MACHINE VISION FOR THE GRADING OF PINE SEEDLINGS

by

MICHAEL PATRICK RIGNEY

Bachelor of Science in Agricultural Engineering

Oklahoma State University

Stillwater, Oklahoma

1985

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the degree of
MASTER OF SCIENCE
December, 1986

MACHINE VISION FOR THE GRADING

OF PINE SEEDLINGS

Thesis Approved:

Glenn Kranzler
Thesis Adviser

Marvin L. Stone

Gerald H. Brusewitz

Norman N. Durham
Dean of the Graduate College

PREFACE

The author wishes to acknowledge the following persons and organizations for the role they played in facilitating this study:

The Department of Agricultural Engineering of Oklahoma State University for providing a research assistantship.

My graduate committee members, Dr. Marvin Stone and Dr. Gerald Bruzewitz for their time and efforts.

My graduate adviser, Dr. Glenn Kranzler, for his gracious support and guidance.

Mr. Douglas Devoe and Mr. Mark Appleman for their significant contributions of time and energy.

My parents Sylvester and Denise Rigney and my brothers; Mark, Steve, Philip, and Lloyd for their infinite love and support.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

The commercial forest industry, along with state and federal agencies, produces hundreds of millions of tree seedlings annually. These seedlings are vital to the reforestation effort which is necessary to ensure future supplies of lumber, paper, and other forest products.

One of the early stages in reforestation is the culture of tree seedlings (Fig. 1). Nursery managers perform many cultural operations to improve the productive potential of the stock grown in nursery beds. Quality pine seedlings are currently valued at $35 per thousand, representing a 2.5 million dollar crop annually for a single Oklahoma nursery. Grading harvested seedlings to remove inferior stock is an important management procedure.

Grading is currently performed manually in an environment that is cold and humid. It is not feasible for human graders to inspect every seedling or to grade seedlings into more than two classes. Grading performance varies widely among graders. Seedling throughput per grader is low, with the average grader processing only 3000 - 3500 seedlings per hour. Research by Lawyer (1981) indicates that grading and sorting account for 19% of total labor cost for a typical nursery. These facts indicate a need for automation of the seedling grading operation.

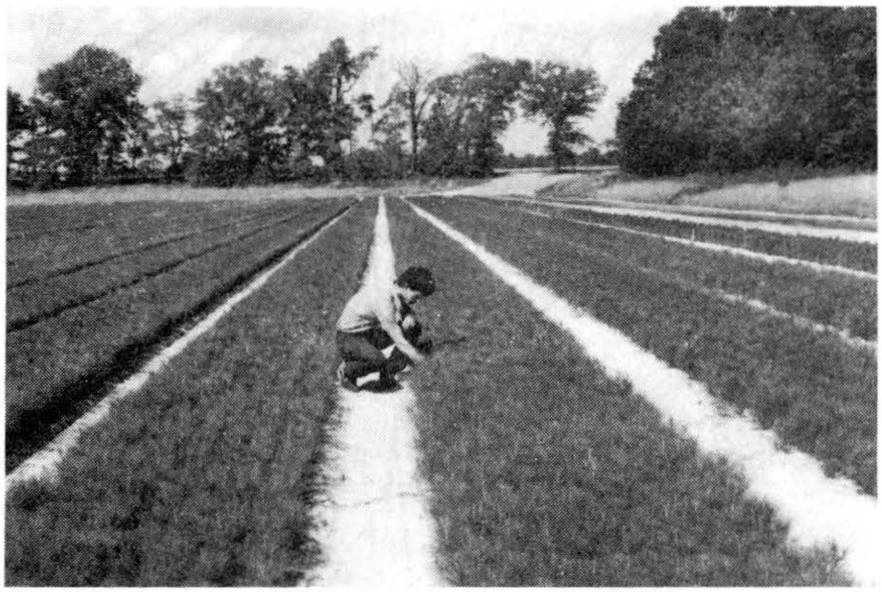


Figure 1. Loblolly Pine Seedlings at the Weyerhaeuser Nursery, Ft. Towson, Oklahoma

Digital image processing has been successfully implemented in many industrial and some agricultural inspection processes. It appears to be an ideal tool for addressing the seedling grading problem. Digital image processing systems have demonstrated high accuracy and throughput and have permitted 100% inspection of products where it was previously not feasible. Beyond the improvement of the grading process, this tool could contribute to the knowledge base of silviculture and increase the productivity of our forests. This result could be realized if nursery managers had the ability to provide seedlings of a prescribed grade for a specific site.

Objectives

The purpose of this research is to demonstrate the ability of machine vision to grade harvested pine seedlings under commercial nursery production conditions. To this end, three specific objectives are adopted:

1. Define a seedling grade classification scheme based on appropriate seedling measurements.
2. Develop and implement a machine vision algorithm for obtaining seedling measurements in real time (here defined as at least one seedling per second.)
3. Evaluate the performance of the algorithm implementation in terms of measurement precision, speed and accuracy of classification, and causes of misclassification.

Assumptions

The machine vision environment will be defined in terms of required lighting, cameras, optics, machine vision system, and seedling transport. Several assumptions have been made about the environment in which the grading algorithm will be employed. Constraints were necessary to reduce the scope of this study to a reasonable breadth.

The first assumption is that only one seedling will appear within the camera field-of-view (FOV) at a given time. It would be possible to grade the seedlings if several were present, however, occlusion would pose a significant problem. The simplest case is to inspect one seedling at a time. This requires that a mechanism for singulating the seedlings be implemented in a commercial application. Manual singulation is used in this study.

A second assumption is that the orientation of the seedling and position of the root collar are constrained. Since seedlings are lifted from the nursery bed with uniform orientation, it is assumed that all seedlings will have the same orientation when they are viewed by the cameras. Another aspect of orientation is the angle of the major axis with respect to the image axes. The singulation technique will be assumed to constrain angular variation. Additionally, root collars are at ground level when the seedlings are lifted. It is assumed that root collars will be minimally displaced relative to each other when they pass beneath the cameras.

These assumptions are consistent with the operation of mechanically lifting seedlings from the nursery bed. For this reason, the automated grading process may best be implemented on the seedling lifter itself. If the seedlings are to be graded in the grading shed (as assumed), mechanical processes may be implemented, or more cameras may be required to meet these assumptions.

Another assumption is that a non-reflective black conveyor belt is used to transport the seedlings under the cameras. This measure is necessary to provide high contrast between the seedling and background, which simplifies image processing and improves measurement accuracy.

Loblolly pine is one of the major tree species used by the commercial forest industry in the southern United States. Because of the morphological differences between species of pine seedlings, this study was limited to the grading of loblolly pine seedlings. It is anticipated that the algorithm developed here could be adapted to the grading of other pine species.

CHAPTER II

REVIEW OF LITERATURE

Introduction

The task of grading pine seedlings with machine vision is a marriage of two major fields of study. The first involves nursery management and seedling grading criteria. In the first part of this chapter the criteria for grading pine seedlings are presented. A review of previous work in the mechanization of sorting and grading seedlings follows. Finally, previous applications of machine vision in inspection and grading processes are presented.

The second part of this chapter considers the field of digital image processing and machine vision. Castleman (1979) defines digital image processing as, "subjecting numerical representations of objects to a series of operations in order to obtain a desired result." The numerical representation is further defined as, "a sampled, quantitized function of two dimensions which has been generated by optical means, sampled in an equally spaced rectangular grid pattern, and quantitized in equal intervals of grey level." Machine vision has been described as the, "implementation of the pattern recognition process for the interpretation of visual data" (Valenty and Kraska, 1984), and "the ability (of computers) to monitor and control visual information" (Preston and Molinari, 1986). In the second part of this chapter, the general techniques of machine vision are discussed, with particular emphasis on those relevant to the task of grading pine seedlings.

Seedling Grading Criteria

Nursery managers perform many cultural operations to increase the quality of stock grown in nursery beds. These include control of seedbed density, irrigation, fertilization, fumigation, undercutting, top pruning, and wrenching (Duryea and Landis, 1984). The final operations performed at most nurseries are lifting, grading, and packaging the seedlings.

In current practice, seedlings are mechanically lifted from the beds and transported to a grading shed in large containers (Beckman, 1986). Timely processing is important to reduce the stress of root exposure. Grading sheds are maintained at low temperature and high humidity to further reduce seedling stress, however, this is an uncomfortable environment for grading personnel, making automation desirable. Graders grasp seedlings from a conveyor belt, remove culls by applying a number of visual grading criteria, and place acceptable seedlings on another conveyor for packaging.

Seedlings may be graded according to physiological and morphological characteristics (Forward, 1982; Duryea and Landis, 1984). Physiological characteristics include root growth capacity, frost hardiness, stress resistance, carbohydrate level, bud dormancy, degree of cold hardening, and nutrient levels in the tissues. Root growth capacity, frost hardiness and stress resistance are performance attributes and may be assessed by evaluating seedling response in an environmental control chamber. Seedlings must be destroyed to assess the remaining physiological characteristics which are material attributes. Physiological characteristics are valuable indicators of seedling quality, however, they are difficult and time-consuming to determine. Seedlings may already be planted before evaluation of physiological characteristics is complete.

Because assessment of physiological characteristics is difficult, morphological characteristics are used in the grading of most nursery stock (Forward, 1982). These characteristics include shoot height and weight, root weight or volume, root fibrosity, stem caliper at the root collar, foliage color, presence of terminal buds, root/shoot volume ratio, and ratio of top height to stem caliper (sturdiness ratio) (Fig. 2). Stem caliper, root volume, shoot height, and root/shoot ratio, are considered the most important morphological characteristics (Forward, 1982). The importance of morphological grades has been shown by Wakeley (1969). In a thirty-year study, grade 1 loblolly seedlings produced twice as much wood volume as grade 3 seedlings.

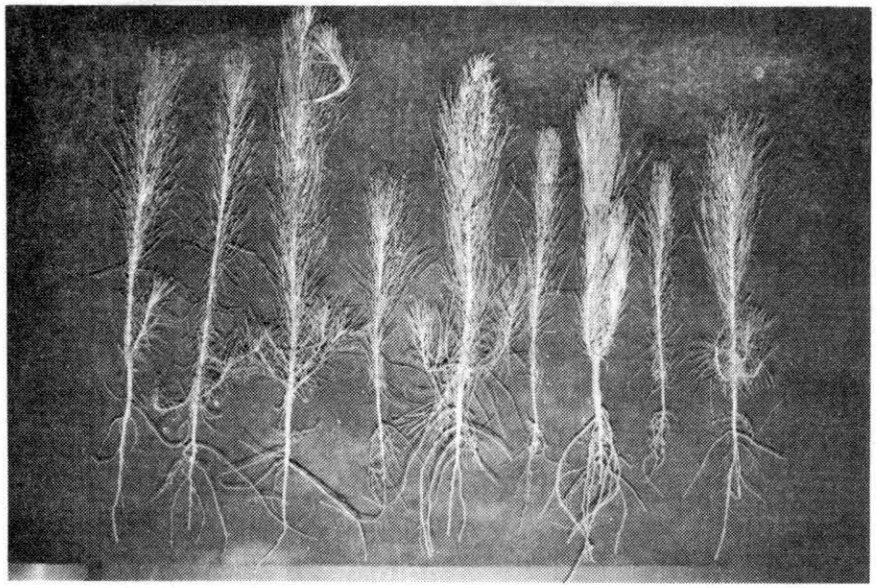


Figure 2. Variation among Loblolly Seedlings

Table 1 is an agglomeration of grading criteria from several sources. There is no one set of criteria for all loblolly seedlings. In fact, different criteria may be specified for different geographical regions and planting sites. Special criteria are sometimes specified by nursery customers, but in general, manually grading seedlings into more than two classes (good and cull) is not practical. An automated grading system would have the capability of grading into several classes, suitable for

TABLE I

GRADING CRITERIA FOR LOBLOLLY PINE SEEDLINGS

SOURCE	Beckman (1986)	May (1982)		Wakeley (1969)			Weaver (1981)	
GRADE	min.	min.	opt.	1	2	3	industry min.	state min.
CRITERION								
Stem Caliper (mm)	3	3.2	5.5	4.8-7.9	3.2-4.8	<3.2	4.3	3.0
Shoot Height (cm)	11	13	25	23-30	15-25	8-30	17	18
Root Laterals (#)	6		30	30+	20+	<15		
Root Length (cm)		12	15				18	12
Root/Shoot Ratio (volume)		0.33	0.40				1.0	0.66

different planting sites.

Seedling Sorting and Grading Mechanization

A digital system for measurement and recording of tree seedling height, stem caliper, root mass area index, and sample number has been described by Buckley et al. (1978). Seedling height and caliper were measured with potentiometric transducers, while root mass area index (root silhouette area) was measured with a moving, 1024 element photoelectric linear array. Accuracy of the area scanner was determined using opaque wires and rectangles of known dimensions. The system required an operator to open the area sensor cover, place the root collar of a seedling in the caliper transducer, close the area sensor cover, position the stem height transducer, and press a button to initiate the area measurement. This apparatus was an improvement over manual measurement techniques, but is not suitable for grading large quantities of seedlings.

Maw et al. (1980) developed a system which sorted plant seedlings on the basis of height. This system required that the seedlings be singulated prior to introduction to the sorting machine. Seedlings were classified as good or cull on the basis of a length measurement made by a row of phototransistors. Cull seedlings were destroyed by a guillotine knife. This system was capable of sorting large numbers of seedlings, but assessed only one of several grading criteria.

A system was developed and tested in the laboratory which could automatically sort and feed pre-singulated and taped seedlings to a planting machine (Ardalan and Hassan, 1981). Two methods of sorting were studied, both of which measured the stem caliper of seedlings secured between two lengths of tape. One method used an opto-electronic emitter-detector pair and determined caliper as a function of seedling velocity and time of emitter blockage. The other system made use of a linear vertical potentiometer attached to a roller which was displaced in the presence of a seedling. Both systems provided satisfactory performance in measuring stem caliper, however, the required taping and transporting of cull seedlings to the planting site added unnecessary cost.

A mechanical pine seedling singulator was developed and tested, providing a 65% singulation success rate (Graham and Rohrbach, 1983). The system made use of a wedge shaped vacuum nozzle and a rotating triangular seedling hopper. The vacuum nozzle was designed to catch only one seedling by sucking it into the wedge where it would block the nozzle orifice. The seedling hopper rotated 1/3 revolution for each seedling selection to prevent bridging and root entanglement. The researchers determined that with two singulators working independently, a single seedling would be available for planting 95% of the time. Such an apparatus could also be used to singulate seedlings prior to automated grading.

Research has been conducted to assess various methods for detecting seedlings (Maw et al., 1985). The goal of the research was to improve greenhouse efficiency by automating seedling sorting. Leaf area, seedling multiplicity, and leaf color were specific items of interest. The use of fiber optics, photo transistors, and digital image processing were investigated. Fiber optics and digital image processing were found to be the most promising tools for acquisition of the needed information.

Other studies have applied opto-electronics to caliper measurement and counting of pine and other seedling plants (Kranzler et al., 1984; Sutton and McLendon, 1985; McLendon and Allison, 1986; Huang et al., 1986). Most techniques measured stem caliper as a function of sensor velocity and time of sensor blockage, though Huang achieved high accuracy with mechanical transport of opto-electronic sensors for detecting stem edges.

Machine Vision Applications

In the last decade there has been a trend to automate many agricultural and industrial inspection tasks through the use of machine vision. The technology has achieved both quality improvement and processing cost reduction. This section presents several inspection and grading applications of machine vision.

Automated apple classification with emphasis on bruise detection has been described by Taylor and Rehkugler (1985). Detection was based on the difference in infrared reflectance between normal and bruised tissue. The accuracy of the system was equivalent to human grading accuracy, but the speed of classification was limited by the image processing system used.

Sarkar and Wolfe (1985) describe algorithms for classification of fresh market tomatoes based on size, shape, color, and surface defects at the stem and blossom ends. Processing techniques included boundary chain coding and gradient transformations. An optical filter was used to aid in color discrimination.

Hines et al. (1986) describes a system for grading container grown horticultural plants. The system must be trained with a set of plants from each variety to be graded. Classification on a scale of 1 to 10 is based on features such as shape, size, symmetry, foliage density, and color.

Wolf and Sandler (1985) describe an algorithm for detecting stems attached to harvested fruit. The boundary chain code of the fruit was transformed into syntactic primitives which indicate the degree of concavity or convexity of small boundary segments. A stem is recognized as concave-convex-concave sequence, preceded and followed by uniform convex curvature of a lesser magnitude.

Meyer and Davison (1985) describe a machine vision system for measuring plant growth in the field or environmental chamber. Measurement of leaf axial dimensions and area, stem and petiole length, canopy closure, and stem diameter were all investigated. Diameter measurements were obtained with the stem magnified to at least 40% of the field of view. Performance was accurate, however care had to be exercised with lighting (shadows), plant positioning, and plant movement due to wind.

High inspection rates attainable with vision systems have been demonstrated in many applications. The inspection of bottlecaps is an excellent example (Schreiber, 1985). Zapata Industries' vision system can inspect 2600 bottlecaps per minute and is responsible for a 33% increase in productivity. The seal, central area, and circumferential flutes are inspected on the inside of each cap. Plans are to add exterior inspection. The system can be reconfigured in 30 seconds to inspect any of 6 different bottlecaps produced.

Cambier and Pasiak (1986) describe an automatic inspection system for packaged foods. Pulsed X-rays are used to detect glass and metal contaminants in jars and cans. Throughput is up to 900 containers per minute with a 95% probability of contaminant detection.

Several vision based inspection systems have been surveyed by Kranzler (1984). A vision based sorter classifies cucumbers into three grades and five sizes at a rate of up to 600 per minute. Up to 200 pizza crusts per minute are inspected for holes, foreign objects, burns, and shape defects. French fry strips are inspected for discoloration at rates of up to 151 kg (333 lb) per minute. Finally, up to 720 eggs per minute are inspected for broken yolks on a processing line which automatically separates the yolks from the albumen. These examples demonstrate the ability of machine vision to perform inspection at the high throughput rates required in food processing plants.

Image Processing Techniques

This section describes the image processing environment and processing techniques. Image processing can generally be divided into four steps. These are: 1) image acquisition, 2) segmentation of the object from the background, 3) measurement of features, and 4) making a decision based on these measurements.

Lighting

An important consideration in the image acquisition task is the design of scene illumination. Different lighting techniques are useful in the acquisition of different object features. Diffuse front lighting reduces specular reflection from the object and is preferred when texture, surface edges, or lettering are of interest. Backlighting provides a high-contrast image of the object silhouette, useful in recognition of object presence or absence, and dimensional measurement (Novini, 1986). Structured lighting, such as a laser line projector, allows measurements in the third dimension to be obtained through triangulation. Fiber optics can be used to direct intense light to specific locations.

