform, including Windows CE ™, and versions of Linux. The downfall of any operating system, for these purposes, is the overhead processing. This project has not used an operating system, instead basing all tasks around the processing of each frame. As the CPU is not using DMA access to read the image frame into the memory buffer, no other processing can be scheduled simultaneously. There is a small amount of idle time available at the end of each line, and a substantial number of cycles at the

end of each frame. During this time, the processor can schedule any off-vision processing required.

There are generally two parts to software on an embedded system: the bootloader and the main program. The bootloader runs first, initialising the hardware and preparing the system for the main application. Generally the bootloader is stored in some non-volatile memory area accessible to the system on start up. In this system, the first 12KB of the Dataflash is automatically downloaded into internal SRAM on reset.

The pseudo code is:

Initialise Stack pointers
Setup GPIO pins as peripherals
Swap clock speed to 180MHz
Initialise External SRAM
Initialise UARTs, TWI, SPI, Clock,
Check for bootloader updates on SD card, serial port or other interface devices
Copy Main program from Dataflash to external SRAM
Set the Program Counter to Main Program initialise routine.

Note that the main program can be anything from a small dedicated process to an embedded O/S such as Windows CE or Linux.

4.3.3 Embedded image processing.

In general, an embedded image processing application is a devoted process. Adding an O/S adds overhead. The benefit of an O/S is more portable code, but at the expense of many CPU cycles.

Generic embedded image processing.
The following pseudo code summarises the steps involved in bringing the system to a functional state and reading, storing and processing images.

Initialise Serial interfaces
Initialise SD card interface

```
        Initialise Interrupt handlers
Main loop:
        Wait for Vsync (beginning of frame)
Row loop:
        Wait for Href (beginning of line)
        For 320 pixels:
                Wait for PClk (pixel valid signal)
                Copy 16 bit pixel values to internal SRAM area
        Copy Line data to external SRAM
        Goto Row loop until 240 lines read
```

Transform Frame. The same code may be used as any DirectShow filter. Perform output stage – report or accumulate results, physical interface, write frame to mass storage device if required.

Check user interface – check for incoming commands from local or remote users.

```
        Goto Main Loop.
```

This embedded hardware platform will provide an entry point into many applications where a machine vision solution was previously not feasible due to cost, power, or size limitations.

4.4 Testing framework

Regardless of the platform used, there must be a rigorous testing framework in place for any application that is expected to be used in real world applications.

The measurement regime for any application will usually be specific for that application, but there will be areas of global consistency:
Ground-truthed accuracy

Robustness and reliability
Measurement statistics
Ongoing accuracy

Each of these will be discussed below as they are critical to the success of the research described in this thesis.

4.4.1 Ground Truthing

After a prototype application or algorithm is written, there must be a way to test the accuracy for that method. Ground truthing provides a means of manual determination of the actual/correct results, which can be checked against the machine vision system results. In most situations, there is merit in the creation of a test harness, consisting of a number of ground truthed video clips which can be processed automatically. The generated results are compiled and compared to the correct results, and an accuracy figure is generated at the completion of each run. In this way, algorithm changes can be tested over a range of inputs without manual intervention. This allows rigorous testing after each significant change to the algorithms.

4.4.2 Robustness and reliability

Practical applications will always require some robustness and reliability trials. When the system is in the field, regardless of the configuration, there will be physical impacts on the system. Consider a system using a laptop PC and webcam. Cooling and airflow issues should be checked, mounting for the camera should be sturdy. For a normal production system, downtime is not an option. The application must be reliable enough to run unassisted whenever required.

4.4.3 Measurement statistics

For any application, there will be different levels of classification results. For example the animal species identification may be binary; the animal is either a sheep or a goat. Other results may be less clear cut. For example classifying a defect in produce grading may depend upon the size of the area, subject to measurement

ambiguities. Other applications, such as the citrus texture measurement discussed previously, are manually subjective and difficult to ground truth effectively.

For each type of application, appropriate statistical methods should be used to determine the accuracy of the automated system. Key Performance Indicators (KPIs) are generally accepted as useful tools in measuring performance in both financial and non-financial situations, and these methods can be applied directly to machine vision applications.

4.4.4 Ongoing Accuracy

In any application there may be accuracy drift over time. This happens not only due to software changes and hardware degradations, but also due to changes in the items under consideration. For example, new types of defects may be present depending upon seasonal effects. For all these reasons, an annual accuracy review is recommended.

5 Macadamia yield monitor application

5.1 Introduction

The primary purpose of the macadamia yield project was to design and develop a vision based yield monitoring device. This device relates quantity of nuts collected to collection location with sufficient accuracy to identify the yield from individual trees. A prototype system for use within the Macadamia industry plant breeding program has been developed as part of the project.

Mechanised yield assessment offers an opportunity to reduce the cost of assessing field trials [2]. This project was commissioned by Horticulture Australia Limited to evaluate yield assessment using mechanised processes (MC03020). It has been estimated by Hardner [3] that the cost reduction by mechanised harvesting could be up to 59%, or $AU109,824 pa. This offers substantial savings to the individual growers undertaking progeny trials, as well as the funding body which provides research funding for progeny trials.

The standard measurement of harvest yield is the weight of nuts collected after several runs of the harvester along rows of trees. As cultivars are generally planted in mixed rows, the count or weight of nuts from each square meter under each tree is required. The usual method of cultivar testing is the manual collection and counting of fallen macadamia nuts by workers. This method is relatively accurate, but is becoming too costly due to rising employment costs and the general reduction of available skilled workers in the agricultural industry. An ideal solution to automating this task would be to mechanically weigh the nuts dynamically as they are collected, however this is not feasible in this situation due to the spatial restrictions inherent in a standard harvester with augers.

Macadamia trees are planted at intervals of 6m in rows 8m apart. The nuts (see Figure 24) are harvested between May and September each year. As the nuts ripen, they fall to the ground which has been cleaned of weeds by a mulcher prior to the start of the season (Figure 25).

Figure 24 Macadamia nuts on the tree.

Figure 25 Macadamia nuts on ground ready for harvesting.

A standard harvester utilises a set of rollers with flexible fingers (pinwheels) spaced at such a distance to capture nuts between the fingers (Figure 26). The nuts are then carried around by the fingers until removed by spikes between the rollers into the auger system. The stream of nuts are transported by a series of augers across the direction of travel to one side of the harvester, then to the back of the harvester where they are collected into a large collector bin.

Figure 26 Macadamia nut harvester and auger.

The nuts at any one point in the auger system have been accumulated from a thin diagonal slice of the field, not a horizontal slice. The exact original position is dependent on the ratio of harvester speed and auger speed (Figure 27). Coupled with the fact that there may be random delays in parts of the stream due to auger action or jammed rollers, this harvest method cannot provide any spatial data accurate to an individual tree.

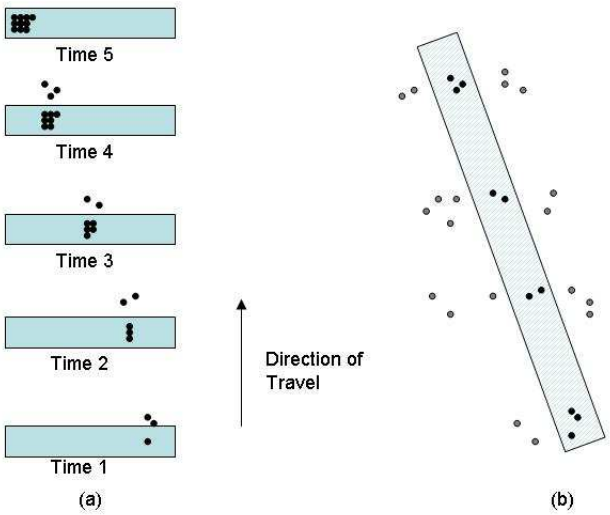


Figure 27 a) Sequential accumulation of nut stream exiting the system at time period 5. b) Original field position for nuts

NCEA initially prepared a report evaluating several options and considered that using a vision system was most likely to succeed. Billingsley [1] verified the feasibility of a vision system and initial project results were reported in Dunn and Billingsley [74]. Because of the complicated Green's function and the requirement for position sensing in both x and y, any sensing system that inspects the rollers must detect lateral position accurately; a single mechanical sensor will not suffice.

A further application of this project in the commercial arena has also been identified. Yield mapping of commercial harvests with this technology will provide an immediately intuitive, visual indication of crop production patterns. Armed with this information, producers may make more informed decisions about resource management functions such as watering and fertilising macadamia trees.

5.2 Key performance indicators

The following KPI measurements have been identified by representatives of the Australian Macadamia Society, CSIRO, QLD DPI and NSW Dept of Agriculture as relevant metrics to measure the performance of the system:

1. Cost.

There must be a positive net effect on the bottom line of the implementation of this project into the harvesting process of varietal trials. The current cost of manual harvesting is $7.70 per tree per harvest [2]. The maximum cost of the machine vision system process should be $5.70 per tree per harvest [2].

2. Count Accuracy

The current accuracy of manual collection of nuts cannot be 100% as there is some ambiguity of nut origin at the crossover points in the field. No matter what system is used, this ambiguity can not be overcome. However, a consistent measurement for each tree will provide the required information.

The accuracy of the vision system should exceed r^2=0.90.

3. Position Accuracy

The yield map reported by the machine vision system should be accurate to within 10cm spatially within the orchard.

4. Speed

The machine vision system and harvester should be capable of 3m/s ground speed (10km/hr) with no loss of accuracy.

5. Size of machine

The harvest of varietal trial nuts has been envisaged as a single person operation. As such, the harvester must be easy to transport and use. The harvester should fit on a trailer for ease of transportation and be loaded/unloaded/operated by an individual.

6. Type of orchard

The machine vision system and harvester should be capable of operating in orchards ranging from 5m x 2m to 10m x 5m spacing between trees.

5.3 Materials and methods

5.3.1 Nut detection hardware

For a machine vision system to work effectively there must be a clear, unobstructed view of the area. Most macadamia nut harvesters have either a single back auger, or dual banks of pinwheels. In either case, there is no feasible mounting point for any type of camera to have a clear view of the nuts before or during the pick up operation.

Hidden Valley Plantation has designed and built a harvester that is ideally suited to a machine vision solution. The harvester has a single, front mounted auger which provides around sixty degrees of visible pinwheels with nuts (Figure 28).

Figure 28 Macadamia nut harvester with single front mounted auger.

The harvester has 1800mm of working width and standard cameras usually have 320 pixels width resolution. To achieve the total coverage required with 3 cameras, each pixel must be at least $2mm^2$. As average nuts have diameters larger than 15mm, this resolution is sufficient to discriminate the nuts from the background.

Cameras with 3.6mm focal length lenses were positioned 600 mm from the rollers in a sheltered cowling. This focal length and position provides 600mm horizontal field of view. The pinwheel and cameras were covered so no sunlight enters, all light was provided artificially. This reduced changes due to shadows as the tractor moves in and out from under trees. Figure 28 displays a picture of the prototype at Hidden Valley Plantations. Note that the unit has a lid to shield it from direct sunlight and the lid is hinged to allow access to the rollers for cleaning.

Initial trials were undertaken early in this project (2003) with standard white rollers on the harvester. Video footage was acquired with a JVC analogue video recorder mounted behind the pinwheels. With use, the standard white rollers become smudged with dirt and almost indistinguishable from the nuts being collected. The solution to this problem was to have the pinwheels moulded with blue tint in the plastic. This provided adequate colour separation to distinguish background (pinwheels) from foreground (nuts and trash).

In terms of image acquisition and data processing, trials commenced using a fan-less 533MHz VIA mini-ITX PC, encased in a sturdy Pelican case. Windows 98 was used as the operating system platform. This system was chosen as it had no moving parts, with a 512 MB Compact Flash card as the storage media. Three Logitech Quickcam Pro 4000 cameras were used as the image sensors. These cameras provided high image quality for a reasonable price (AU$150.00). As each camera required a USB host port to provide the bandwidth for full streaming video, a PCI USB host controller card was inserted to provide the third USB port.

This system was designed to be 'headless', i.e., with no monitor attached. This provided extra ruggedness, but presented some difficulties in user interface.

After problems with the speed of this system became obvious, the project moved to a 1 GHz VIA motherboard. This still presented challenges with the user interface and setup time. A DELL Latitude D600 laptop was then utilised as the main processing unit. A parallel connection to a 24*2 segment LCD and switch provided a user interface mounted within easy reach and view of the driver. The Rugged Outdoor

Camera (ROC) system developed at NCEA was then customised for use in this project.

The final hardware configuration consists of four ROCs – three 'Nutcams' and one 'Treecam'. Each unit was connected via 2 serial UART lines, one channel upstream and one channel downstream. The Treecam handled the accumulation and storage of all data onto a single SD card. The data may be downloaded at the end of each run, or each day for post processing.

5.3.2 Location hardware

After the image processing phase, the location of the identified nuts must be recorded to the required accuracy. Initial project designs included a bicycle wheel attached to the tractor for odometry purposes. The location of the tractor on the farm could then be recorded or identified manually at the start of each run. The wheel proved to be unsatisfactory due to bouncing and skipping.

Global Positioning System (GPS) options were investigated. Base level GPS (eg Garmin 18) are accurate to +- 3m, which would not provide individual row information at any given point. Differential GPS (DGPS), works on normal satellite signals, as well as an error correction signal, either from another satellite service (eg SBAS) or radio beacon (coastguard beacons are available off the east coast of Australia. See Appendix 1). DGPS provides sub-meter accuracy within the field. A DGPS system (CSI wireless Minimax) was purchased to provide location information to the system.

The interface is RS-232 with standard NMEA signals (see www.nmea.org). The 'GPGGA' string was chosen as the input, which provides latitude, longitude, height, number of satellites, and signal quality. Initial problems occurred due to the aerial being mounted close to the harvester. Multipath problems from signal bounce caused loss of accuracy. The aerial was subsequently mounted on a 2m pole, bringing the antenna up to head height (Figure 29 and Figure 30). This significantly increased the accuracy attained.

Figure 29 Treecam and GPS receiver.

Figure 30 GPS antenna and radar mounted on the macadamia nut harvester

Even with this advanced GPS, the system could not discriminate position to the required in-field position of 10cm. A ground speed radar (Dickey-John) was sourced to provide odometry information (Figure 31).

Figure 31 Ground speed radar mounted on the macadamia nut harvester

The radar provides a frequency modulated square wave relative to the ground speed. The radar was mounted at the front of the harvester, aimed at the ground at 30degrees to the horizontal. The output of the radar was connected to a Timer/Counter input pin. Each change of state adds to a register, which accumulates the total distance and current speed. After trials and calibration, each change of state was triggered after 6mm travelled (Figure 32). This allowed lateral distance to be measured to the required accuracy, but not transverse distance between the rows.

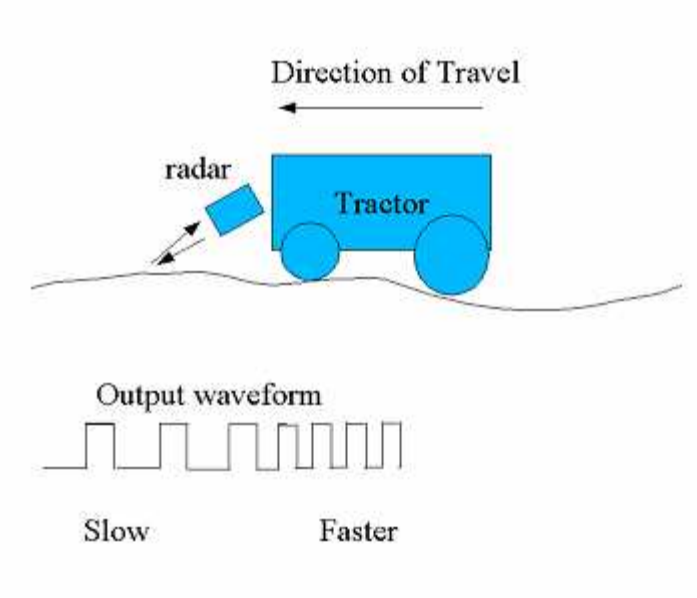


Figure 32 Ground speed radar showing radar direction and output waveforms

A new location method has been devised to locate the tractor accurately both laterally and transversely in a particular row by mounting a camera on the harvester focused perpendicular to the direction of travel (Figure 33). This Treecam identifies tree trunks streaming past and records trunk width and position. The position in the image that the tree is identified is directly and linearly related to the angle from the camera to the tree. If the trees are identified at 60frames per second, and the tractor is travelling at maximum speed of 2m/s along a run 0.5m away from the tree line, then the system should detect at least 10-15 frames of identified tree moving from left to right in the image. Once we have identified a tree, we can use triangulation methods to determine lateral distance, as we know current speed from the radar.

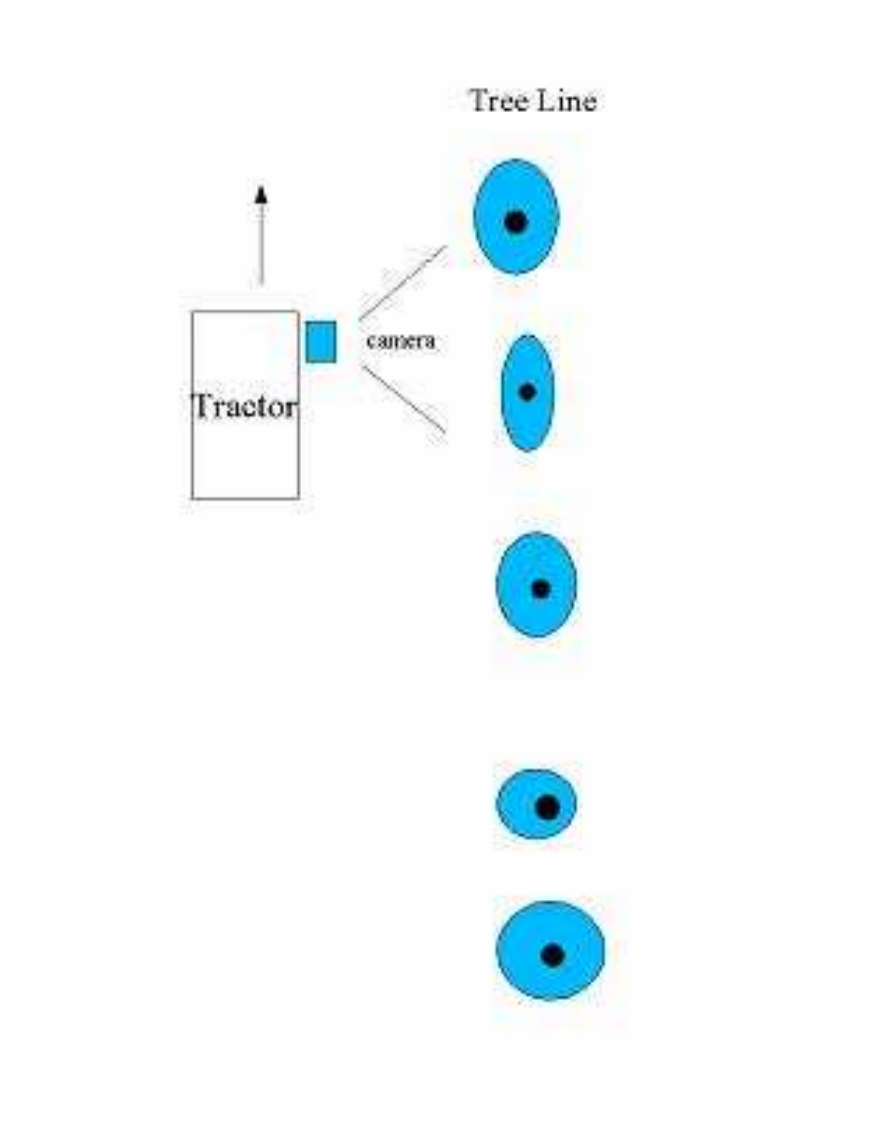


Figure 33 TreeCam – local positioning by tree detection.

Radio Frequency Identification (RFID)

A system has been developed based on the Texas Instruments TIRIS™ Radio Frequency Identification (RFID) architecture. The reader uses Electromagnetic coupling to charge a passive tag. The tag, once charged, has enough energy to burst transmit the data stored, typically a unique 64 bit number, back to the reader.

With unique tags at known positions in the plantation, this technology can completely remove the requirements for GPS input. Consider a plantation with a single tag on the first tree of each row. As the tractor passes the tree, the system identifies exactly which row in the plantation is currently being harvested. The other sensors can then position the harvester accurately down the row.

Table 1 summarises the advantages, disadvantages and accuracy of the different location methods. Note that for the full system, a fusion of several items is required.

Table 1 Location technologies.

System	Advantages	Constraints	Accuracy
GPS	Standard units available, cheap	Occasional dropouts, Variable reliability under tree cover	+- 3m
DGPS	More accurate, standard units available	More expensive, Variable reliability under tree cover	+-<1m
RFID	Cheaper than GPS, reliable in all areas.	Requires knowledge of unique tag placement within the plantation. Reader must pass within 1 m of tree	<1m
Standard Odometry	Fitted standard on most new tractors	Indicates current speed only, not position	5% km/hr
Radar Odometry	Robust, non contact	Indicates current speed, not position	+-3% m/s
Tree Triangulation	Robust, non contact	Must know distance between trees and current speed.	+-12mm

A fusion of the radar odometry, tree triangulation and either RFID or DGPS produces accuracy in advance of any current systems in the market.

5.3.3 Nut detection software

The main requirement for this software was to process an image in real time to determine the presence and location of nuts on the roller. This was achieved using colour and shape information.

5.3.3.1 Colour

Red/Green/Blue intensity plots for sample nuts and the blue roller background is shown in Figure 34. As shown, in the left-most plot, there is a distinct separation in the Red/Blue colour channels that will allow accurate identification of non-background areas. Figure 35 illustrates that trash such as leaves and twigs picked up by the rollers cannot be separated from nuts based on colour data alone. This means that there must be some shape checking to ensure only nuts are counted.

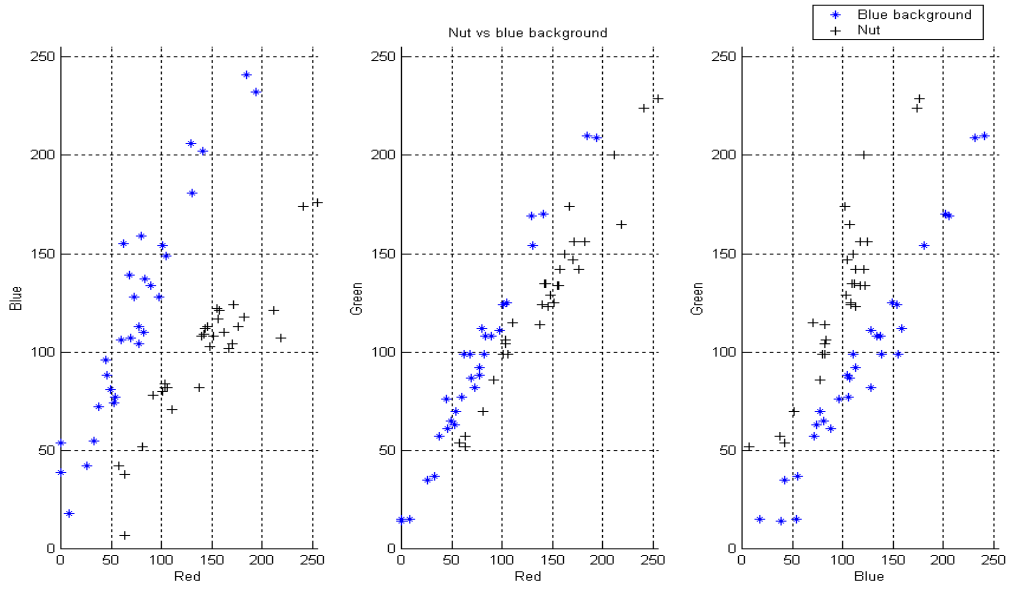


Figure 34 Intensity contrast for nuts on a blue background.

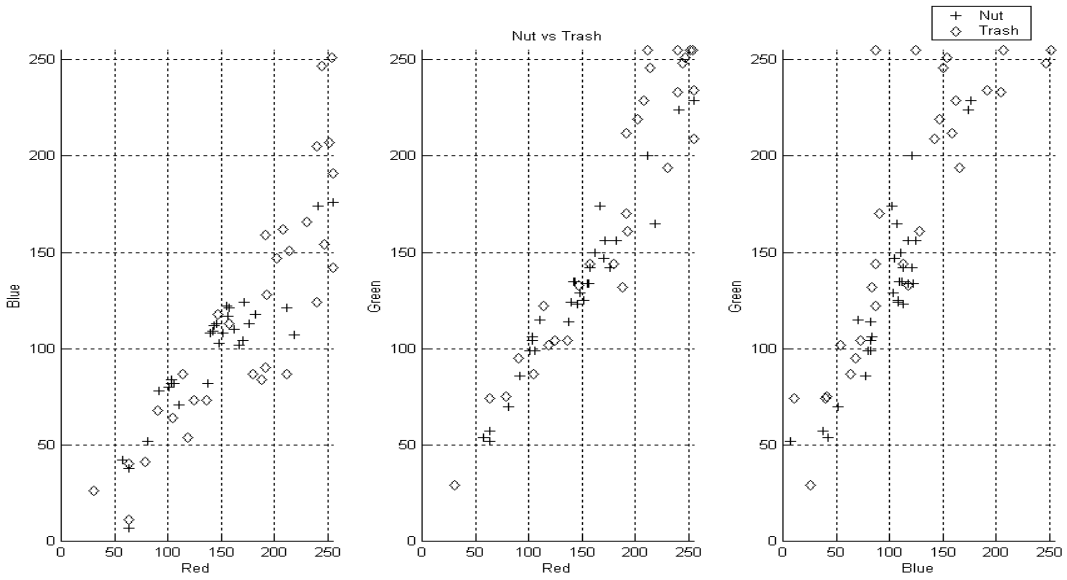


Figure 35 Intensity contrast for trash on a blue background

5.3.3.2 Lighting effects

Given that the harvesting of macadamia nuts is an outdoor occupation, there will generally be varying degrees of light present on the roller and nuts as the harvester goes in and out of shadow under trees.

To reduce the effects of trees and varying light conditions, a cover was manufactured for the harvester top to completely cover the pinwheels and cameras. Lights were mounted between the cameras to provide complete and constant light coverage of the pinwheels.

Three 12V fluorescent lights were attached to the system initially, but enhanced foreground/background contrast was realised with four 12V/50W yellow filtered halogen lamps.

5.3.3.3 Shape

For each frame to be processed, the shape algorithm performs the following tasks:

Divide the frame into a grid at a reasonable pixel size, depending on the distance from the camera to the rollers. In the prototype system, this value was around 4 pixels, which was slightly less than half the diameter of the average macadamia nut. This ensures that no nuts will be missed by a fast scan of the image.

At each of the grid points, search for a non-blue value, indicating the presence of something on the rollers – either a nut, or trash.

From this identified object point, step vertically up, down and right until a boundary is reached. This provides at least three points on the edge of a potential circle. If any direction is more that twice the diameter of an average nut, this point should be skipped as it is probably trash. Figure 36 displays an example grid search with the first circle identified.

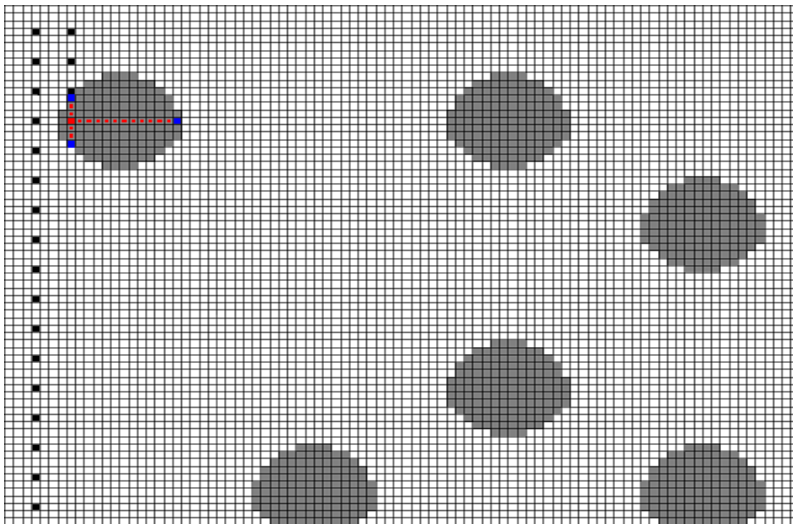


Figure 36 An example scan of an image grid

After a potential circle has been identified, we need to verify whether or not a circle actually exists in this position. Given that the image is discrete and the circle boundary is not an exact line, we can check two points, at the radius plus or minus a percentage T_B from the centre for a boundary in this region. If the inner pixel meets the target value and the outer pixel does not, then there is an edge between the two.

In a digital circle, there are approximately 2*pi*radius pixels on the edge and thus 2*pi*radius discrete angles to check. Given that there is a size limitation, the circle can be approximated by only 16 points, at angles 0, pi/8, 2*pi/8, 3*pi/8 radians, rotated by pi/2 radians 4 times to complete the circle. After checking and summing the matching angles, if the count is over a threshold percentage of the angles then there is a circle described by (a,b,r). These circle parameters are stored to ensure that the same circle is not found more than once. No further points on the main grid in this area are checked.

Figure 37 below, is the virtual search space of an example image. This is not actually generated by the system, as only selected grid points are evaluated. Figure 38 shows the end result of this processed example frame, with the identified nuts circled. Note that the two nuts close together have been identified correctly, and the small leaf in the image has been discounted correctly.

Figure 37 An example of virtual search space. The black pixels have been identified as the background, the white as non-background

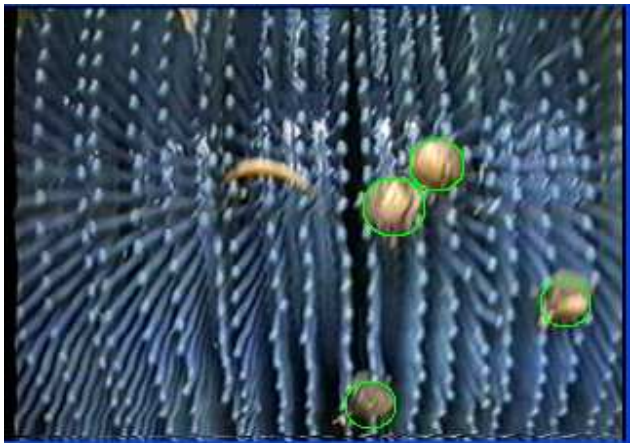


Figure 38 An example frame after being processed

5.3.4 Tree detection software

The algorithm for tree detection is based solely on the relative brightness intensity of portions of the image. For each row, the average pixel intensity is measured. Any pixel with less than half of the average intensity is marked as dark – or trunk. The columns are then processed with a low pass filter to find the darkest vertical area. This area is marked as a tree trunk and the horizontal position recorded. Note that in the absence of a visible tree in the closest row, trees in further rows will be identified. These can be easily differentiated by the rate of change of the position of the identified tree.

Figure 39 displays an example frame from a webcam mounted sideways on the harvester, aimed at the row of trees. Figure 40 shows the example frame processed for trunk position.

Figure 39 An example of an input frame showing a tree trunk

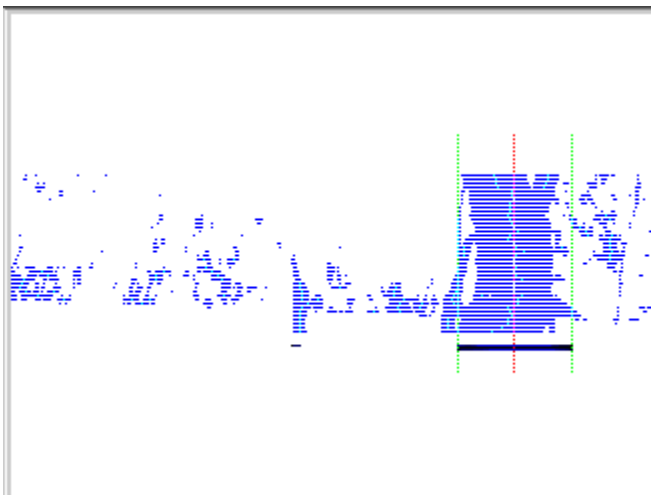


Figure 40 An example of the processed frame showing the identified tree trunk

The centre of the tree is identified at harvester height, not ground height. This means that trees not growing straight will need to be measured in the field at harvester height, not as planted.

The imaging unit is linear so the distance from the centre of the image to the centre of the tree trunk can be converted directly to an angle. The field of view of a standard camera with a 4.5 mm lens is 40 degrees. This means that a tree is in view from 20degrees ahead of the harvester to 20 degrees behind.

Note the trunk has been identified with the centre and width. Plotting only the centres for each frame down a row of trees produces graphs such as Figure 41. The interval between each frame is 0.033 seconds.

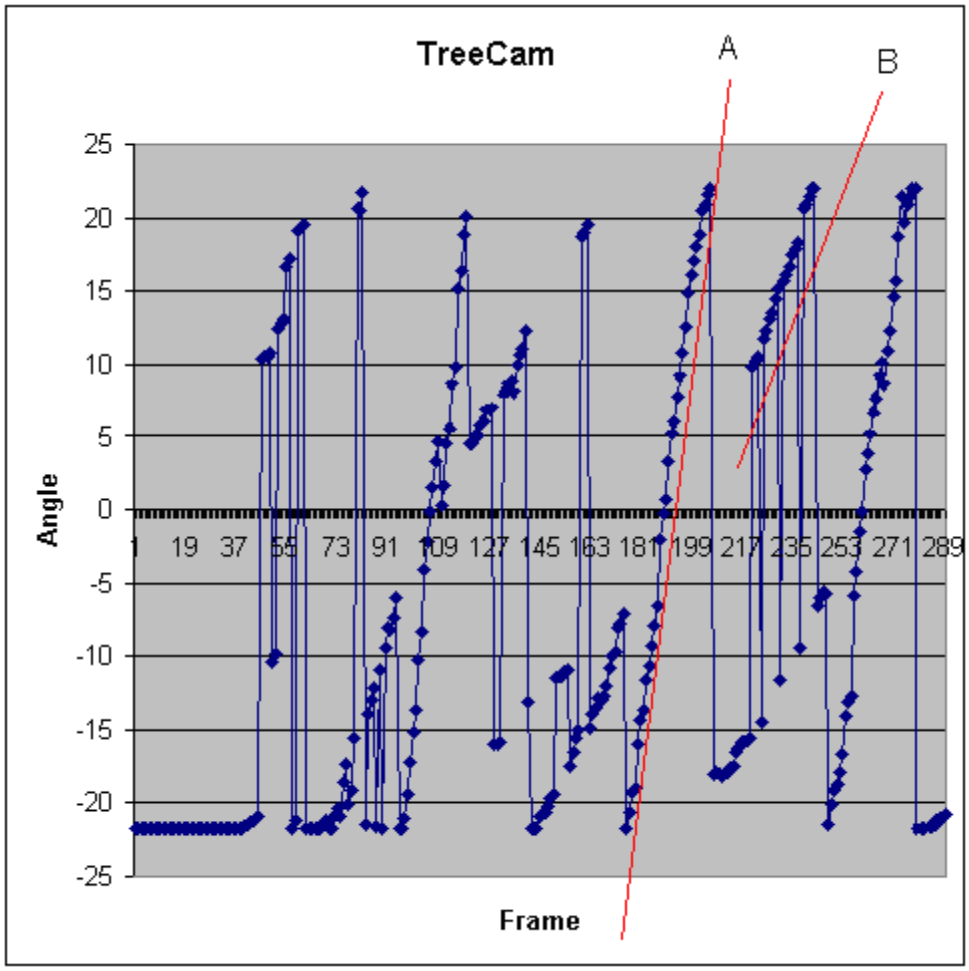


Figure 41 An example of Treecam data plotting tree sightings for each frame

Note that the slope of the locus formed by the identification of trunks, together with current speed information, will provide lateral distance from the camera to the trunk. In the case of 'A' above, the tree identified is clearly a tree in the current tree line, but 'B' is one row over. This system is invulnerable to noise as there is feedback from the odometer. If, for example, someone walked past the Treecam, the lack of odometer data will show that the tractor is not moving and invalidate the reading.

5.3.5 Sensor fusion

In the previous sections, the various sources of data available to this system have been described. The next task is to merge the data from the following sensors:

- 3x Nut identification cameras,
- 1x Tree identification camera,
- 1x Differential GPS, and
- 1x Radar odometer

Each sensor is providing data independently and at various speeds. To accumulate the data each sensor stamps the message with an internal timestamp. These timestamps, accurate to microsecond level, allow post process algorithms to align the data streams.

Each data item is logged in the following format:
<Microsecond>, <data type>, <data>
And the file is labelled with the date/time stamp of the start of the run.

The different types of data are:

NUT – This signifies the identification of a single new nut.
Associated Data: Camera id, horizontal position of the centre, vertical position of the centre, nut diameter.

TREE – This signifies the detection of a tree from the tree camera.
Associated Data: horizontal position of the centre of the tree, width of the tree.

ODO – Each low-high change of state of the signal from the ground speed radar is logged.
Associated Data: none

GPS – The GPS system has been configured to output the standard 'GPGGA' string data at 5Hz.

Associated Data: Full GPGGA string. GPGGA, Time, Latitude, S, Longitude, E, signal type, #satellites, HDOP (signal quality), height, M, filler.

Figure 42 is an example of graphing GPS data directly.

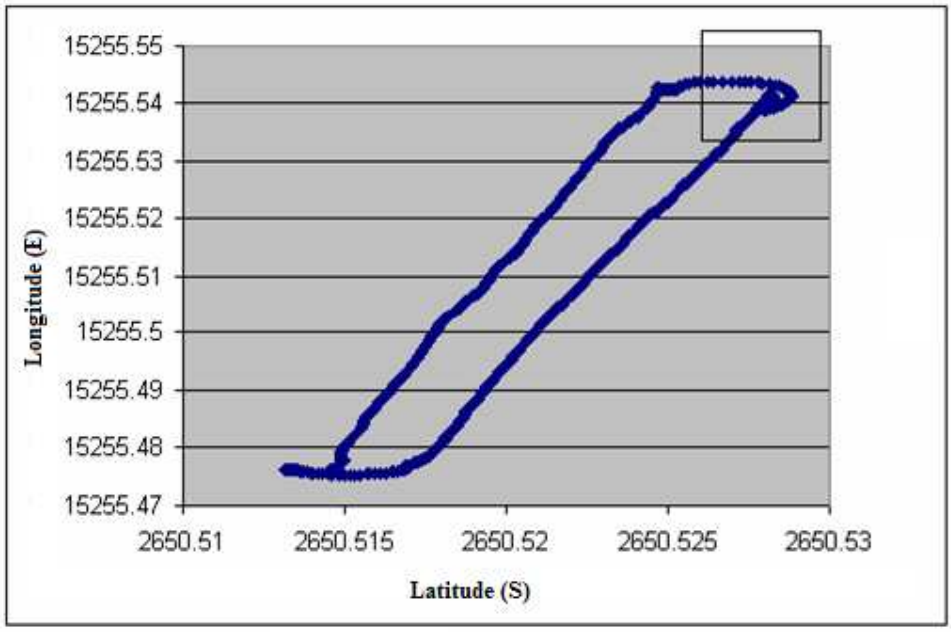


Figure 42 An example of the GPS data

Figure 43 displays a close view of the area in the square above.

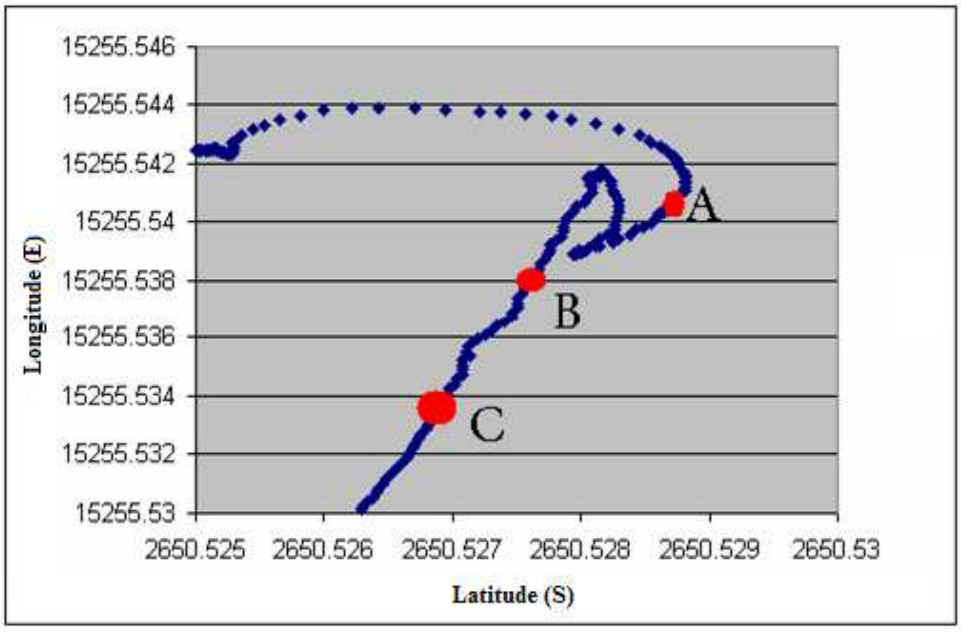


Figure 43 A close view of the GPS sample data

Figure 37 clearly demonstrates where the harvester has turned at the end of the run at point A, reversed to get in line for the next run, and continued. Note also, the deviation in the line between point B and C, less than a meter off the true straight line course, but obviously demonstrating that GPS alone is not enough for 10cm accuracy.

At a speed of 2m/sec, the ground speed radar produces spikes at around 5 ms intervals, or 10mm. The obvious initial way to use this data is to take the heading from the last two or more GPS positions, and interpolate the odometry points between them. Problems arise, however, as the GPS data is not reliable under tree coverage and accuracy decreases, at times drastically. This may cause substantial heading errors, invalidating this method.

Post-processing algorithms are used for amalgamating the data. In this case, we have standard 'tracks' along the field which provide a template for the movement. The GPS data is used to identify which track we are on. The tree data is then used, together with the instantaneous speed, to determine lateral and transverse positions of the tractor.

5.3.6 Post processing algorithm

The most important part of data fusion is accurate timing. If unsynchronised data is accumulated, there is no method of rescheduling back into the original reference time sequence. Each sensor system uses an internal high-resolution timer to timestamp messages.

In the post-processing step, each piece of data is examined and used to assist in the transfer of raw data to accurate yield maps. The steps required in post processing are detailed below. This process performs a fusion of the sensor data to give a best estimate of the absolute location at all times.

A. Determine Harvester Location and Heading (World Co-ordinate System)

1. Either raw GPS records or predetermined GPS path points are passed through a b-spline type smoothing algorithm
    - In the case of Raw GPS records the distance between control points is dynamically determined by the HDOP signal quality field in $GPGGA.
    - In the case of predetermined paths, the controls points are aligned with known locations.
2. Odometer readings are placed on interpolated positions on the b-spline curve.
3. Treecam records are examined simultaneously, and when a successful triangulation occurs, this determines closest tree by comparing position with a reference map.
4. Positioning error is calculated (difference between Treecam and map position), and initial GPS signal records in current run are back-adjusted to allow for the error.
5. Steps 1-4 are repeated until calculated Treecam position matches the reference map within 5mm
6. When a Treecam position is finalised as per steps 1-5, the cumulative position error is applied to all future raw GPS records, and the algorithm then continues to work through the records in a progressive fashion until done.
7. At any point, heading is assumed to be aligned with line connecting closest corrected odometer positions.

This method of progressive post-processing with Treecam has been very successful, capable of maintaining position even with very poor GPS signals. However success is dependent on two factors:

- A high percentage of Treecam 'hits': Provided trees are reasonably maintained, as is usual in varietal trials, 80% is quite achievable which is adequate for the algorithm.
- A high accuracy in the GPS position at the start of each row: Differential GPS is adequate, but other options need to be considered where DGPS coverage is poor.

B. Determine the position of nuts within the harvester (Local Co-ordinate System)

1. The positions of separate components (cameras, GPS etc) are recorded using a local co-ordinate system, measured in metres with the GPS antenna as the

origin and direction of travel as the y-axis. This information together with component serial numbers etc is stored in a harvester configuration file.

2. When a nut is detected, pixel (x, y) is converted to the local co-ordinate system and offsets are applied to allow for the relative positions of components.

C. Transform Nut position to World co-ordinates, and assign to tree in reference map.

1. A transform is applied that combines nut position (local CS), harvester position and heading (world CS) to calculate nut position in World/GPS co-ordinates.
2. Nuts appearing in the same place but from different cameras/frames are filtered
3. Nuts are then assigned to trees in a reference map. The actual method for this depends on the end user's requirements - for example it might be simply the closest tree, or the nut may have to lie within a polyline marking the canopy boundary

5.4 Results

5.4.1 Test setup

In review, the relevant KPI's to be rigorously tested against performance are:

- Count Accuracy – the number of nuts counted.
- Position Accuracy – position of nuts in field.
- Accuracy at various speed

To test all these performance indicators in a single trial setup, the following experiment design has been devised.

In any given row, the outline of a checkerboard is drawn on the ground with paint. A series of 1m by 1m squares are measured out and the outline drawn. For initial testing, nuts are laid only on the horizontal boundary lines at 10cm intervals. In subsequent testing, all the nuts are removed from alternate squares and the nuts remaining in the other squares are manually counted.

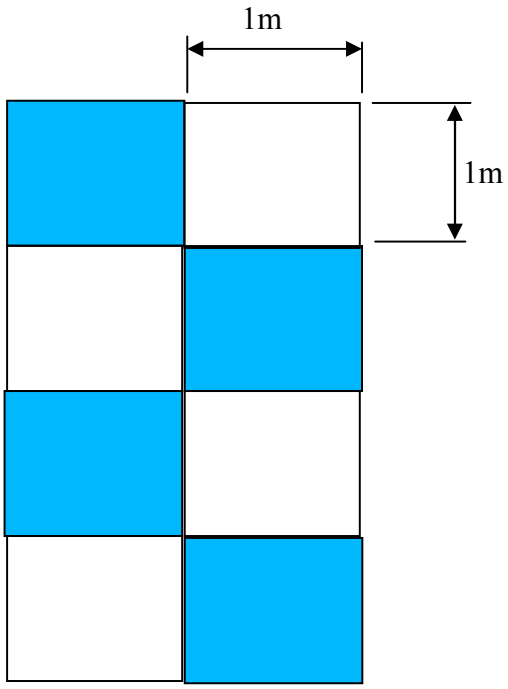


Figure 44. Trial Layout. White squares have nuts removed, dark squares have nuts counted.

The distance from the nearest tree to the checkerboard start is measured, and the checkerboard can be reproduced at various intervals down the row. Note that the accuracy tested is a higher degree than the macro level 1m squares, as it is the boundaries between the squares that will be used to identify location accuracy issues.

After the harvester passes over the checkerboard, the remaining nuts in each square are manually counted and noted. In this way we can test the accuracy of the vision system, the accuracy of the location system and the efficiency of the rollers in a single test. This test can be repeated at different speeds to compare accuracy figures and ensure that the speed KPI is met.

If the output of the machine vision system should be a coverage map of the entire row, the checkerboard should be clearly visible at the appropriate locations in the row. Virtual checkerboards can be overlaid onto the yield map, and the nuts identified in each region counted. The error can be determined directly between the vision system and the test layout.

5.4.2 Field trials

Field trials were undertaken over 2004, 2005 and 2006 harvest seasons.

Treecam testing

The accuracy of the Treecam has exceeded 90%. Given that there are high weeds and low hanging branches partially obscuring the tree trunks, this is considered a reasonable degree of accuracy.

Results from four trials conducted on 24th August 2006 are presented here as representative data (Table 2). The full data for these trials may be found on the software CD in the *Case Study Results\Macadamia Yield Map* folder.

Table 2 The accuracy of the Treecam process. The average accuracy is 92%

File/run (time)	Real trees	Identified	Missed
075023.*	34	30	4
081409.*	34	33	1
083737.*	34	31	3
085221.*	34	32	2

Figure 45 A satellite view of Hidden Valley Plantations

Figure 45 illustrates the extents of the Hidden Valley Plantation macadamia orchards. Testing has progressed in most of the numbered fields.

Figure 46, Figure 47, and Figure 48 display the end result for a single run of data (trial 075023). Commercial harvests will have hours of data rather than minutes, and will be displayed as a 'blob' map overlaid onto the orchard. The various levels of overlay colour directly relate to intensity of nuts collected over the area.

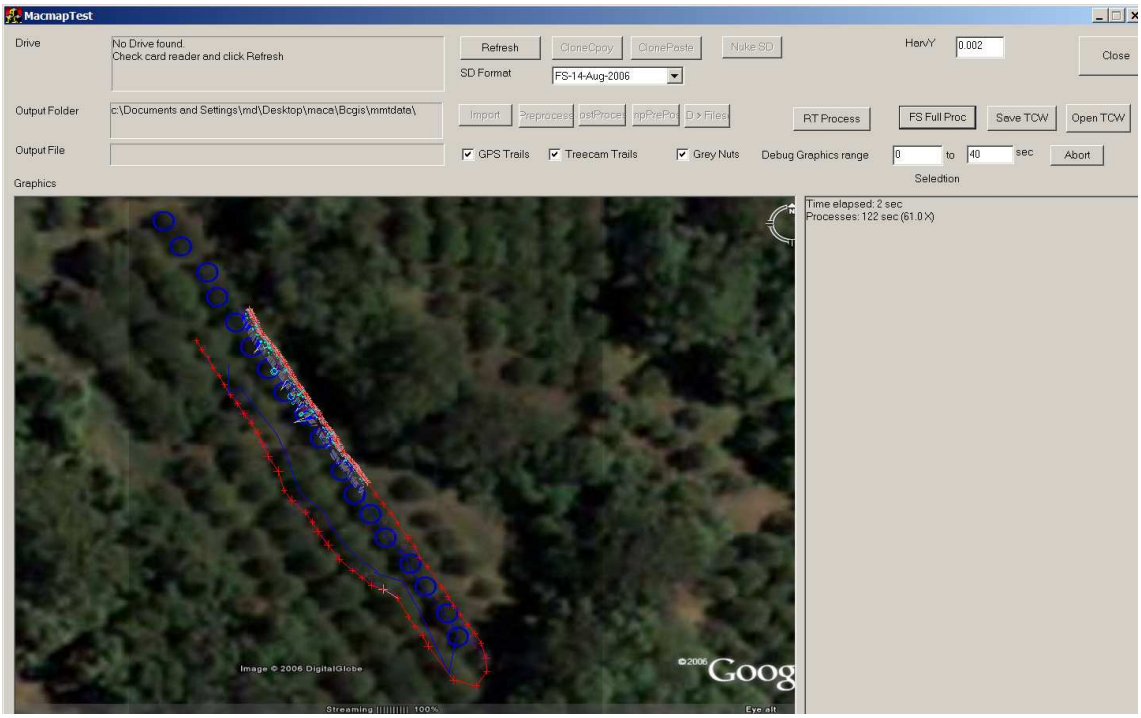


Figure 46 An example processed yield map overview

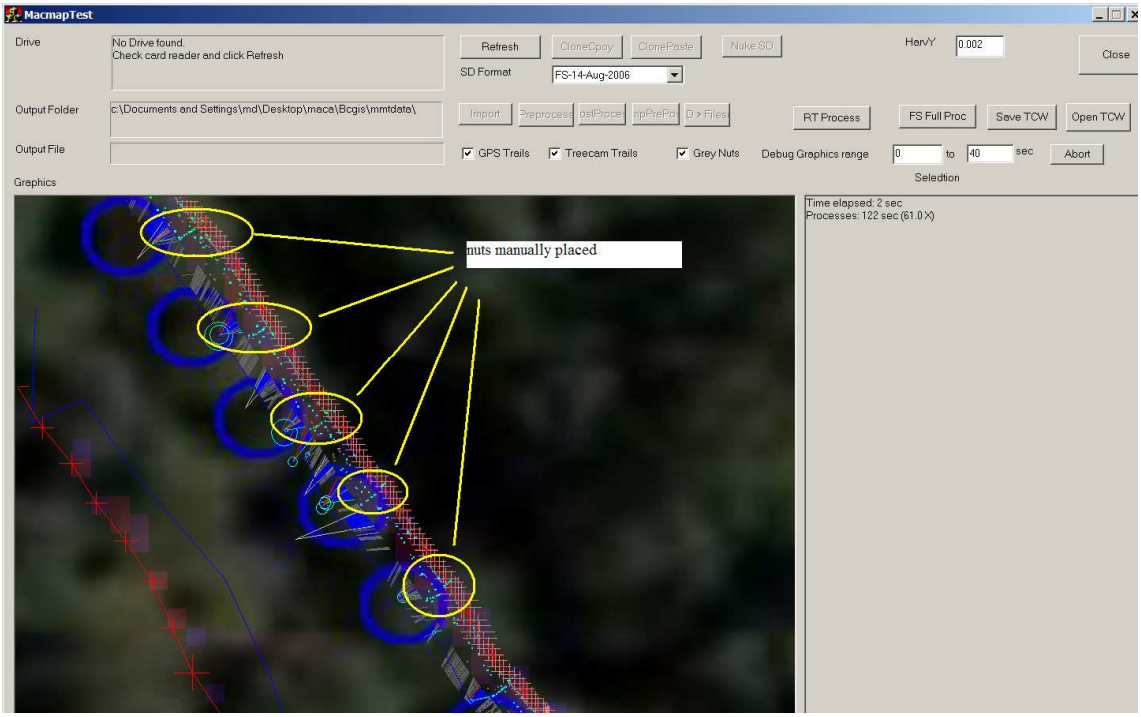


Figure 47 An example processed yield map displaying identified trees and nuts

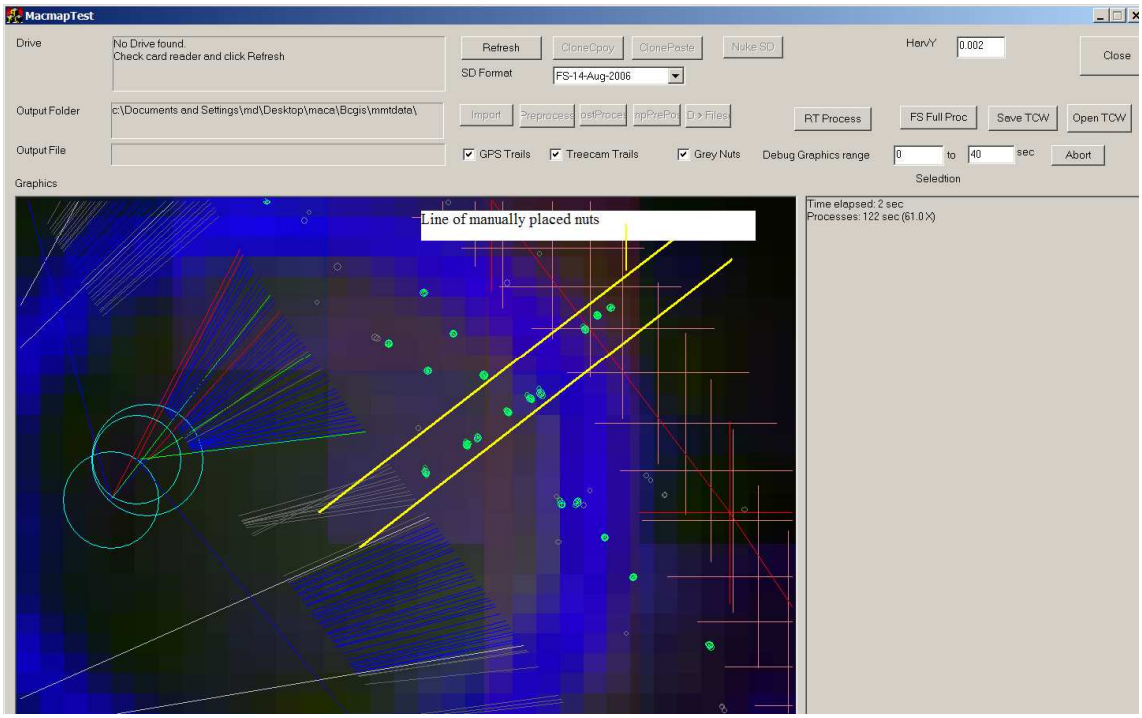


Figure 48 An example processed displaying identified nuts under one tree

In the same set of trials, nuts were laid in a line adjacent to each tree across the direction of travel. The nuts were spaced 10 cm apart. The area between the lines of nuts was cleared of visible nuts. The number of nuts were counted before and after the harvester passed over the area. As the results show, 85% of nuts were correctly identified in this sequence of tests. The missed nuts are due to occlusions from leaves and trash, insufficient lighting, or transverse nut movement. These results also indicate nuts found between the tree lines. These will be a combination of real nuts not cleared from the area, and spurious counts from trash. The results of this trial are displayed in Table 3.