The type of light source is another consideration in illumination specification. Incandescent sources have a peak energy output in the near-infrared, corresponding to the peak sensitivity of solid-state image sensors. Florescent lamps provide diffuse light with less heat (infrared) than incandescent lamps. Xenon tubes provide very intense strobed lighting that can "freeze" the motion of moving objects. The spectral content of a xenon flash is similar to that of daylight. Small light emitting diodes can also be strobed and are useful in illuminating small objects. Diffusers may be added to all of these light sources to achieve more uniform illumination and reduce specular reflection. X-rays are unique in their ability to differentially penetrate various substances, providing an image that conventional illumination cannot.

Optical filters can control the wavelengths of light illuminating the scene and/or reaching the camera. Paulsen and McClure (1985) suggests using an infrared blocking filter on the camera so that the light reaching the sensor is of the same wavelengths detected by the human eye. The infrared image will otherwise have a "washout" effect on the visible light image (Dunbar, 1986). Color filters can sometimes be used to increase contrast between subject and background. Mersch (1984) describes the use of polarizing filters for the elimination of specular reflections, minimization of diffuse reflection while preserving specular reflection, and increasing the contrast of translucent objects.

Image Acquisition

A scene to be acquired for image processing is focused with a lens onto a sensor. For a given image size (FOV), a variety of lenses is available to achieve a desired standoff (camera-to-subject distance). Typically, images are acquired with tube-type (vidicon) or solid-state image sensors. Tube-type cameras have been used in the television industry for years, but solid-state devices have recently been preferred for image processing, because their performance is not degraded by geometric distortion and lag. Image lag appears as a ghost of a bright object after it has moved, and results from electric charge remaining on the sensor after an image scan.

Solid-state sensors are available as linear or rectangular arrays containing from 64 to over one million picture elements (pixels). Photons absorbed by the sensor are converted to an electrical charge which is transferred from the camera in the RS-170 television format at 30 frames per second. Some specialized vision systems perform image acquisition at higher speeds by bypassing the RS-170 format.

The image must next be digitized before it can be stored or processed in a digital computer. The analog video signal entering the digitizer is converted into an array of pixels with discrete grey levels. A typical image with 256 lines (rows) and 256 columns of pixels, each having one of 256 grey levels, requires 64K bytes of memory for storage.

Sources of Error

Many possible sources of error are attributed to the image acquisition components of image processing systems (Tappan et al., 1986; Chu, 1986). Vision applications designers may exercise control through lighting design and choice of optics, however, a significant portion of system errors may be attributes of the sensor itself.

Lens optics may contribute several types of error (Doty, 1986). Optical aberrations cause fine detail to be reproduced with low contrast. The effect can be reduced by using a small aperture. Diffraction blurs a sharp thin edge and is most pronounced at small apertures. A low quality lens may introduce distortion or attenuation of the light hitting the edges of the sensor. Generally, lenses introduce less error than the solid-state sensors on which the image is focused.

Unless a very expensive, high quality solid-state device is used in the camera, it will have many "dead" pixels which are much less sensitive to light than the average pixel (Novini, 1985). Camera manufacturers assign defective pixels the intensity value of an adjacent pixel or the average of several adjacent pixels.

Pixel geometry also has an influence on accuracy. In some image sensors up to one-half of the imaging area is not sensitive to light, but used to transfer image data from the pixels. Often the pixels are rectangular, and sometimes alternate rows are shifted one-half pixel. This procedure enhances the image for television viewing, but is not desirable for machine vision applications.

After the image is acquired by the sensor, other errors may be introduced when it is digitized by the processing system. If the image pixel density is greater than the sensor pixel density, some adjacent image pixels will have come from the same sensor pixel. If the sensor has a higher pixel density, some of the resolution will be lost.

Generally, measurements made with digital systems are limited by the spatial resolution or sampling frequency. The Nyquist criterion states that the high frequency detail retained in an image is limited to one-half of the sampling frequency. Since edges are high frequency phenomena, we can expect that an edge location can, at best, be approximated to plus-or-minus one pixel and a length or diameter to within two pixels. Measurement precision may be improved, however, through averaging.

Segmentation

Segmentation of the subject from the background is the most difficult task in many image processing applications. In the simplest case, an object may be adequately segmented from its background with proper illumination. With the use of backlighting, an object appears as a silhouette on a white background. A histogram is a plot of the frequency distribution of the grey levels in an image (Baxes, 1984) (Fig. 3). The histogram of a backlit object would contain high numbers of light pixels (background) and dark pixels (object), but relatively low numbers of grey levels in between. Such a histogram is bimodal. A binary segmented image may be obtained by thresholding the image at a grey level between the two modes (the antimode). All pixels darker than the threshold are mapped to black and all pixels equal to or lighter than the threshold are mapped to white.

More frequently, the subject and the background in an image contain common grey levels, and edge detectors must be applied to locate object boundaries (Ballard and Brown, 1982). In such an image, edges appear as local areas characterized by a rapid change in grey level (Fig. 4). Edges are high frequency phenomena and may segmented through high-pass filtering with Fourier transforms. More commonly, edges are detected through

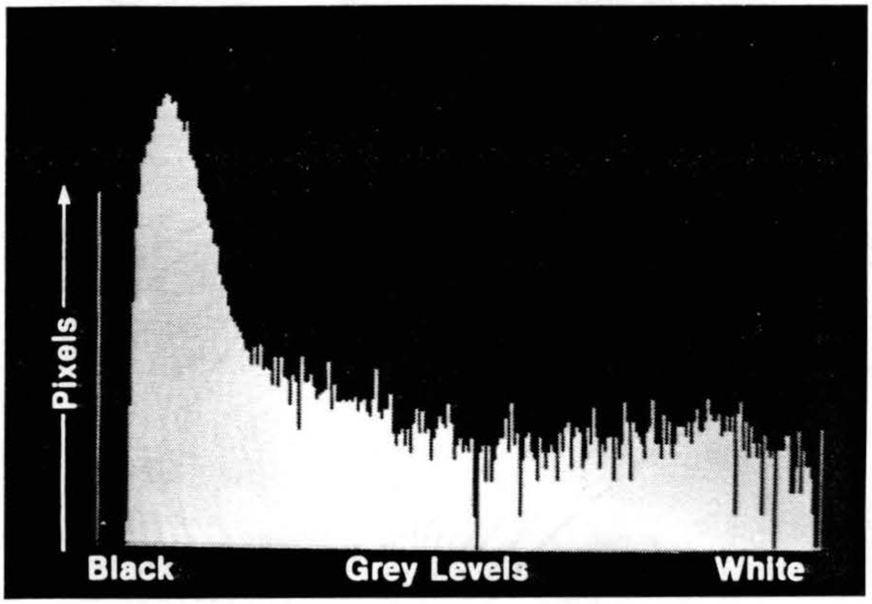


Figure 3. Histogram of Seedling Image

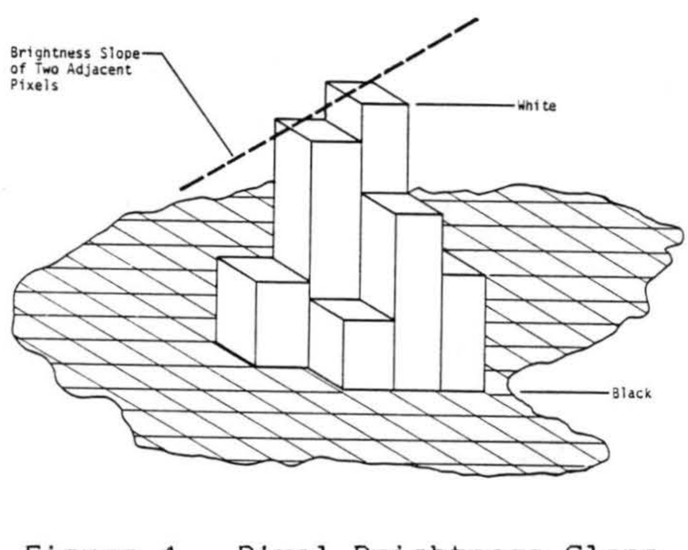


Figure 4. Pixel Brightness Slope
in a Digital Image
(Baxes, 1984)

convolution of the image with edge masks or templates. Levine (1985) discusses edge masks proposed by Roberts, Sobel, Prewitt, and Kirsh. Convolution of an edge mask with a pixel and it neighbors provides an index of the magnitude and direction of the intensity gradient at that pixel. Different templates must be applied to detect edges at different orientations. These templates typically vary in size from 2 X 2 pixels to 7 X 7 pixels (Fig. 5).

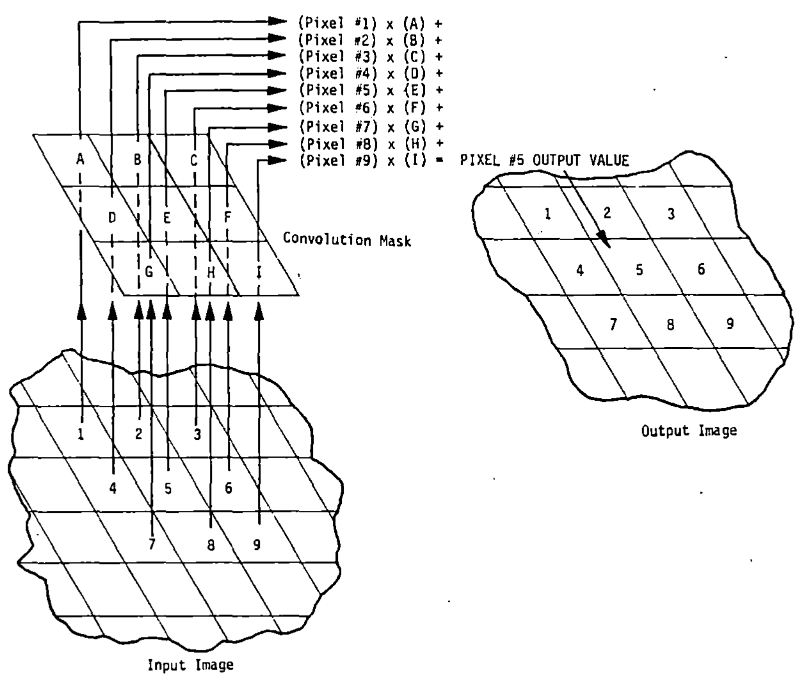


Figure 5. Spatial Convolution (Baxes, 1984)

Pixels with an intensity that does not correspond to the intensity in the real scene contribute noise. Small gradient masks may interpret this noise as evidence of an edge. The effect of averaging makes larger gradient masks less sensitive to noise, but at a cost of longer processing time. Low-pass filtering can reduce noise, however it also blurs details in the image.

Convolution of an image with a gradient mask yields an image which is an approximation of the first partial derivative of the original image in the direction of the mask gradient. Convolution of an image with the Laplacian edge detector yields an approximation of the second derivative of the image (Englander, 1986). The Laplacian therefor detects changes in gradient and is not sensitive to areas of constant gradient which correspond to areas of uniformly changing grey level in the original image (Fig. 6). The Laplacian is not dependent on edge orientation, but is very sensitive to noise.

A different Laplacian is required for detection of increasing changes in gradient as opposed to decreasing changes in gradient. Depending on the detector used, the transformed image will contain either gradient magnitudes or magnitudes of gradient changes. Larger magnitudes correspond to more pronounced edges, while smaller magnitudes correspond to noise and weak edges. The gradient image may be thresholded to obtain a binary image of the strongest edges.

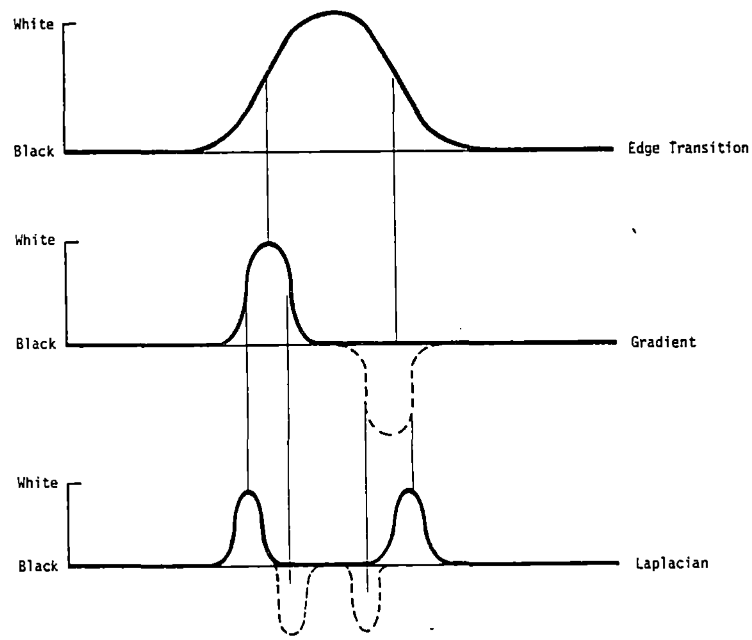


Figure 6. Response of Gradient and Laplacian Edge Detectors (Baxes, 1984)

Feature Extraction

After an image has been segmented into object(s) and background, various features of the object must be extracted on which processing decisions can be based. The features selected are highly dependent on the specific application. Length and width dimensions are readily computed for simple objects. These measurements may be determined from the runlength code of the image. The runlength code is a series of numbers representing the locations of transitions between object and background on each pixel grid line. The first and last lines containing the object may be used to calculate the length, while the maximum object runlength from all lines would represent the width.

This procedure is not adequate for objects in random orientation. Another method makes use of the minimum enclosing rectangle (MER) (Castleman, 1979). As the object is rotated in small increments thru an angle of 90 degrees, the area of the MER is computed. When the area is minimized, the length and width of the object are taken as those of the rectangle. The orientation of the principal axes of the object may be derived from the angle at which the MER was minimized.

Moments of an object are useful features for size and shape determination. The general equation for the moment of a two-dimensional function is,

$$M_{jk} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} X^j \, Y^k \; F(X,Y) \; dx \, dy \; .$$

For the case of a discrete image function the general moment equation is,

$$M_{jk} = \sum_X \sum_Y X^j \, Y^k \; I(X,Y) \; .$$

The parameter $j+k$ is known as the order of the moment. The zeroth order moment is,

$$M_{00} = \sum_X \sum_Y I(X,Y) \; .$$

In a binary image with object pixels equal to one and background pixels equal to zero, the zeroth moment is the area of the object. If the object pixels retain grey levels while the background pixels are equal to zero, the zeroth moment is called the integrated optical density (IOD).

The centroid of the object can be found by dividing the first moments by the zeroth moment:

$$\overline{X} = \frac{M_{10}}{M_{00}} \quad , \quad \overline{Y} = \frac{M_{01}}{M_{00}} \quad .$$

Moments calculated with the centroid as the origin are called central moments:

$$\mu_{jk} = \sum_{x} \sum_{y} \; (X-\overline{X})^{j} \; (Y-\overline{Y})^{k} \;\; I(X,Y) \; .$$

The principal axes x' and y' can be found at an angle ϕ from the x and y axes by the equation,

$$\tan 2\phi \;=\; \frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \; .$$

Moments which are divided by the area (or IOD), and calculated relative to the principle axes with the centroid as origin, are invariant to size, orientation, translation of the object. This property makes moments useful in pattern recognition. There is an infinite set of moments which completely specify a function f(x,y). A selected subset of these moments can be used to discriminate between different shapes.

Tabatabai and Mitchell (1984) describe a method of edge location to subpixel accuracy in which the first three moments of a one-dimensional data set (containing an edge) are matched to an ideal step edge having the same moments. This method can also be applied to width measurement where the moments of the object cross-section are matched to a square wave with a width assumed to be that of the object.

Many other features may be extracted from an image, though their utility is highly dependent on the application. Shape encoding is very useful in object recognition and includes such techniques as boundary chain code, Fourier transforms and derivatives of chain code, and medial axis transforms (Ballard and Brown, 1982). Measurement of parameters such as perimeter, circularity, rectangularity, or elongation may be useful in specific applications.

CHAPTER III

METHODS AND PROCEDURES

Introduction

The development of a machine vision pine seedling grading algorithm required the assembly of proper equipment, investigation of processing techniques, and extensive programming. This chapter initially describes the equipment used for the laboratory development and testing of the grading algorithm. The next section describes the selection of grading criteria and the grading scheme employed in the algorithm. This description is followed by a discussion of an investigation of several caliper measurement techniques. The algorithm developed for grading pine seedlings is then described. Finally, evaluation of algorithm performance is discussed.

Description of Equipment

This section describes the equipment used for the laboratory implementation of the seedling grading algorithm. The main components are a conveyor belt, machine vision computer, cameras, lenses, and lights.

To simulate grading shed operations, a conveyor belt was constructed on which singulated seedlings could be transported beneath a pair of cameras. The belt was 56 cm (22 in) wide with a 91 cm (36 in) travel and powered by a variable speed drive. A frame above the conveyor belt supported lighting and two cameras (Fig. 7).

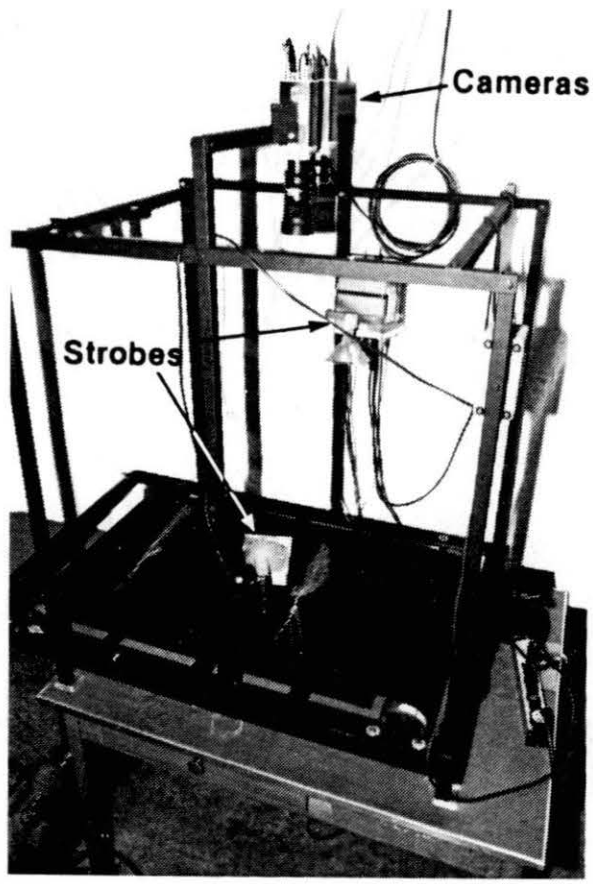


Figure 7. Conveyor Belt, Cameras, and Strobe Lamps

The belt material as received from the supplier was highly reflective and was dulled with a disk sander. This treatment allowed strong illumination of the seedlings without specular reflection from the belt surface. Specular reflection would otherwise reduce the contrast between the seedling and the belt, making segmentation difficult. Polarizing filters were investigated as a means of removing specular reflection, but their use would have required more powerful light sources.