.

Table 3 Nutcam results

run/tree	**075023 real count**	Identified	**081409 real count**	Identified	**083737 real count**	Identified	**085221 real count**	Identified
5	**11**	10	**10**	9	**10**	9	**10**	9
6	**14**	9	**12**	11	**12**	9	**12**	11
7	**14**	10	**13**	12	**11**	10	**11**	9
8	**13**	10	**15**	9	**11**	11	**12**	12
9	**14**	12	**14**	13	**14**	13	**11**	11
10	**12**	12	**12**	10	**14**	9	**7**	7

Total	**78**	63	**76**	64	**72**	61	**63**	59
Percent		80.77%		84.21%		84.72%		93.65%

5.4.3 KPI results

Current results are as follows:

1. Position Accuracy – position of nuts in field. **KPI exceeded.** The tractor in-field position is accurate to within 12mm, using DGPS and Treecam positioning.

2. Count Accuracy – Nuts counted. **KPI not yet accomplished.** Work is continuing for improving the accuracy. The correlation of nuts collected to nuts counted is high, however an exact count is preferred for research purposes. Further algorithm improvement using all the colour channels is expected to increase the accuracy of correct nut detection.

3. Speed. **KPI met.** The position trials have been performed at various speeds, up to a maximum of 10km/hr.

4. Cost. **KPI met**. The estimates given in (Hardner, 2004) have been confirmed and are detailed in Table 4 below for 1600 trial trees at 4 sites in 2 regions over 4 harvests:

Table 4 Estimates for the cost of the system

Cost Breakdown	$/hr	Hours/tree/harvest	$/tree/harvest
Harvest labour – machinery operator	30	0.01	$0.30
Machinery operating costs –fuel, disposables	10	0.01	$0.05
Transport labour – Driver/operator	30	0.02	$0.62
Transport operating costs	10	0.02	$0.21

– fuel, disposables			
Annual maintenance	100	0.0025	$0.25
Harvester depreciation	_	_	$0.32
Vision system depreciation	_	_	$0.32
Nut mass assessment	20	0.138	$2.75
Total			$4.82

5. Size of Orchard. **KPI met**. The trials have been successfully conducted on various fields at Hidden Valley Plantation, ranging from 5m x 2m to 8m x 5m.

6. Size of Machine. **KPI met**. This prototype development has proven that the technology can be fitted to any machine with a front auger, including smaller units. Further work is also being undertaken to attempt to install a vision system on a machine with a back mounted auger, as these system are much more widespread in production harvesting.

5.5 Conclusions

This project has proved the concept and has had considerable hours of testing. Many refinements have been made to the original ideas, including a method of determining in-field position to an unprecedented accuracy.

A commercial unit is being built for the 2007 harvest season. This unit will be used to compile yield reports for all state government varietal trial plots.

The techniques created here for in-field location may be generalised to any orchard crop where the collection is undertaken by a machine travelling along the rows. This will prove valuable to many industries that cannot take full advantage of the latest GPS technology due to signal interference from overhead branches.

The ability to identify and count items on a mobile harvester can also be applied to diverse applications, ranging from sugar billets to lettuce. Some pre-grading can also be performed by a harvester system, reducing the workload and variability of produce at the packing shed.

5.5.1 Model summary

Below is a summary of this case study in the terms identified in Chapter 4.

Image Acquisition.

Treecam - Use ROCs for 30fps image acquisition in YUV space.

Nutcam - Use ROCs for 60fps image acquisition in RGB space.

Pre Process.

Nutcam – binarise the image based on blue/non blue target pixels.

Treecam – none.

Analysis.

Treecam – Identify dark vertical stripes (tree trunks).

Nutcam – Identify and track circles in binarised image.

Post Process.

Treecam – Report microseconds, x position and width of tree trunk. Store accumulated data to SD card.

Nutcam – Report microseconds, x/y position and diameter of all identified nuts. Send data downstream via serial communication.

Offline Processing.

The data accumulated from all sensors are converted to a single time sorted datastream and mapped graphically as a yield map.

6 Citrus texture application

6.1 Introduction

A machine vision system has been created as a non-destructive method to obtain a reliable measure of surface texture from sample fruit. The citrus industry uses surface texture as an indicator of quality, together with colour and other attributes that are harder to measure such as sweetness and juice content. While machine vision has long been used in grading citrus fruit, it is currently limited to defect and colour classifications [40, 75].

The skin texture of citrus fruit is a combination of three different types of spatial variation. Sub-millimetre wrinkles (scent glands) are found over the entire skin and are irregular, but have relatively constant coverage. Small dimples 1mm-5mm in depth are randomly spaced around the fruit. It is the depth and quantity of these which have the greatest impact on the skin texture grade. The third type of variation is deformation from the normal spheroid shape. These lumps or flat spots can be caused by rough handling, or may be due to variety. These three parameters are generally referred to as roughness, waviness and form in production machining. Figure 49 displays an example of each of these parameters.

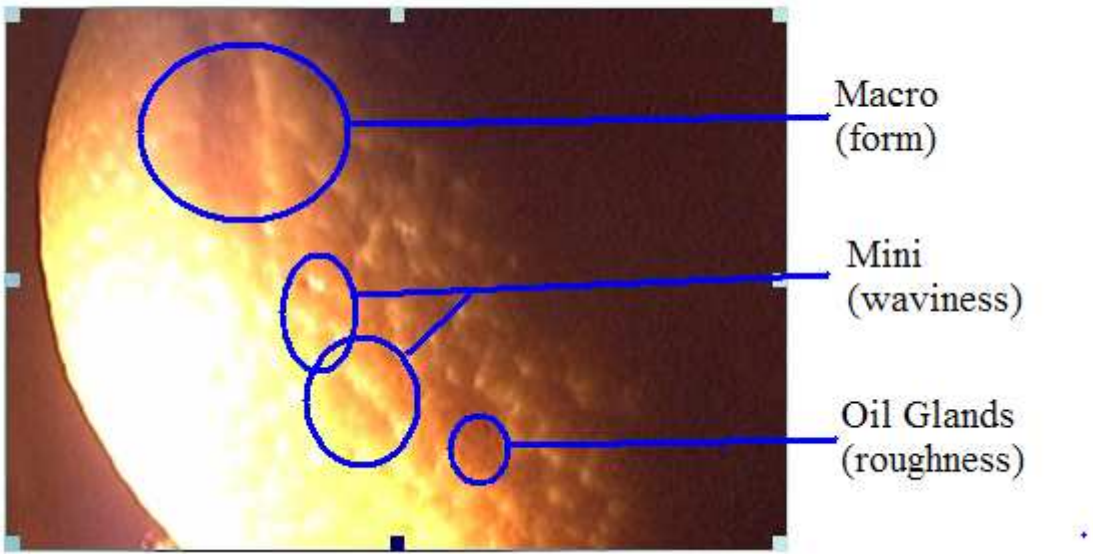


Figure 49 Example citrus fruit image

The best measure of the texture of a fruit is given by the arithmetical mean deviation of the profile R_a by Leach[76]. Where R_a is defined as

$$R_a = \frac{1}{N}\sum_{i=1}^{n}\left|Z_i - \overline{Z}\right| \qquad (6.1)$$

Where Z is the distance measured from the centre of the object.

This can be measured directly from a fruit using an expensive stylus instrument where a needle, similar to that of a record player, touches the skin of the fruit as it revolves. The changes in position are amplified and recorded. A serious problem is that this method only provides a single sample from one 'latitude' around the fruit, which may or may not be representative of the entire surface.

Using a simple plan image of the fruit will not provide adequate results as the classical texture measurements will not account for dimple depth, merely contrast changes in skin colour.

Given the random spacing of the dimples on the fruit and the reflective quality of freshly picked fruit, we can use specular reflection to calculate an effective measure of texture. If the fruit to be measured were perfectly smooth then, as in Figure 50 below, the illuminated area would appear as slowly varying intensity, and the terminator as an even edge.

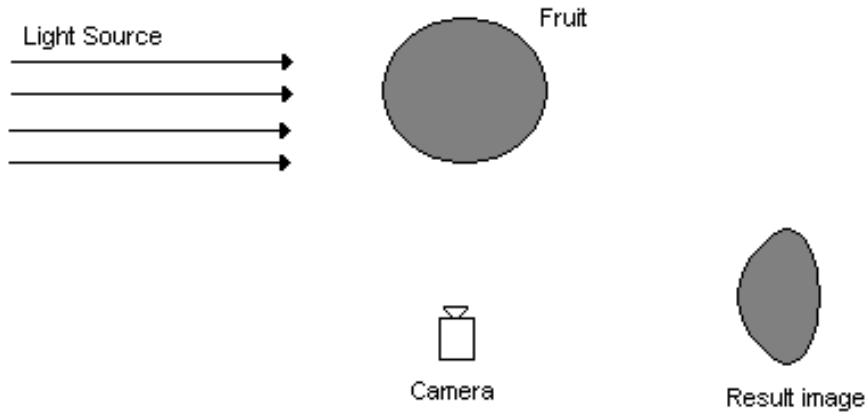


Figure 50 Light terminator on smooth fruit showing the resulting image

6.2 Key performance indicators

In a practical application, the most important performance indicators are classification success. This is harder to measure when the current methodology is subjective. In this project, the requirements were:

- Accuracy. The classifications produced by the system should correlate with manual classification $r^2 > 0.90$

- Repeatability. An important requirement is for similar fruit to yield similar results. This can be measured directly by processing the same fruit multiple times. The resulting measurement should be within 3-sigma at 10% of the midpoint of the scale.

- Cost. The system should be affordable as a research tool. This system would still require refinement to be implemented as a production grading tool.

6.3 Materials and methods

A prototype texture measurement system was designed using a Logitech webcam, a 12V, 50W halogen bulb and a 12V DC motor and gearbox combination (Figure 51). The cost of the unit was under $AU200, with the webcam the most expensive component at around $140.

Figure 51 Prototype citrus texture measurement system.

The web camera captured images at 30 frames per second as the fruit was revolved. The light source was positioned at 90 degrees to the camera to provide a clear terminator line. The fruit was turned at 4 seconds per revolution, which provides 120 frames per sample fruit revolution.

The unit is placed under a light hood to remove ambient light effects.

For a textured fruit (Figure 52), the light intensity reaching each pixel of the camera is a function of the angle of the surface at the observed point. Measuring the intensity of light reaching the camera then provided an indication of the angle between this normal and the axis of the camera.

Figure 52 Original image of a citrus fruit

Taken in horizontal rows, the intensity graph produces a map of the specular reflection. The intensity levels thus measured are superimposed on those due to any macro deformations in the shape of the fruit. To remove the effect of those deformations, the deviation from the moving mean of the intensity graph is calculated.

Relating this back to equation 6.1, the Row Texture (rt) for a row with n pixels is determined as:

$$rt = \frac{1}{n}\sum_{i=1}^{n} I_i - \bar{I}_i \qquad (6.2)$$

For the sample image, we can process *m* lines to determine the frame texture (ft) thus:

$$ft = \frac{1}{m}\sum_{i=1}^{m}(rt_i) \tag{6.3}$$

This measure is then summed around each rotation of the fruit and again averaged to provide a single number for each fruit (R_{visual}). This texture measure is related directly to the physical texture of the fruit by a constant.

$$R_a = kR_{visual} \tag{6.4}$$

Figure 53 shows a sample frame for which the intensity graphs for each colour channel for the 40 central rows are overlaid onto the grey scale representation of the fruit. The blue channel exhibits the widest response and these intensity levels are processed to determine the texture. The blue graph at the top of the figure represents individual deviations from the mean.

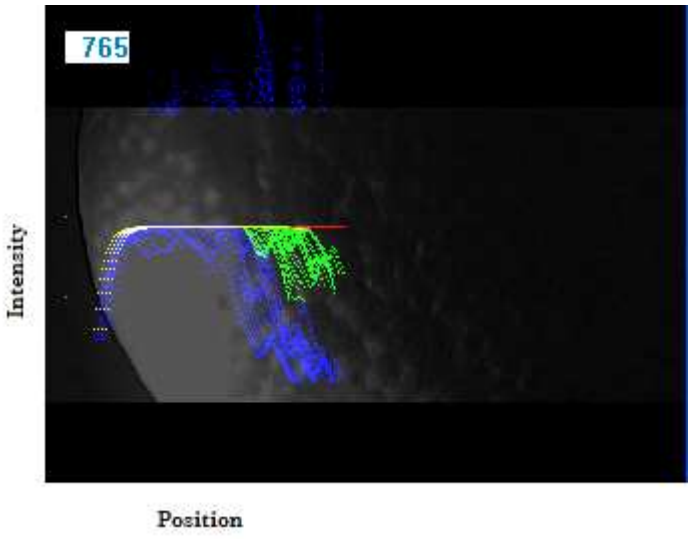


Figure 53 An example processed image of a citrus fruit

From Figure 53, it is clear that several well-defined brighter areas exist in this portion of the fruit. The size and location of these areas are averaged over each frame and are accumulated over the series of frames to provide a single number representative of the texture of the sample fruit (R_{visual}).

6.4 Results

After measuring the fruit continuously, R_{visual} for the fruit is seen to settle to a final value, as shown in Figure 54.

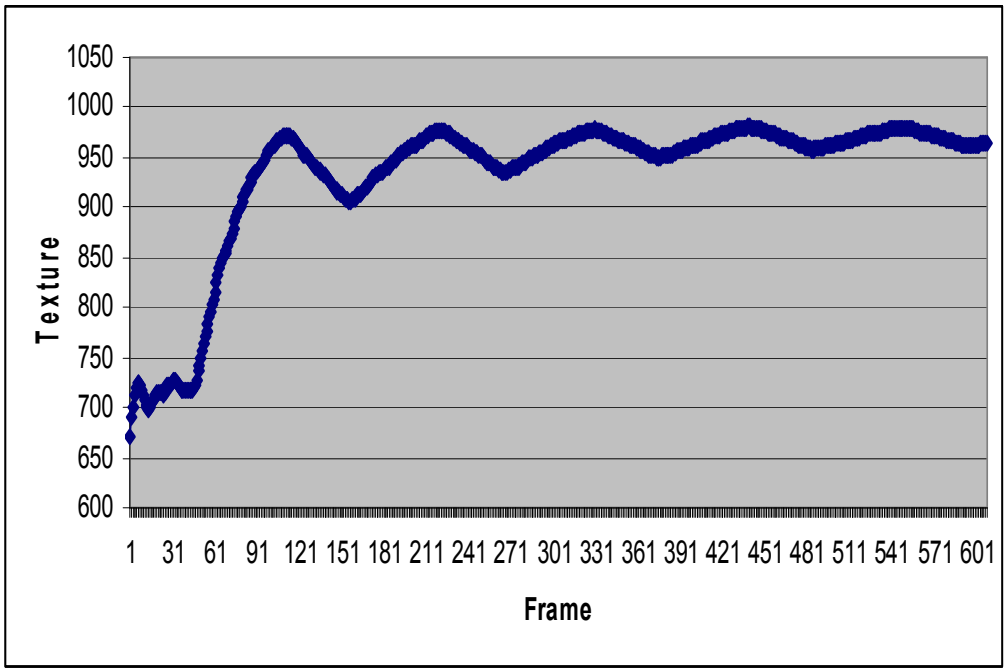


Figure 54 Temporal measure of the texture of a citrus fruit

This demonstrates the fact that each sample fruit must revolve for around 6 seconds, or 180 frames, to determine a representative steady state value.

Repeatability is a major factor for the success of this system. Every fruit should have a repeatable measure of surface texture to within minimum limits.

After 20 individual tests of 12 different citrus fruit with random placement on the system, the following results were produced (Table 5). A subjective manual grading exercise was also undertaken by 6 consumers with the instructions to order the fruit from smoothest to roughest. The results were tabulated by simple majority votes classification (Table 6).

Table 5 Results for the machine vision grading of three types of citrus fruit

Fruit	Mean	Std Dev	Fruit Type
1	16835	719	Navel Orange
2	35374	789	
3	22260	787	
4	31269	823	
5	35362	714	Valencia Orange
6	33386	731	
7	12088	441	
8	15356	608	
9	14496	500	Murcott Mandarin
10	12743	427	
11	19523	522	
12	28410	669	

Table 6 Results from the manual grading of three types of citrus fruit

Manual Grading (smoothest-roughest)	Fruit Type
1-3-4-2	Navel Orange
7-8-6-5	Valencia Orange
10-9-11-12	Murcott Mandarin

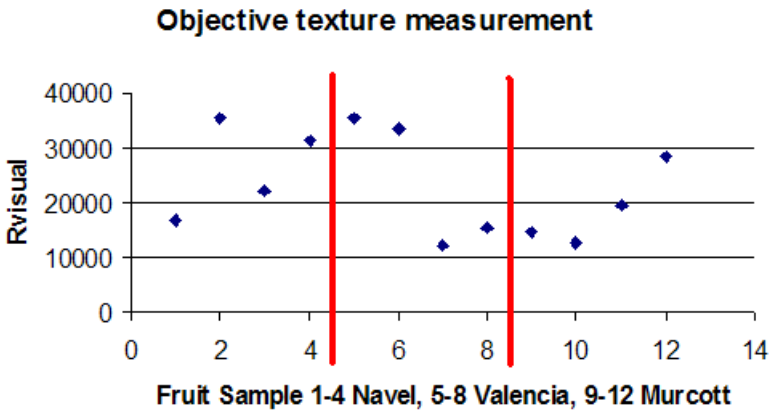


Figure 55 Graphical representation of trial results

The results (Figure 55) indicate a strong correlation between the machine vision results and the subjective visual measurements. In every case, the subjective measurements are upheld by the R_{visual} measurement produced by the system.

The midpoint of the arbitrary measurement scale is 25000, so the repeatability KPI requires all 3-sigma values (3 standard deviations) to be less than 2500. Table 7 demonstrates the 3-sigma results.

Table 7 3-sigma results for machine vision grading of three types of citrus fruit

Fruit	Std Dev	3sigma
1	719	2157
2	789	2367
3	787	2361
4	823	2469
5	714	2142
6	731	2193
7	441	1323

8	608	1824
9	500	1500
10	427	1281
11	522	1566
12	669	2007

The system has been used in research work at the Department of Primary Industries, Bundaberg, Queensland. Malcolm Smith has used the device for determining texture differences between citrus fruit rootstocks.

'I measured 15 individual fruit from each of these 47 trees and they were measured in a random order. The 15 fruit sample was collected from each tree at random from all around the tree at about chest height; this was done without knowledge of the rootstock treatment (a code number is used throughout the sample harvesting and assessment process to prevent any bias, and it is only after the assessment is complete that the code number is matched to the rootstock treatment).' (M Smith, pers. Comms) [77].

Malcolm Smith reports the results: 'The machine was able to detect highly significant differences in skin texture between rootstocks. The differences correspond with comments in my notes regarding the visual texture of various samples, and I am in no doubt that the machine is detecting real differences in skin texture. What is particularly exciting about the results is that the LSD (0.05) comes out at around 283, which is far lower than I would have expected. This gives me confidence that the machine (and our sampling procedures) now enables us to detect quite small differences between rootstocks, even when the variability WITHIN the samples is quite high. In short, if the rootstock is affecting skin texture then we will detect it.'

6.4.1 KPI results

1. Accuracy. KPI exceeded. Small scale experiments show high correlation (r^2>0.9). Large scale experiments are infeasible due to cost constraints.

2. Repeatability. KPI exceeded. Results from 20 repeats of 12 fruit show 3-sigma values within 10% mid-scale value.

3. Cost. KPI Exceeded. At under $AU200 per unit, this system provides a valuable tool for researchers utilizing surface texture as a measurement parameter.

6.5 Conclusions

This project has demonstrated the usefulness of machine vision techniques to areas where standard measurement systems are too expensive or infeasible due to time constraints.

A full prototype system has been developed and deployed to a research location, where it is in constant use.

Results show a high degree of repeatability and accuracy, providing a cost effective tool for measurement of citrus skin texture.

This technique may be applied in any situation where the texture comprises of components with visible difference at the terminator of directed light. Further experimentation in this field will be directed to correlation of single and multiple images to final accurate results with a view to decreasing the measurement time requirements.

Further experimentation is also required to determine *k*, the scaling figure between R_{visual} and the physical measurement R_a.

6.4.2 Model summary

Below is a summary of this case study in the terms identified in Chapter 4.

Image Acquisition.

The image acquisition was performed with a Logitech QuickCam Pro 4000 webcam. A dedicated PC application was developed and deployed for this project.

Pre Process.

Detect citrus edge and reduce to separate colour channels.

Analysis

Intensity averaging over multiple rows and frames to produce R_{visual}.

Post Process

Summation and recording of objective texture measurement for each fruit.

Offline Processing

none

7 Animal identification application

7.1 Introduction

In an open agricultural environment such as the Australian rangelands, resource management is a key issue. There is an increasing demand to ensure that the available feed and water are utilised in the most efficient and effective manner. Feral animals and other wildlife exploit resources provided for livestock, which means that more must be supplied to compensate, or the livestock will get less access to these resources.

The Great Artesian Basin Strategic Management Plan is a joint Australian Federal/State government and landowner initiative, which aims to replace open flowing bore drains with pipe and trough systems [8]. The widespread implementation of this scheme provides an opportunity to control invasive species at an ecosystem level. Figure 56 illustrates the extent of the land covering the Great Artesian Basin [Source: Queensland Environmental Protection Agency].

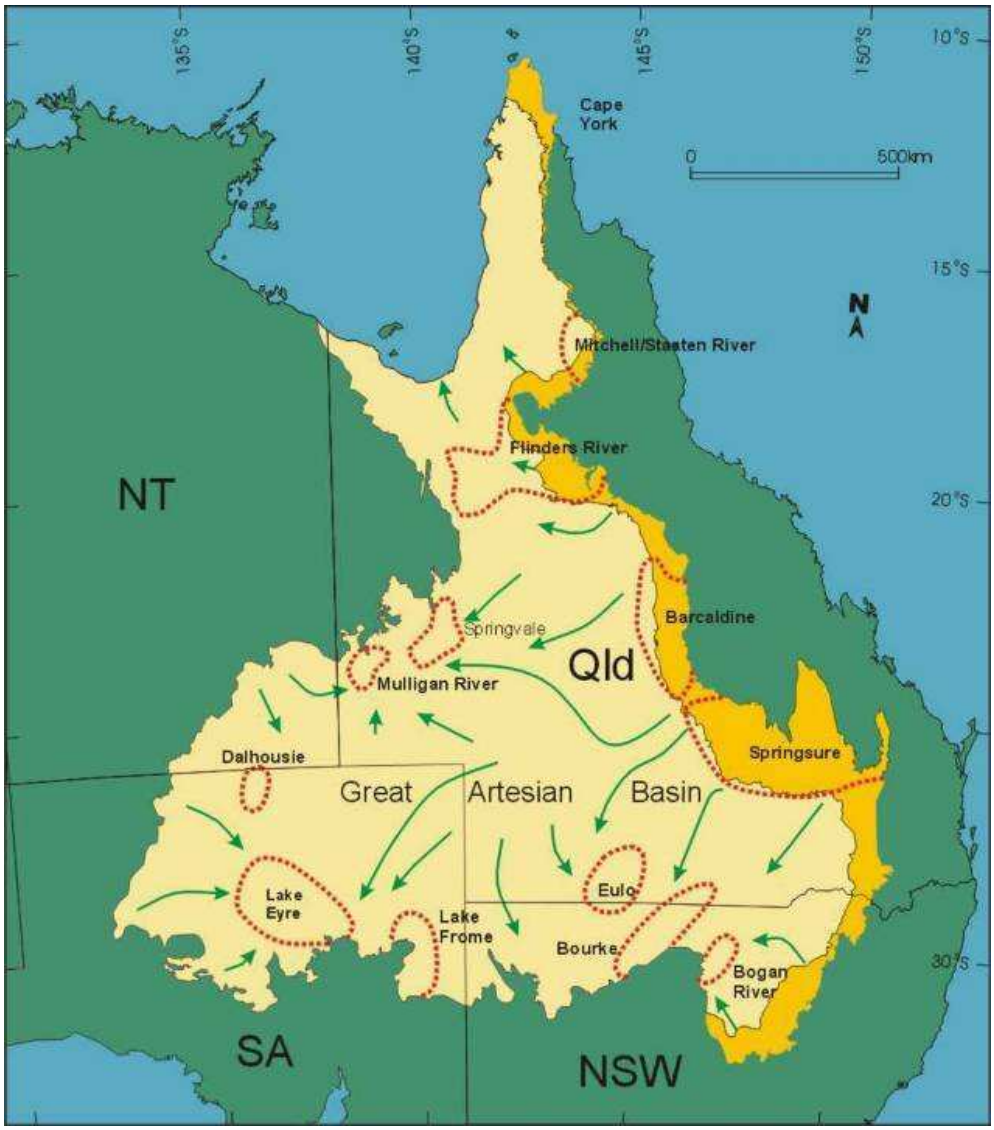


Figure 56 Map illustrating the extent of the Great Artesian Basin

A machine vision solution was proposed by Neal Finch, School of Animal Studies, from the University of Queensland. The project proposed was to erect a fence around the new pipe and trough water points, and use automated gates in the fence to control access by animals to that water. The only solution that could be guaranteed to work with both domestic and wild animals was a computer controlled system that could detect not only the presence of an animal, but also the species, and also control the automated gates.