The image processing computer used for this investigation was the International Robomation/ Intelligence (IRI) D256 machine vision system (IRI, 1985a). This system digitizes images into an array of 256 (H) X 240 (V) pixels with 256 grey levels. Four frame buffers are available for image processing. The D256 employs a Regulus operating system (Unix look-alike) and includes vision software written in the C programming language. The resident Iconic Kernel System is a library of function calls which set parameters and perform image processing functions. A 40 Mbyte Winchester hard disk is used for program, data, and image storage. A 5-1/4 inch floppy disk drive is available for archive creation and retrieval. An external output was interfaced to a strobe illumination source to provide synchronous operation with the RS-170 television format of the cameras (Appendix D).

The D256 has a coprocessor which performs computationally intensive operations such as the addition, subtraction, or multiplication of two images. The coprocessor also performs image convolution, runlength encoding, and moments calculations. The D256 can convolve an image with a mask as large as 7 X 7 pixels. The time required for convolving a 3 X 3 mask with an image is 38 milliseconds. A coprocessor window may be defined which limits coprocessor operations to a selected set of lines, thus reducing processing time.

Two Hitachi KP-120U cameras were used for image acquisition. The KP-120U is a black-and-white solid-state television camera with a 320 (H) X 244 (V) pixel sensor. One camera was used to obtain a close-up image of the root collar, having a FOV approximately 12.8 cm (5 in) square and a pixel resolution of approximately 0.5 mm (0.20 in). A Tokina 12.5 - 75 mm zoom lens set at a focal length of 48 mm and an aperture of f2.0 was used on camera 1. A second camera, with an Optronix 12.5 mm lens and an aperture of f2.0, gave a FOV approximately 51 cm (20 in) square, and acquired an image of the entire seedling. Both cameras were mounted 106 cm (42 in) above the conveyor belt. The wide-angle camera was centered and the close-up camera was placed 10 cm (4 in) off center over the expected location of the root collar.

Three types of illumination were used in this study. Four 75 W incandescent flood lamps were used in an investigation of the precision of edge detection techniques. Illumination for the grading algorithm was provided by fluorescent room lighting and strobed xenon flash. The relatively low-level room lighting was sufficient for detection of the seedlings in the FOV of camera 2. When a seedling was detected, strobe lamps were used for the acquisition of an image with each camera. Strobe illumination was provided by a General Radio Strobotac and Stroboslave, which were triggered by the IRI D256. A short flash duration of six microseconds allowed a sharp image of the moving seedling to be obtained. Strobe lamps were mounted on either side of the conveyor belt, in line with the cameras. The Strobotac's lamp was positioned 58 cm (23 in) above the belt surface, illuminating the seedling shoot and needles. The lamp of the Stroboslave was positioned 31 cm (12 in) above the belt and illuminated the roots and root collar. Higher intensity illumination was found to be desirable in the root zone, because the roots have a lower reflectance than the needles.

Selection of Grading Criteria

It was necessary to limit the number of seedling grading criteria in order to achieve the goal of grading seedlings in real time and to constrain the scope of this study to a reasonable breadth. Measurement of stem caliper at the root collar is an obvious choice, because it is the most important morphological quality indicator. Shoot height and root volume are also important quality indicators and were chosen as additional grading criteria for this study. To meet processing time constraints, a decision was made to emphasize caliper measurement and obtain only rough indices of shoot height and root volume. A classification scheme based on values of these three parameters is discussed below.

A classification scheme was formulated, based on grading criteria cited in the literature (Table 1). Seedlings are graded into three classes; acceptable, cull, and not gradable. It is assumed that seedlings which are not gradable will be grouped with culls, but they are classified separately as an indicator of the algorithm's ability to grade seedlings. In a commercial implementation, the cutoff values between classes could be easily altered from the values chosen for this study.

The orientation of the major axis of each seedling is measured and is used to correct the measured caliper and shoot height for angular variation. Although seedling orientation is assumed to be constrained, this provision increases the robustness and accuracy of the algorithm. Investigation showed acceptable algorithm performance for seedling orientations within thirty degrees of vertical. When the orientation of the seedling is greater than thirty degrees, the seedling is classified as not gradable.

The stem caliper of a seedling is acceptable if it measures from 3.0 to 8.0 mm. Seedlings with a measured caliper between 2.8 and 3.0 mm are acceptable if the measured root area index is significantly larger than the cutoff value. Under this condition, it is assumed that the caliper measurement was erroneously small, and that a larger root area indicates a larger stem caliper.

The root area index of a seedling is acceptable if it is greater than 200 pixels. This corresponds to an area of approximately 9.7 sq. cm (1.5 sq. in). This value was chosen after consultation with experts in seedling production. The purpose of this measurement is to enable rejection of seedlings with significantly undersized or missing root masses. Seedlings with calipers between 2.8 and 3.0 mm must have an area index greater than 250 pixels to be classified as acceptable.

The shoot height of a seedling is acceptable if its measurement is greater than 16 cm. This value is larger than some minimums found in the literature. A larger cutoff is used because the algorithm measures the distance from the root collar to the end of the needles, which is not always the true shoot height.

Seedlings which the algorithm fails to grade are classified as not gradable. Acceptable seedlings are classified as A1 or A2, depending on whether the measured diameter is greater or less than 3.0 mm, respectively. Cull seedlings are classified as C1, C2, or C3, depending on measured caliper. Table 2 presents this classification scheme.

TABLE II

GRADING SCHEME FOR LOBLOLLY PINE SEEDLINGS

Caliper (mm)	Root Area Index (pixels)		Shoot Height (cm)	Grade
3.0 - 8.0	> 200		> 16	A1
2.8 - 3.0	> 250		> 16	A2
< 2.8 or > 8.0	any		any	C1
3.0 - 8.0	< 200	or	< 16	C2
2.8 - 3.0	< 250	or	< 16	C3

Investigation of Caliper Measurement Techniques

The importance of accurately measuring stem caliper and the difficulty of doing so with low pixel resolution prompted an investigation of several caliper measurement techniques. Six methods of measuring caliper were investigated for precision and speed.

Each method was applied to two sets of thirty images of a wooden dowel having a nominal three millimeter caliper. The caliper of the dowel varied between 2.95 and 3.05 millimeters when rotated about its axis. Dowel orientation was vertical in one image set, while in the other set the dowel was oriented fifteen degrees either side of vertical. The dowel was rotated about its axis and displaced horizontally between images. Wood stain was applied to the dowel to approximate the color of a seedling stem.

Images of the dowel were acquired with camera 1, having a FOV of 12.8 cm and a pixel resolution of 0.5 mm. Incandescent flood lamps were used to provide even illumination. For each image and technique, the dowel caliper was taken as the average of the calipers measured on 35 consecutive pixel grid lines near the center of the image.

Method 1: Binary Thresholding

The high contrast, grey-level image of the dowel was thresholded to produce a binary image in which the dowel was represented by white pixels, and the conveyor belt (background) by black pixels. Choice of threshold value had a strong influence on the measured caliper due to the grey-level gradient at each edge. A grey-level threshold of 120 was used, resulting in a mean dowel measurement of 3.0 mm (for vertical dowel images). The binary image was next runlength encoded. On any image line, the caliper of the dowel (in pixels) was taken as the distance from the first transition to the second transition of the runlength code, corresponding to the left and right edges of the dowel, respectively. Transitions in the runlength code correspond to pixel locations of intensity changes (black-to-white, and white-to-black) in the binary image.

Method 2: Moments

The method developed by Tabatabai and Mitchell (1984) for edge location with subpixel accuracy was applied to the measurement of stem caliper. The grey level-image was initially thresholded and runlength encoded as in method one. The first and second transitions were used to determine the center of the dowel on each line. The maximum seedling caliper encountered in a grading situation is expected to be approximately eight millimeters, which corresponds to about sixteen pixels. A one-dimensional data set (grey levels) from each line, centered about the dowel midpoint, was used to measure dowel caliper with this technique. A data set of thirty pixels was chosen to insure inclusion of the entire stem and a reasonable amount of background.

Method 3: Modified Laplacian Edge Detector (3 X 3)

The 3 X 3 Laplacian edge mask resident in the Iconic Kernel Package is given below.

$$\begin{bmatrix} 0 & 4 & 0 \\ 4 & -16 & 4 \\ 0 & 4 & 0 \end{bmatrix}$$

A modified version of this mask with increased sensitivity to vertical edges and decreased sensitivity to horizontal edges is given below.

$$\begin{bmatrix} 2.625 & 0.0 & 2.625 \\ 2.625 & -16.0 & 2.625 \\ 2.625 & 0.0 & 2.625 \end{bmatrix}$$

The grey-level image was convolved with this modified mask, resulting in an image with grey levels which are an index of the change in grey-level gradient. This mask detects positive changes in gradients which have horizontal components, corresponding to the edges of a vertical dowel or seedling stem. The grey-level image was next thresholded to show only the strongest gradient changes. A threshold of 30 was chosen to clearly show the edges. The image was runlength encoded, and the caliper was taken as the distance between the first and third transitions, on lines containing four transitions. Two transitions mark each edge, with the first and third transitions corresponding to the first transition at each edge. Lines which contained either more or less than four transitions were not considered. These correspond to lines in the original image with noise or milder changes in grey-level gradient at the dowel edges, respectively.

Method 4: Modified Laplacian Edge Detector (5 X 5)

A 5 X 5 edge mask which detects horizontal, quadratic (Laplacian) changes in grey level is given below.

$$\begin{bmatrix} 0 & 2 & -4 & 2 & 0 \\ 0 & 2 & -4 & 2 & 0 \\ 0 & 2 & -6 & 2 & 0 \\ 0 & 2 & -4 & 2 & 0 \\ 0 & 2 & -4 & 2 & 0 \end{bmatrix}$$

This mask is more sensitive to vertical edges, but gave inferior performance in detecting the edges of the dowel oriented at fifteen degrees from vertical. Caliper was measured in the same manor as in method 3, but with a threshold of 25.

Method 5: Gradient Edge Detector

Two gradient edge masks were developed for the detection of the positive and negative gradients at the left and right edges of the dowel, respectively. Performance is increased by minimizing the width of the mask, since we have a priori knowledge that the edge width is small with respect to pixel size. Both masks translate the location of the edge to the right because they are not symmetrical. The grey-level image is convolved with each mask and the two resulting images are summed. This image is then thresholded at grey level 120, followed by runlength encoding. The caliper on any line is taken as the distance between the first and third transitions in the runlength code. Again, this operation is performed on lines with exactly four transitions. The gradient masks are given below.

$$\begin{bmatrix} 0 & -3 & 3 \\ 0 & -3 & 3 \\ 0 & -3 & 3 \end{bmatrix} \qquad \begin{bmatrix} 0 & 3 & -3 \\ 0 & 3 & -3 \\ 0 & 3 & -3 \end{bmatrix}$$

Method 6: Modified Gradient Edge

Detector

A minor modification of the masks used in method 5 reduced the sensitivity of this method to noise. The threshold was reduced to 100 and caliper measurement proceeded as in method 5. The gradient masks are given below.

$$\begin{bmatrix} 0 & -3 & 3 \\ 0 & -4 & 3 \\ 0 & -3 & 3 \end{bmatrix} \qquad \begin{bmatrix} 0 & 3 & -3 \\ 0 & 3 & -4 \\ 0 & 3 & -3 \end{bmatrix}$$

Performance Evaluation

Performance of the six methods was evaluated for precision and speed. Precision was determined from the variance of measurements after scaling to a mean of 3.0 mm (Table III). An F value of 1.99 was calculated for the comparison of methods two and three. An F value of 1.85 is significant at the 90 percent confidence level, indicating that method two is statistically more precise.

TABLE III

SUMMARY OF STATISTICS FROM SIX CALIPER MEASUREMENT METHODS

METHOD	Average Measured Caliper (mm)	Stdev Measured Caliper (mm)	Stdev Scaled Caliper (mm)
1	2.996	0.1193	0.1194
2	2.895	0.0643	0.0666
3	3.762	0.1178	0.0940
4	3.811	0.1396	0.1099
5	2.610	0.1174	0.1350
6	2.730	0.1442	0.1585

The speed of each method was evaluated by measuring the time required to perform 100 consecutive caliper measurements on a dowel. The image of the dowel was stored in a frame buffer, and the caliper measuring subroutine was repeatedly called. Method 2 required 0.54 seconds per measurement, compared to the remaining methods which required approximately 0.03 seconds. Calculations in method 2 required a large number of time-consuming floating point operations. Attempts to convert these calculations to faster integer arithmetic were not successful.

Method 3 was chosen as the preferred method for use in the algorithm. The positive Laplacian operation detects the outer gradient changes at the dowel edges (Fig. 6). This results in a larger pixel distance between the thresholded edges (unscaled mean in Table III). Although the pixel variance was approximately the same for all methods (other than method 2), when the pixel caliper from method 3 was scaled to a metric measurement, a lower metric variance resulted. An F-test comparing methods three and four yielded a value of 1.37, which was not significant at the 90 percent confidence level. Although method three was not statistically more precise than the next best method (4), it was selected because it was fast and had the best precision, next to method two. Appendix A contains data, programs, and analysis documenting this investigation.

Description of Algorithm

The grading algorithm is composed of several sections which consist of grading subroutines and support subroutines. The grading subroutines are: waitfor( ), orient( ), col1( ), col2( ), diam( ), root( ), and grade( ). The support subroutines are: main( ), calibrate( ), statfile( ), edge( ), setthr( ), wish( ), scales( ), pixel( ), keysnap( ), and tinue( ). Most of these subroutines call standard C functions and/or Iconic Kernel functions. Appendix B contains the program listing of the grading algorithm.

Whenever possible, all grading subroutines are applied to every seedling. This is not done, however, if a seedling fails to meet the orientation criterion or a grading subroutine is unable to complete its task. Barring such failures, all measurements are made on each seedling even if, for example, the measured stem caliper is found to be unacceptable. This procedure is followed because time must be allocated to measure all parameters, and the statistics collected on cull seedlings could be of value in a commercial implementation.

Measurement of stem caliper is normally performed at the root collar. This location is desirable for visual gauging, because edges of the stem are usually not occluded by needles and/or branches at this location. Two concerns influenced the FOV chosen for the root collar image. The first was the accuracy of caliper measurement, and the other was the probability that the root collar would pass through the FOV as the seedling traveled down the conveyor.

Very accurate measurements have been obtained with machine vision gauging systems under controlled conditions. In this application, however, the position of the root collar cannot be tightly constrained, and a wide FOV is necessary. A decision was made to make the FOV as large as possible, while maintaining a measurement precision of at least 0.5 mm (0.20 in). If the pixel resolution is set equal to 0.5 mm, a measurement precision better than 0.5 mm should be attained through image processing. A pixel resolution of 0.5 mm yields a FOV of 12.8 cm square (5.0 in sq.). It is assumed that since the root collars are all at the same level when they are mechanically lifted from the seedbed, they will not be displaced more than a few centimeters when they pass through the FOV.

Seedling Detection

The waitfor( ) subroutine is called and initialized with a threshold grey level, the address of window coordinates, and a minimum detection area. With the strobe-sync disabled, a loop is entered in which successive images are acquired with camera 2 (wide FOV). Each image is multiplied by a template, which defines a window in which seedling detection will trigger subsequent operations. The FOV of this "waitfor" window overlaps and is smaller than the FOV of camera 1 (Fig. 8). All pixels inside the window remain unchanged, while the exterior pixels are set to zero. A coinciding hardware window is also implemented (44 lines out of 240), which proportionately reduces coprocessor processing time.

After thresholding at grey level 25, the coprocessor calculates the area in the windowed, binary image. A low threshold is necessary for seedling detection under conditions of low illumination intensity. When the area in the thresholded image exceeds fifty pixels (default), the strobe-sync is enabled, and an image is obtained from each camera (with strobe illumination). The image from camera 1 (close-up) is placed in frame buffer 1, and the image from camera 2 (wide-view) is placed in frame buffer 4. The horizontal position of the seedling in both images is a function of conveyor velocity and location of the template window. Control is then returned to the main program.

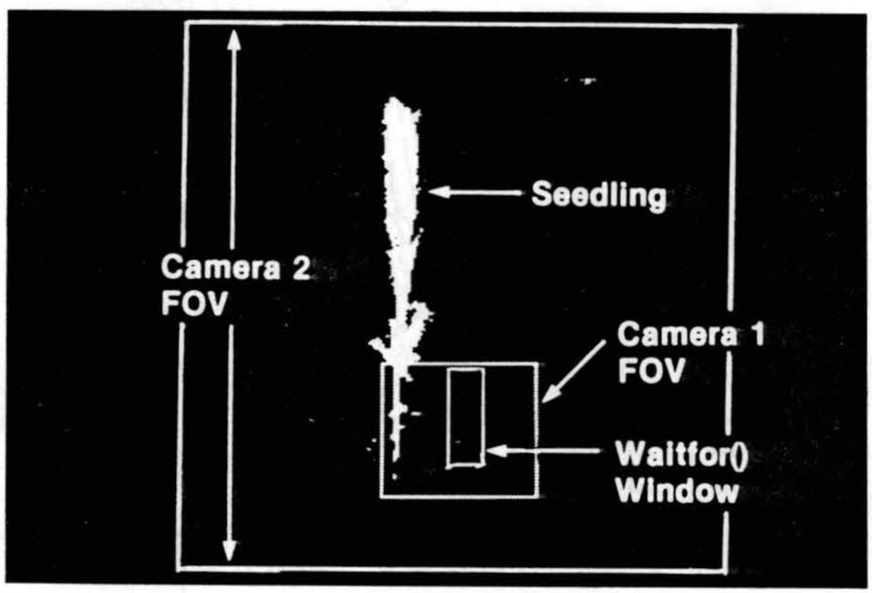


Figure 8. Field of View of Camera 2, FOV of Camera 1, and Waitfor( ) window

Seedling Orientation

The orient( ) routine is called and initialized with a threshold value and the addresses of variables which hold degree and radian measures. Image buffer 4 (wide-view) is copied to buffer 3 and thresholded at grey level 170 (Fig. 9). A high threshold used here segments the major axis of the seedling from the grey level image. The Iconic Kernel procedure, Imoments( ), calculates the first three moments and derives the angle between the seedling major axis and the vertical axis of the FOV. Imoments( ) returns this angle in radians, which is converted to degrees, and both angular measures are assigned to their respective variables. Control is then returned to the main program. The degree measure is recorded in the statistics file, and the radian measure is used in subsequent calculations. If the orientation angle is greater than thirty degrees, no further measurements are made and the seedling is classified as not gradable.

Location of Root Collar

Accurate location of the root collar is crucial for the subsequent measurement of stem caliper, shoot height, and root area index. This task is not trivial, because there is large variation in seedling silhouettes (Fig. 2). The best case is shown in Figure 10, where image lines with only two transitions (left and right stem edges) are good candidates for the root collar location. In some cases, however, there are no lines with only two transitions (Fig. 11). Subroutines col1( ) and col2( ) work together to locate the root collar.

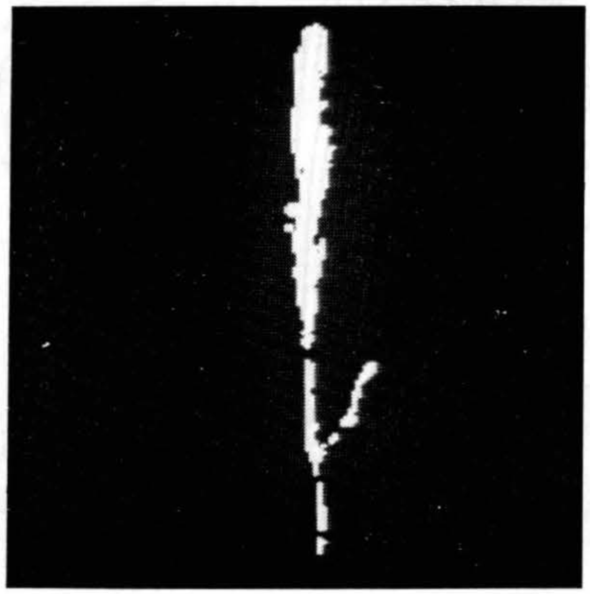


Figure 9. Orient( ) Subroutine
Threshold at
Grey Level 170

Col1( ) is passed a threshold, the addresses of variables which are assigned the collar line and midpoint (column) location, and the address of the window size variable (number of lines) for caliper measurement. Frame buffer 1 (close-up) is copied to buffer 2 and thresholded at grey level 90. This yields a binary image showing the stem, roots, branches, and needles (Fig. 11). This image

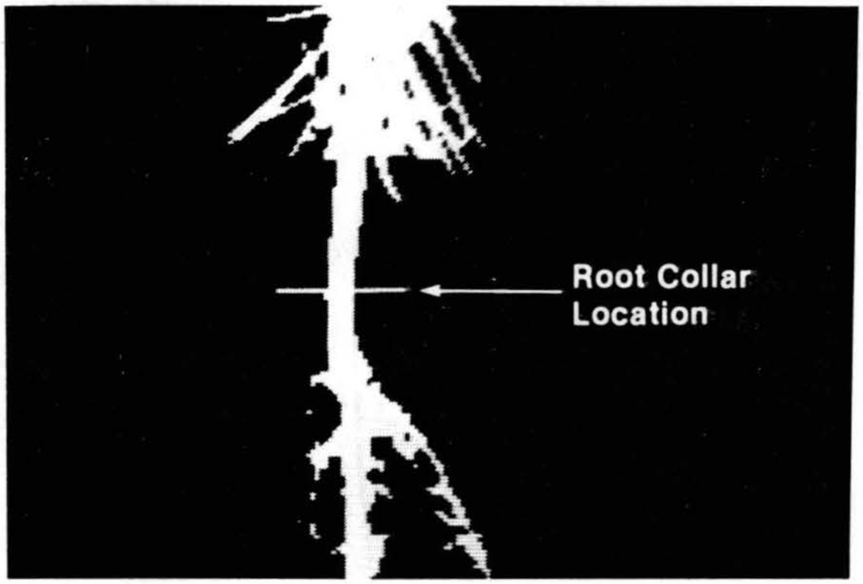


Figure 10. Algorithm Finds Root Collar. Threshold Grey Level: 90

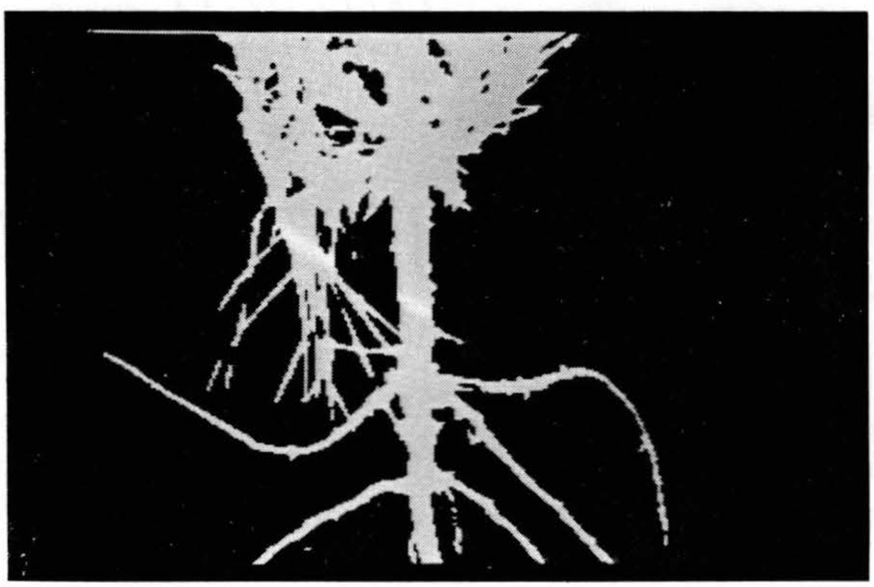


Figure 11. Algorithm Fails to Find Root Collar. Threshold Grey Level: 90

is next runlength encoded.

Col2( ) is then called, passed the same addresses passed to col1( ), and the variable, ntrans. If the number of transitions on a line is less than or equal to ntrans, that line is a candidate for the root collar location. If col2( ) fails to find the root collar using ntrans, it returns a 0 to col1( ), and col2( ) is called again with a larger value of ntrans. Ntrans takes values of two, four, and finally six. When col2( ) is successful, it returns a 1 to col1( ). Col1( ) returns a 1 to main( ) when col2( ) is successful, or returns a 0 if col2( ) fails with ntrans equal to six. If col1( ) returns a 0 to main( ), col1( ) is called again with a threshold of 140. At this threshold, only the stem and major branches and roots are visible (Fig. 12). If no collar can be located at this second threshold, the seedling is classified as not gradable.

Col2( ) inspects every line containing the ntrans number of transitions, or less. The number of transitions on a line is always an even number. The transitions occur in pairs; black-to-white, and white-to-black. On each candidate line, the maximum distance between pairs of transitions is determined. If this distance is between five and eighteen pixels (2.5 and 9 mm), the pixel line number and midpoint (column number) between the two transitions are stored.

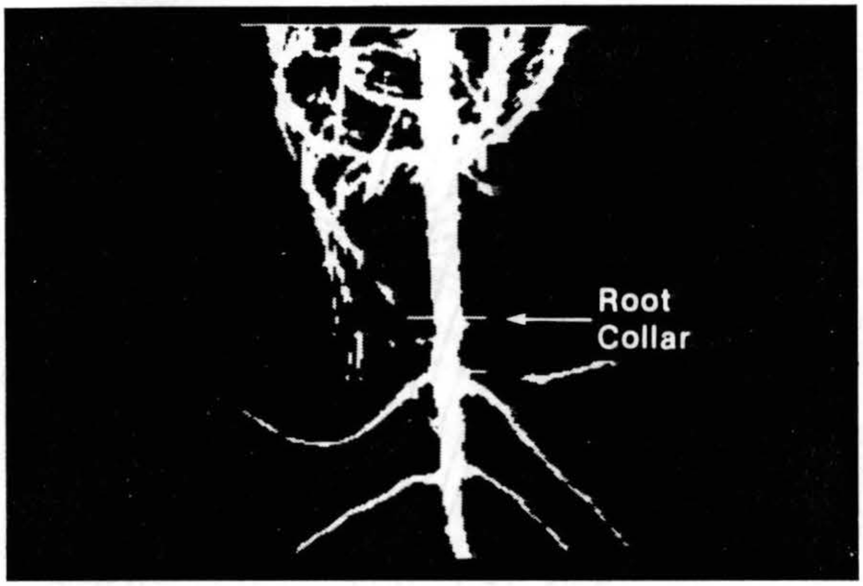


Figure 12. Algorithm Finds Root Collar.
Threshold Grey Level: 140

After all lines have been inspected, the list of stored line and column numbers is processed. For sets of consecutive lines, the line numbers and column numbers are summed and stored. The number of lines in each set of consecutive lines is also stored. The set of consecutive lines with the largest number of members, and having at least six members, is assumed to contain the root collar. Col2( ) will return a 0 to col1( ) if there are no sets of consecutive lines, if there are more than thirty sets, or if the largest set has fewer than six members. Otherwise, the collar is located at the average of the line numbers in the largest set. The midpoint of the root collar is located at the average of the column numbers.

Additionally, one-half of the number of members in the set, or a maximum of ten, is assigned to a variable which defines the size (number of lines) of the caliper measurement window. If successful, a 1 is returned to col1( ).

Measurement of Stem Caliper

Six parameters and two addresses are passed to the subroutine diam( ), which calculates the stem caliper. The parameters are; the line number of the root collar, the collar midpoint (column number), the size of the caliper measurement window, a scale factor, the stem orientation angle, and a threshold value. Additionally, the address of the convolution coefficient matrix, and the variable which holds the measured caliper, are passed.

Initially, a hardware window is implemented about the root collar. Window size is defined in the root collar subroutines. The image in buffer 1 is convolved with the modified Laplacian edge detector, and the result is placed in buffer 2 (Fig. 13). Image buffer 2 is next thresholded at grey level 57, resulting in a binary image of the strongest edges. Runlength encoding is then performed. The convolution and runlength encoding operations are applied only to that portion of the image inside the hardware window.

For lines which are candidates for caliper measurement, and contain four or more transitions, the consecutive odd transitions which bracket the midpoint of the collar are found (Fig. 14). Odd transitions correspond to the left members of transition pairs. If these transitions are within ten pixels (horizontally) of the collar midpoint, the distance between the transitions is summed with other such distances, and a counter is incremented.

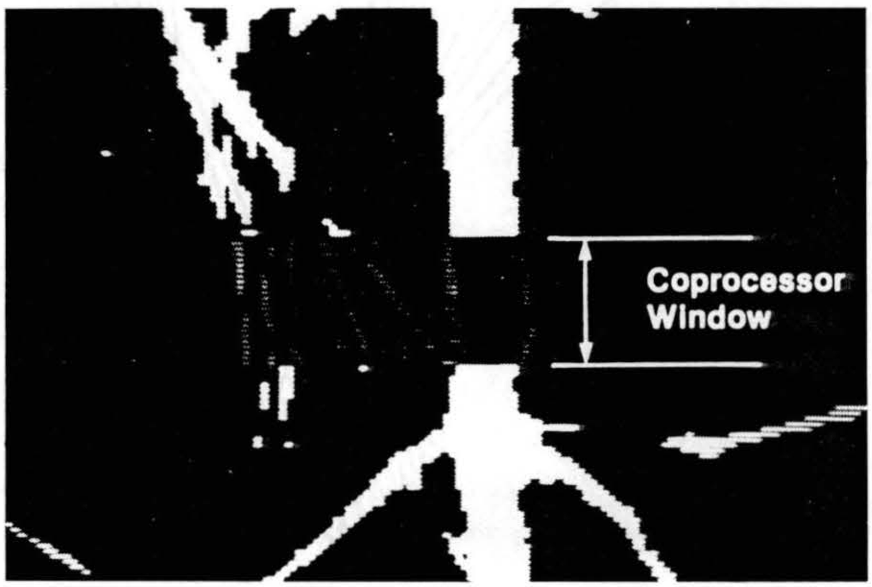


Figure 13. Modified Laplacian Edge Detector Applied in Hardware Window

When the processing of candidate lines is complete, and at least one line has provided a distance measure, the stem caliper is calculated. The sum is multiplied by the caliper scale factor, the cosine of the orientation angle,

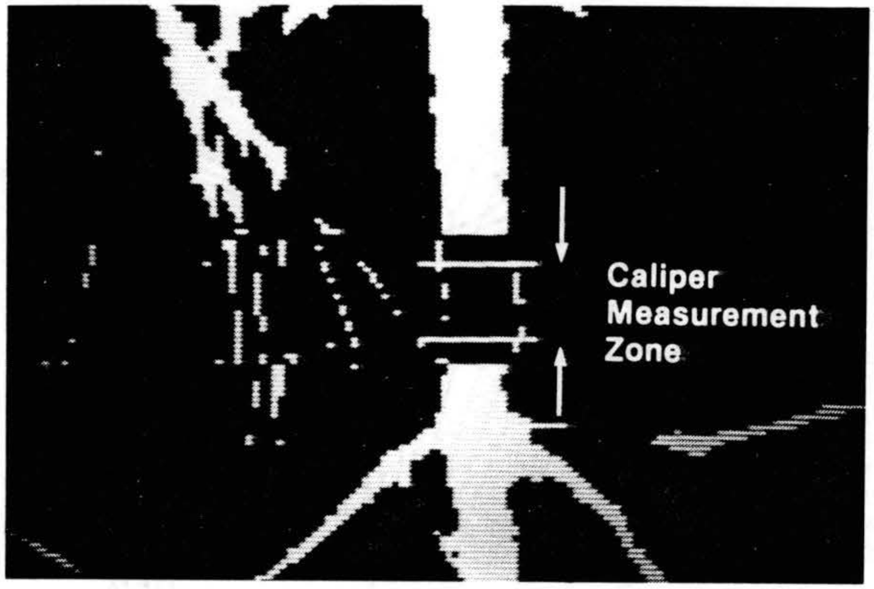


Figure 14. Root Collar Zone Thresholded at Grey Level 57 after Convolution

and divided by the summation counter to yield the stem caliper. If no lines provided a distance measure, a 0 is returned to main( ). Main( ) again calls diam( ), but passes a threshold at grey level 40, and the process proceeds as before. The lower threshold yields a binary image with a greater number of edge pixels. If diam( ) is still unable to obtain a measurement, a 0 is again returned to main( ), and the seedling is classified as not gradable. When diam( ) is successful, a 1 is returned to main( ).

Measurement of Root Area Index

The subroutine root( ) calculates both root area index and shoot height. Five parameters are passed; the line number of the root collar, a threshold for root area processing, a threshold for height processing, a scale factor for camera 2, and the orientation angle of the seedling. Additionally, addresses for the area and height measurement variables are passed. The root area index is measured first, followed by shoot height measurement.

Before root( ) is called, an equation in main( ) calculates the line number of the root collar in image buffer 4 (wide view). This equation uses the line number of the root collar found in image buffer 1 (close-up) and transformation coordinates defined in the calibrate( ) subroutine.

When root( ) is called, a hardware window is implemented from the root collar to the bottom of the image (wide-view). The image in buffer 4 is then convolved with a 5 X 5 Laplacian edge detector (predefined in the Iconic Kernel Package), and the resulting image is placed in buffer 3. Thresholding this image at grey level 48, yields a binary image with maximum root area but minimum noise. This is followed by a coprocessor calculation of the area inside the hardware window (Fig. 15). This area is assigned to the area measurement (pixel) variable and the hardware window is disabled.

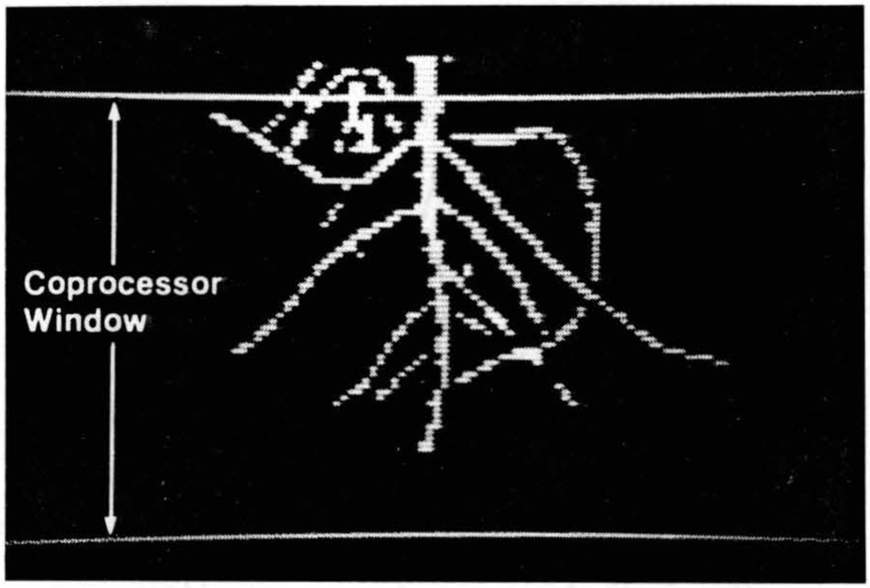


Figure 15. Root Zone Thresholded at Grey Level 40 after Convolution with Laplacian Edge Detector

Measurement of Shoot Height

For shoot height measurement, the image in buffer 4 is copied to buffer 3, which is then thresholded at grey level 100 (Fig. 16). The binary image is then runlength-encoded. Starting at the top of the image, each line is checked to determine if the maximum distance between paired transitions exceeds five pixels. The seedling top is assumed to be located when four consecutive lines meet this criterion. If this condition is not met before the root collar line is reached, a 0 is returned to main( ), and the seedling is classified as not gradable. The shoot height is calculated (cm) as the pixel distance between the root collar and the seedling top (Fig. 16), multiplied by a scale factor, and divided by the cosine of the orientation angle. The calculated height is assigned to the appropriate variable, and a 1 is returned to main( ).

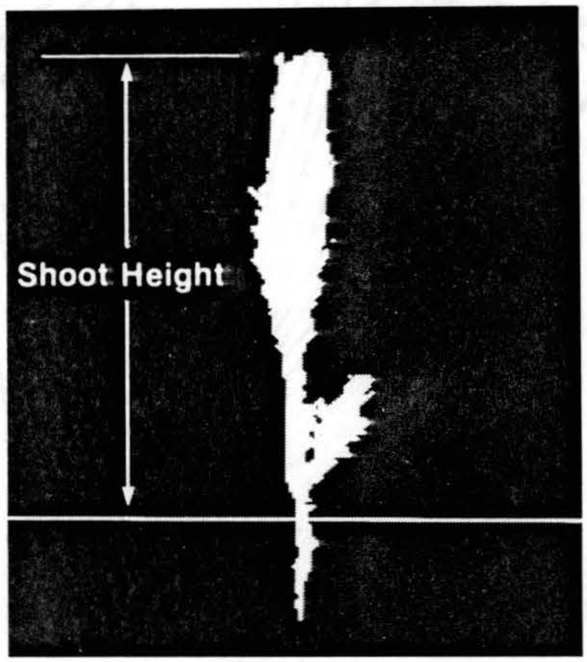


Figure 16. Measurement of Shoot Height Grey Level: 100

Recording Seedling Statistics

The grade( ) subroutine writes the measured seedling parameters, classification, and count, to a file which has been opened with the statfile( ) subroutine. A pointer (address) to the file is passed to this routine along with stem caliper, root area index, shoot height, root collar line number (from buffer 1), collar midpoint column, and angle of orientation. Additionally, the address of the appropriate classification counter is passed, and the counter is incremented. Three counters hold the total number of seedlings assigned to each classification.

Main Program

Main( ) is the name of the controlling program, which is standard practice in the C programming language. All threshold and parameter variables are declared at the beginning of main( ), along with a several loop and counter variables. Subsequent statements open the video interface, initialize the Iconic Kernel variables, and enable the coprocessor. Before entering the main loop of the program, the calibrate( ) and statfile( ) subroutines are called.

Inside the main program loop, values returned by subroutines (0's and 1's) are tested to control program flow. If all grading subroutines are successful in their respective tasks, a series of if-else statements is used to call the grade( ) subroutine with the appropriate parameters. Whenever a subroutine fails (0 returned), grade( ) is called with parameters successfully measured, and the seedling is classified as not gradable.