There is a national trend towards tagging all production animals with Radio Frequency Identification (RFID) devices (Figure 57). These devices allow the opportunity for greater control over the animal and for information to be recorded for

that animal. This creates the possibility that a control system could simply detect the presence of an animal wearing a tag and allow or deny access to that animal accordingly. There are two problems with this approach. The first problem is that RFID readers are not 100% reliable. There will be occasions where the tag is missed or misread due to the relative positions of the tag and the antenna. In this scenario, the animal would eventually get to water if it continued to try until the tag was read correctly.

Figure 57 Standard RFID cattle ear tag device (25mm diameter) in comparison to a 20c piece

The more serious issue is in loss rates of RFID tags. The most common tags are incorporated into a device that is attached to the animal's ear. These devices are estimated to have a loss rate (in cattle) of between 1% per year by a technical consultant [78] to 10% per year by a Cattle Breeder [79]. For animals with no tag, there would be no access to water. For valuable animals this is unacceptable, and indeed for any system there should be a bias toward errors allowing access rather than denying access.

An important feature of a machine vision solution is that there is no physical contact, reducing the need for system maintenance and also stress on the animals being

monitored. Computing capabilities have developed to the point that systems processing real-time video data are now feasible.

Previous work using machine vision has been done in detecting lameness and behaviour prediction in livestock [80]. This research used statistical modelling and complex predictive algorithms to calculate the most likely position and orientation of the object in the next time step. This method used prior knowledge of the object and features of the object in the determination.

The main species to be determined in this animal identification project were: sheep, goat, cattle, horse, pig, emu, and kangaroo. The algorithms, however, are generic and extensible to any objects passing a set camera position in random time. This is a completely different proposition to a conveyor belt producing objects at a nominally set time, within minimum and maximum limits.

The research has allowed the development of a prototype low cost system incorporating image capture hardware and processing software for use in agricultural applications in remote locations. The intelligent machine vision systems developed allow control of a gate remotely after identifying the species passing in front of the camera. Software developed includes algorithms for shape and movement classification which can be compared against a database of animal classes for animal categorisation and subsequent real time control of access.

7.2 Key performance indicators

The end result of this project was a complete unit to control access to a watering point using a camera (with appropriate hardware and software to identify animals or read a RFID ear tag) and an automated gate at the end of the laneway. This unit will have the capability to record RFID ear tags of appropriately tagged animals that enter the watering point and use that capability integrated into the units' software to control the gate. The system also has a 'fail safe' default control that opens the gate (access to water) if the unit fails.

The following requirements have been identified as necessary for the successful completion of this project:

Accuracy

The animal recognition software should perform in excess of 90% accuracy to correctly identify an animal from the species list under perfect conditions. Perfect conditions are defined as no dust or rain, and a single animal travelling at under 2.2m/sec.

Under normal conditions the machine vision software will exceed 85% accuracy to correctly identify an animal. Normal conditions are defined as any field conditions excluding conditions where visibility is less than 10 metres (eg. dust storms and heavy rain).

Robustness

The ability of a complete unit to run for a continuous period of 12 months under field conditions with a less than 5% hardware failure rate. Any failure should cause the system to revert to 'safe' mode, allowing complete access to water for all animals.

Cost

The unit price of the system should be under $AU5,000. This will allow access to the technology to all sized production landholdings. This includes the automated gate and laneway/race, power supplies and camera, but not the fencing around the water point, or the water point infrastructure.

Power Consumption

The unit should be capable of stand alone processing for indefinite periods. This will be accomplished with matched batteries and solar panels.

7.3.1 Hardware

The initial prototype of this system was constructed at the University of Queensland Gatton campus (Figure 58). A series of yards was constructed connected by a short laneway. One area contained a water trough; the other area was for feeding. A blue tarpaulin was used as a backdrop for the machine vision component. The automated gate was an off-the-shelf 12V DC gate motor.

Figure 58 Prototype of the animal identification system.

After several trials and changes of configuration, the system now has a set format. A standard section of portable laneway has been designed in conjunction with RPM Rural Products. This section of multi species fencing has an identification area built in (Figure 59). The raised area in the middle of the blue wall of the laneway is the RFID tag reader antenna. This system has been patented under Australian patent 2004218711 “Control of Animals using Electronic Recognition Technology”.

Figure 59 Portable laneway incorporated into the fence surrounding a water point.

The PC platform was trialled initially to support the animal identification system. However there are limitations on the use of PC technology. A laptop consumes up to 3A in processing mode, slightly less with the computer screen powered down. This equates to 72Ampere-hours per day, a substantial amount of power for a remote system.

Several units were assembled for testing the proof of concept (Figure 60). These units consisted of a HP nx6120 notebook with 12V power adapter, two deep cycle 120Ah truck batteries connected in parallel, and a 60W solar panel (Figure 61).

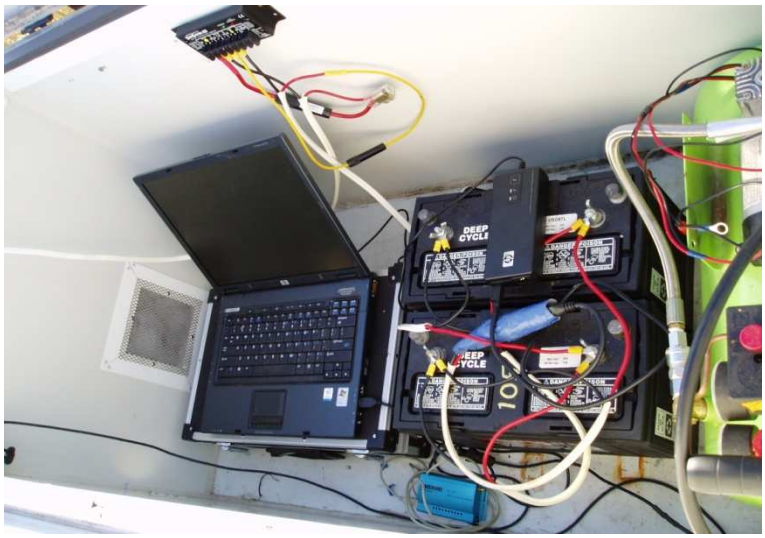


Figure 60 Laptop platform showing HP nx6120 notebook plus 2 deep cycle batteries and compressor to operate the gate.

Figure 61 60W solar panels used to supply the laptop and air compressor

The units were placed in custom built insulated boxes, (Figure 62). This helped reduce heat build up from solar radiation and provided protection from the elements. Internal heat build up was dissipated by way of a rotary vane, powered if the internal temperature reached 50° C.

Figure 62 Insulated boxes that were used to house the prototype animal identification system

A 50mm hole was cut in the side, and a camera mount installed for the web cameras to be fitted. Another hole directly beneath was fitted with an IR panel for night

lighting of the laneway. The IR panel contained a light sensitive switch to ensure operation only during periods of darkness.

Using this methodology, software could be written and trialled with relative ease, with the resources of a full PC available. Motion detection algorithms were written to record images of the animals proceeding along the laneway past the camera. These images were then used to generate a library for algorithm development.

The project then progressed to using the ROC platform for video capture and recording. Mounted in tough Pelican cases (model 1040), the system developed at NCEA was customised specifically for this project (Figure 63). The units are 190mm x 130mm x 54mm and attached to a swivel mount for fast and easy deployment.

Figure 63 A ROC prototype enclosed in a black Pelican case

7.3.2 RFID hardware

There are only two substantially different types of RFID hardware for reading National Livestock Identification Scheme (NLIS) tags on the Australian market. Fixed units for reading RFID tags are usually permanently mounted on raceways or crushes and log the identity of the animal each time it passes the reader. Portable units are handheld and can be used to scan animals with RFID tags in the field.

This project acquired an EDiT-ID race reader for development and testing purposes. The data transmitted by most readers is in standard RS232 ASCII format text, consisting only of the unique identifier read from the tag and carriage return. The antenna was mounted on the back side of the laneway and painted blue (Figure 59). As animals pass the antenna, the tags are read and the tag ID recorded.

7.3.3 Enclosure design.

Initial laneway structure was not considered important – merely a fenced off lane wide enough for all animals to pass through. However, there are many facets to this part of the project that have required research. One major finding is that the laneway must be orientated such that animals walking down the laneway can directly see the water directly. Animals trained to respect fences (generally production animals such as cattle and sheep) will not willingly enter a laneway unless they have been trained to it, or they have direct line of sight to the water.

There are two general uses for this system; exclusion and drafting (Figure 64 and Figure 65). In most situations, the main use of this technology will be in excluding unwanted species from accessing the watering point. However, there may be some situations where a species should be drafted into a holding pen. An example of this is collecting/trapping feral goats for harvesting.

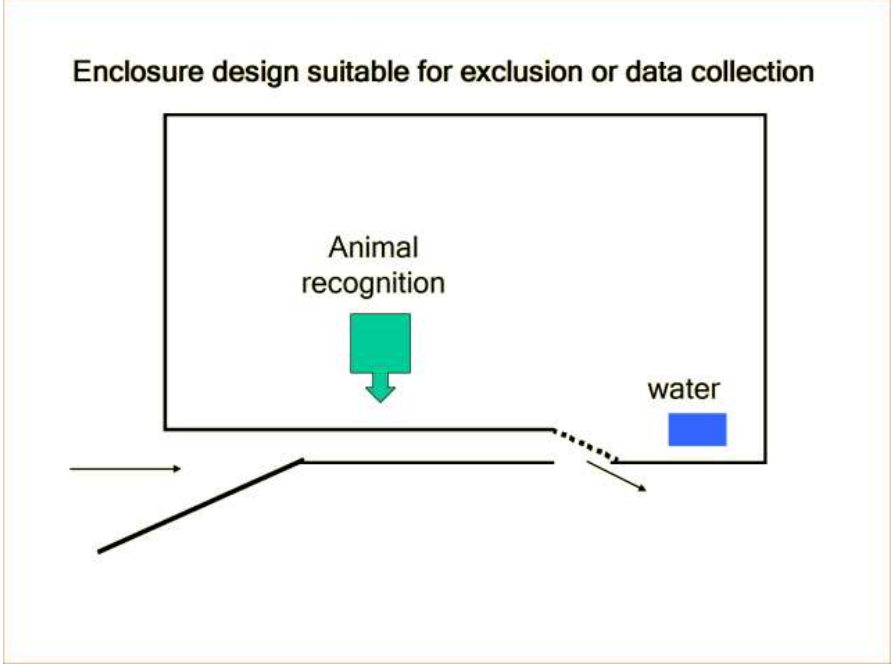


Figure 64 . Basic design of an enclosure for exclusion of animals from water

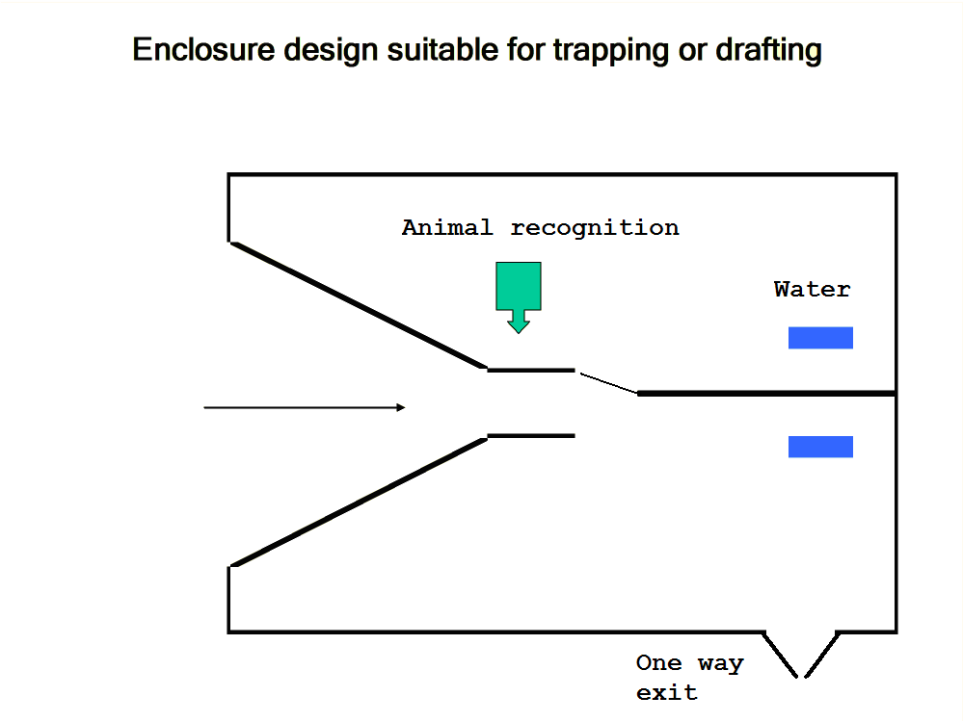


Figure 65 Basic design of an enclosure for either allowing animal access to water and then release or allowing animal access to water and trapping

7.3.4 Laneway design.

The layout of the laneway also plays an important role in the success of the complete system. If the laneway is too wide, animals (especially sheep and goats) pass through with overlapping parts of their bodies, obscuring the outlines. A novel set of animal separators was designed and implemented by Neal Finch, School of Animal Studies, University of Queensland (Figure 66).

Figure 66 Laneway designed for animals to traverse in single file with 'fingers' at each end

The 'fingers' on the entry and exit to the laneway are mounted on springs and are entirely a visual deterrent to animals entering side by side. This system was initially successful, but after a short period of acclimatisation, some animals, especially goats, ignored the fingers and pushed their way through in multiple groups.

A V-Race was provided by Rural Pacific Management Ltd (RPM) who were commercial partners in the project. The base of the V-race is 400mm wide, with the back plane angled to 800mm at the top (see Figure 59). This configuration only allows sheep and goats to physically pass in single file, yet also allows Brahman bulls weighing in excess of one tonne to traverse the laneway. Tests indicate over 90% of animals pass single file through this system.

7.3.5 Algorithms

As per the generic model discussed in Chapter 4, there are several discrete steps in the process.

7.3.5.1 Image input.

A calibration technique has been used to ensure that the camera field of view was the same in all experiments. The camera was positioned 3m from the centre of the lane,

at a height of 1m. A target cross painted on a metal plate is placed at 1m high in the centre of the laneway and the camera is focused upon that cross. This setup routine ensured size invariance to initial stages of the project. It is anticipated that this routine may be discarded in later phases. The image is captured in RGB colour space.

7.3.5.2 Pre-process

Depending on the background of the image, there are many methods of identifying 'interesting' objects. These methods are discussed below.

Colour ratios.

The initial method was to use a blue background by covering the inside of the laneway, furthest from the camera with a sheet of marine ply painted blue (Figure 67).

Figure 67 Example image illustrating blue background, with a goat in the laneway

In this manner, the background can be separated from the foreground by a simple threshold on the blue chromaticity information present in the image.

$$b_{chroma} = \frac{B}{R + G + B} \qquad (7.1)$$

A threshold was applied to convert the resultant image into a binary image. In most algorithms, binary target/non target details were required for analysis (Figure 68).

Figure 68 The resultant binary image of a goat in the laneway.

Note that this process also removes shadows present in the image, as they have the same surface reflectance as the background, simply darker.

This method is highly accurate and effective in daylight, but it is not at night. When lit with IR lights, a standard image sensor (with IR block filter removed) will respond most vigorously in the red colour channel and to a lesser extent the other two channels. Any automatic white balance algorithms will then initiate colour balancing on the image. This means that every pixel in the image will have a similar colour ratio.

Under IR lights, however, some objects are more reflective than others. For example, TiO_2, (titanium dioxide), is particularly reflective. This compound is added to most light coloured paint as a matter of course and thus the blue background (painted) is highly reflective. This provides a discrimination method for night viewing as the animal reflectance will generally be less than the painted background. Again applying a simple threshold will provide a distinct target function.

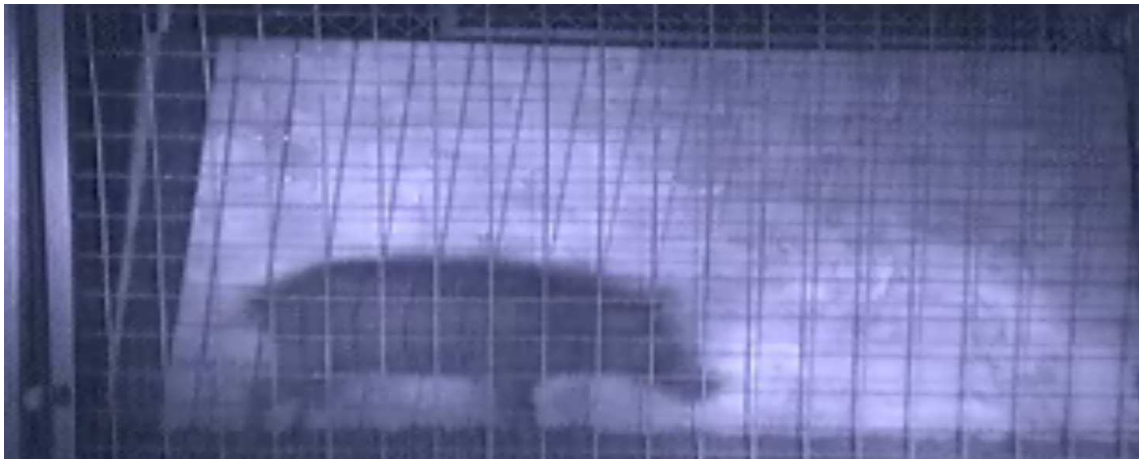


Figure 69 Example frame, with a blue background to the laneway, showing a feral pig in the laneway at night

Figure 70 Example frame resultant image of a feral pig in the laneway at night

Note that even though there is noise in the original image (Figure 69), caused by areas of inconsistent IR lighting, there is enough information to determine the shape of the animal, and thus perform the template match for species identification (Figure 70).

Motion detection.

Another method of detecting objects is motion detection. The method used most commonly is reference image subtraction. This is where a standard 'reference' image is subtracted from every frame. The resulting image is substantially non-zero only where there is a major difference.

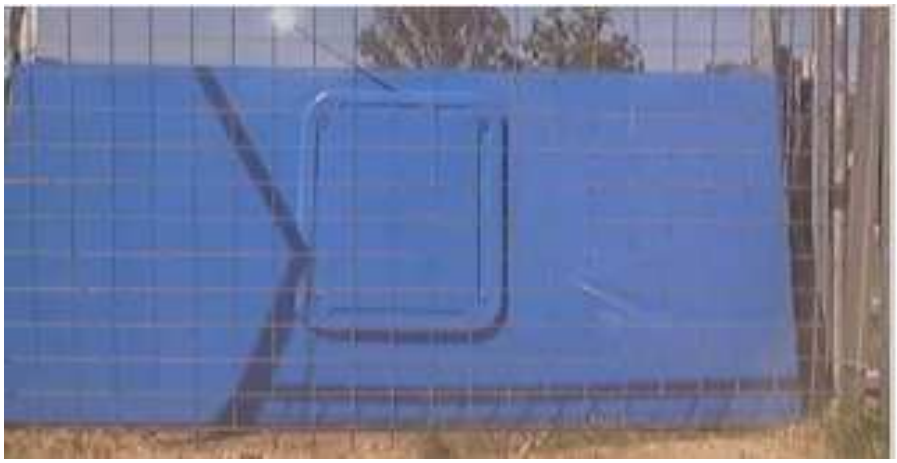


Figure 71 Initial reference image of the blue background

Figure 72 Current image of a goat in the laneway

Figure 73 Resultant image of a goat in the laneway after subtraction of the reference image

While this method works just as well at night or day, there are some substantial problems. Shadows thrown by objects are detected as part of the object (as seen in

Figure 73). More drastically, any changes in ambient lighting from the reference image will add an offset to all pixels. This can however be mostly removed by judicious use of mean shifting. Another problem is the selection and updating of the reference image. If for example an image with an object in it was taken as the reference image, every image after the object moved would show the negative of the object as a target area.

Another method is successive subtraction, where each image is subtracted from the next. This removes noise generated by large image changes, however only shows the moving edges.

Figure 74 Motion image 1 of a goat moving down the laneway with a blue background

Figure 75 Motion image 2 of a goat moving down the laneway with a blue background

Figure 76 Resultant binary image from subtracting image 1 from image 2 and threshholding

An advantage of this method is that it offers 'front' and 'back' edge detection for free, however flat areas along the top of the object are not detected, as the horizontal motion does not have contrast changes. Also, if the animal stops moving, it effectively disappears from this algorithm. Another problem is that moving shadows are detected as edges.

Multiple variable temporal reference images

A method devised during the course of this project promises excellent results. The basis of this technique is a set of reference images with differing levels of temporal adjustment. Both temporal mean filtering [81] and temporal median filtering **[82]** can be used. The application of mean filtering is described below.

The general algorithm for mean filtering is described by the function $\mathbf{MeanFilter}(T_i)$

$$MeanPI_{x,y} = MeanPI_{x,y} + (MeanPI_{x,y} - CurrentPI_{x,y})/T_i \qquad (7.2)$$

Where:

$MeanPI_{x,y}$ is the intensity of the accumulated mean frame pixel;

$CurrentPI_{x,y}$ is the intensity of the current frame pixel; and

T_iT is the required level of filtering.

The process is applied sequentially to each pixel of the image.

For example, T_i=2 provides an average of the last 2 frames. T_i=100 provides a low pass filter effect on the image. This means that noise in a single frame will not affect the image at all, and an object that moves into the frame and stops will slowly become visible over multiple frames. When the object leaves, it will slowly fade from the image.

In this way it is possible to build a set of reference images with high pass and low pass attributes. Each incoming frame can then be subtracted from each of the set of references, giving a set of temporally significant images.

Consider a set of images with T = [10, 100]. It is obvious that we have both a low pass filter, and a relatively high pass filter. The results can be combined using logical operations, for example the following equation.

$$\text{ResultImage} = \text{maximum}\left(\text{MeanFilter}(T_1) + \text{MeanFilter}(T_2)\right) \quad (7.3)$$

This provides a combination of both filters to use as a reference image. The input image is then subtracted from this reference, and a threshold applied to produce the binary image for further processing.

One problem with this method is 'artifacting' – the production of areas above the threshold produced by bias towards the object as it moves across the image and is incorporated into the reference image set. The solution to this is to dynamically increase the time constant for target pixels. In this way, any pixels identified as 'targets' will add much less weight to the reference image than non-target pixels. If, however, the target pixel is a new fixture on the image stream, it will still eventually be combined into the reference image.

Prior knowledge

Given the specifics required for this project there are a number of 'known' items. These include:

It is known that target objects will be travelling across the frame.
The background may be either a single flat colour or a natural scene.
The field of view is known and will be lit by Near Infrared Radiation (NIR) lights at night. This removes most of the variation from the image processing task.

The temporal distance between image frames is known. The maximum speed animals will travel through the laneway was determined as 2m/sec. Given these two items, this allows the application of tracking techniques to follow the animal in successive frames, rather than detection and identification of an animal in every frame.

Temporal continuity of the video stream ensures that an animal cannot appear in mid-frame; it will enter the image from one side, and must also leave one side of the image. Consequently, any object that appears or disappears from the centre of the image must be noise.

7.3.5.3 Image processing

At this point in the process stage, the image is segmented into notional target and non-target areas.

Object Shape.

To determine the objects shape, the object needs to be found, and the edge traced. While edge detection algorithms could be used to detect some edges, they offer no directional information and also there is no guarantee of a closed shape (i.e. finishing where it starts). Directional information can be gained by using Sobel filters [83], however this requires in 12 pixel accesses per step. Testing pixels directly for target/non target values as described in the algorithm below results in an average of just over 2 pixel accesses per step.

This project has implemented an edge tracing routine that is based on standard chain directions, but accumulates into an s-psi graph. This ensures a closed shape is found. The s-psi chain is then normalised for length and matched against a library of templates for classification.

The following pseudo code details the algorithm.

For each Image received.

Determine threshold for target/non-target discrimination

For each previously found object, determine current position and validate existence and correct identification.

For each point on a grid

Check pixel = *target* and not in previously found object list

Step <up> until *non target* pixel found. This is now a point on the edge.

Call GetOutline to trace the entire shape of the object into s-psi chain

Normalise s-psi into unit size chain (256 links)

Check list of objects found last frame for match

Determine direction of travel

Apply a-priori information as probability modifiers

Correlate with classes

Decide on class.

Getoutline:

While not at start point

Change direction leftwards and check pixel until non target reached

Change direction rightwards and check pixel until target reached

Step this direction and increment distance travelled

Record distance and direction in s-psi chain

Object Matching.

Once the object has been traced, the result is a full s-psi chain. Some of the techniques derived in Chapter 3 can be applied to process this information. These techniques are discussed below:

Remove the legs

The trace of the legs provides limited information about the shape of the animal, and will serve only to corrupt the rest of the s-psi chain in a time-variant application. The

top, front and back edges of the animal will generally change only marginally between frames, while the legs of the animal are constantly moving. The leg information is removed by tracing left and right from the start point of the s-psi chain until the outline drops to $1/5^{th}$ maximal height. After this point, the object is no longer a closed chain, and directionality will affect subsequent processing.

Determine the direction that the animal is travelling

The direction of travel can be easily determined by where the object enters the screen. Any animal will travel nose first in normal movement, so the leading edge may be assumed to be the nose of the animal. The s-psi chain is then reconfigured starting at the leading edge and finishing at the tail.

Determine size and orientation

Use the bounding box algorithms to determine the size and orientation of the s-psi chain. All animals will fit within a square to rectangular orientation, with height/width ratio from 1:1 to 1:2.

Search for Features

Each species of animal has general features that may be used to bias the probability matrix toward or away from determination. For example, most goats have tails that will generally be extended up and back from the animals rump, but sheep tails will always hang down if they are intact. As per Chapter 3, a tail is identified directly from the s-psi graph by a simple inexpensive search for an area of the chain with a local maxima around $2\pi/4$ between areas of constant psi.

Apply heuristic rules

Cattle and horses are always larger than 1 m. Goats and sheep are always smaller than 1.5m. Pigs and kangaroos generally drink at night, other animals usually by day. These decision support rules are applied as probability modifiers to the final classification process.

Correlate the s-ψ chain to the template library

A library of shape templates has been created from sample video clips of each of the animal species. The s-ψ chain is normalised and smoothed prior to correlation. This provides size invariance to the algorithm so that immature and smaller animals are still correctly identified. The s-ψ chain is then matched to each item in the library. The correlation is done repeatedly with phase shifts of the current chain of between -10 and +10 steps. The best match values for each library species is added to the evidence based matching routine.

Match

If the accumulated probability for any species falls above a set threshold, treat the object as identified. Continue building evidence each frame by tracking the object to ensure that the correct decision has been made.

7.3.6 Template library

Below are the templates of the animals identified in this project. The images on the left are the outlines in the space domain, the matching s-ψ graph is on the right. Figure 77 displays an example s-ψ graph, with the matching tail, head and back sections circled.