Algorithm Calibration

Proper calibration of threshold values, scale factors, and image transformation coordinates is essential to optimum algorithm performance. The calibrate( ) subroutine initializes sixteen such parameters with default values. The user is then provided an opportunity to alter the default values interactively.

The user is first requested to place a seedling in the field of view and snap frames as required, while making camera and lighting adjustments until a satisfactory image is obtained. This procedure is performed for both cameras. The user is next given the opportunity to alter eight default thresholds. The appropriate binary image (after edge detection, if necessary) is displayed, and the user is prompted to change the threshold (up or down) while observing the binary image. A message is displayed on the monitor to aid in the selection of an appropriate threshold. The threshold value is assigned to the appropriate variable after keying a carriage return. The keysnap( ), wish( ), and setthr( ) subroutines are used in this procedure.

The user is next given the opportunity to alter the default image transformation coordinates and scale factors. The user is instructed to place a calibration dowel vertically in the field of view of camera 1 (close-up). The user snaps as many frames as required to properly position the dowel. When the user is satisfied and keys the carriage return, the system automatically obtains an image from each camera.

The operator is next requested to enter the length and caliper of the dowel in millimeters (120 and 3.0 mm). Subsequently, an image from camera 2 (wide-view) is displayed. The user is instructed to move a cursor to the top of the dowel and key the carriage return. This step is duplicated for the bottom of the dowel. The procedure is then repeated for the image from camera 1 (close-up). From these operations, two corresponding points have been found in the two images. The four line numbers obtained are assigned to the image transformation coordinate variables. Pixel scale factors are also calculated for each image using the pixel distance between the ends of the dowel and the length of the dowel. These data are also used to scale and position the FOV of camera 1 in the image from camera 2 to aid in altering the position of the waitfor( ) window (Fig. 8). The subroutine pixel( ) is used in this procedure.

The caliper scale factor is determined by processing the dowel image in a manner similar to that used in the subroutine diam( ). The image is convolved with the modified Laplacian edge detector and thresholded at grey level 57 (or the altered caliper threshold). The binary image is next runlength-encoded. On forty central lines, which contain exactly four transitions, the average distance between the odd transitions is calculated. The caliper scale factor is calculated as the actual dowel caliper divided by the average pixel distance between the edges.

Additional Subroutines

The statfile( ) subroutine, called at the beginning of the program, requests a filename from the user for recording seedling statistics. The file is opened, and a header is printed at the beginning of the file identifying the parameters listed.

The subroutine tinue( ) is used in the "slow" version of the program, and halts program execution wherever called. A frame number is passed to this routine and the image in that frame is displayed for user inspection. The user is prompted to press any key to continue program execution. This routine is inserted after each processing step to allow observation of algorithm performance.

Additional Iconic Kernel functions are called in the slow version. These functions display such features as window borders or root collar location in specific images.

Method of Performance Evaluation

The performance of the algorithm was evaluated using a set of 100 loblolly pine seedlings obtained from the Weyerhaeuser nursery at Ft. Towson, Oklahoma. The seedling calipers ranged from 2.3 to 6.0 mm, with a subset of twenty having calipers between 2.8 and 3.3 mm.

Two tests were performed to evaluate algorithm performance. One test was designed to evaluated ability of the algorithm to correctly classify seedlings as acceptable or cull. This test also provided statistics on measurement precision and accuracy. The second test measured the time required to grade a seedling.

Evaluation of classification performance is based on the grading of 100 seedlings. A seedling was manually placed on the conveyor and passed beneath the cameras. The seedling was repeatedly returned to the belt for twenty repetitions. Conveyor belt speed was 0.46 m/s (1.5 ft/s). Assuming a seedling spacing of 46 cm (18 in) on the belt, this speed would provide a seedling throughput of one per second.

An effort was made (not always successfully) to place the root collar in the field of view of camera 1. No attempt to rigidly constrain the position of the root collar was made, because collar position could probably not be tightly constrained in a commercial implementation. An effort was made to place the seedlings vertically in the FOV, however, angular variation did occur. A few seedlings were presented to the cameras at orientations between 15 and 30 degrees from vertical. No seedling failed to be classified because of orientation. An effort was made to rotate the seedlings about their longitudinal axes, in order to present different views of each seedling to the cameras.

A separate file was created for statistics on each seedling. Actual seedling caliper, measured manually with a micrometer, was also stored in the file for each seedling. For most seedlings, the caliper at the root collar varied along and about the longitudinal axis. This variation was approximately 0.1 mm, and "actual" caliper was reported to the nearest 0.1 mm.

The second test evaluating algorithm performance measured the time required to grade a seedling. The grading program was modified by eliminating calibration steps and using default thresholds and scale factors. A loop in the main( ) program was modified to grade a seedling 100 times. A seedling was placed in the FOV so that it was inside the waitfor( ) window. On every loop, the waitfor( ) subroutine was satisfied on the first pass and two images were obtained. Subsequent processing proceeded as described in the previous section.

The Rugulus operating system function, Time, was used to call the modified program. After the modified program had graded the seedling and filed statistics 100 times, the Time function displayed the total elapsed time, time spent by the microprocessor in running the program, and time spent by the microprocessor in support of running the program. The sum of the final two statistics was used for calculating algorithm speed.

CHAPTER IV

RESULTS, CONCLUSIONS, AND RECOMMENDATIONS

Introduction

This chapter begins with a discussion of the performance of the grading algorithm. Performance will be discussed in terms of speed of the algorithm, accuracy of seedling classification, and seedling parameter measurement precision. This discussion is followed by a summary of the objectives and results of this study. The final section of this chapter presents recommendations for improvements to the algorithm, improvements in the grading environment, and areas in which further study might be beneficial.

Performance Evaluation Results

The algorithm developed in this study performed well in terms of both speed and accuracy. The time required to grade a seedling was approximately 0.25 seconds. This interval easily meets the goal of grading seedlings in real time (at least one per second).

The detection performance of the waitfor( ) subroutine was excellent. Seedlings were detected every time they passed through the field-of-view. A conveyor speed of

0.46 m/s (1.5 ft/s) was used in the evaluation of grading performance, corresponding to a throughput of one seedling per second. An informal investigation revealed that image capture was reliable at a conveyor speed of 1.0 m/s (3.28 ft/s). This conveyor speed would allow a commercial implementation to realize a throughput rate exceeding three seedlings per second.

The classification error rate averaged 5.7 percent for the set of 100 seedlings. A total of 2.3 percent of the seedlings in this set were not gradable (Table IV). This is acceptable performance, bettering manual grading operations which have an average misclassification rate of seven to ten percent (Beckman, 1986). As expected, a large part of the classification error was due to seedlings which straddled the borderline between acceptable and cull. Such seedlings comprised 17 percent of the grading test set.

Results from the grading of the 100 seedlings were divided into two data sets and analyzed. A set of 17 marginal seedlings (with respect to caliper and root mass) showed an average misclassification rate of 23.2 percent (Table V). In this set, acceptable seedlings were classified as culls 31.7 percent of the time, while culls were classified as acceptable 18.6 percent of the time. There is no significant commercial penalty for misclassification of borderline seedlings.

TABLE IV

PERCENT MISCLASSIFICATION OF 100 SEEDLINGS

Acceptable Seedlings 1380		Cull Seedlings 620	
Classified Cull	Not Gradable	Classified Acceptable	Not Gradable
66	34	49	13
4.7%	2.5%	7.9%	2.1%

Total Misclassification	5.7%
Total Not Gradable	2.3%

TABLE V

PERCENT MISCLASSIFICATION OF 17 SEEDLINGS

Acceptable Seedlings 120		Cull Seedlings 220	
Classified Cull	Not Gradable	Classified Acceptable	Not Gradable
38	0	41	9
31.7%	0%	18.6%	4.1%

Total Misclassification	23.2%
Total Not Gradable	2.6%

The remaining 83 seedlings showed an average misclassification rate of 2.2 percent (Table VI). In this set, acceptable seedlings were misclassified as cull 2.2

percent of the time, while culls were misclassified as acceptable 2.0 percent of the time.

TABLE VI

PERCENT MISCLASSIFICATION OF 83 SEEDLINGS

Acceptable Seedlings 1260		Cull Seedlings 400	
Classified Cull	Not Gradable	Classified Acceptable	Not Gradable
28	34	8	4
2.2%	2.7%	2.0%	1.0%

Total Misclassification	2.2%
Total Not Gradable	2.3%

Measurement precision was good, considering the pixel resolutions of cameras 1 and 2, which were 0.5 mm and 2.2 mm, respectively. The coefficient of variation (CV) (standard deviation divided by mean) of caliper measurements ranged from 1.3 to 35.3 percent for different seedlings, averaging 7.6 percent. The CV of the root area index ranged from 3.6 to 61.3 percent and averaged 12.2 percent. Shoot height CV ranged from 0 to 15.3 percent and averaged 4.1 percent for all seedlings.

The few seedlings which showed the largest deviations in measured parameters were characterized either by needles hanging down past the root collar, or by roots bent upward past the root collar, or both. The subroutines which located the root collar performed inconsistently on such seedlings. These seedlings were also responsible for the largest number of "not gradable" classifications.

Appendix C contains data supporting this evaluation. The mean, standard deviation, and coefficient of variation of the caliper, area index, and shoot height measurements for each seedling are tabulated. A table summarizing the manual and algorithm grade classifications of each seedling follows.

Conclusions

This study has shown that machine vision can provide accurate real-time grading of pine seedlings. A seedling grade classification scheme was defined. Seedlings were classified as acceptable or cull on the basis of minimum acceptable stem caliper, root area, and shoot height.

A real-time machine vision algorithm which measures seedling stem caliper, root silhouette area, and shoot height was developed and implemented. Emphasis was placed on accurate caliper measurement. Seedlings were assumed to be singulated and transported on a non-reflective black conveyor belt, with shoot orientation and root collar position loosely constrained.

Tests with loblolly pine seedlings revealed excellent performance. Seedlings were graded in approximately 0.25 seconds, with an average classification error rate of 5.7 percent. The coefficient of variation of measurements on 100 seedlings averaged 7.6, 12.2, and 4.1 percent for stem caliper, root area, and shoot height, respectively.

The machine vision algorithm developed in this study could serve to improve commercial grading operations. Seedlings could be inspected and graded with a lower classification error rate than is achieved with current manual operations. Measurement precision is adequate to allow grading into multiple classes, taylored to specific planting sites. In addition, comprehensive measurement statistics obtained in a commercial implementation would provide a valuable data base and nursery management tool.

Recommendations

This section presents recommendations for improvement of the grading environment, the algorithm, and possibilities for future research. One needed improvement which was apparent during the development of the algorithm is a higher level of illumination. Strobe illumination is desirable to stop the motion of the moving seedlings, however the strobe sources available for this study did not provide the intensity or the uniformity of illumination which is desirable. As stated in the literature review, lighting is a very important component of a vision system. Lenses used in this study were operated at close to maximum aperture, resulting in a shallow depth of field. Higher illumination levels would allow lens openings to be decreased, resulting in improved lens performance.

Consultation with forest nursery experts revealed that seedlings might be expected to carry considerable moisture on needles and roots when graded. This condition would significantly change the reflectance properties of the seedlings. It might only require changing thresholds, or it could necessitate the use of polarizing filters to eliminate specular reflections.

Recommended changes to the algorithm are primarily related to grading criteria and are coincidental with recommendations for further research. First, the shoot area could easily be measured, allowing a calculation of the root/shoot volume ratio (index). A calculation of the sturdiness ratio (caliper/height) is also straightforward. Second, a data base collected with the algorithm would enable statistical determination of optimal cutoff values, and hence improve classification performance. A training routine could be developed to assist in cutoff selection after a set of training seedlings had been processed. Finally, the accuracy of grading demonstrated by this algorithm suggests use for classification of seedlings into several acceptable grades. Additional grades might be optimal for specific types of planting sites.

REFERENCES CITED

Ardalan, S. H. and A. E. Hassan. 1981. Automatic feeding and sorting of bare root seedlings. ASAE Paper No. 81-1082. ASAE. St. Joseph, MI 49085.

Ballard, D. H. and C. M. Brown. 1982. Computer vision. Prentice-Hall, Inc., Englewood Cliffs, NJ 07632.

Baxes, G. A. 1984. Digital image processing. Prentice-Hall, Inc., Englewood Cliffs, NJ 07632.

Beckman, W. 1986. Personal contact. Weyerhaeuser Nursery, Fort Towson, OK.

Buckley, D. J., W. S. Reid and K. A. Armson. 1978. A digital recording system for measuring root area and dimensions of tree seedlings. Transactions of the ASAE. 21(2):222 - 226.

Cambier, J. L. and D. C. Pasiak. 1986. On-line X-ray inspection of packaged foods. Vision '86 conference proceedings. Machine vision association of SME. Dearborn, MI. 1:1 - 14.

Castleman, K. R. 1979. Digital Image Processing. Prentice-Hall, Inc., Englewood Cliffs, NJ 07632.

Chu, Y. C. 1986. Resolution and accuracy in grey level machine vision. Vision '86 conference proceedings. Machine Vision Association of SME. Dearborn, MI. 6:1 - 12.

Doty, J. L. 1986. Practical optics for machine vision systems. Robotics Engineering. 8(6):4 - 7.

Dunbar, P. 1986. Machine vision - an examination of what's new in vision hardware. Byte. 11(1):161 - 173.

Duryea, M. L. and T. D. Landis. 1984. Forest nursery manual: production of bareroot seedlings. Forest Research Laboratory, Oregon State University, Corvallis, OR. pp. 232 - 255.

Englander, A. C. 1986. Edge detection techniques for industrial machine vision. Vision '86 Conference

Proceedings. Machine Vision Association of SME. Dearborn, MI. 5:85 - 105.

Forward, P. W. 1982. Stock production specifications - bareroot stock. Artificial Regeneration of Conifers in the Upper Great Lakes Region. Michigan Technological University, Houghton, MI. pp. 260 - 268.

Graham, L. F. and R. P. Rohrbach. 1983. Mechanical singulation of bare root pine seedlings. ASAE Paper No. 83-1093. ASAE. St. Joseph, MI 49085.

Hines, R. L., F. E. Sistler and M. E. Wright. 1986. A vision system for grading container-grown plants. ASAE Paper No. 86-3043. ASAE. St. Joseph, MI 49085.

Huang, B. K., A. Kato, Y. Kano and E. P. Harris. 1986. Non-contact micrometer for continuous monitoring of plant and fruit growth. ASAE Paper No. 86-3033. ASAE. St. Joseph, MI 49085.

International Robomation/Intelligence. 1985a. D256 Reference Manual. IRI. 2281 Las Palmas Drive. Carlsbad, CA. 92008.

International Robomation/Intelligence. 1985b. Iconic Kernel System Manual. IRI. 2281 Las Palmas Drive. Carlsbad, CA. 92008.

Kranzler, G. A. 1984. Digital image processing for agriculture. ASAE Paper No. 84-5513. ASAE. St. Joseph, MI 49085.

Kranzler, G. A., D. R. Devoe and J. P. Gentry. 1984. Opto-electronics and image processing for seedling counting in tree nursery beds. ASAE Paper No. 84-1091. ASAE. St. Joseph, MI 49085.

Lawyer, J. N. 1981. Mechanization of nursery production of bare root deciduous planting stock. Forest Regeneration. ASAE publication 10-81, pp. 30 - 37. ASAE. St. Joseph, MI 49085.

Levine, M. D. 1985. Vision in man and machine. McGraw-Hill, Inc., New York, NY.

Maw, B. W., C. W. Suggs and P. Harris. 1980. Progress in the mechanical singulation and sorting of seedlings. ASAE Paper No. 80-1055. ASAE. St. Joseph, MI 49085.

Maw, B. W., S. J. Thomson and W. A. Cromer. 1985. Optical methods for detecting seedlings. ASAE Paper No. 85-1086. ASAE. St. Joseph, MI 49085.

May, J. T., E. W. Belcher, Jr., C. E. Cordell, T. H. Filer, Jr., and D. South. 1982. Southern Pine Nursery Handbook. Forest Service, USDA.

McLendon, B. D. and J. M. Allison. 1986. Field evaluation of pine seedling counter. ASAE Paper No. 86-1094. ASAE. St. Joseph, MI 49085.

Mersch, S. H. 1984. Polarized lighting for machine vision applications. Proceedings, Third Annual Applied Machine Vision Conference. SME. Chicago, IL. 4:40 - 54.

Meyer, G. E. and D. A. Davison. 1985. An electronic image plant growth measurement system. ASAE Paper No. 85-3548. ASAE. St. Joseph, MI 49085.

Novini, A. 1985. Before you buy a vision system. Manufacturing Engineering. 49(3):42 - 48.

Novini, A. 1986. Fundamentals of machine vision lighting. Vision. 3(2):18 - 20.

Paulsen, M. R. and W. F. McClure. 1985. Illumination for computer vision systems. ASAE Paper No. 85-3546. ASAE. St. Joseph, MI 49085.

Preston, C. and F. Molinari. 1986. Applications proliferate as digital image processing enters mainstream design. Electronic Design. 34(5):21 - 29

Sarkar, N. and R. R. Wolfe. 1985. Feature extraction techniques for sorting tomatoes by computer vision. Transactions of the ASAE, 28(3):970 - 979.

Schreiber, R. R. 1985. Quality control with vision. Vision. 2(4):1 - 2.

Steel, R. G. D. and J. H. Torrie. 1980. Principles and procedures of statistics; a biometrical approach. McGraw-Hill, Inc., New York, NY.

Sutton, J. F. and B. D. McLendon. 1985. Development of an optoelectronic pine tree seedling counter. ASAE Paper No. 85-3044. ASAE. St. Joseph, MI 49085.

Tabatabai, A. J. and O. R. Mitchell. 1984. Edge location to subpixel values in digital imagery. Transactions on Pattern Analysis and Machine Intelligence. IEEE. PAMI-6(2):188 - 201.

Tappan, J. H., M. E. Wright and F. E. Sistler. 1986. Error sources in digital image analysis. ASAE Paper No. 86-3040. ASAE. St. Joseph, MI 49085.

Taylor, R. W., and G. E. Rehkugler. 1985. Development of a system for automated detection of apple bruises. Proceedings of the Vision 85 Conference, 6:55 - 65.

Valenty, C. and P. Kraska. 1984. Dedicated hardware for pattern recognition aids machine vision. Computer Technology Review, Summer.

Wakeley, P. C. 1969. Results of southern pine planting experiments established in the middle twenties. Journal of Forestry. 67:237 - 241.

Weaver, G. H., B. Izlar, F. S. Broerman, and K. Zydias. 1981. Southern forest industry and state forest tree nursery practices. Forest Regeneration. ASAE Publication no. 10-81. pp. 53 - 55. ASAE, St. Joseph, MI. 49085.

Wolfe, R. R. and W. E. Sandler. 1985. An algorithm for stem detection using digital image analysis. Transactions of the ASAE. 28(2):641 - 644. ASAE. St. Joseph, MI 49085.