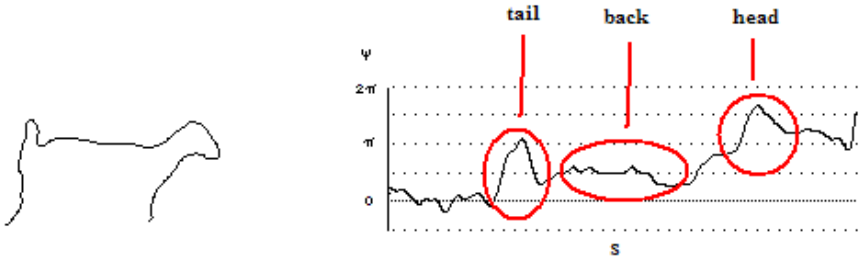
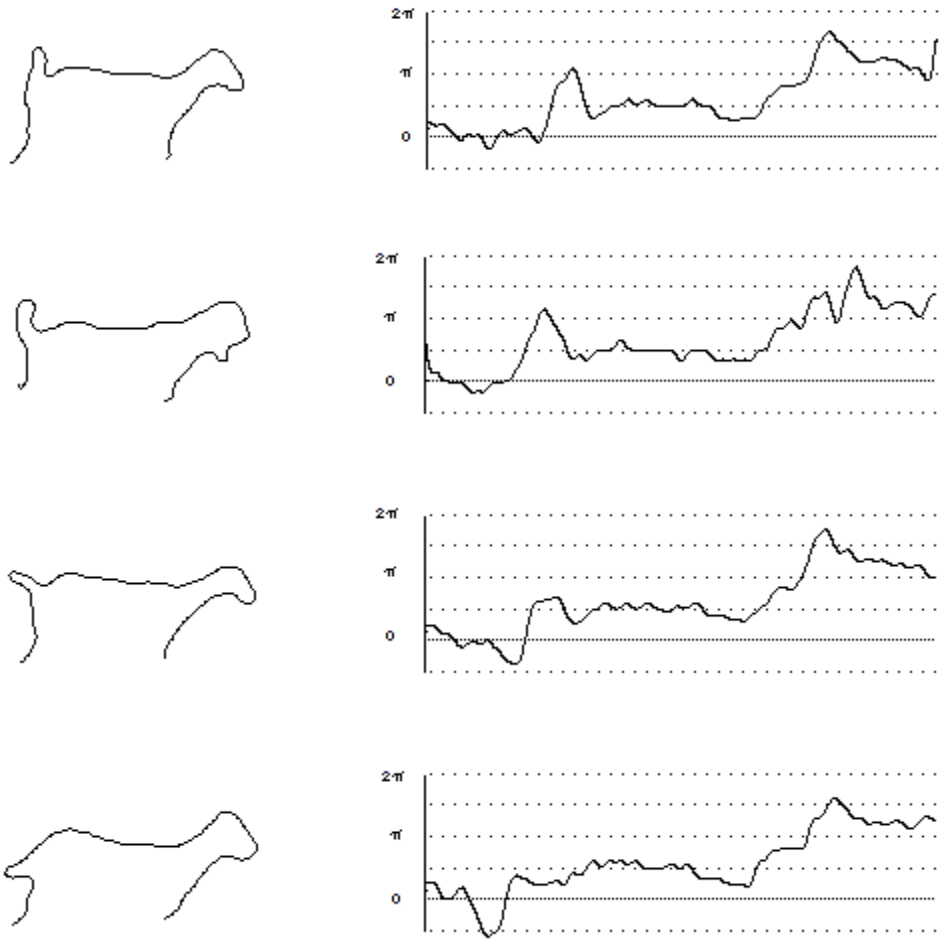
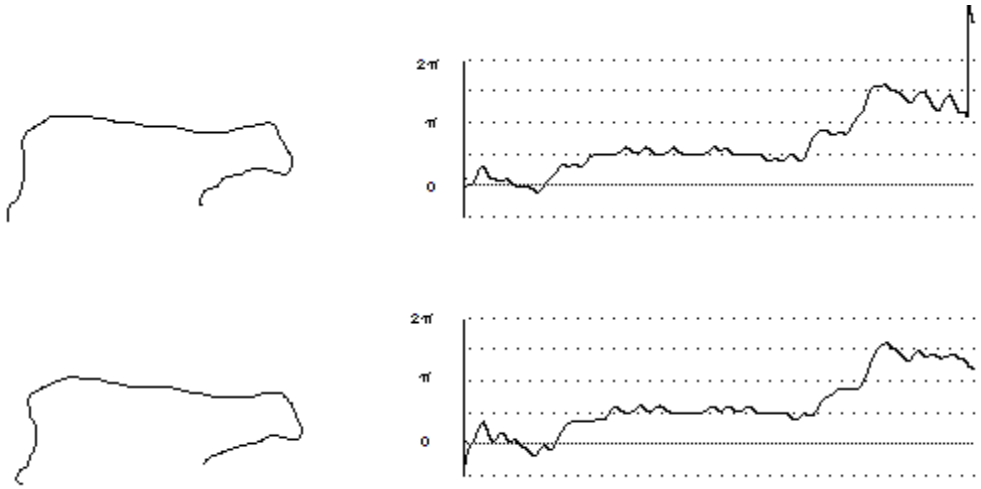


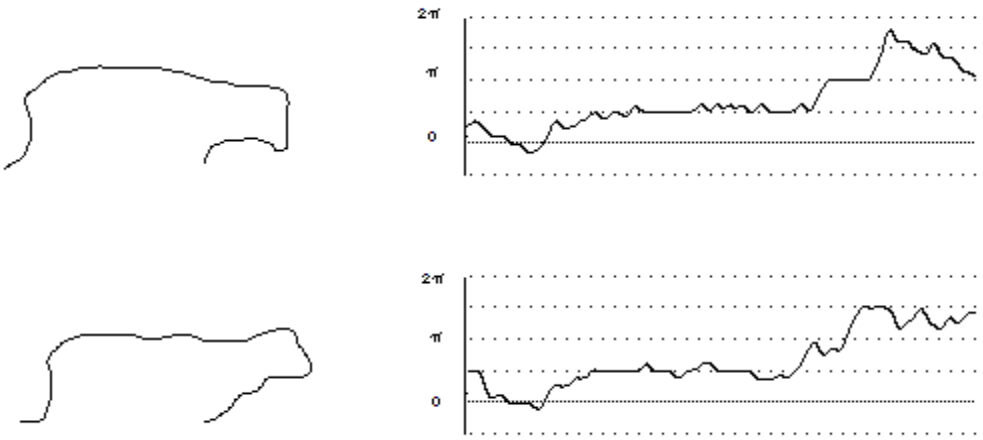
Figure 77 Example template s-ψ graph

7.3.6.1 Goats

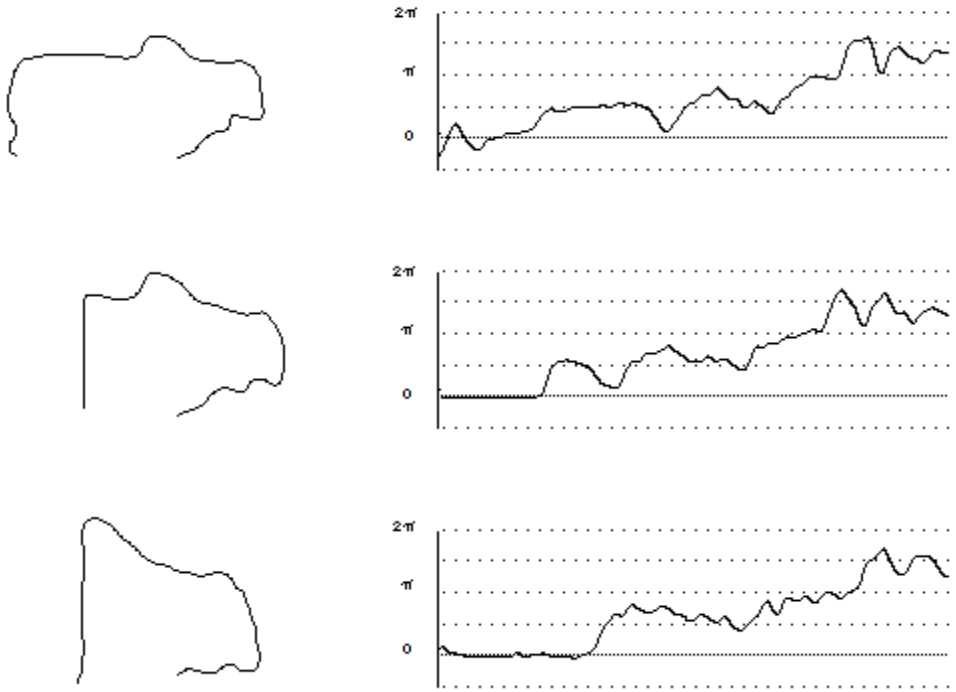


7.3.6.2 Sheep

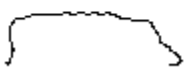


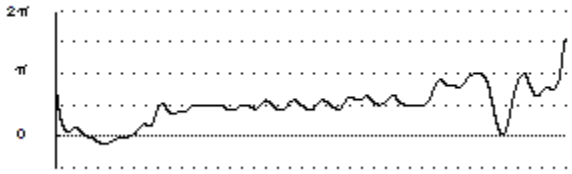


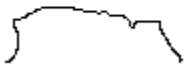
7.3.6.3 Cattle

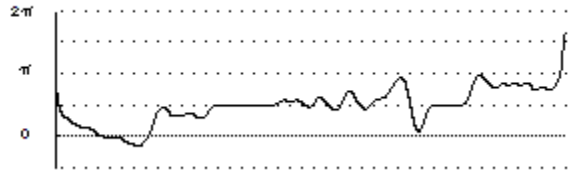


7.3.6.4 Pigs

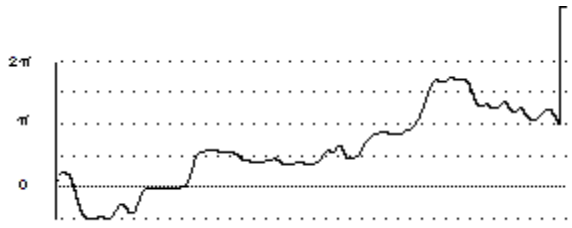


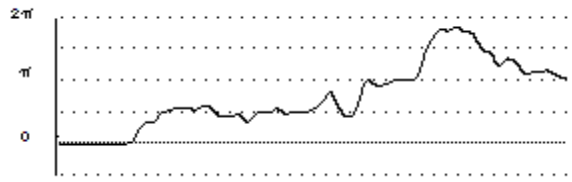




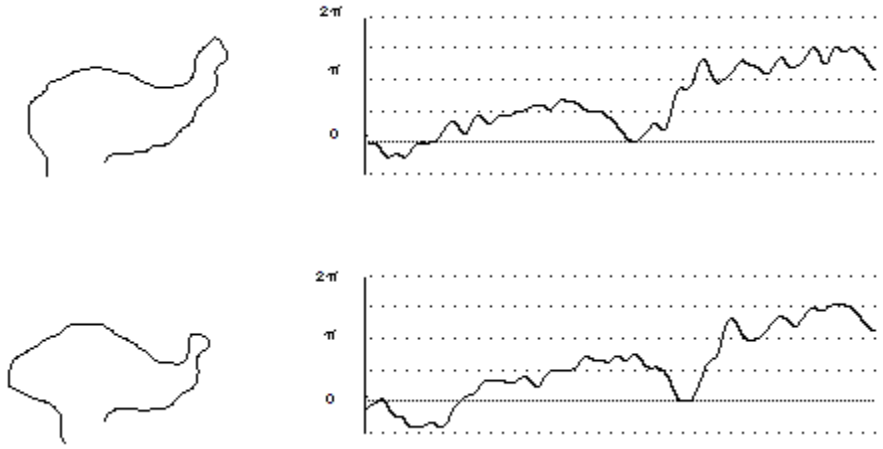


7.3.6.5 Horses

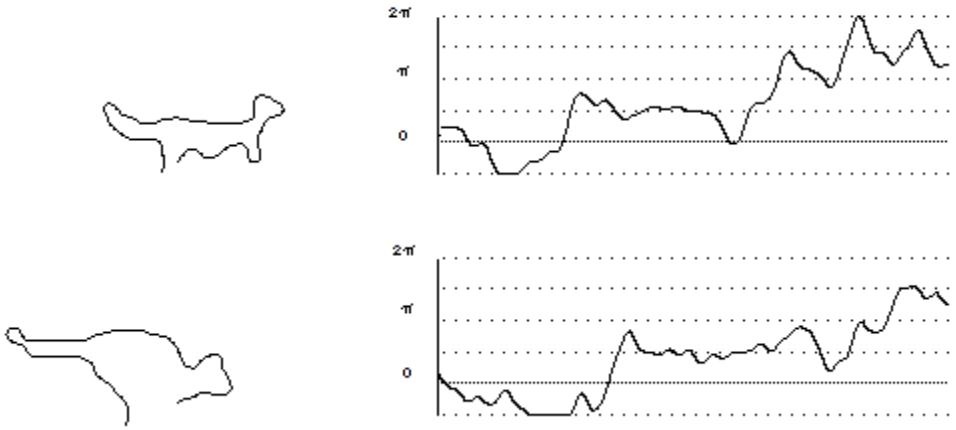




7.3.6.6 Emus



7.3.6.7 Kangaroos



Note that larger animals (cattle and horses) may be identified from the outline of the front of their body, as they are generally too long to fit in a single image frame.

7.4 Results

During the course of this project, over 5000 animal events were recorded. Over 1400 animal events were ground-truthed manually. This was done by a human identifying the species of each animal in the video clip. The manual classification details were stored in a file for comparison with the automatic classification. The results are

tabulated in tables 8, 9 and 10. As shown in Table 10, all animals identified except sheep are correctly classified in over 90% of instances.

Animal events marked as 'missed' are the animals that were not identified by the system. This occurred for a variety of reasons, including multiple animals and fast travelling animals. In particular, the high level of missed classification for kangaroos is due to fast hopping animals being in the field of view for only one or two frames.

Table 8 Confusion matrix

Real \ Log	Goat	Sheep	Pig	Kangaroo	Emu	Cow	Horse	Missed	Total
Goat	540	10	3	9	1	0	0	21	584
Sheep	29	169	12	0	0	0	0	3	213
Pig	0	1	72	0	0	0	0	5	78
Kangaroo	0	0	0	16	1	0	0	9	26
Emu	0	0	0	1	69	2	0	12	84
Cow	0	0	0	0	4	233	11	54	302
Horse	0	0	0	0	0	0	0	0	0
Total									1287

Table 9 Complete system accuracy

Real \ Log	Goat	Sheep	Pig	Kangaroo	Emu	Cow	Horse	Missed
Goat	92%	2%	1%	2%	0%			4%
Sheep	14%	79%	6%					1%
Pig		1%	92%					6%
Kangaroo				62%	4%			35%
Emu				1%	82%	2%		14%
Cow					1%	77%	4%	18%

Table 10 Cross-Classification accuracy

Real \ Log	Goat	Sheep	Pig	Kangaroo	Emu	Cow	Horse
Goat	96%	2%	1%	2%	0%		
Sheep	14%	80%	6%				
Pig		1%	99%				

Kangaroo				94%	6%		
Emu				1%	96%	3%	
Cow					2%	94%	4%

A sample of full results indicating date, time and file is given in Appendix D. The raw data generating the above tables is provided on the CD under the folder *Case Study Results\Animal Species Identification.*

7.4.1 KPI results

The following results against the KPI targets are detailed below:

1. Accuracy. KPI not yet met. While the results cross both ideal and normal environmental conditions, the accuracy for sheep recognition is below the required level.

2. Robustness. KPI met. Several systems have been run over a period of at least six months with no failure. The only hardware failure encountered to date has been with solar panel regulators and blown fuses.

3. Cost. KPI exceeded. The unit price of the complete system has been evaluated at around $AU4000 by the commercial partner to the project.

4. Power Consumption. KPI met. The camera/computer system draws under 100mA at 12V, requiring only a 10W solar panel and 12Ah battery to ensure continuous operation.

7.5 Conclusions

This project has proved the concept that a machine vision system can accurately identify animals to species level and has had a considerable number of hours testing. There are currently six complete system units in the field being tested.

The system hardware has progressed from full PC and associated support equipment to a stand-alone unit capable of indefinite run time on site. The system is in final commercialisation negotiations and will be available to end users in 2007.

Algorithms for detection, tracking and identification have been developed and implemented on both a PC platform and an embedded system (ROC) platform.

The animal shape algorithms have proven to work effectively in a real world situation. The main limiting factors to this technology are now environmental issues generic to remote sensing technology in natural environments. Specifically, dust masking the shape of the animals, variable lighting conditions and animal behaviours (bunching, moving fast through the field of view) caused the majority of misclassifications.

7.5.1 Model summary

Below is a summary of this case study in the terms identified in Chapter 4.

Image Acquisition

The image acquisition was initially performed with a Logitech Quickcam Pro 4000 webcam. Phase 2 of this project moved to image capture by thROC.

Pre Process

Determine method and threshold for image segmentation. The most effective algorithms were using blue background for discrimination.

Analysis

Edge tracing, probability modifiers, classification by template matching.

Post Process

Activate gate, record images and NLIS if available.

Summarise activations and record. Transfer to remote site if applicable.

Offline Processing

Testing only: Record video sequence for template library updates.

8 Generalising shape analysis

8.1 Introduction

Shape analysis has been demonstrated in the previous chapters on a per-project basis. There are, however, many common threads between all these projects that may be considered part of a generic methodology for shape analysis.

When discussing shape analysis, we are usually interested in **classifying** an object by use of its shape. This can be represented by geometry, points, lines and planes, or by higher order descriptors, such as Freeman's chain, s-psi curves, scale space or Bezier curves.

In many applications, there can be rotational, translational and scale factors applied to the target shape. This means that we need to find invariant representations of the shape before we apply higher order analysis techniques, or suffer the penalty of searching for the object at every conceivable angle and position.

So to analyse the shape, we need to Detect, Describe and Match (or measure). These requirements are discussed separately, in detail, in the following sections.

8.2 Shape detection

As with any process, the first requirement is a place to start. For machine vision applications, that must be contrast. It is necessary to be able to differentiate 'something' from 'something else'. The lowest level of this contrast will be colour. Regardless of the colour space (YUV, RGB, Greyscale, LAB etc), there must always be a means of differentiation in (at least one of) the wavelengths that the image sensor is detecting. For example, a white goat against a blue background will be discernable regardless of the colour space, but a white goat against a white background will be almost indistinguishable unless the image sensor is capable of NIR or thermal IR wavelength detection.

Some feasible methods of direct object detection are:

- colour discrimination;
- motion discrimination;
- location, or
- frequency domain analysis

In all but the simplest applications, there will be some level of shape analysis. An example exception is in simple colour sorting applications, where the grade is simply a percentage measurement of the pixels that meet target colour requirements.

While it may be the shape that is important in an application, the method of measurement may or may not require outlines. For example, the citrus application showed that the shape of the dimples is important, but could be measured without outlines *per se*.

8.3 Shape descriptions

There are several methods of describing a shape. As discussed above, the description should be invariant to translation, rotation and scale to ensure that the matching process may be done efficiently.

Using the outline is the most common, and this may be accomplished using techniques such as s-psi chain and Freeman's chain. B-Spline chains and Bezier curves also have uses in reducing the data to describe the edge.

Other methods use statistical properties to approximate the shape. Descriptors such as area, perimeter, compactness and orientation may be combined to provide a method of differentiating shapes.

Euler number, profiles and moments are other descriptors that can be used effectively.

Methods involving parameter space are common, such as Curvature Scale Space, described by Mokhtarian [84, 85].

8.4 Shape matching or measurement

Procrustes Analysis is a technique for minimising the sum of the Euclidean distances of one shape from another. This is done by centring the shape (removing the translation component), and rotating and scaling each shape to find the minimum difference in circles enclosing each shape.

Template matching is commonly used for edge shape descriptors such as s-ψ. The s-ψ technique has a bonus in this area as it is already translation invariant and can be made rotation invariant by using ds,dψ chains as described in chapter 3. Scale invariance can be added by normalising the templates and the matching chain to a set number of points. This reduces 2D shape matching to a 1D minimisation function.

Statistical techniques may be used for simple geometrical descriptors such as area and circularity. In this circumstance, the simple Sum of Squares is adequate to provide a distance function given several numerical descriptors, providing they can be scaled to similar units.

8.4.1 Higher dimensionality

There is no reason shape analysis has to be complete in two dimensions. If the goal of a shape analysis technique is classification, we can use time as another dimension. Consider a video stream with an object moving across the frame at a constant (or non-constant) rate. Evidence gathering techniques can be used to build a probability tree of the different object possibilities. Updated every frame, it is possible to have a higher confidence level in the classification than from a single snapshot. Both speed and motion type (constant, jerky, sinusoidal) can be used as data.

Even the third normal space dimension, depth of image, can be used if either stereo vision is used (two cameras/multiple cameras), or side-to side movement is built into the system.

Even in 2D images, there are multiple dimensions available. Consider a standard colour camera with an RGB space image output. Using standard image processing techniques it is possible to:

- convert to grey scale and use contrast differences to detect edges;
- use a single colour channel;
- use the difference between channels, or a mathematical combination thereof (eg 1.5R-2.2B+0.7G).

Unfortunately, combining colour channels can result in too much information, as there are infinite combinations. Add in multi spectral data and it gets more complicated. There are standard cameras that provide infra-red sensitivity. Hyper spectral data is also provided by satellite image capture devices. These devices capture many wavelengths of light, of which the human visible range is only a small part. Tuneable filters could also be added to a standard camera to allow only certain bands of light.

All of these sources of data need to be considered for a complete system.

8.5 Toolkit

This section aims at providing a starting point for an investigation into a new application. As for many applications, the overriding concern should be to keep the system as simple as possible to operate.

8.5.1 Image capture

The image sensor used should meet minimum environmental standards depending on the application. Most common webcams can withstand normal conditions, but will need housing for external applications.

The image sensor should have approximately double the resolution required to provide minimum accuracy. For example, if an object is to be measured to 1mm accuracy, ensure that there are 2pixels per mm in the design.

Ensure that the image is stored in a fashion conducive to further processing. The simplest and most logical method is a frame-buffer area in memory, with each horizontal pixel adjacent in memory, in per pixel blocks. For example, RGB should be stored as RGBRGBRGBRGB. Similarly, YUV format should be stored as YUVYUVYUV etc. This reduces the amount of mathematical operations required for most processing algorithms as the stride between horizontal or vertical pixels is known.

8.5.2 Pre-processing

One of the most important aspects of the system is to describe the target definition correctly and generic enough to cover situations that will occur in practice.

A colour space conversion application is included in the software (copy the files from the CD Folder *'Applications/MVColourSpaceExplorer'*, run *RegisterMe.bat* then MVColourSpaceExplorer.exe) for use in colour space investigations. A mathematical conversion from RGB (sRGB) space to any other linear colour space can be entered. Each new colour parameter will be displayed in greyscale in a separate window. Auto conversion to YUV, XYZ and HSV are provided.

Figure 78 demonstrates the user interface for the colour space conversion utility. Figure 79 displays the output for a certain frame using various colour space mappings.

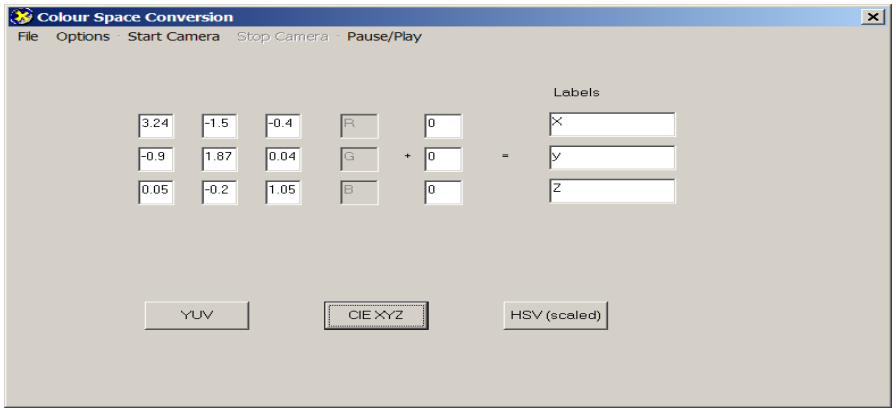


Figure 78 Colour Space Explorer

Figure 79 Example colour Space processed frames. From top: RGB, YUV,Cie XYZ, HSV

For a real application, a target function can be defined with the aid of colour or motion segmentation, or a combination of both as used in the project examining identification of animal species.

Possible tasks for the pre-processing step include:

- Colour conversion/segmentation
- Motion detection/segmentation
- Noise reduction

8.5.3 Analysis

At this point, the application requirements must be considered for the best method of proceeding. The level of analysis complexity will depend on the objects to be detected, described and matched. If the requirement is for a simple 'is there an object in front of the camera?', a correspondingly simple target pixel count may suffice. More complex requirements will obviously entail more effort.

The general analysis families are:

- Feature Extraction
    - Line detection
    - Hough transforms
    - Parameter estimation
    - Shape
- Geometry
    - Chain codes
    - Shape
    - Curvature Scale Space
- Statistical
    - Histograms
    - Probabilities
- Model based
    - Contours
    - prior knowledge
- Artificial Neural Networks

Use the templates provided (described in Chapter 4 and Appendix B) to implement any of the above pre-processing and analysis techniques into processing filters. Each technique can be implemented as a linear algorithm, suitable for implementation in this format. Some ANN techniques are more suitable for parallel processing, but linear techniques can be performed (although they will obviously be slower).

8.5.4 Post processing

Use the application templates provided (described in Chapter 4 and Appendix B) to implement a complete machine vision solution. The post processing step will usually be done at the application level, accumulating or using information from the filters. The application has the appropriate resources to utilise the file system and hardware input/output techniques.

8.6 Conclusions

This chapter has provided a generic model framework for real world machine vision applications. The tools required to construct a complete machine vision software solution have been provided and described.

An overview of shape analysis has shown that most machine vision applications will have some shape functionality required at one or more levels. The basic requirements of shape detection, description and matching or measurement can be applied to any object detection routine. An examination of all the available input data should be undertaken before commencing system design.

Chapter 9 Conclusions

9.1 Introduction

This thesis has dealt with practical applications of machine vision. New algorithms and methodologies have been introduced, particularly for s-psi theory and application.

9.2 Contributions

This thesis has formalised the S–psi operations in machine vision. This will allow the use of S-psi processing in more mainstream applications. S-psi has been overlooked recently as a viable method for shape analysis as there has been no research or advances in the field.

Machine vision has been implemented into several new areas in agriculture. As technologies are introduced to the real fields, there is a growing awareness of the benefits produced by automation and increased information produced by machine vision applications. The uptake of technology is accelerating as it becomes more accessible and, importantly, robust and reliable.

This research has provided practical solutions to current problems. The specific applications described in this document, namely macadamia, animal identification and citrus texture analysis, are real world problems solved by the judicious use of machine vision. In each of these areas, a viable solution has been developed and is now in use.