APPENDIX A

SIX TECHNIQUES FOR THE MEASUREMENT OF DOWEL CALIPER

MEASUREMENTS ON 30 DOWEL IMAGES USING
BINARY THRESHOLDING AT GREY LEVEL 120
METHOD 1

CALIPER MM	DEVIATION FROM 3.0	ORIENTATION RADIANS	COLUMN POSITION
2.6300	-0.37	0.004	104
3.0390	0.04	0.007	107
3.0243	0.02	0.011	114
3.1266	0.13	0.010	139
2.9801	-0.02	0.019	144
3.1267	0.13	0.008	156
3.1405	0.14	0.024	143
2.7317	-0.27	0.021	134
2.9800	-0.02	0.021	125
2.9797	-0.02	0.026	116
3.0677	0.07	0.020	105
3.0812	0.08	0.033	117
3.0234	0.02	0.026	129
3.0237	0.02	0.022	138
3.0823	0.08	0.021	148
3.0959	0.10	0.033	153
3.0663	0.07	0.037	138
3.0504	0.05	0.046	128
3.0211	0.02	0.048	115
3.0096	0.01	0.015	128
3.0243	0.02	0.011	143
3.0094	0.01	0.018	147
3.0241	0.02	0.016	135
2.8725	-0.13	0.064	109
2.8909	-0.11	0.038	101
2.8778	-0.12	0.021	119
3.0238	0.02	0.021	121
2.9217	-0.08	0.019	125
3.1560	0.16	0.004	133
2.8053	-0.19	0.002	137

STATISTICS ON CALIPER MEASUREMENTS
FROM METHOD 1

CALIPER MM	SCALED CALIPER	ABS. VAL. OF DEV. OF SCALED CAL. FROM 3.0	
2.630000	2.633336	0.366664	
3.039000	3.042855	0.042855	
3.024300	3.028136	0.028136	
3.126600	3.130566	0.130566	
2.980100	2.983880	0.016120	
3.126700	3.130666	0.130666	
3.140500	3.144483	0.144483	
2.731700	2.735165	0.264835	
2.980000	2.983780	0.016220	
2.979700	2.983479	0.016521	
3.067700	3.071591	0.071591	
3.081200	3.085108	0.085108	
3.023400	3.027235	0.027235	
3.023700	3.027535	0.027535	
3.082300	3.086210	0.086210	
3.095900	3.099827	0.099827	
3.066300	3.070189	0.070189	
3.050400	3.054269	0.054269	
3.021100	3.024932	0.024932	
3.009600	3.013417	0.013417	
3.024300	3.028136	0.028136	
3.009400	3.013217	0.013217	
3.024100	3.027936	0.027936	
2.872500	2.876144	0.123857	
2.890900	2.894567	0.105433	
2.877800	2.881450	0.118550	
3.023800	3.027635	0.027635	
2.921700	2.925406	0.074594	
3.156000	3.160003	0.160003	
2.805300	2.808858	0.191142	
2.996199	3.000000	0.086263	mean
0.014231	0.014263	0.006566	var
0.119292	0.119429	0.081033	stdev

MEASUREMENTS ON 30 DOWEL IMAGES USING
MOMENTS TECHNIQUE ON 30 PIXELS, 35 LINES
METHOD 2

CALIPER MM	DEVIATION FROM 3.0	ORIENTATION RADIANS	COLUMN POSITION
2.7181	-0.28	0.017	103
2.9139	-0.09	0.024	106
2.9843	-0.02	0.040	113
2.9671	-0.03	0.047	138
2.8958	-0.10	0.056	142
2.8796	-0.12	0.048	154
2.9831	-0.02	0.057	142
2.7397	-0.26	0.054	133
2.9207	-0.08	0.054	124
2.9007	-0.10	0.051	115
2.9199	-0.08	0.031	105
2.9271	-0.07	0.056	115
2.9281	-0.07	0.066	128
2.9442	-0.06	0.058	137
2.9499	-0.05	0.063	146
2.9275	-0.07	0.067	151
2.9059	-0.09	0.066	137
2.9311	-0.07	0.067	126
2.9476	-0.05	0.060	114
2.9261	-0.07	0.056	126
2.9116	-0.09	0.054	142
2.8890	-0.11	0.058	145
2.9057	-0.09	0.055	134
2.7758	-0.22	0.052	108
2.8119	-0.19	0.038	101
2.8594	-0.14	0.053	117
2.8497	-0.15	0.052	119
2.9020	-0.10	0.051	124
2.9043	-0.10	0.038	132
2.8408	-0.16	0.041	136

STATISTICS ON CALIPER MEASUREMENTS
FROM METHOD 2

CALIPER MM	SCALED CALIPER	ABS. VAL. OF DEV. OF SCALED CAL. FROM 3.0	
2.718100	2.816341	0.183659	
2.913900	3.019217	0.019217	
2.984300	3.092162	0.092162	
2.967100	3.074340	0.074340	
2.895800	3.000463	0.000463	
2.879600	2.983678	0.016322	
2.983100	3.090918	0.090918	
2.739700	2.838722	0.161279	
2.920700	3.026263	0.026263	
2.900700	3.005540	0.005540	
2.919900	3.025434	0.025434	
2.927100	3.032894	0.032894	
2.928100	3.033931	0.033931	
2.944200	3.050612	0.050612	
2.949900	3.056519	0.056519	
2.927500	3.033309	0.033309	
2.905900	3.010928	0.010928	
2.931100	3.037039	0.037039	
2.947600	3.054135	0.054135	
2.926100	3.031858	0.031858	
2.911600	3.016834	0.016834	
2.889000	2.993417	0.006583	
2.905700	3.010721	0.010721	
2.775800	2.876126	0.123874	
2.811900	2.913531	0.086469	
2.859400	2.962748	0.037252	
2.849700	2.952697	0.047303	
2.902000	3.006887	0.006887	
2.904300	3.009271	0.009270	
2.840800	2.943476	0.056525	
2.895353	3.000000	0.047951	mean
0.004137	0.004439	0.002062	var
0.064321	0.066624	0.045411	stdev

MEASUREMENTS ON 30 DOWEL IMAGES USING
MODIFIED LAPLACIAN EDGE DETECTOR
METHOD 3

CALIPER MM	DEVIATION FROM 3.0	ORIENTATION RADIANS	COLUMN POSITION
3.4628	0.46	0.004	104
3.6819	0.68	0.008	107
3.6818	0.68	0.010	114
3.9741	0.97	0.008	139
3.9150	0.91	0.020	144
3.8573	0.86	0.006	156
3.7249	0.72	0.023	143
3.6812	0.68	0.020	134
3.8126	0.81	0.021	125
3.7685	0.77	0.025	116
3.8566	0.86	0.020	105
3.7530	0.75	0.033	117
3.8414	0.84	0.026	129
3.7541	0.75	0.022	138
3.7396	0.74	0.021	147
3.8002	0.80	0.033	153
3.7380	0.74	0.036	138
3.7511	0.75	0.046	128
3.7362	0.74	0.047	115
3.8861	0.89	0.015	127
3.7548	0.75	0.011	143
3.7253	0.73	0.017	147
3.7400	0.74	0.015	135
3.6453	0.65	0.064	109
3.5626	0.56	0.037	101
3.6813	0.68	0.020	119
3.6959	0.70	0.020	121
3.6814	0.68	0.018	125
4.0619	1.06	0.002	133
3.8865	0.89	0.002	137

STATISTICS ON CALIPER MEASUREMENTS
FROM METHOD 3

CALIPER MM	SCALED CALIPER	ABS. VAL. OF DEV. OF SCALED CAL. FROM 3.0	
3.462800	2.761615	0.238386	
3.681900	2.936348	0.063652	
3.681800	2.936269	0.063731	
3.974100	3.169381	0.169381	
3.915000	3.122248	0.122248	
3.857300	3.076232	0.076232	
3.724900	2.970641	0.029359	
3.681200	2.935790	0.064210	
3.812600	3.040583	0.040583	
3.768500	3.005413	0.005413	
3.856600	3.075673	0.075673	
3.753000	2.993052	0.006948	
3.841400	3.063551	0.063551	
3.754100	2.993929	0.006071	
3.739600	2.982365	0.017635	
3.800199	3.030694	0.030694	
3.738000	2.981089	0.018911	
3.751100	2.991536	0.008464	
3.736200	2.979653	0.020347	
3.886100	3.099200	0.099200	
3.754800	2.994487	0.005513	
3.725300	2.970960	0.029040	
3.740000	2.982684	0.017316	
3.645300	2.907160	0.092840	
3.562600	2.841206	0.158794	
3.681300	2.935870	0.064130	
3.695900	2.947514	0.052486	
3.681400	2.935950	0.064050	
4.061900	3.239402	0.239402	
3.886500	3.099519	0.099519	
3.761713	3.000000	0.068126	mean
0.013887	0.008829	0.004027	var
0.117841	0.093963	0.063462	stdev

MEASUREMENTS ON 30 DOWEL IMAGES USING
MODIFIED LAPLACIAN EDGE DETECTOR
METHOD 4

CALIPER MM	DEVIATION FROM 3.0	ORIENTATION RADIANS	COLUMN POSITION
3.5943	0.59	0.004	104
3.6819	0.68	0.008	107
3.6818	0.68	0.010	114
4.2079	1.21	0.008	139
3.9968	1.00	0.020	144
3.8719	0.87	0.006	156
3.7541	0.75	0.023	143
3.7105	0.71	0.020	134
3.8711	0.87	0.021	125
3.7831	0.78	0.025	116
3.8712	0.87	0.020	105
3.8114	0.81	0.033	117
3.8414	0.84	0.026	129
3.9733	0.97	0.022	138
3.7980	0.80	0.021	147
3.7728	0.77	0.033	153
3.7380	0.74	0.036	138
3.8241	0.82	0.046	128
3.7800	0.78	0.047	115
3.9007	0.90	0.015	127
3.7841	0.78	0.011	143
3.7691	0.77	0.017	147
3.7838	0.78	0.015	135
3.6891	0.69	0.064	109
3.6210	0.62	0.037	101
3.6813	0.68	0.020	119
3.7397	0.74	0.020	121
3.6814	0.68	0.018	125
4.1349	1.13	0.002	133
3.9742	0.97	0.002	137

STATISTICS ON CALIPER MEASUREMENTS
FROM METHOD 4

CALIPER MM	SCALED CALIPER	ABS. VAL. OF DEV. OF SCALED CAL. FROM 3.0	
3.594300	2.829590	0.170410	
3.681900	2.898553	0.101447	
3.681800	2.898474	0.101526	
4.207900	3.312643	0.312643	
3.996800	3.146456	0.146456	
3.871900	3.048129	0.048129	
3.754100	2.955392	0.044608	
3.710500	2.921068	0.078932	
3.871100	3.047500	0.047500	
3.783100	2.978222	0.021778	
3.871200	3.047579	0.047578	
3.811400	3.000501	0.000501	
3.841400	3.024118	0.024118	
3.973299	3.127956	0.127956	
3.798000	2.989952	0.010048	
3.772800	2.970114	0.029886	
3.738000	2.942718	0.057282	
3.824100	3.010499	0.010499	
3.780000	2.975782	0.024218	
3.900700	3.070802	0.070802	
3.784100	2.979009	0.020991	
3.769100	2.967201	0.032799	
3.783800	2.978773	0.021227	
3.689100	2.904221	0.095779	
3.621000	2.850610	0.149390	
3.681300	2.898081	0.101919	
3.739700	2.944056	0.055944	
3.681400	2.898159	0.101841	
4.134900	3.255174	0.255174	
3.974200	3.128664	0.128664	
3.810763	3.000000	0.081335	mean
0.019489	0.012083	0.005236	var
0.139604	0.109922	0.072364	stdev

MEASUREMENTS ON 30 DOWEL IMAGES USING
GREY LEVEL GRADIENT EDGE DETECTOR
METHOD 5

CALIPER MM	DEVIATION FROM 3.0	ORIENTATION RADIANS	COLUMN POSITION
2.5277	-0.47	0.004	104
2.8345	-0.17	0.008	107
2.9221	-0.08	0.010	114
2.6591	-0.34	0.008	139
2.5976	-0.40	0.020	144
2.6592	-0.34	0.006	156
2.6878	-0.31	0.023	143
2.3227	-0.68	0.020	134
2.6440	-0.36	0.021	125
2.7314	-0.27	0.025	116
2.7317	-0.27	0.020	105
2.6140	-0.39	0.033	117
2.4831	-0.52	0.026	129
2.5417	-0.46	0.022	138
2.6440	-0.36	0.021	147
2.6286	-0.37	0.033	153
2.6137	-0.39	0.036	138
2.6418	-0.36	0.046	128
2.5979	-0.40	0.047	115
2.6297	-0.37	0.015	127
2.7321	-0.27	0.011	143
2.5127	-0.49	0.017	147
2.6005	-0.40	0.015	135
2.4205	-0.58	0.064	109
2.4968	-0.50	0.037	101
2.6295	-0.37	0.020	119
2.4834	-0.52	0.020	121
2.5565	-0.44	0.018	125
2.6141	-0.39	0.002	133
2.5423	-0.46	0.002	137

STATISTICS ON CALIPER MEASUREMENTS
FROM METHOD 5

CALIPER MM	SCALED CALIPER	ABS. VAL. OF DEV. OF SCALED CAL. FROM 3.0	
2.527700	2.905376	0.094624	
2.834500	3.258017	0.258017	
2.922100	3.358706	0.358706	
2.659100	3.056410	0.056410	
2.597600	2.985720	0.014280	
2.659200	3.056524	0.056525	
2.687800	3.089398	0.089398	
2.322700	2.669746	0.330254	
2.644000	3.039053	0.039053	
2.731400	3.139512	0.139512	
2.731700	3.139857	0.139857	
2.614000	3.004571	0.004571	
2.483100	2.854113	0.145888	
2.541700	2.921468	0.078532	
2.644000	3.039053	0.039053	
2.628600	3.021353	0.021352	
2.613700	3.004226	0.004226	
2.641800	3.036525	0.036525	
2.597900	2.986065	0.013935	
2.629700	3.022617	0.022617	
2.732100	3.140317	0.140317	
2.512700	2.888135	0.111865	
2.600500	2.989054	0.010946	
2.420500	2.782159	0.217841	
2.496800	2.869859	0.130141	
2.629500	3.022387	0.022387	
2.483400	2.854457	0.145543	
2.556500	2.938479	0.061521	
2.614100	3.004686	0.004686	
2.542300	2.922158	0.077842	
2.610023	2.999999	0.095547	mean
0.013786	0.018217	0.008768	var
0.117412	0.134970	0.093639	stdev

MEASUREMENTS ON 30 DOWEL IMAGES USING
GREY LEVEL GRADIENT EDGE DETECTOR
METHOD 6

CALIPER MM	DEVIATION FROM 3.0	ORIENTATION RADIANS	COLUMN POSITION
2.6300	-0.37	0.004	104
3.1267	0.13	0.008	107
3.0244	0.02	0.010	114
2.7030	-0.30	0.008	139
2.7532	-0.25	0.020	144
2.7615	-0.24	0.006	156
2.8630	-0.14	0.023	143
2.3665	-0.63	0.020	134
2.6878	-0.31	0.021	125
2.8629	-0.14	0.025	116
2.9070	-0.09	0.020	105
2.6578	-0.34	0.033	117
2.6437	-0.36	0.026	129
2.6586	-0.34	0.022	138
2.7755	-0.22	0.021	147
2.6870	-0.31	0.033	153
2.7013	-0.30	0.036	138
2.6856	-0.31	0.046	128
2.7292	-0.27	0.047	115
2.6297	-0.37	0.015	127
2.8928	-0.11	0.011	143
2.6588	-0.34	0.017	147
2.8196	-0.18	0.015	135
2.5663	-0.43	0.064	109
2.5698	-0.43	0.037	101
2.7902	-0.21	0.020	119
2.6733	-0.33	0.020	121
2.7610	-0.24	0.018	125
2.6884	-0.31	0.002	133
2.6300	-0.37	0.002	137

STATISTICS ON CALIPER MEASUREMENTS
FROM METHOD 6

CALIPER MM	SCALED CALIPER	ABS. VAL. OF DEV. OF SCALED CAL. FROM 3.0	
2.630000	2.889948	0.110052	
3.126700	3.435741	0.435741	
3.024400	3.323330	0.323330	
2.703000	2.970163	0.029837	
2.753200	3.025325	0.025325	
2.761500	3.034445	0.034445	
2.863000	3.145977	0.145977	
2.366500	2.600403	0.399597	
2.687800	2.953460	0.046540	
2.862900	3.145867	0.145867	
2.907000	3.194326	0.194326	
2.657800	2.920495	0.079505	
2.643700	2.905002	0.094998	
2.658600	2.921375	0.078625	
2.775500	3.049829	0.049829	
2.687000	2.952582	0.047418	
2.701300	2.968295	0.031705	
2.685600	2.951043	0.048957	
2.729200	2.998952	0.001048	
2.629700	2.889618	0.110382	
2.892800	3.178723	0.178723	
2.658800	2.921594	0.078406	
2.819600	3.098288	0.098288	
2.566300	2.819952	0.180048	
2.569800	2.823798	0.176203	
2.790200	3.065982	0.065982	
2.673300	2.937527	0.062473	
2.761000	3.033896	0.033895	
2.688400	2.954120	0.045880	
2.630000	2.889948	0.110052	
2.730153	3.000000	0.115448	mean
0.020795	0.025109	0.011320	var
0.144203	0.158457	0.106393	stdev

```
/* PROGRAM mdi.c : calculates caliper using moments */

#include <ikp.h>
#define rllen 10

main()
{
     int i;
     vopen("/dev/vdg");
     ikpInit();
     Ivoff();
     lcon();
     for( i = 1 ; i <= 10 ; ++i )
          diam();
}

diam()
{
     int j,i = 0;
     int codd,ceven,cntr,dist,line,ntrans,n;
     int dthr = 130;
     float x,m1,m2,m3,sig2,sbar,p1=0;
     float stemdi,scale1= .526;
     char rnblk[rllen * 1024];
     Ifcopy(1,3);
     Ip_frame(3);
     lp_single(dthr);
     Ibinary();
     Ip_frame(1);
     n = Irlen(3,rnblk,rllen);
     if(n == 0) printf("\nRunlength failure");
     for(line = 160;line <= 170; ++line){
          ntrans = Ig_rx(line);
          if( ntrans == 2 ){
               m1 = m2 = m3 = 0;
               codd = Ig_ry(line, 0 );
               ceven = Ig_ry(line, 1 );
               cntr = ( codd + ceven ) / 2;
               for( j = cntr-15 ; j < cntr+15 ; ++j ){
                    x = (float) Ig_pix(line,j);
                    m1 += x / 30.0;
                    m2 += x * x / 30.0;
                    m3 += x * x * x / 30.0;
                    }
               sig2 = m2 - m1 * m1;
               sbar = (m3 + 2*m1*m1*m1 - 3*m1*m2) / (sig2 * sqrt
     (sig2));
               p1 += .5 - .5*sbar*sqrt(1/(4+sbar*sbar));
               ++i;
               }
          }
     stemdi = scale1 * p1 * 30.0 / (float) i;
}
```