A toolkit for Do It Yourself machine vision applications has been provided. Another model for generic applications has been introduced and described. This approach is suggested for any machine vision system, using templates provided as an immediate starting point. A model consisting of: Image capture, Pre-process, Analysis, Post-process and Offline processing has been described and recommended as the skeleton of any implementation of vision processing.

This research has resulted in the creation of a cheap, low-powered embedded machine vision platform for use in remote or mobile applications. The Rugged Outdoor Camera will be a useful tool for research students, DIY machine vision hobbyists, or for inclusion in real-world production systems.

9.3 Publications, reports and articles from this thesis

The following list of papers, presentations and reports have been generated from the work presented in this thesis. [74, 86-96]

2003.

Banhazi, T., **Dunn**, M., Cook, P., & Durack, M. (2003). *Review of Precision Livestock Farming (PLF) technologies for the Australian pig industry* (Project report). Toowoomba: Australian Pork Ltd.

Dunn, M., Billingsley, J., & Finch, N. (2003). Machine Vision Classification of Animals. In *Mechatronics and Machine Vision 2003:Future trends* (pp. 157-163). Baldock, UK: Research Studies Press Ltd.

Dunn, M., & Billingsley, J. (2003). *Machine vision system for counting macadamia nuts.* Paper presented at the Australasian Conference on Robotics and Automation, Brisbane.

2004.

Dunn, M., & Billingsley, J. (2004). *A Machine Vision System for Surface Texture Measurements of Citrus.* Paper presented at the Proceedings 11th IEEE conference on Mechatronics and Machine Vision in Practice, Macau.

Dunn, M., Billingsley, J., Raine, S., & Piper, A. (2004, November). *Using Machine Vision for Objective Evaluation of Ground Cover on Sporting Fields.* Paper presented at the Proceedings 11th IEEE conference on Mechatronics and Machine Vision in Practice, Macau.

2005.

Bjursell, A., **Dunn**, M., Withers, K., Senior, G., Lundie-Jenkins, G., & Billingsley, J. (2005, December 2005). *Application of Machine Vision Technology to identification of Oestrus in the Julia Creek Dunnart (Sminthopsis douglasi).* Paper presented at the 22nd Annual Meeting – Australian & New Zealand Society for Comparative Physiology & Biochemistry, Dunedin.

Dunn, M. (2005). *Implementation of Precision Agriculture Technologies in the Macadamia Industry* (Project Report). Toowoomba: Horticulture Australia Ltd.

Dunn, M., Billingsley, J., & Bell, D. (2005). *Implementation of Precision Agriculture Technologies in the Macadamia Industry.* Paper presented at the Proceedings 12th IEEE conference on Mechatronics and Machine Vision in Practice, Manila.

2006.

Billingsley, J., & **Dunn**, M. (2006). Strange vision - machine vision applications at the NCEA. *Sensor Review, 25*(3), 202-208.

Dunn, M., Billingsley, J., & Bell, D. (2006). Vision based macadamia yield assessment. *Sensor Review, 26*(4), 312-317.

N. Finch, P. J. Murray, M. **Dunn**, and J. Billingsley, "Control of access to water using machine vision classification of animals," *Australian Journal of Experimental Agriculture*, vol. 46, pp. 837-839, 2006.

Galea, M. (2006, May 2006). Vision tool keeps ferals away. *Australian Farm Journal, 16,* 22-23.

9.4 Further research

This research has shown the scope of s-psi algorithms with relation to shape analysis, and it is envisaged that the use of these algorithms extends beyond this field. An example might be modelling object behaviour from identified position and configuration using s-psi techniques.

Further work is anticipated into the Animal Species Identification Project. This project will be extended into other countries to deal with native and introduced species relevant to that area. A new method of automatic library accumulation with user supervision will be incorporated to make this system stand alone and customisable. In this way, the system may be used for any sequence of items moving past the camera (or indeed, a moving camera), with the only limitation being on the size of the object.

The macadamia implementation will be extended to use RFID technology for macro field placement. There is also scope here to investigate automatic image registration of aerial or satellite photographs (from Google Earth™ for example) using a farm map gathered from Treecam readings.

NCEA is currently investigating several new machine vision technology applications that will require exploration using the models, templates and techniques described in this document.

References

[1] J. Billingsley, "The Counting of Macadamia Nuts," presented at Mechatronics and Machine Vision, Chiang Mai, Thailand, 2002.

[2] C. Hardner, "Review of method and costs of mechanical harvesting for assessment in genetic improvement program," CSIRO, Brisbane 2004.

[3] C. Hardner, "Cost of manual and mechanised yield assessment," CSIRO, Brisbane 2005.

[4] R. E. Plant, "Site-Specific Management: The Application Of Information Technology To Crop Production.," *Computers and Electronics in Agriculture*, vol. 30, pp. 9-29, 2001.

[5] N. Zhang, W. M., and N. Wang, "Agriculture – a Worldwide Overview.," *Computers and Electronics in Agriculture*, vol. 36, pp. 113-132, 2002.

[6] J. V. Stafford, R. M. Lark, and H. C. Bolam, "Using Yield Maps to Recognise Fields into Potential Management Units.," presented at Using Yield Maps to Recognise Fields into Potential Management Units., Madison., 1999.

[7] B. B. Beattie and L. J. Revelant, "Product Description Language Oranges: Reference Guide," NSW Department of Primary Industries, Sydney 1995.

[8] F. a. F. Dept of Agriculture, "Great Artesian Basin Sustainability Initiative," 2006.

[9] E. R. Davies, *Machine Vision Theory Algorithms, Practicalities*, 2nd ed. London: Academic Press, 1997.

[10] J. Billingsley, *Essentials of Mechatronics*: John Wiley & Sons, 2006.

[11] N. Zuech, "Machine Vision Trends - 2006," vol. 2006, 2006.

[12] I. Global Information, "Advances in Machine Vision Systems," vol. 2006: Technical Insights, Inc, 2005.

[13] P. Aimonen, "Machine Vision News," vol. 2006, 2004.

[14] J. A. Marchant, C. M. Onyango, and M. J. Street, "Computer vision for potato inspection without singulation," *Computers and Electronics in Agriculture*, vol. 4, pp. 235-244, 1990.

[15] M. S. Howarth and S. W. Searcy, "Algorithms for grading carrots by machine vision," presented at ASAE International Summer Meeting, New Orleans, LA, 1989.

[16] G. F. Figueiredo, T. W. Dickerson, E. R. Benson, G. L. Van Wicklen, and N. Gedamu, "Development of Machine Vision Based Poultry

Behavior Analysis System," presented at 2003 ASAE Annual International Meeting, Las Vegas, Nevada, 2003.

[17] I. Philipp and T. Rath, "Improving plant discrimination in image processing by use of different colour space transformations," *Computers and Electronics in Agriculture*, pp. 1-15, 2002.

[18] A. Plebe and G. Grasso, "Localization of spherical fruits for robotic harvesting," *Machine Vision and Applications*, vol. 13, pp. 70-79, 2001.

[19] D. Stajnko, M. Lakota, and M. Hocevar, "Estimation of number and diameter of apple fruits in an orchard during the growing season by thermal imaging," *Computers and Electronics in Agriculture*, pp. 31-42, 2004.

[20] R. O. Duda and P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," *Communications of the ACM*, vol. 15, pp. 11-15, 1972.
[21] H. Erives and G. J. Fitzgerald, "Automated registration of hyperspectral images for precision agriculture," *Computers and Electronics in Agriculture*, pp. 103-119, 2005.
[22] D. Sergeant, R. Boyle, and M. Forbes, "Computer visual tracking of poultry," *Computers and Electronics in Agriculture*, vol. 21, pp. 1-18, 1998.
[23] J. S. Zelek, B. N., and R. Kanwar, "Watching Pigs," University of Guelph, Ontario, Canada 2000.
[24] A. R. Frost, A. P. French, R. D. Tillett, T. P. Pridmore, and S. K. Welch, "A vision guided robot for tracking a live, loosely constrained pig," *Computers and Electronics in Agriculture*, pp. 93-106, 2004.
[25] L. Noldus, A. J. Spink, and R. Tegelenbosch, "Computerised video tracking, movement analysis and behaviour recognition in insects," *Computers and Electronics in Agriculture*, pp. 201-227, 2002.
[26] B. S. Bennedsen, D. L. Peterson, and A. Tabb, "Identifying defects in images of rotating apples," *Computers and Electronics in Agriculture*, 2005.
[27] E. R. Davies, M. Bateman, D. R. Mason, J. Chambers, and C. Ridgway, "Design of efficient line segment detectors for cereal grain inspection," *Pattern Recognition Letters*, vol. 24, pp. 413-428, 2003.
[28] D. Warren, "Image analysis in chrysanthemum DUS testing," *Computers and Electronics in Agriculture*, vol. 25, pp. 213-220, 2000.
[29] J. W. Funck, Y. Zhong, D. A. Butler, C. C. Brunner, and J. B. Forrer, "Image segmentation algorithms applied to wood defect detection," *Computers and Electronics in Agriculture*, vol. 41, pp. 157-179, 2003.
[30] B. K. Yadav and V. K. Jindal, "Monitoring milling quality of rice by image analysis," *Computers and Electronics in Agriculture*, vol. 33, pp. 19-33, 2001.
[31] J. A. Marchant, H. J. Andersen, and C. M. Onyango, "Evaluation of an imaging sensor for detecting vegetation using different waveband combinations," *Computers and Electronics in Agriculture*, pp. 101-117, 2001.
[32] S. K. Maxwell, J. R. Nuckols, M. H. Ward, and R. M. Hoffer, "An automated approach to mapping corn from Landsat imagery," *Computers and Electronics in Agriculture*, pp. 43-54, 2004.
[33] V. Alchanatis, L. Ridel, A. Hetzroni, and L. Yaroslavsky, "Weed detection in multi-spectral images of cotton fields," *Computers and Electronics in Agriculture*, pp. 243-260, 2005.
[34] P. Tantaswadi, "Machine Vision for Automated Visual Inspection of Cotton Quality in Textile Industries Using Color Isodiscrimination Contour," *Nectec Technical Journal*, vol. 1, 1999.
[35] J. C. Noordam, G. W. Otten, A. J. M. Timmermans, and B. v. Zwol, "High-speed potato grading and quality inspection based on a color vision system," *Machine Vision Applications in Industrial Inspection VIII*, vol. 3966, pp. 206-220, 2000.
[36] D. G. Sena Jr, F. A. C. Pinto, P. A. Queiroz, and P. A. Viana, "Fall Armyworm Damaged Maize Plant Identification using Digital Images," *Biosystems Engineering*, vol. 85, pp. 449-454, 2003.
[37] S. Jeyamkondan, N. Ray, G. Kranzler, and N. Biju, "Beef Quality Grading Using Machine Vision," presented at SPIE 4203-Biological Quality and Precision Agriculture II, Boston, 2000.

[38] P. Granitto, P. Verdes, and H. A. Ceccatto, "Large-Scale Investigation of Weed Seeds Identification by Machine Vision Techniques," *International Journal of Pattern Recognition and Artificial Intelligence*, 2003.

[39] J. A. Marchant and C. M. Onyango, "Segmentation of row crop plants from weeds using colour and morphology," *Computers and Electronics in Agriculture*, vol. 39, pp. 141-155, 2003.

[40] N. Aleixos, J. Blasco, F. Navarron, and E. Molto, "Multispectral inspection of citrus in real-time using machine vision and digital signal processors," *Computers and Electronics in Agriculture*, pp. 121-137, 2002.

[41] W. S. Lee, D. C. Slaughter, and D. K. Giles, "Robotic Weed Control System for Tomatoes," *Precision Agriculture*, vol. 1, pp. 95-113, 1999.

[42] G. Meyer, J. Neto, D. D. Jones, and T. W. Hindman, "Intensified fuzzy clusters for classifying plant, soil, and residue regions of interest from color images," *Computers and Electronics in Agriculture*, pp. 161-180, 2004.

[43] D. E. Kline, C. Surak, and P. A. Araman, "Automated hardwood lumber grading utilizing miltiple sensor machine vision technology," *computers and Electronics in Agriculture*, vol. 41, pp. 139-155, 2003.

[44] N. Kondo, U. Ahmad, M. Monta, and H. Murase, "Machine vision based quality evaluation of Iyokan orange fruit using neural networks," *Computers and Electronics in Agriculture*, pp. 135-147, 2000.

[45] D. Guyer and X. Yang, "Use of genetic artificial neural networks and spectral imaging for defect detection on cherries," *Computers and Electronics in Agriculture*, pp. 179-194, 2000.

[46] P. K. Goel, S. O. Prasher, R. M. Patel, J. A. Landry, R. B. Bonnell, and A. A. Viau, "Classification of hyperspectral data by decision trees and artificial neural networks to identify weed stress and nitrogen status of corn," *Computers and Electronics in Agriculture*, pp. 67-93, 2003.

[47] Y. Uno, S. O. Prasher, R. Lacroix, P. K. Goel, Y. Karimi, A. A. Viau, and R. M. Patel, "Artificial neural networks to predict corn yield from Compact Airborne Spectrographic Imager data," *Computers and Electronics in Agriculture*, pp. 149-161, 2005.

[48] D. Moshou, C. Bravo, J. West, S. Wahlen, A. McCartney, and H. Ramon, "Automatic detection of 'yellow rust' in wheat using reflectance measurements and neural networks," *Computers and Electronics in Agriculture*, vol. 44, pp. 173-188, 2004.

[49] V. C. Patel, R. W. McLendon, and J. W. Goodrum, "Color Computer Vision and Artificial Neural Networks for the Detection of Defects in Poultry Eggs," *Artificial Intelligence Review*, vol. 12, pp. 163-176, 1998.

[50] M. Siddaiah, M. A. Lieberman, S. E. Hughs, and N. R. Prasad, "Automation in Cotton Ginning," presented at First International Conference on Intelligent Technologies (Intech'2000), Bangkok, Thailand, 2000.

[51] I. M. Johnston, "The magic of old tractors." Sydney: New Holland Publishers (Australia) Pty Ltd, 2004, pp. p26.

[52] J. Billingsley and M. Schoenfisch, "A vision-guided agricultural tractor," presented at Australian Robot Association Conference, Robots for Competitive Industries, Brisbane, 1993.

[53] J. Billingsley and M. Schoenfisch, "The successful development of a vision guidance system for agriculture," *Computers and Electronics in Agriculture*, vol. 16, pp. 147-163, 1997.

[54] J. Billingsley, "Automatic guidance of agricultural mobiles at the NCEA," *Industrial Robot*, vol. 27, pp. 449-457, 2000.
[55] K. Fitzpatrick, D. Pahnos, and W. V. Pype, "Robot windrower is first unmanned harvester," *1997*, vol. 24, pp. 342-348, 1997.
[56] R. D. Tillett, C. M. Onyango, and J. A. Marchant, "Using model-based image processing to track animal movements," *Computers and Electronics in Agriculture*, vol. 17, pp. 249-261, 1997.
[57] D. C. Slaughter, P. Chen, and R. G. Curley, "Vision Guided Precision Cultivation," *Precision Agriculture*, vol. 1, pp. 199-216, 1999.
[58] T. Hague, J. A. Marchant, and R. D. Tillett, "Ground based sensing systems for autonomous agricultural vehicles," *Computers and Electronics in Agriculture*, vol. 25, pp. 11-28, 2000.
[59] H. T. Sogaard and H. J. Olsen, "Determination of crop rows by image analysis without segmentation," *Computers and Electronics in Agriculture*, vol. 38, pp. 141-158, 2003.
[60] R. Vaughan, N. Sumpter, A. Frost, and S. Cameron, "Robot sheepdog project achieves automatic flock control," presented at Fifth International Conference on the Simulation of Adaptive Behaviour, 1998.
[61] C. P. Schofield, J. A. Marchant, R. P. White, N. Brandl, and M. Wilson, "Monitoring Pig Growth using a Prototype Imaging System," *Journal of Agricultural Engineering Research*, vol. 72, pp. 205-210, 1999.
[62] R. P. White, D. J. Parsons, C. P. Schofield, D. M. Green, and C. T. Whittemore, "Use of visual image analysis for the management of pig growth in size and shape," British Society of Animal Science 2003, 2003.
[63] J. Wu, R. D. Tillett, N. McFarlane, X. Ju, J. P. Siebert, and C. P. Schofield, "Extracting the three-dimensional shape of live pigs using stereo photogrammetry," *Computers and Electronics in Agriculture*, vol. 44, pp. 203-222, 2004.
[64] A. Peacock and R. Boyce, "Biomimetic robotics heralds new era in dairy farming," *Industrial Robot*, vol. 30, pp. 414-416, 2003.
[65] H. Freeman, "Computer processing of line-drawing images," *ACM Computing Surveys*, vol. 6, pp. 57-98, 1974.
[66] S. Hoque, K. Sirlantzis, and M. C. Fairhurst, "A new Chain-code Quantization Approach Enabling High Performance Handwriting Rrecognition based on Multi-Classifier Schemes," presented at Proceedings of the Seventh International Conference on Document Analysis and Recognition, 2003.
[67] V. Govindaraju, Z. Shi, and J. Schneider, "Feature Extraction Using a Chaincoded Contour Representation of Fingerprint Images," presented at Audio- and Video-Based Biometric Person Authentication, Guildford, UK, 2003.
[68] L. J. Latecki and R. Lakamper, "Shape Similarity Measure based on correspondence of visual parts," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, 2000.
[69] E. M. Arkin, L. P. Chew, D. P. Huttenlocher, K. Kedem, and J. S. B. Mitchell, "An efficiently computable Metric for Comparing Polygonal Shapes," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, pp. 209-216, 1991.
[70] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and Machine Intelligence*, vol. 8, pp. 679-698, 1986.

[71] G. Schuster and A. Katsaggelos, "An efficient boundary encoding scheme which is optimal in the rate distortion sense," presented at International Conference on Image Processing, Santa Barbara, CA, 1997.
[72] J. Iivarinen and A. Visa, "Shape Recognition of Irregular Objects," presented at Intelligent Robots and Computer Vision {XV}: Algorithms, Techniques, Active Vision, and Materials Handling, 1996.
[73] M. D. Pesce, *Programming Microsoft DirectShow for Digital Video and Television*: Microsoft Press, US, 2003.
[74] M. Dunn and J. Billingsley, "Machine vision system for counting macadamia nuts," presented at Australasian Conference on Robotics and Automation, Brisbane, 2003.
[75] M. Recce, A. Plebe, J. Taylor, and G. Tropiano, "Video Grading of Oranges in RealTime," *Artificial Intelligence Review*, vol. 12, pp. 117-136, 1998.
[76] R. Leach, "The Measurement of Surface Texture using Stylus Instruments," National Physixs Laboratory, London 2001.
[77] M. Smith, "personal communication," 2005.
[78] Alliance Consulting and Management, "Cost analysis of NLIS compliance for beef producers," Meat & Livestock Australia Ltd, Brisbance 2004.
[79] B. Walker, "Personal communications," NCEA, Ed., 2005.
[80] D. Magee and R. Boyle, "Feature tracking in real world scenes (or how to track a cow)." presented at Proc. IEE Colloquium on Motion Analysis and Tracking, 1999.
[81] Y. Weiss, "Deriving intrinsic images from image seqences," *IEEE Trans. Pattern Analysis and Machine Intelligence*, pp. 68-75, 2001.
[82] H. Y. Shum and R. Szeliski, "Construction and refinement of panoramic mosaics with global and local allignment," *Proceedings IEEE CVPR*, pp. 953-958, 1998.
[83] E. R. Davies, "Designing efficient line segment detectors with high orientation accuracy," presented at Sixth Internation Conference on Image Processing and Its Applications, Dublin, Ireland, 1997.
[84] F. Mokhtarian, "Silhouette-Based Isolated Object Recognition through Curvature Scale Space," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, pp. 539-544, 1995.
[85] F. Mokhtarian and H. Murase, "Silhouette-Based Object Recognition through Curvature Scale Space," presented at International Conference on Computer Vision, 1993.
[86] T. Banhazi, M. Dunn, P. Cook, and M. Durack, "Review of Precision Livestock Farming (PLF) technologies for the Australian pig industry," Australian Pork Ltd, Toowoomba, Project report September 2003 2003.
[87] M. Dunn, J. Billingsley, and N. Finch, "Machine Vision Classification of Animals," in *Mechatronics and Machine Vision 2003:Future trends*. Baldock, UK: Research Studies Press Ltd, 2003, pp. 157-163.
[88] M. Dunn and J. Billingsley, "A Machine Vision System for Surface Texture Measurements of Citrus," presented at Proceedings 11th IEEE conference on Mechatronics and Machine Vision in Practice, Macau, 2004.
[89] M. Dunn, J. Billingsley, S. Raine, and A. Piper, "Using Machine Vision for Objective Evaluation of Ground Cover on Sporting Fields," presented at Proceedings 11th IEEE conference on Mechatronics and Machine Vision in Practice, Macau, 2004.

[90] A. Bjursell, M. Dunn, K. Withers, G. Senior, G. Lundie-Jenkins, and J. Billingsley, "Application of Machine Vision Technology to identification of Oestrus in the Julia Creek Dunnart (Sminthopsis douglasi)." presented at 22nd Annual Meeting – Australian & New Zealand Society for Comparative Physiology & Biochemistry, Dunedin, 2005.
[91] M. Dunn, "Implementation of Precision Agriculture Technologies in the Macadamia Industry," Horticulture Australia Ltd, Toowoomba, Project Report October 2005 2005.
[92] M. Dunn, J. Billingsley, and D. Bell, "Implementation of Precision Agriculture Technologies in the Macadamia Industry," presented at Proceedings 12th IEEE conference on Mechatronics and Machine Vision in Practice, Manila, 2005.
[93] J. Billingsley and M. Dunn, "Strange vision - machine vision applications at the NCEA," *Sensor Review*, vol. 25, pp. 202-208, 2006.
[94] M. Dunn, J. Billingsley, and D. Bell, "Vision based macadamia yield assessment," *Sensor Review*, vol. 26, pp. 312-317, 2006.
[95] N. Finch, P. J. Murray, M. Dunn, and J. Billingsley, "Control of access to water using machine vision classification of animals," *Australian Journal of Experimental Agriculture*, vol. 46, pp. 837-839, 2006.
[96] M. Galea, "Vision tool keeps ferals away," in *Australian Farm Journal*, vol. 16, 2006, pp. 22-23.
[97] M. Dunn, "Video and image processing tools," in *Engineering*. Toowoomba: USQ, 2002, pp. 178 leaves.

Appendix A: CD contents

The structure of folders contained on the CD is below:

- Case Study Results
    - Animal Species Identification
    - Macadamia Yield Map
- DirectShow
    - Applications
    - Filters
        - Binaries
        - MVCannyEdge
        - MVHoughCircle
        - MVSobelEdge
    - Include
    - Lib
    - Templates
        - VB
        - VC6
        - VC7
- Media Samples

The contents of these folders are as follows:

Case Study Results/Animal Species Identification

This folder contains the raw and processed (ground-truthed) data from the animal species identification project.

<name>_log.csv These files contain the algorithm logs for each frame of video in file <name.avi>

Example entry

136 191 150 Cow

The format of this message is:

Frame #

X position of the centre of the identified animal

Y position of the centre of the identified animal

Identified species

<name>_fgt0.csv These files contain the manually ground truthed logs for each frame of video in file <name.avi>

Example entry

716 159 157 Goat

The format of this message is:

Frame #

X position of the centre of the identified animal

Y position of the centre of the identified animal

Identified species

Results.csv This file contains each animal event, summarised by start frame and end frame, for each matching pair of algorithm and ground truth log files.

Case Study Results/Macadamia Yield Map

This folder contains the raw and processed data from 4 trials of the macadamia yield map project.

<name>.GPS These files contain the raw gps data as logged by the Treecam. Note that this data is little endian.

<name>_gps.txt These files contain the gps data converted to normal text mode.

Example string:

$GPGGA,215152.20,2650.18766,S,15255.00311,E,2,06,1.1,64.1,M,41.0,M,8.0,0007*6B

<name>.NUT These files contain the raw nut data as logged by the Treecam. Note that this data is little endian.

<name>_nut.csv These files contain the gps data converted to human readable comma separated value files (in hexadecimal values).

Example entry:

f0 0e 1 7 84 27 0 0 0b 14 6

The format of this message is:

1 Header (always 0xf0)
2 Nutcam serial #
3 Message type (1=nut message)
4 Time since trial start (in microseconds)
9 nut found X position
10 nut found Y position
11 nut found size (radius)

<name>.ODO These files contain the raw Odometry data as logged by the Treecam. Note that this data is little endian.

<name>_odo.csv These files contain the odometry data converted to human readable text form.

Example entry:

3992 9

The format of this message is:

Time since trial started (in microseconds)
Accumulated # radar clicks (1click=5mm)

<name>.TRE These files contain the raw Treecam data. Note that this data is little endian.

<name>_tre.csv These files contain the Treecam data converted to human readable text form.

Example entry:
21617 304 4

The format of this message is:

Time since trial started (in microseconds)

Identified X position of tree trunk

Identified tree trunk width

<name>final.jpg This is a visual image of the post processed data illustrating nuts and trees identified in that trial.
<name>final.tcw This is a drawing file of the end results as above by accessible by TurboCad reader applications.

DirectShow/Applications

This folder contains the Colour Space Explorer program for investigation of optimal colour space algorithms to use in new machine vision applications. To use this program, copy the entire folder to a local disk and run the batch file *registerme.bat*. This should produce a message box similar to Figure 80.

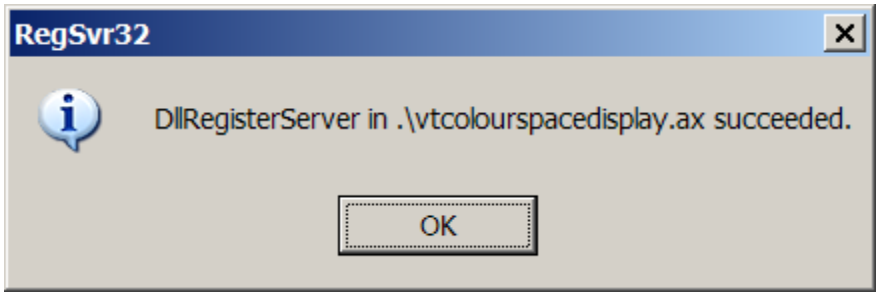


Figure 80 DirectShow filter registration success message

The executable file *MVColourSpaceExplorer.exe* can now be used.

This folder also contains the sample executable *MachineVisionApplication.exe*, a standard program to process video from a camera or file. This application can also incorporate the sample filters described below.

DirectShow/Filters/Binaries

This folder contains the compiled versions of the sample Machine Vision Template Filters. Copy the complete folder to a local disk and run the batch file *RegisterFiles.bat*. This will register the algorithms to run on the local PC. These filters will now be available for selection under *Options->Select Filter* menu of the machine vision template applications.

DirectShow/Filters/MVCannyEdge

This filter contains source code for a sample implementation of Canny's Edge Detection algorithms.

DirectShow/Filters/MVHoughCircle

This filter contains source code for a sample implementation of a Hough Transform for circle detection.

DirectShow/Filters/MVSobelEdge

This filter contains source code for a sample implementation of a Sobel Filters for edge detection.

DirectShow/Include

The contents of this folder should be copied to <DirectShow SDK Install Path>/Include before commencing programming work with the DirectShow templates provided on this CD.

DirectShow/Lib

The contents of this folder should be copied to <DirectShow SDK Install Path>/Lib before commencing programming work with the DirectShow templates provided on this CD.