RESULTS OF TIME TEST OF CALIPER MEASUREMENT TECHNIQUES

Method Tested	Seconds
Method 1: Grey Level Thresholding	0.022
Method 2: Moments	0.560
Method 3: Laplacian Edge Detector (3X3)	0.028
Method 4: Laplacian Edge Detector (5X5)	0.031
Method 5: Gradient Edge Detector	0.038
Method 6: Modified Gradient Edge Detector	0.035

APPENDIX B

PINE SEEDLING GRADING ALGORITHM

PROGRAM LISTING

```
/**********************************************************/
/*                   PROGRAM seed1.c                     */
/*                     written by                        */
/*                  Michael P. Rigney                    */
/**********************************************************/


#include <ikp.h>              /*iconic kernel library          */
#include <math.h>             /*math library                   */
#include <stdio.h>            /*standard I/O library           */
#define MIN 3                 /*min acceptable stem diameter   */
#define MAX 8                 /*max acceptable stem diameter   */
#define MINL 18               /*minimum stem length in cm      */
#define MINA 200              /*minimum root area in pixels    */
#define rllen 10              /*# blocks for runlength data    */

main()
{
    char c;
    int i=0,j=0,k=0;          /*init accept/cull counters      */
    int wtr,wa;               /*waitfor binary and area thresh*/
    int rtr,ltr;              /*root area and length bin thrs */
    int ctr,ctr2;             /*collar thresholds 1 and 2      */
    int dtr,dtr2;             /*diameter thresholds 1 & 2      */
    int otr,ang;              /*orientation  thr and angle     */
    int a1,b1,a2,b2;          /*scale & frame conversion coors*/
    int first,last,loop;      /*loop variables and counter     */
    int n;                    /*variable for returned values   */
    short w[4];               /*waitfor window coordinates     */
    short right[10];          /*convolution coefficient array */
    int collar,collar2;       /*root collar location in f1, f4*/
    int center,lines;         /*root collar location in f1     */
    int roota,length;         /*root area and stem length      */
    float rad;                /*orientation angle in radians   */
    float stemdi;             /*stem diameter                  */
    float s1=.410,s2=2.1818;       /*scale factors             */
    FILE *fp;                 /*pointer to statistics file     */


    vopen("/dev/vdg");        /*open video interface           */
    ikpInit();                /*initialize ikp variables       */

    Icon();                   /*turn coprocessor on            */

    lwoff();
    edge(right);              /*initialize convolution matrix */

    /*****      CALL CALIBRATION SUBROUTINE            ******
    ******                                             *****/
cal:
calibrate(&s1,&s2,&a1,&b1,&a2,&b2,&wtr,w,&wa,&otr,&ctr,&ctr2,
    &dtr,&dtr2,&rtr,&ltr);
run:                      /*label for repeating measurements  */
```

```
statfile(&fp);      /*open & initialize statistics file */
printf("Enter the number of seedlings to be measured:");
scanf("%d \n",&last);

/*****              MAIN PROGRAM LOOP                  ******
******                                                 *****/

for( loop = 1 ; loop <= last ; ++loop)
{

/*****           CALL WAITFOR SUBROUTINE               ******
******                                                 *****/

waitfor(wtr,w,wa);
tinue(1);

/*****           CALL ORIENTATION SUBROUTINE           ******
******                                                 *****/

orient(otr,&ang,&rad);
if( ang < -30 || ang > 30 ){
    printf("Orientation greater than 30 degrees!!!\n");
    grade(&fp,0.0,0,0,0,0,ang,&k,"NONE");
    continue;
}

/*****          CALL ROOT COLLAR SUBROUTINE            ******
******                                                 *****/

n = col1(&collar,&center,&lines,ctr);
if( n == 0 ) n = col1(&collar,&center,&lines,ctr2);
if( n == 0 ) {
    printf("Can not find root collar!!!\n");
    grade(&fp,0.0,0,0,0,0,ang,&k,"NONE");
    continue;
}

/*****        CALL STEM CALIPER SUBROUTINE             ******
******                                                 *****/

n = diam(collar,center,lines,right,s1,&stemdi,dtr,rad);
if(n==0) n= diam(collar,center,lines,right,s1,&stemdi,dtr
2,rad);
if( n == 0 ) {
    printf("Can not measure stem calliper!!!\n");
    grade(&fp,0.0,0,0,collar,center,ang,&k,"NONE");
    continue;
}

/*****  Calculate collar location in image from cam2 */

collar2 = a2 + (collar-a1)*(b2-a2)/(b1-a1);
tinue(4);
```

```
/*****          CALL ROOT AREA INDEX /                    ******
******         SHOOT LENGTH SUBROUTINE                    *****/

n = root(collar2,rtr,ltr,&roota,&length,s2,rad);
if( n == 0 ) {
    printf("Can not measure root area or stem length!!!\n
");
    grade(&fp,stemdi,0,0,collar,center,ang,&k,"NONE");
    continue;
}

/*****              ASSIGN GRADE              ******
******                                        *****/

if( stemdi > MIN && stemdi < MAX )
    if( roota > MINA && length > MINL )
        grade(&fp,stemdi,roota,length,collar,center,ang,&
i,"A1");

    else
        grade(&fp,stemdi,roota,length,collar,center,ang,&
j,"C2");

else
    if( stemdi < MIN && stemdi > 2.8 )
        if( roota > MINA + 50 && length > MINL )
           grade(&fp,stemdi,roota,length,collar,center,an
g,&i,"A2");

        else
           grade(&fp,stemdi,roota,length,collar,center,an
g,&j,"C3");

else
    grade(&fp,stemdi,roota,length,collar,center,ang,&j,"C
1");
}

/*****                 END OF MAIN PROGRAM      ******
******                        LOOP              *****/

fclose(fp);                /*close statistics file */

printf("Enter c to recalibrate, else <cr> :\n");
if( c = getchar() == 'c' ) goto cal;

printf("Enter m to measure more seedlings, else <cr> to e
xit :\n");
if( c = getchar() == 'm' ) goto run;
}
```

```
/*****          WAITFOR SUBROUTINE          ******
******                                      *****/

waitfor(thr,w,wa)
int thr,wa;                    /*threshold, area              */
short *w;                      /*window coordinates           */
{
    MOMRES abc;                /*struct for moment data       */
    short r = 0;               /* area variable               */

    Ip_camera(2);              /*cam 2 current                */
    Ip_tr(thr);                /*init moment threshold        */
    Ip_frame(2);               /*frame 2 current              */
    Itbwin(w);                 /*w is hardware window         */
    Iinside();                 /*0's out, 4's inside window*/
    Ip_frame(1);               /*frame 1 current              */
    while( r < wa ){           /*while frame 1 area < wa      */
        Isnap();               /*snap frame 1                 */
        Ifmul(1,2,1);          /*mask f1 with f2              */
        Iarea(&abc);           /*compute frame 1 area         */
        r = abc.Im0;
    }
    Istbon();                  /*enable strobe lamp sync      */
    Ip_camera(1);              /*cam 1 current                */
    Isnap();                   /*snap into frame 1            */
    Ip_camera(2);              /*cam 2 current                */
    Ip_frame(4);               /*frame 4 current              */
    Isnap();                   /*snap into frame 4            */
    Istboff();                 /*disable strobe sync          */
    Iwoff();                   /*hardware window off          */
}
```

```
/*****              ORIENTATION SUBROUTINE          ******
******                                              *****/


orient(otr,ang,rad)
int otr,*ang;                 /*threshold, degree addr     */
float *rad;                   /*radian address             */
{
    MOMRES abc;               /*struture for moment data   */

    Ip_frame(3);              /*frame 3 current            */
    Ifcopy(4,3);              /*copy f4 to f3              */
    Ip_tr(otr);               /*init moments threshold     */
    Ip_bdim(0,8);             /*hardware win line 8 down   */
    Imoments(&abc);           /*calc first three moments   */
    *rad = abc.Imajor[1];     /*orientation of major axis */
    *ang = (int) (57.3 * *rad);
    printf("Stem orientation = %d degrees,  = %f radians.\n",
    *ang,*rad);
    Ip_single(otr);           /*threshold for display      */
    Ibinary();
    Iwoff();                  /*hardware window off        */
    tinue(3);
}
```

```
/*****                ROOT COLLAR SUBROUTINE  ******
******                                        *****/


coll(col,cent,num,cthr)
int *col,*cent,*num,cthr;
{
    int n;
    char rnblk[rllen * 1024];

    Iwoff();
    Ifcopy(1,2);                /*copy f1 to f2             */
    Ip_frame(2);                /*frame 2 current           */
    Ip_single(cthr);            /*init threshold            */
    Ibinary();                  /*threshold f2              */
    n = Irlen(2,rnblk,rllen);   /*runlength encode          */
    if( n == 0 ){
        printf("Runlength failure!!!\n");
        return(0);
    }


    /******************************************************/
    /*  Try to find the root collar.  First on lines with *
     *  two transitions, then on lines with four or less  *
     *  transitions, and finally on lines with six or     *
     *  less transitions.                                 */
    /******************************************************/

    n = col2(2,col,cent,num);
    if( n == 0 ) n = col2(4,col,cent,num);
    if( n == 0 ) n = col2(6,col,cent,num);
    if( n != 0 ){
         printf("root collar is on line %d : center = %d\n",*
    col,*cent);
         Ihoriz(*col,*cent - 15,30);
    }
    tinue(2);
    return(n);
}
```

```
col2(numtrans,col,cent,num)
int numtrans,*col,*cent,*num;
{
    int i,k,n,trans,dist,max,set,line[200],centr[200],lnset[3
    ][30];
    /*****************************************************/
    /*  For each line with numtrans or less, find the      *
     *  transition pair with the largest span, and if      *
     *  that span is between 5 and 18 pixels, store the    *
     *  line number and center pixel of the span.          *
    /*****************************************************/
    int j = 0;
    for( i = 0 ; i < 240 ; ++i ){
        if( (n = Ig_rx(i)) <= numtrans ){
            max = 0;
            for( trans = 0 ; trans < n ; trans += 2 ){
                dist = Ig_ry(i,trans+1) - Ig_ry(i,trans);
                if(dist > max){
                    max = dist;
                    centr[j] = Ig_ry(i,trans) + dist / 2;
                }
            }
            if( max > 5 && max < 18 ) line[j++] = i;
        }
    }
    for( i = 0 ; i < 3 ; ++i )
        for( k = 0 ; k < 30 ; ++k ) lnset[i][k] = 0;

    /*****************************************************/
    /*  For sets of consecutive lines, in the array        *
     *  formed above, sum the line numbers, sum the        *
     *  center columns, and keep a count of the number     *
     *  of consecutive lines.                              *
     *  Bail if more than 30 or less than 1 set of lines.  *
    /*****************************************************/
    k = 0;
    for( i = 0 ; i < j ; ++i ){
        if( line[i+1]-line[i] == 1 ){
            lnset[0][k] += line[i];
            lnset[1][k] += centr[i];
            lnset[2][k] += 1;
        }
        else{
            lnset[0][k] += line[i];
            lnset[1][k] += centr[i];
            lnset[2][k] += 1;
            ++k;
        }
        if( k == 30 ) return(0);
    }
    if( k == 0 ) return(0);
```

```
/*******************************************************/
/*  Find the largest set of consecutive lines.  The    *
 *  collar is the average of the lines and is centered*
 *  at the average of the centers.                     *
 *  Limit the diameter measurement area to the number *
 *  of lines in that set or a maximum of 20 lines      *
/*******************************************************/

    max = 0;
    for( i = 0 ; i < k ; ++i ){
        if( lnset[2][i] > max ){
            max = lnset[2][i];
            set = i;
        }
    }
    if( lnset[2][set] < 6 ) return(0);
    *col = lnset[0][set] / lnset[2][set];
    *cent = lnset[1][set] / lnset[2][set];
    if( lnset[2][set] > 19 ) *num = 10;
    else *num = lnset[2][set] / 2;
    return(1);
}
```

```
/*****            DIAMETER SUBROUTINE         ******
******                                        *****/

diam(collar,c,lines,filt,scale1,stemdi,dthr,ang)
int collar,c,lines,dthr;
short *filt;
float scale1,*stemdi,ang;
{
    int codd,ceven,dist,ntrans,line,n;
    int sum = 0, i = 0;
    char rnblk[rllen * 1024];

    Ip_bdim(0,collar-lines);        /*hardware window on     */
    Ip_bdim(2,collar+lines+1);      /* about root collar     */
    Itbcoefs(filt);                 /*convol coefs to buffer*/
    Iconvol(1,2);                   /*convolution into f2    */
    tinue(2);
    Ip_frame(2);                    /*frame 2 current        */
    Ip_single(dthr);                /*threshold at dthr      */
    Ibinary();
    n = Irlen(2,rnblk,rllen);       /*runlength encode f3    */
    if(n == 0){
        printf("\nRunlength failure");
        return(0);
    }
```

```
/*****************************************************/
/*      For all candidate lines, get the number of      *
*       transitions.  If the number of transitions      *
*       is equal to or less than ntrans, find the       *
*       location of left members of transition pairs.   *
*       If two consecutive left members bracket the     *
*       center of the root collar, and are within 9     *
*       pixels of the center, the stem diameter on      *
*       that line is the distance between them.         *
/*****************************************************/

    for(line = collar-lines ; line < collar+lines ; ++line){
        ntrans = Ig_rx(line);       /*# of transitions - row*/
        if( ntrans >= 4 )
            for( n = 0 ; n < ntrans-2 ; n += 2 ){
            codd = Ig_ry(line,n);
            ceven = Ig_ry(line,n+2);
            if((codd < c && c-codd < 10)&&(ceven > c && ceven
    -c < 10)){
                sum += ( ceven - codd );
                ++i;
                break;
            }
        }
    }
    Iwoff();
    if( i != 0 ){
        *stemdi = scale1 * (float) sum * cos(ang) / (float) i
    ;
        printf("stem diameter = %f\n",*stemdi);
        Ihoriz(collar-lines-1,c-10,20);
        Ihoriz(collar+lines+1,c-10,20);
    }
    tinue(2);
    return(i);
}
```

```
/*****          ROOT AREA INDEX SUBROUTINE          *******
******          LENGTH MEASUREMENT                  ******/

root(collar,rthr,lthr,area,length,s2,ang)
int collar,rthr,lthr,*area,*length;
float s2;
{
    int n,ntrans,tnum,line,max,dist,codd,ceven;
    char rnblk[rllen * 1024];
    MOMRES abc;

    Ip_frame(3);                /*frame 3 current              */
    Idark();                    /*frame 3 all 0's              */
    Ip_bdim(0,collar);          /*set top of win at collar     */
    Itcoefs(5);                 /*5 x 5 Laplacian high pass */
    Iconvol(4,3);               /*convolution into frame 3     */
    tinue(3);
    Ip_single(rthr);            /*threshold f3 at rthr         */
    Ibinary();
    Iarea(&abc);                /*find area inside window      */
    *area = abc.Im0;
    printf("root area = %d\n",abc.Im0);
    Irect(Ig_winadr());         /*draw window                  */
    Iwoff();                    /*turn window off              */
    tinue(3);

    Ifcopy(4,3);                /*copy frame 4 to frame 3      */
    Ip_single(lthr);            /*threshold at lthr            */
    Ibinary();
    n = Irlen(3,rnblk,rllen);/*runlength encode frame 3    */
    if(n == 0){
        printf("\nRunlength failure");
        return(0);
    }
```

```
    /****************************************************/
    /*  Starting at the top of the image, find four      *
     *  consecutive lines with a maximum span of at      *
     *  least 5 pixels between a transition pair.        *
     *  Call the fourth line the shoot top.              *
    /****************************************************/

    line = 0;
    n = 0;
    while( n < 4 ){
         max = 0;
         ntrans = Ig_rx(++line);
         if( ntrans == 0 ) n = 0;
         for(tnum = 0; tnum < ntrans; tnum += 2){
              codd = Ig_ry(line, tnum);
              ceven = Ig_ry(line, tnum + 1);
              dist = ceven - codd;
              if( dist > max ) max = dist;
         }
         if( max > 5 ) ++n;
         if( line == collar ) break;
    }
    *length = ((collar - line) * s2 / cos(ang)) / 10;
    printf("stem length = %d centimeters.\n",*length);
    Ihoriz(collar,0,255);
    tinue(3);
    return(n);
}
```

```
/**********************************************************/
/*  This subroutine opens a file for the storage of      *
 *  statistics on the graded seedlings.  The user is     *
 *  asked for a filename and a header is written at the  *
 *  top of the file.                                     *
/**********************************************************/

statfile(ptr)
FILE **ptr;
{
    char name[15], fname[40];
    FILE *fopen();
    printf("Enter the filename for seedling statistics : ");
    scanf("%s \n",name);
    strcpy(fname,"/lusr/usr/mike/images/");
    strcat(fname,name);
    *ptr = fopen(fname, "w");
    fprintf(*ptr,"  STEM      ROOT     STEM     ROOT    COLLAR  S
    TEM\n");
    fprintf(*ptr,"CALIPER    AREA   LENGTH  COLLAR  CENTER  A
    NGLE  GRADE COUNT\n\n");
    fprintf(*ptr,"   mm     pixels    cm      line    column
    deg\n\n");
}


/**********************************************************/
/*  This subroutine is passed a list of seedling          *
 *  measurements, an assigned grade, and pointers to the  *
 *  statistics file and the grade counter.  Various       *
 *  statistics are printed on the terminal with the grade *
 *  and count.  All statistics are written to the         *
 *  statistics file.                                      *
/**********************************************************/

grade(ptr,di,area,len,col,ctr,ang,count,class)
float di;
int area,len,col,ctr,ang,*count;
char class[20];
FILE **ptr;
{
    printf("Stem diameter = %3.2f mm; root area = %d pixels\n
    ",di,area);
    printf("length = %3d cm; angle = %2d deg;  %4s #%d\n",len
    ,ang,class,++*count);

    fprintf(*ptr,"%6.1f%9d%7d%9d%8d%7d%11.7s#%d\n",di,area,le
    n,col,ctr,ang,class,*count);
}
```

```
/*********************************************************/
/*      This subroutine halts execution of the algorithm   *
 *      whenever called, and displays the frame number      *
 *      passed to it.                                       *
/*********************************************************/

tinue(num)
int num;
{
     char c;
     Ipf(num);
     printf("Press any key to continue :\n");
     Ieoff();
     c = getchar();
     Ieon();
}


/*********************************************************/
/*      This subroutine initializes the modified Laplacian *
 *      edge detector used in the diameter subroutine.     *
/*********************************************************/

edge(r)
short *r;
{
     int i;
     *r = 3;
     *(r+2) = *(r+8) = 0;
     for( i = 2 ; i < 9 ; i += 3 )
          *(r+i-1) = *(r+i+1) = Icfixed(2.625);
     *(r+5) = Icfixed(-16.0);
}
```

```
calibrate(s1,s2,xa1,xb1,xa2,xb2,wtr,w,wa,otr,ctr,ctr2,dtr,dtr
     2,rtr,ltr)
int *xa1,*xb1,*xa2,*xb2,*wtr,*wa,*otr,*ctr,*ctr2,*dtr,*dtr2,*
     rtr,*ltr;
short *w;
float *s1,*s2;
{
     int n = 7, img;
     int ya1,yb1,ya2,yb2;
     float fratio;
     short filt[10],fov[4];
     char c;
char *msg1 = "Set this threshold high to show";
char *msg2 = "visible needles and make the root collar the";
char *msg3 = "only area with a small number of transitions.";

char *msg4 = "Set this threshold low to show";
char *msg5 = "only the stem and large branches.";
char *msg6 = "just enough edges to allow caliper measurement.
     ";
char *msg7 = "well defined edges for caliper measurement.";
char *msg8 = "the seedling top for length measurement.";
char *msg9 = "as many roots as possible but minimize noise.";

char *msg0 = "the stem and branches clearly.";
char *msgA = "the major axis of the stem.";
     *wtr = 25;  *wa = 50;     *ctr = 90;  *ctr2 = 140;
     *dtr = 57; *dtr2 = 40; *rtr = 45; *ltr = 100; *otr = 170;

     *xa1 = 7;   *xb1 = 235;  *xa2 = 136;  *xb2 = 191;
     *w = 136;   *(w+1) = 125; *(w+2) = 180; *(w+3) = 140;
     ya1 = 118;  ya2 = 128;   fratio = .24123;

     edge(filt);
     Ipcoefs(filt);
     Ivon();
     printf("Turn strobe lamps on.\n");
     printf("Place a seedling in the field of view.\n");
     printf("Adjust camera TWO and strobe lamps.\n");
     keysnap(2,4,1);
     printf("Adjust camera ONE.\n");
     keysnap(1,1,1);
```

```
printf("Do you wish to callibrate the scale factors? (y/n
) : ");
scanf("%c \n",&c);
if( c == 'y' || c == 'Y' )
     scales(s1,s2,xa1,xb1,xa2,xb2,&ya1,&yb1,&ya2,&yb2,*dtr
,filt);

fratio = (*xb2 - *xa2) / (float) (*xb1 - *xa1);
*fov      = (int) (*xa2 - *xa1 * fratio);
*(fov+1) = (int) (ya2 - ya1 * fratio);
*(fov+2) = (int) (*xa2 + (255 - *xa1) * fratio);
*(fov+3) = (int) (ya2 + (255 - ya1) * fratio);
printf("ya2 = %d, ya1 = %d\n",ya2,ya1);
printf("xa1= %d, xb1= %d, xa2= %d, xb2= %d\n",*xa1,*xb1,*
xa2,*xb2);

n = wish(&n,"default");
if( n == 1 ) {
     printf("Place a seedling in the field of view.\n");
     keysnap(2,2,0);
     keysnap(1,1,1);
     printf("For next image; hold key down for several con
secutive snaps.\n");
     keysnap(2,4,1);
     n = wish(wtr,"waitfor");
     if( n == 1 ) {
          printf("%s %s\n\n",msg4,msg0);
          setthr(2,3,wtr);
          printf("You may change the waitfor window:\n");
          printf("comands: u d l r s t w n : <cr> when fini
shed.\n");
          Irect(fov);
          Iwmov(w);
          n = wish(wa,"waitfor area");
          if( n == 1 ) {
               printf("Enter new value :");
               scanf("%d \n",wa);
          }
     }
     n = wish(ctr,"root collar");
     if( n == 1 ) {
          printf("%s %s %s\n\n",msg4,msg2,msg3);
          setthr(1,2,ctr);
     }
     n = wish(ctr2,"second root collar");
     if( n == 1 ) {
          printf("%s %s\n\n",msg1,msg5);
          setthr(1,2,ctr2);
     }
```

```
        .
        n = wish(dtr,"stem diameter");
        if( n == 1 ) {
            printf("%s %s\n\n",msg1,msg6);
            Itbcoefs(filt);
            Iconvol(1,3);
            setthr(3,2,dtr);
        }
        n = wish(dtr2,"second stem diameter");
        if( n == 1 ) {
            printf("%s %s\n\n",msg4,msg7);
            Itbcoefs(filt);
            Iconvol(1,3);
            setthr(3,2,dtr2);
        }
        n = wish(rtr,"root area");
        if( n == 1 ) {
            printf("%s %s\n\n",msg4,msg9);
            Itcoefs(5);
            Iconvol(4,2);
            setthr(2,3,rtr);
        }
        n = wish(ltr,"stem length");
        if( n == 1 ) {
            printf("%s %s\n\n",msg1,msg8);
            setthr(4,3,ltr);
        }
        n = wish(otr,"orientation");
        if( n == 1 ) {
            printf("%s %s\n\n",msg1,msgA);
            setthr(4,3,otr);
        }
    }
    Ivoff();
}
```

```
scales(one,two,xa1,xb1,xa2,xb2,ya1,yb1,ya2,yb2,thr,filt)
short *filt;
float *one,*two;
int *xa1, *xa2, *xb1, *xb2,*ya1,*yb1,*ya2,*yb2, thr;
{
    int imnum,line,ntrans,n,sum = 0,i = 0;
    char c,rnblk[rllen * 1024];
    float length, diameter, scale;
    printf("Place calibration dowel in field of view.\n");
    keysnap(1,1,1);
    keysnap(2,4,1);

    Ipf(4);
    printf("Enter the length of the dowel in millimeters :");

    scanf("%f",&length);
    printf("Enter the diameter of the dowel in millimeters :"
    );
    scanf("%f",&diameter);
    pixel(xa2,ya2, "TOP");
    pixel(xb2,yb2, "BOTTOM");
    *two = length / (float) (*xb2 - *xa2);
    printf("\n\nPixel scale factor = %f mm/pixel\n",*two);
    Ipf(1);
    pixel(xa1,ya1, "TOP");
    pixel(xb1,yb1, "BOTTOM");
    scale = length / (float) (*xb1 - *xa1);
    printf("\n\nPixel scale factor = %f mm/pixel\n",scale);

    Itbcoefs(filt);
    Iconvol(1,2);
    Ipf(2);
    Ip_single(thr);
    Ibinary();
    n = Irlen(2,rnblk,rllen);
    for( line = 110 ; line < 150 ; ++line ){
        ntrans = Ig_rx(line);
        if( ntrans == 4 ){
            sum += Ig_ry(line,2) - Ig_ry(line,0);
            ++i;
        }
    }
    Ihoriz(99,Ig_ry(99,0)-20,50);
    Ihoriz(151,Ig_ry(99,0)-20,50);
    *one = diameter * (float) i / (float) sum;
    printf("\n\nDiameter scale factor = %f mm/pixel\n",*one);

}
```

```
setthr(src,dst,thr)
int src,dst,*thr;
{
     Ifcopy(src,dst);
     Ipf(dst);
     Ip_single(*thr);
     printf("Set threshold :: u = up , d = down , <cr> when fi
     nished\n");
     Immov(3);
     *thr = Ig_single();
     printf("Threshold chosen at %d.\n",*thr);
}


wish(thr,name)
int *thr;
char name[20];
{
     char c;
     printf("Do you wish to change the %s threshold: %d :",nam
     e,*thr);
     scanf("%c \n",&c);
     if( c == 'y' || c == 'Y' ) return(1);
     else return(0);
}


pixel( ex,wy, point )
int *ex,*wy;
char point[10];
{
     printf("Move the cross hair to the %s of the dowel :\n u=
     up, d=down, l=left, r=right, <cr> when done.\n",point);
     Ipmov();
     *ex =  Ig_p0(0);
     *wy =  Ig_p0(1);
}


keysnap(camera,frame,stb)
int camera,frame,stb;
{
     char c;
     Ip_camera(camera);
     Ipf(frame);
     if( stb == 1 ) Istbon();
     printf("Press any key to snap, <cr> when finished.\n");
     Ieoff();
     while( c = getchar() != 10 ) Isnap();
     Ieon();
     Istboff();
}
```