DirectShow/Templates/VB

This folder contains the file required to create DirectShow Machine Vision Template applications using Microsoft™ Visual Basic 6 or 7.

DirectShow/Templates/VC6

This folder contains the file required to create DirectShow Machine Vision Template applications and filters using Microsoft™ Visual Studio 6.

DirectShow/Templates/VC7

This folder contains the files required to create DirectShow Machine Vision Template applications and filters using Microsoft™ Visual Studio 7 (.net)

Media Samples

This folder contains samples of avi files from the various projects.

Appendix B: Programming in DirectShow

There are two areas that will be briefly touched upon here: the Application, and the Filter.
The following sections require at least rudimentary knowledge of programming in C or C++ and, preferably, Microsoft™ Visual Studio.

Setting up the build environment

Download the DirectX Software Development Kit (SDK) from the Microsoft Download Area (www.microsoft.com/downloads). For preference, use version October 2005 or earlier. Versions after this require download and install of the Windows Platform SDK also.

Install to a high level directory for ease of access (eg C:\DXSDK). The help files have been removed from this SDK into the Windows Platform SDK, which is available for download from the same site.

Copy the contents of the folder *Lib* from the CD to *<install directory>/Lib.*
Copy the contents of the folder *Include* from the CD to *<install directory>/Include.*

Compile the baseclasses. DirectShow is founded on a series of baseclasses, or helper classes. These are the building blocks of almost all tasks most users will perform in DirectShow. This is accomplished by opening the project file relevant to the version of Visual Studio from *<install directory>\Samples\C++\DirectShow\BaseClasses.* Build in both Debug and Release Configuration.

The search directories must be added to the search path. Add <install directory>\Include to the list of include directories. Add *<install directory>\Lib* to

the list of library include directories. In Visual Studio .Net this functionality is found in *Tools->Options->Projects->VC++ Build Directories*. In Visual studio 6 follow *Tools->Options->Directories.*

Application programming

Obviously, we can use GraphEdit to create a filter graph, save it and run it every time machine vision processing is required, but this is not user friendly, nor is it conducive to customisation. An application can build a filter graph in exactly the same way as a user manually adding filters and connecting pins.

There are a number of steps that can be followed to create an application. Alternatively, a generic template for an application has been included in the software on the CD attached to this document.

New Applications

Include the following header into an application:

```
#include <direct.h>
```

And the following libraries into the linker input section:

<install directory>\Samples\C++\DirectShow\BaseClasses\debug\strmbasd.lib
Quartz.lib

In the Initialise routine add:

```
IGraphBuilder *          m_pGrB;     //General interfaces for building graphs
ICaptureGraphBuilder2 *  m_pBld;     //Specific interfaces for video capture

CoInitializeEx(NULL, COINIT_APARTMENTTHREADED);
        //required to initialise the Common Object Model (COM)
```

```
HRESULT hr = CoCreateInstance(CLSID_FilterGraph, NULL, CLSCTX_INPROC,
                    IID_IGraphBuilder, (void **)&m_pGrB);
        //create the IGraphBuilder interface

HRESULT hr =CoCreateInstance(CLSID_CaptureGraphBuilder2, NULL,
      CLSCTX_INPROC, IID_ICaptureGraphBuilder2,
      (void **)&m_pBld);
        //create the ICaptureGraphBuilder2 interface

m_pBld->SetFiltergraph(m_pGrB);
        //attach the interfaces
```

At this point, we have an accessible graph builder in m_pGrB and a Capture Graph Builder, m_pBld, which is a helper interface specifically for video and audio capture filter graphs.

Now each filter can be added by:

```
IBaseFilter * filter;
filter= CoCreateInstance(CLSID_VideoRenderer, 0, CLSCTX_INPROC_SERVER,
            IID_IBaseFilter, reinterpret_cast<void**>(&pF));
m_pGrB->AddFilter(filter,"short description");
```

Replacing <CLSID_VideoRenderer> with the GUID of the required filter.
The filters can be joined together by rendering a single output pin:

```
m_pBld->RenderStream(NULL, NULL, filter_pin, NULL, NULL);
```

Alternatively, the filters can be joined by manually connecting each output pin to an input pin:

```
pl1 = GetPin(filter1,"Output");
pl2 = GetPin(filter2,"Input");
m_pGrB->Connect(pl1,pl2);
```

This offers flexibility for creating filter graphs to meet specific purposes.

Templates

As an easy starting point, a template user application is included on the CD. This provides immediate entry to machine vision programming. The template provides access to both video capture device streams and reading from video media files. In addition, standard or custom filters may be added to the processing chain.

VC6:

From the CD folder *'Templates/VC6'*, copy file '*mvtemplateapp.axw'* to folder 'Program Files\Microsoft Visual Studio\Common\MSDev98\Template'.

Under *'File->New->Project'*, there is now a new template '*Machine Vision Application Appwizard'* which will create a working application.

VC7:

From the CD folder 'Templates/VC7', copy complete folder 'MVtemplateAppVC7' to folder 'Program Files\Microsoft Visual Studio .NET 2003\Vc7\VCWizards'.
Also copy single files 'MVtemplateAppVC7.vsdir' and 'MVtemplateAppVC7.vsz' to folder 'Program Files\Microsoft Visual Studio .NET 2003\Vc7\vcprojects'.

Under '*File->New Project'*, there is now a new template '*MVtemplateApp'* which will create a working application.

VB6/VB7:

From the CD Folder '*Templates/VB*', copy complete folder *MVAppVB* to any install path. Copy file *VBDXInterface.dll* to '*Windows/System32'* directory. Open

MVAppVB.vbp for a working application in Visual Basic. (Upgrade will commence automatically for VB7 and VB.NET)

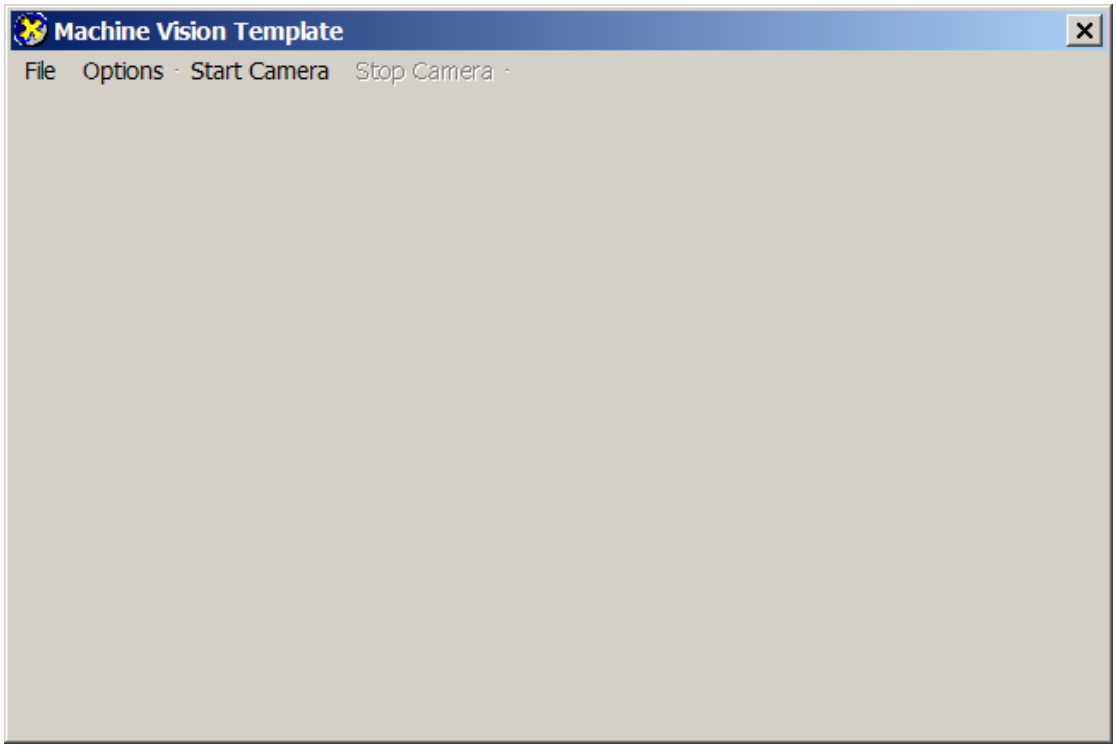


Figure 81 Example screen shot from the machine vision template application

This simple application will allow playback of files *(File->Open Media File),* or viewing output from a video capture device *(File->Select Capture Device).* In addition, a single process filter may be added to the filter graph *(Options->Select Filter).*

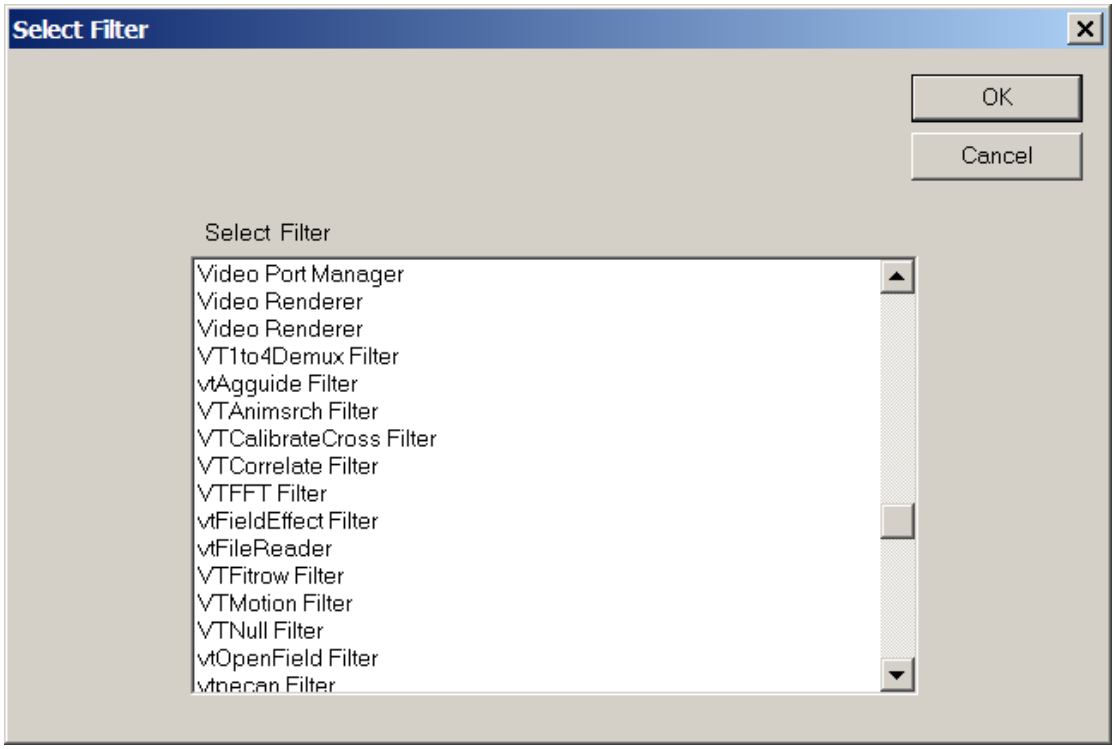


Figure 82 Select Filter dialog box.

All standard installed filters and user created filters are displayed for selection.

Using this tool, an Input-process-output filter graph may be created and run programmatically. Customisation from this point is simply a matter of extensions to the code provided.

MSDN documentation provides samples for tasks such as:

- Adding a file writer filter to capture the processed images to file;
- Adding a playback panel for fast forward, rewind, play and pause of media files, and
- Splitting into multiple streams for different processing.

Filter programming

For any but the simplest routines, customised filters will be required to process the video streams. Again a standard template has been created for immediate use and is included on the CD attached.

In a normal filter, code is required to implement the input pin, output pin and the handle the processing. The template provided, however, deals with all the background processing requirements and allows the user to focus on the image processing algorithms.

There are only two functions provided and required in file <*projectname*>.*cpp*:
InitVars: This is called once at the creation of the filter. Memory areas and global variables may be created and initialised.
TransformFrame: This is where the processing is carried out. This function is called once for each frame that passes through, regardless of whether the source is providing 1000 frames per second, or 1 frame per day.

Pointers are provided to the memory location at the start of the input frame and the output frame. To simply copy the frame, the user can loop through each pixel and copy the values from input to output.

Once the source project has been compiled, there will be a filter called <*projectname*> available to add to filter graphs, either in Graphedit, or user applications.

In addition to the template file, there are a number of example filters provided, to do such processing as colour channel separation, colour segmentation, edge detection and more. The details are listed in Appendix A.

Templates

As an easy starting point, a template user filter is included on the CD.

VC6:

From the CD folder '*Templates/VC6'*, copy file '*mvtemplatefilter.axw'* to folder 'Program Files\Microsoft Visual Studio\Common\MSDev98\Template'.

Under *File->New Project*, there is now an option for a new '*MV Template Filter'*

VC7:

From the CD folder 'Templates/VC7', copy complete folder 'MVtemplateFilterVC7' to folder 'Program Files\Microsoft Visual Studio .NET 2003\Vc7\VCWizards'.
Also copy the individual files 'MVtemplateFilterVC7.vsdir' and 'MVtemplateFilterVC7.vsz' to folder 'Program Files\Microsoft Visual Studio .NET 2003\Vc7\vcprojects'.

Under *File->New Project*, there is now an option for a new '*MV Template Filter'*

VB6/VB7:

Functionality is not available in these languages.

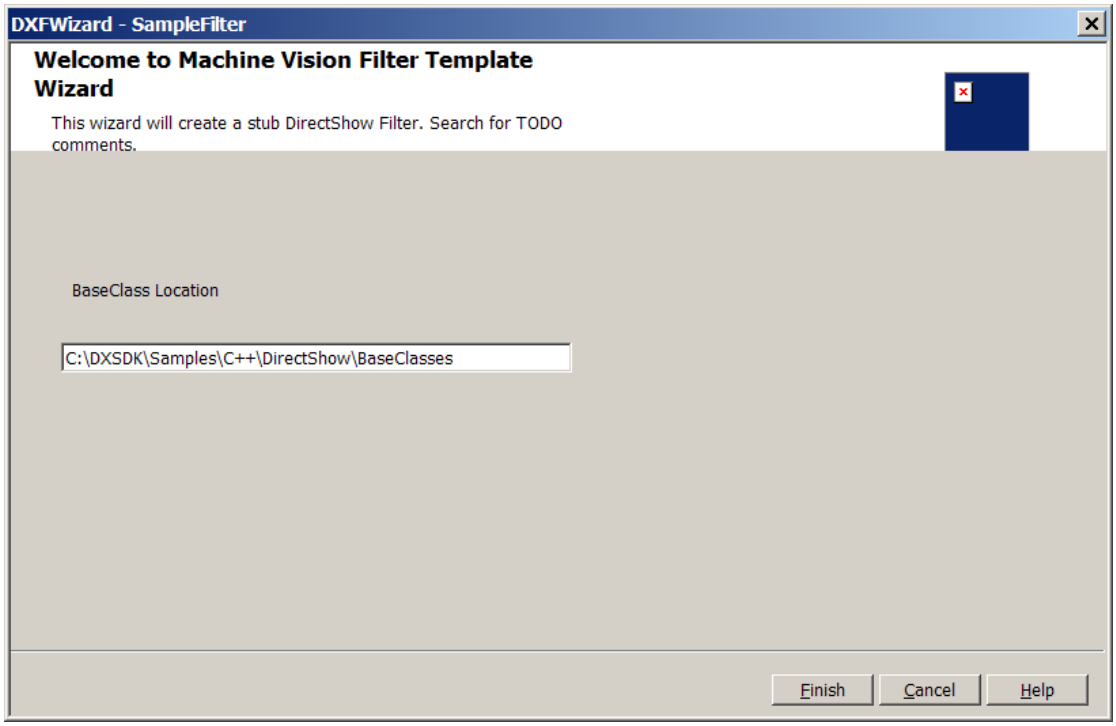


Figure 83 New Filter Wizard.

Appendix C: ROC Hardware Design

This section contains the detailed schematic drawings, PCB layout diagrams and Parts list for the ROC.

Schematic diagrams

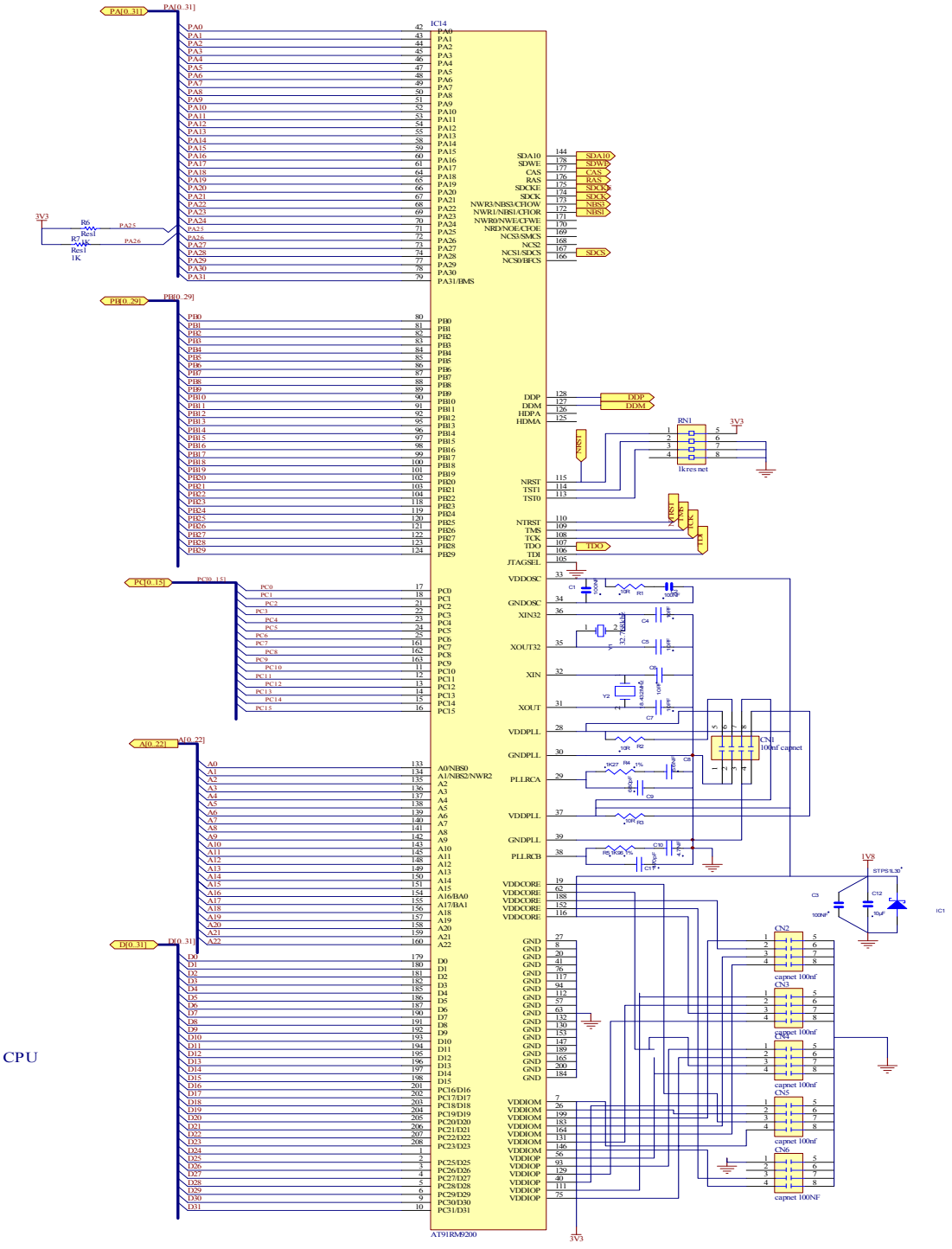
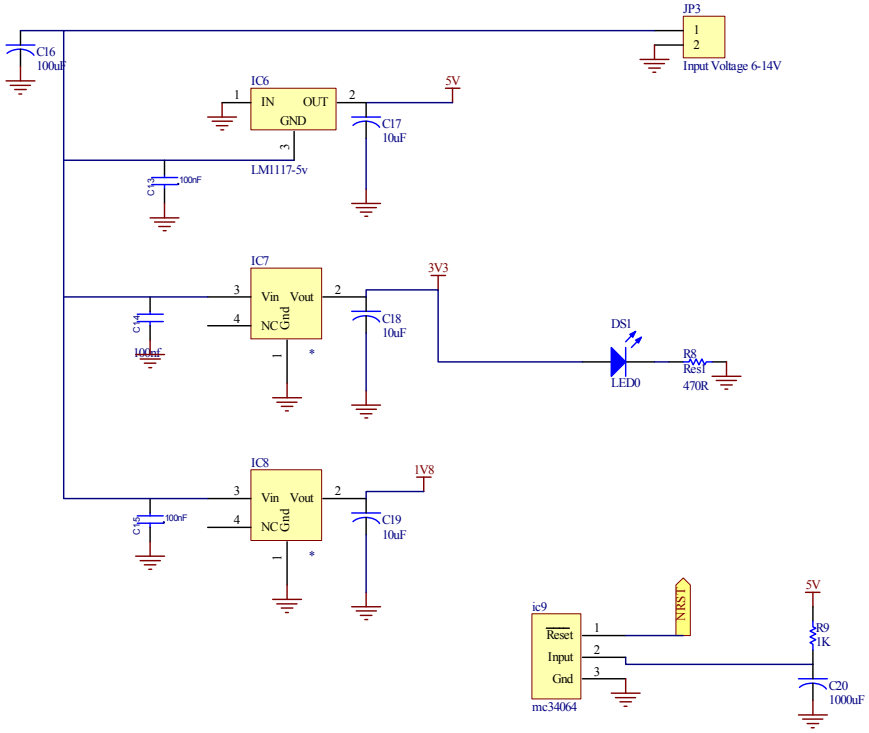


Figure 84 Processor schematic diagram

The processor block contains the central processing unit and immediate support. The main connections are the Data and Address bus (D and A), the Peripheral Buses (PA, PB and PC) and the power supplies (VDDIOM, VDDIOP, and VDDCORE). To generate the correct frequency for the clock, the Phase Lock Loop (PLL) components are required (PLL, XIN, XOUT). The input frequency is 18.432 MHz and the PLL system can generate any frequency up to 200MHz using a combination of software multipliers and dividers.



Power

Figure 85. Power supply schematic

This section generates the three different power levels required by the various part of the circuit. With an acceptable input voltage range of between 6 and 16 Volts, output voltages of 5, 3.3 and 1.8 Volts are produced by low-dropout switching regulators.

IC9 is a low voltage regulator that holds the CPU in reset until the input voltage has reached acceptable levels.

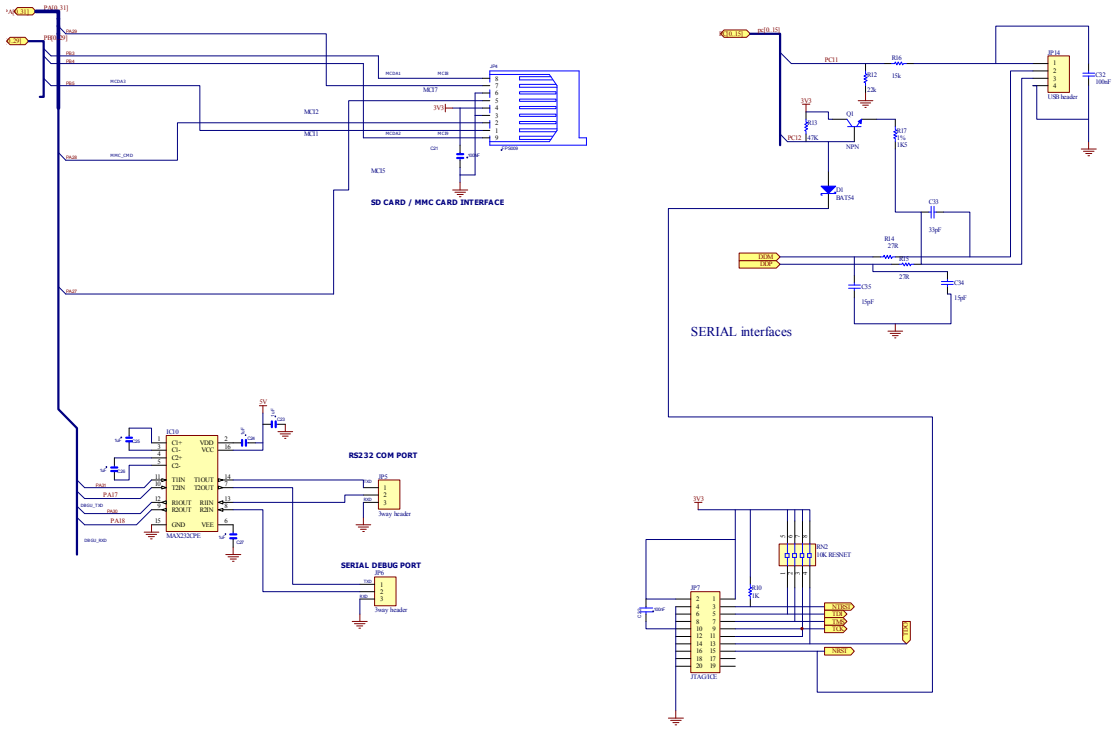


Figure 86 Serial interface schematic

This section of the design contains the serial input and output interfaces. There are two Universal Serial Asynchronous Receivers/Transmitters (USARTs) with RS232 output levels to interconnect to PC's and other serial devices.

The Joint Test Action Group (JTAG) interface is a means of debugging and downloading code to the system.

A USB device interface is provided to connect the unit to any USB host.

The Secure Digital (SD) card interface adds mass storage capabilities.

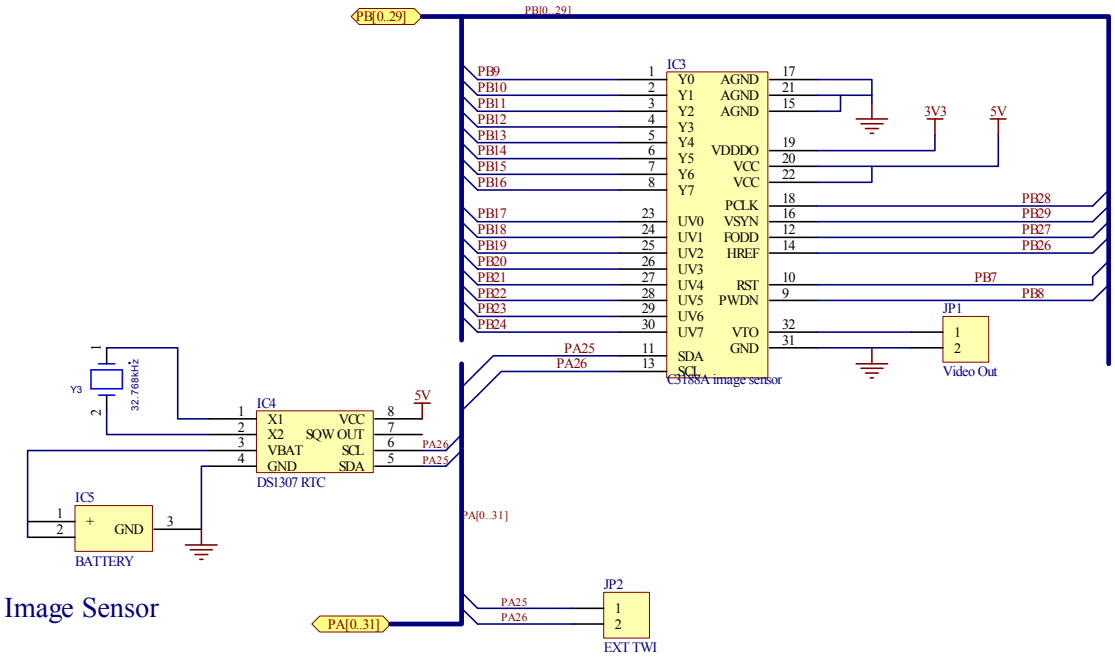
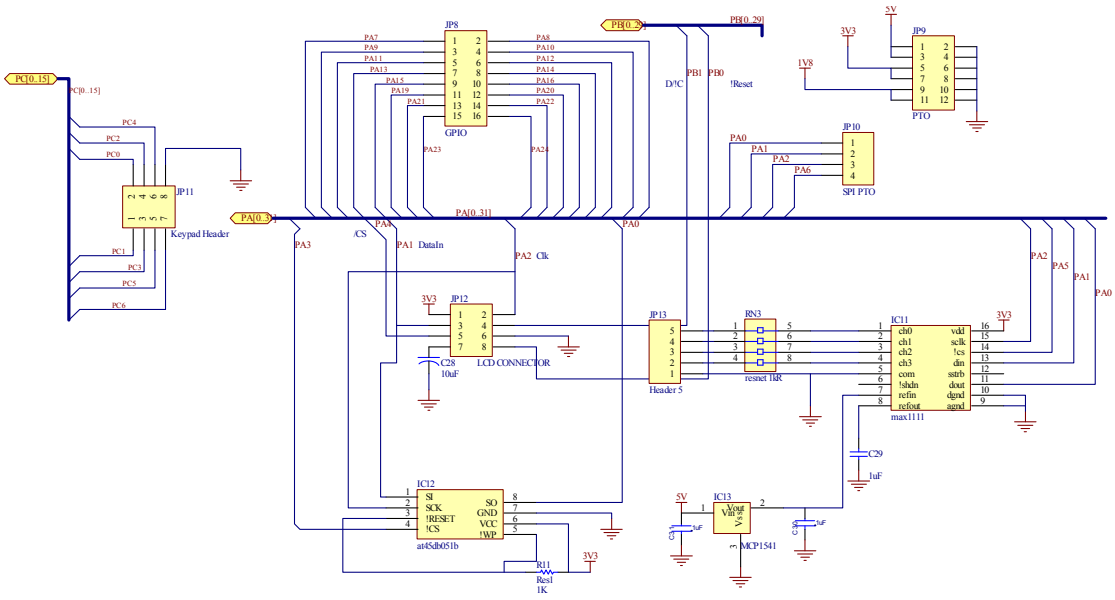


Figure 87 Image Sensor schematic

The image sensor, based on OV7620 sensor chips from Omnivision, is connected to specific peripheral interface pins. Control of the camera is provided by Two Wire Interface (TWI). This module provides analogue video out signals as a preview of the video stream.

Any Image sensor could be used for this device, including newer mega pixel embedded cameras similar to those emerging in mobile phones.

A Real Time clock is added to the system with battery backup to provide up to 10 years of real time functionality. This unit also contains 50 bytes of data storage for serial numbers and other initialisation data.



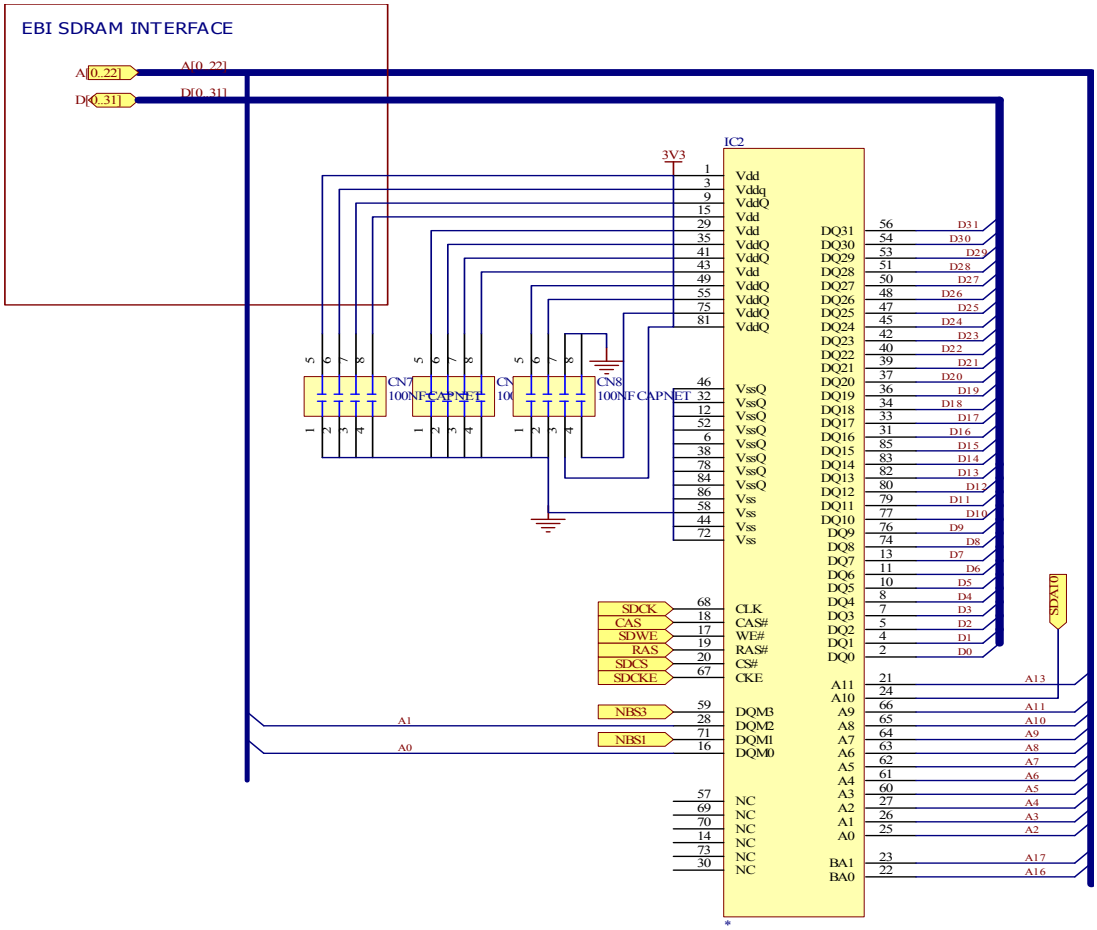
LCD screen + GPIO headers

Figure 88 Interface schematic

This unit provides a number of general input and output interfaces, including keypad, LCD screen, and a 4 channel Analogue to Digital (ADC) converter.

Facility to use 1.8, 3.3 and 5V voltages on external devices is provided.

A 500kB Dataflash (AT45DB051B) is provided as code storage. On reset, bootloader code is downloaded from the first 24 sectors of this device. The bootloader initialises the hardware and downloads the operating program for various other areas in the Dataflash chip.



Memory

Figure 89 SDRAM schematic

An external 16MB SDRAM memory is attached to the system. This chip runs at 100MHz, providing fast volatile data storage.

Printed circuit board layout

The circuit board has been designed as a 4 layer board. Most components are surface mount, providing a low profile device. Most signal and power tracks are run on the

top and bottom layers, with the central 2 layers as ground plane for signal shielding purposes.

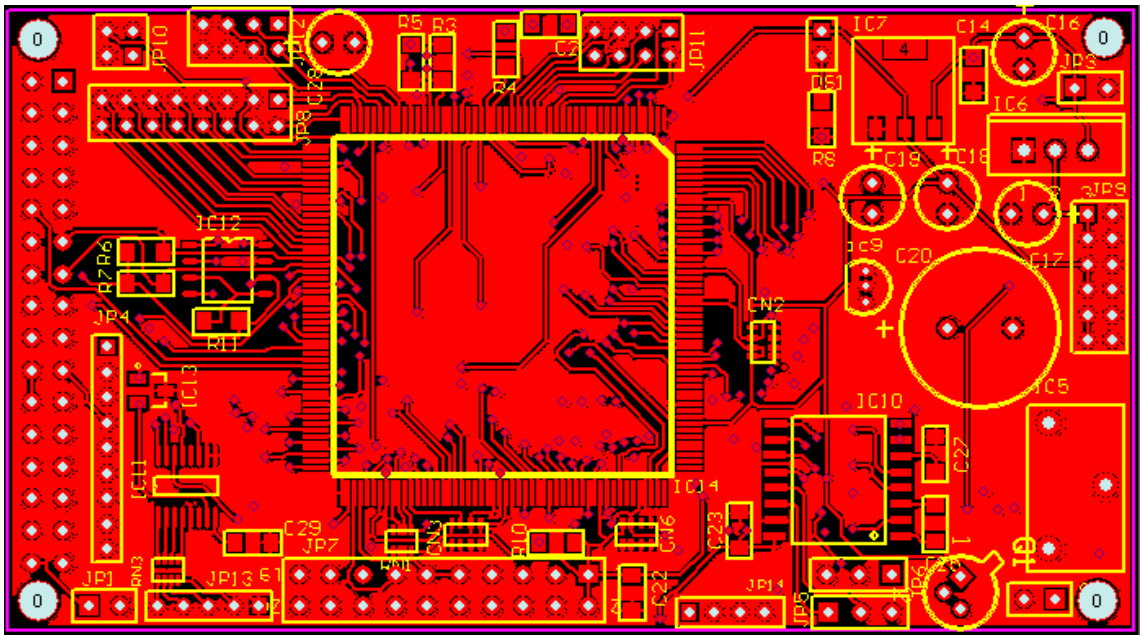


Figure 90 PCB Top Layer

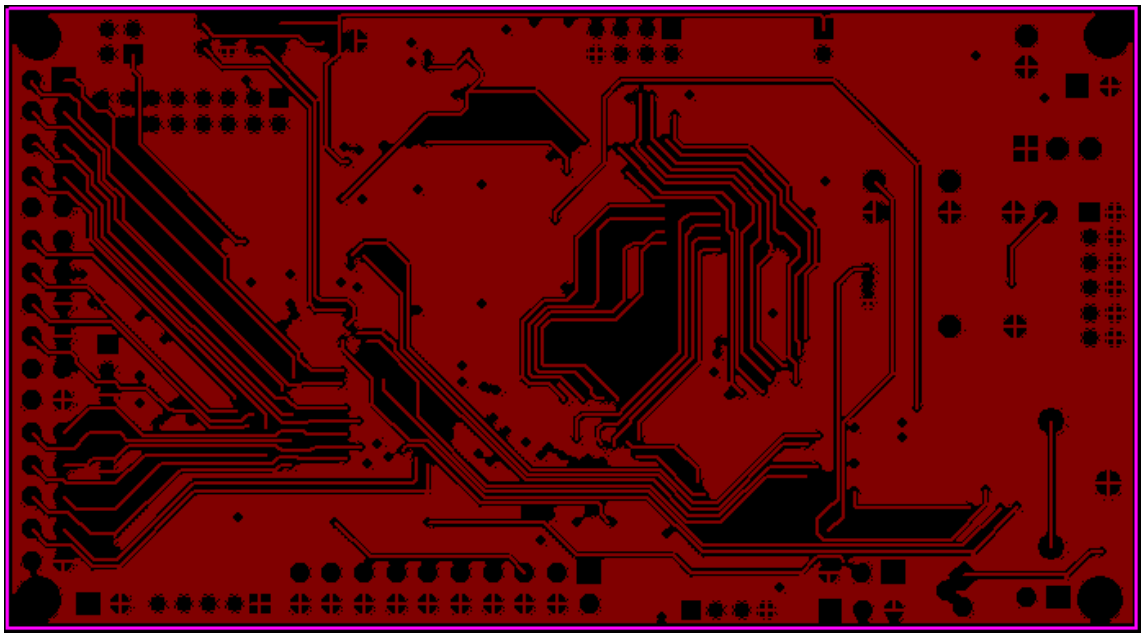


Figure 91 PCB Mid Layer 1

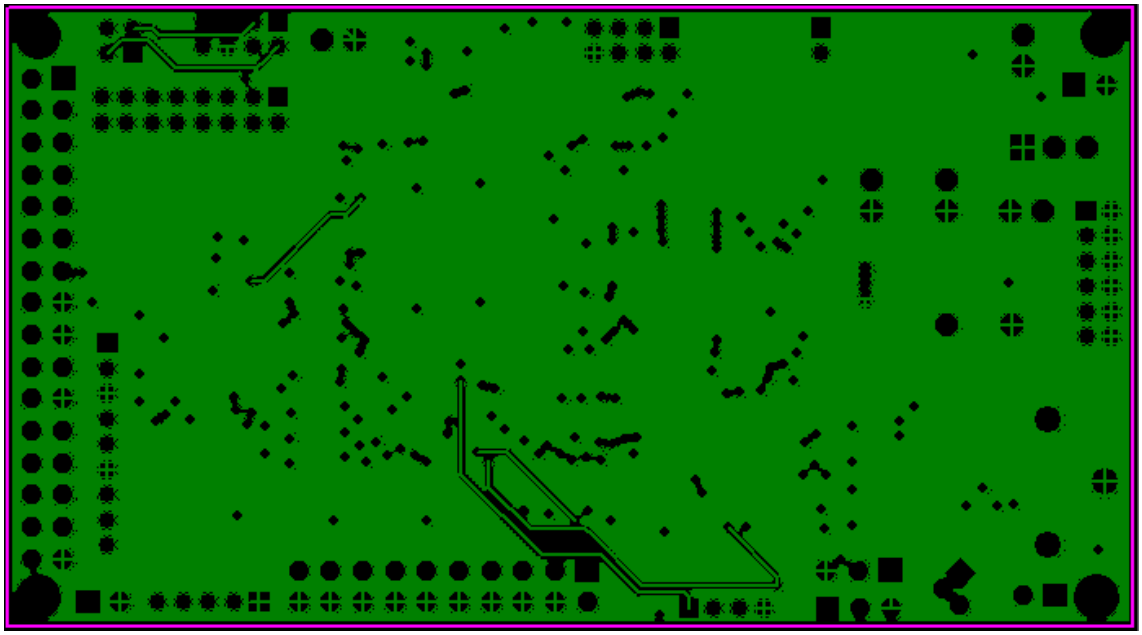


Figure 92 PCB Mid Layer 2

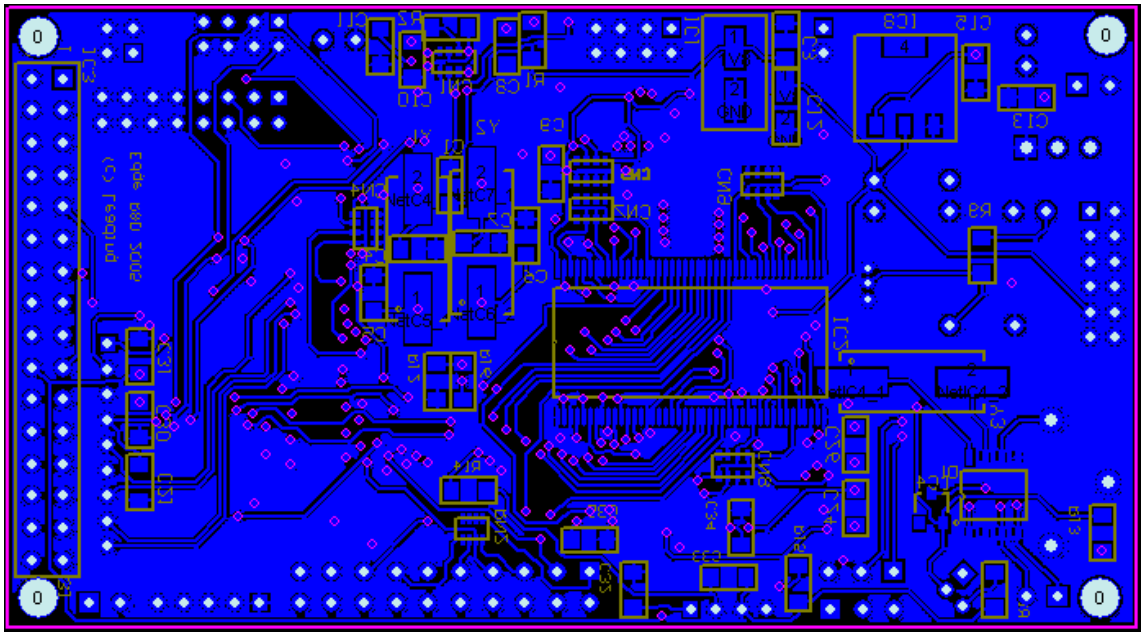


Figure 93 PCB Bottom Layer

Parts list

Table 11 ROC Parts List

Type	Board Designator	Footprint	Qty on Board	Value
Capacitor	C1/2/3/13/14/15/21	0805	7	100nF
Capacitor	C10	0805	1	4.7nF
Capacitor	C11	0805	1	470pF
Capacitor	C12	smt cap	1	10µF
Capacitor	C16	RB0.1/0.2	1	100µF
Capacitor	C17/18/19	RB0.1/0.2	3	10µF
Capacitor	C20	Rb0.2/0.5	1	1000µF
Capacitor	C23/24/25/26/27	0805	5	1µF
Capacitor	C4/5/6/7	0805	4	10pF
Capacitor	C8	0805	1	5.6nF
Capacitor	C9	0805	1	680pF
Capacitor Network	CN1/2/3/4/5/6/7/8/9	0612	9	100nf
STPS1L30U shottky diode	IC1	jedec do-214ac	1	
MAX232CWE	IC10	SOIC16	1	
at45db051b	IC12	SOIC8 6.2MM	1	
AT91RM9200	IC14	PFQP208	1	
MT48lC8M16A2TG-7E memory 128Mb	IC2	TSOP86	1	
C3188A image sensor	IC3	HDR2X16	1	
DS1307 RTC	IC4	SOIC8 6.2MM	1	
CR2032SLF battery	IC5	BATTERY	1	
LM1117-5v	IC6	T03B	1	
lm1117-3v3	IC7	SOT223	1	
lm1117-1v8	IC8	SOT223	1	
mc34064	IC9	TO226	1	
MMC-SD card socket	JP4	2MM9WAY	1	

Resistor	R1/2/3	0805	3	10R
Resistor	R12	0805	1	22K
Resistor	R14	0805	2	27R
Resistor	R4	0805	1	1K27
Resistor	R5	0805	1	1K96
Resistor	R6/7/11	0805	3	1K
Resistor	R9	0805	1	390R
Resistor network	RN1	0804	1	1K
Crystal 32.768kHz	Y1	HC49/4H_SMX	2	32.768khz
Crystal 18.432MHz	Y2	HC49/4H_SMX	1	18.432MHz

Appendix D: Animal Identification Project Data

The ground truth system is a custom developed application for a user to manually identify animals in a video.

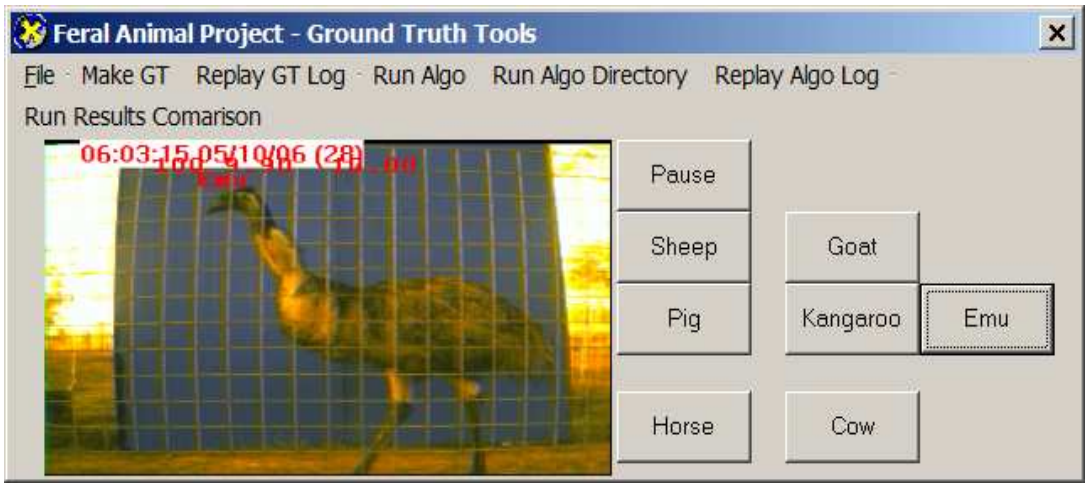


Figure 94 Example animal species groundtruth tool

As an animal passes across the screen, the user selects the animal species and follows the animal with the cursor. The ground truth tool logs the frame number, cursor position, and species to a file. This data can be replayed at any later time for verification purposes.

This tool will also apply the animal identification algorithms to a file or directory of files, allowing batch processing after any changes to the algorithms. This data is also logged to a comma separated file for comparison purposes.

The results comparison is run on a frame by frame basis. The ground truthed species is compared against the algorithm identified species, and logged to the results file. An example of each file is presented below.

Ground truth file

Frame	x	y	Species
701	137	146	Goat
702	104	157	Goat
703	112	157	Goat
704	115	157	Goat
705	118	157	Goat
706	124	157	Goat
707	127	157	Goat
708	129	157	Goat
709	132	157	Goat
710	136	157	Goat
711	138	157	Goat
712	140	157	Goat
713	146	157	Goat
714	149	157	Goat
715	152	157	Goat
716	159	157	Goat
717	161	157	Goat
718	163	157	Goat
719	167	157	Goat
720	170	157	Goat
721	173	157	Goat
722	176	157	Goat
723	178	156	Goat
724	180	156	Goat
725	181	155	Goat
726	182	154	Goat
727	185	154	Goat
728	189	154	Goat
729	191	154	Goat
730	195	154	Goat
731	198	154	Goat
732	202	154	Goat
733	206	154	Goat
734	207	154	Goat
735	213	154	Goat
736	214	153	Goat
737	216	153	Goat
738	216	153	Goat
739	218	153	Goat
740	220	153	Goat
741	221	153	Goat
742	223	153	Goat
743	225	152	Goat
744	230	152	Goat
745	233	152	Goat
746	235	152	Goat
747	235	152	Goat
748	237	152	Goat
749	238	152	Goat
750	238	152	Goat
751	240	152	Goat
752	241	152	Goat
753	241	152	Goat
754	242	152	Goat
755	245	152	Goat
756	246	152	Goat
757	246	152	Goat
758	248	152	Goat
759	248	152	Goat
760	248	152	Goat
761	250	152	Goat
762	250	152	Goat

763	250	152	Goat
764	252	152	Goat
765	257	152	Goat
766	259	152	Goat
767	262	152	Goat
768	262	152	Goat
769	263	152	Goat
770	263	152	Goat
771	263	152	Goat
772	264	152	Goat
773	264	152	Goat
774	264	152	Goat
775	264	152	Goat
776	264	152	Goat
777	264	152	Goat
778	265	152	Goat
779	267	152	Goat
780	267	152	Goat
781	268	152	Goat
782	268	152	Goat
783	270	152	Goat
784	270	152	Goat
785	270	152	Goat
786	270	151	Goat
787	271	151	Goat
788	271	151	Goat
789	272	151	Goat
790	272	150	Goat
791	273	150	Goat
792	274	150	Goat
793	275	150	Goat
794	275	150	Goat
795	275	150	Goat
796	275	150	Goat
797	275	150	Goat
798	275	150	Goat
799	275	149	Goat
800	275	149	Goat
801	275	149	Goat
802	275	149	Goat
803	279	149	Goat
804	286	149	Goat
805	287	149	Goat
806	288	149	Goat
807	296	150	Goat
808	298	150	Goat
809	300	150	Goat
885	300	150	Goat
886	55	164	Goat
887	100	167	Goat
888	125	168	Goat
889	125	168	Goat
890	129	168	Goat
891	140	169	Goat
892	170	171	Goat
893	172	171	Goat
894	182	169	Goat
895	189	168	Goat
896	198	164	Goat
897	205	162	Goat
898	211	162	Goat
899	219	162	Goat
900	230	162	Goat
901	241	162	Goat

902	256	163	Goat
903	265	163	Goat
904	275	163	Goat
905	285	163	Goat
906	294	164	Goat
907	303	164	Goat
908	308	164	Goat
909	314	164	Goat
910	319	164	Goat
915	164	173	Goat
916	196	177	Goat
917	201	177	Goat
918	211	177	Goat
919	216	177	Goat
920	225	177	Goat
921	232	177	Goat
922	239	177	Goat
923	243	177	Goat
924	245	177	Goat
925	250	177	Goat
926	253	177	Goat
927	255	177	Goat
928	258	177	Goat
929	262	177	Goat
930	269	177	Goat
931	277	177	Goat
932	285	177	Goat
933	295	177	Goat
934	310	175	Goat
935	316	175	Goat
936	317	175	Goat
937	317	175	Goat
938	318	175	Goat
939	318	175	Goat
940	319	175	Goat
941	320	175	Goat

Algorithm Identified File

Frame	x	y	Species
704	94	178	Goat
705	101	177	Goat
706	103	178	Goat
707	106	177	Goat
708	108	177	Goat
709	111	177	Goat
710	116	177	Goat
711	119	177	Goat
712	128	177	Goat
713	124	177	Goat
714	125	176	Goat
715	127	177	Goat
716	131	177	Goat
717	136	177	Goat
718	139	177	Goat
719	140	177	Goat
720	143	177	Goat
721	149	176	Goat
722	149	176	Goat
723	152	177	Goat
724	154	177	Goat
725	157	177	Goat

726	165	177	Goat
727	163	176	Goat
728	166	176	Goat
729	175	176	Goat
730	171	176	Goat
731	174	175	Goat
732	182	176	Goat
733	181	177	Goat
734	185	177	Goat
735	184	177	Goat
736	184	176	Goat
737	188	178	Goat
738	193	177	Goat
739	194	177	Goat
740	195	178	Goat
741	195	178	Goat
742	198	177	Goat
884	77	185	Goat
885	88	186	Goat
886	101	186	Goat
887	109	188	Goat
888	118	190	Goat
889	126	190	Goat
890	134	190	Goat
891	145	188	Goat
892	153	188	Goat
893	163	188	Goat
894	174	186	Goat
895	183	186	Goat
896	192	186	Goat
897	202	186	Goat
905	74	191	Sheep
906	85	192	Sheep
907	99	190	Sheep
908	109	189	Sheep
909	121	188	Sheep
910	131	189	Sheep
911	141	191	Sheep
912	148	191	Goat
913	155	192	Goat
914	164	192	Goat
915	174	190	Goat
916	185	191	Goat
917	191	190	Goat
919	208	190	Goat

Results File

C:\GT Animals\dec2\2005_12_2_7_39_20_fgt0.csv						
frame start	frame fin	real species	error species	correct	wrong	missed
701	809	Goat		1	0	0
885	910	Goat	Sheep	1	1	0
915	941	Goat		1	0	0

In this way, 91 files have been processed, with a total of 1287 single animal events to produce the results in Tables 8, 9 and 10.

The log files accumulating this data are on the CD under the folder */Case Study Results/Animal Species Identification.*