APPENDIX C

STATISTICS FROM THE MEASUREMENT OF 100 SEEDLINGS

STATISTICS FOR GRADABLE SEEDLINGS
WITH CALIPERS BETWEEN 2.0 AND 8.0 MM
AND ORIENTATIONS LESS THAN 30 DEGREES

CALIPER	AREA	LENGTH	
pev01 STATISTICS: actual caliper = 3.0			
20 observations			
3.2	179	22	mean
0.51	18	0.9	stdev
15.8	10.0	3.9	CV %
pev02 STATISTICS: actual caliper = 3.0			
17 observations			
3.0	196	28	mean
0.17	26	0.6	stdev
5.7	13.1	2.2	CV %
pev03 STATISTICS: actual caliper = 4.0			
20 observations			
4.6	224	23	mean
0.54	10	0.6	stdev
11.6	4.6	2.8	CV %
pev04 STATISTICS: actual caliper = 4.6			
20 observations			
4.2	289	32	mean
0.16	17	1.2	stdev
3.7	6.0	3.8	CV %
pev05 STATISTICS: actual caliper = 6.0			
16 observations			
5.6	537	36	mean
0.57	46	1.7	stdev
10.3	8.7	4.7	CV %
pev06 STATISTICS: actual caliper = 3.7			
20 observations			
4.0	243	31	mean
0.74	19	1.0	stdev
18.4	7.9	3.2	CV %
pev07 STATISTICS: actual caliper = 3.3			
20 observations			
3.4	220	31	mean
0.15	8	0.5	stdev
4.4	3.6	1.6	CV %
pev08 STATISTICS: actual caliper = 4.2			
20 observations			
4.0	274	28	mean
0.40	47	1.4	stdev
9.9	17.0	5.0	CV %

pev09 STATISTICS: actual caliper = 3.0
20 observations

3.3	176	24	mean
0.16	17	0.4	stdev
4.9	9.6	1.5	CV %

pev10 STATISTICS: actual caliper = 3.7
19 observations

3.3	115	27	mean
1.16	70	3.5	stdev
35.3	61.3	12.8	CV %

pev11 STATISTICS: actual caliper = 3.8
20 observations

3.7	187	24	mean
0.18	17	0.4	stdev
4.9	9.2	1.9	CV %

pev12 STATISTICS: actual caliper = 4.1
20 observations

4.2	294	34	mean
0.13	23	0.8	stdev
3.1	7.9	2.4	CV %

pev13 STATISTICS: actual caliper = 3.2
20 observations

3.3	286	27	mean
0.11	17	0.5	stdev
3.3	5.9	1.9	CV %

pev14 STATISTICS: actual caliper = 3.0
20 observations

3.0	159	26	mean
0.09	11	0.5	stdev
3.2	6.7	1.9	CV %

pev15 STATISTICS: actual caliper = 3.3
17 observations

3.5	185	28	mean
0.83	72	4.3	stdev
23.9	38.8	15.3	CV %

pev16 STATISTICS: actual caliper = 4.4
16 observations

5.1	638	31	mean
1.17	195	4.3	stdev
22.8	30.6	13.9	CV %

pev17 STATISTICS: actual caliper = 5.3
20 observations

5.2	327	24	mean
0.30	19	1.2	stdev
5.8	5.7	5.0	CV %

pev18 STATISTICS: actual caliper = 2.9
20 observations

3.1	170	25	mean
0.42	60	2.8	stdev
13.5	35.3	11.0	CV %

pev19 STATISTICS: actual caliper = 2.7
20 observations

2.9	118	21	mean
0.28	12	0.7	stdev
9.6	10.5	3.1	CV %

pev20 STATISTICS: actual caliper = 3.1
20 observations

3.2	173	29	mean
0.13	14	0.6	stdev
4.2	8.0	2.0	CV %

pev21 STATISTICS: actual caliper = 3.0
20 observations

3.2	205	28	mean
0.37	29	1.6	stdev
11.5	14.0	5.6	CV %

pev22 STATISTICS: actual caliper = 3.3
20 observations

3.4	214	30	mean
0.08	14	0.5	stdev
2.3	6.5	1.6	CV %

pev23 STATISTICS: actual caliper = 3.0
20 observations

3.2	141	22	mean
0.22	21	0.9	stdev
6.8	14.8	4.2	CV %

pev24 STATISTICS: actual caliper = 4.7
20 observations

4.7	271	37	mean
0.56	21	2.3	stdev
11.9	7.7	6.1	CV %

pev25 STATISTICS: actual caliper = 2.5
18 observations

3.0	117	23	mean
0.42	21	1.7	stdev
14.1	17.7	7.4	CV %

pev26 STATISTICS: actual caliper = 3.7
20 observations

3.7	173	28	mean
0.23	24	1.5	stdev
6.3	14.1	5.4	CV %

pev27 STATISTICS: actual caliper = 3.9
20 observations

4.4	258	29	mean
0.52	44	1.7	stdev
11.8	17.2	5.9	CV %

pev28 STATISTICS: actual caliper = 3.9
20 observations

3.6	224	27	mean
0.62	55	2.3	stdev
17.1	24.7	8.5	CV %

pev29 STATISTICS: actual caliper = 3.0
20 observations

3.2	235	34	mean
0.16	33	0.6	stdev
5.1	14.2	1.6	CV %

pev30 STATISTICS: actual caliper = 2.3
20 observations

2.7	205	20	mean
0.37	38	1.7	stdev
13.9	18.4	8.3	CV %

pev31 STATISTICS: actual caliper = 2.8
20 observations

2.9	135	20	mean
0.13	13	0.2	stdev
4.5	9.7	1.1	CV %

pev32 STATISTICS: actual caliper = 3.9
20 observations

4.1	188	26	mean
0.15	14	0.5	stdev
3.6	7.3	2.0	CV %

pev33 STATISTICS: actual caliper = 3.3
20 observations

3.4	212	26	mean
0.17	14	0.2	stdev
5.0	6.5	0.9	CV %

pev34 STATISTICS: actual caliper = 4.5
20 observations

3.8	225	32	mean
0.26	42	1.5	stdev
6.9	18.5	4.6	CV %

pev35 STATISTICS: actual caliper = 4.1
20 observations

4.1	319	34	mean
0.19	24	0.8	stdev
4.6	7.4	2.2	CV %

```
pev36  STATISTICS:  actual caliper = 4.5
  20 observations
   4.9       252     31     mean
  0.39        12    0.4     stdev
   7.9       4.9    1.4     CV %

pev37  STATISTICS:  actual caliper = 2.8
  20 observations
   3.1       124     20     mean
  0.65        15    1.1     stdev
  20.8      11.8    5.7     CV %

pev38  STATISTICS:  actual caliper = 4.0
  20 observations
   3.8       208     31     mean
  0.70        60    3.0     stdev
  18.4      28.6    9.6     CV %

pev39  STATISTICS:  actual caliper = 5.7
  17 observations
   5.7       611     27     mean
  0.28        54    0.5     stdev
   4.8       8.8    1.8     CV %

pev40  STATISTICS:  actual caliper = 5.2
  20 observations
   5.2       300     31     mean
  0.07        13    0.7     stdev
   1.3       4.3    2.2     CV %

pev41  STATISTICS:  actual caliper = 3.5
  20 observations
   3.9       188     23     mean
  0.86        16    0.9     stdev
  22.3       8.5    3.7     CV %

pev42  STATISTICS:  actual caliper = 4.9
  19 observations
   4.7       264     35     mean
  0.30        27    1.9     stdev
   6.3      10.2    5.5     CV %

pev43  STATISTICS:  actual caliper = 4.0
  19 observations
   3.9       158     28     mean
  0.42        34    2.0     stdev
  10.6      21.7    7.3     CV %

pev44  STATISTICS:  actual caliper = 4.8
  20 observations
   4.7       320     28     mean
  0.18        20    0.8     stdev
   3.8       6.1    2.9     CV %
```

pev45 STATISTICS: actual caliper = 5.0
20 observations

5.0	251	33	mean
0.59	32	1.7	stdev
11.9	12.6	5.0	CV %

pev46 STATISTICS: actual caliper = 4.0
19 observations

3.9	234	29	mean
0.15	18	0.5	stdev
3.8	7.7	1.6	CV %

pev47 STATISTICS: actual caliper = 3.9
20 observations

4.1	250	35	mean
0.45	94	2.2	stdev
11.2	37.8	6.2	CV %

pev48 STATISTICS: actual caliper = 3.5
20 observations

3.5	203	27	mean
0.05	16	0.4	stdev
1.4	7.9	1.5	CV %

pev49 STATISTICS: actual caliper = 3.3
20 observations

3.2	153	24	mean
0.15	17	0.4	stdev
4.8	11.4	1.9	CV %

pev50 STATISTICS: actual caliper = 4.9
19 observations

4.9	371	25	mean
0.64	59	1.8	stdev
13.3	15.9	7.2	CV %

pev51 STATISTICS: actual caliper = 3.1
19 observations

3.7	175	27	mean
0.63	39	2.9	stdev
17.2	22.1	10.9	CV %

pev52 STATISTICS: actual caliper = 3.0
20 observations

3.3	164	28	mean
0.21	16	0.7	stdev
6.3	9.9	2.4	CV %

pev53 STATISTICS: actual caliper = 2.3
20 observations

2.5	99	20	mean
0.12	11	0.4	stdev
4.9	10.9	2.2	CV %

pev54 STATISTICS: actual caliper = 4.2
19 observations

3.8	333	31	mean
0.13	48	0.6	stdev
3.4	14.5	1.9	CV %

pev55 STATISTICS: actual caliper = 3.1
19 observations

2.8	182	27	mean
0.16	18	0.9	stdev
5.7	9.7	3.3	CV %

pev56 STATISTICS: actual caliper = 3.6
19 observations

3.4	249	29	mean
0.20	40	0.5	stdev
5.9	16.0	1.6	CV %

pev57 STATISTICS: actual caliper = 4.3
19 observations

4.0	536	33	mean
0.56	44	1.2	stdev
14.2	8.2	3.6	CV %

pev58 STATISTICS: actual caliper = 4.2
20 observations

3.6	307	33	mean
0.16	35	1.3	stdev
4.3	11.5	4.0	CV %

pev59 STATISTICS: actual caliper = 4.4
20 observations

3.8	550	31	mean
0.16	55	0.9	stdev
4.2	10.0	2.9	CV %

pev60 STATISTICS: actual caliper = 4.5
20 observations

4.2	515	32	mean
0.45	80	2.8	stdev
10.7	15.6	8.7	CV %

pev61 STATISTICS: actual caliper = 4.1
20 observations

3.7	385	33	mean
0.17	32	1.0	stdev
4.5	8.3	3.2	CV %

pev62 STATISTICS: actual caliper = 4.4
19 observations

3.6	459	26	mean
0.14	28	0.6	stdev
4.0	6.0	2.2	CV %

pev63 STATISTICS: actual caliper = 4.4
19 observations

3.5	512	29	mean
0.13	48	0.7	stdev
3.7	9.4	2.3	CV %

pev64 STATISTICS: actual caliper = 4.0
20 observations

3.5	453	31	mean
0.20	34	0.7	stdev
5.6	7.5	2.3	CV %

pev65 STATISTICS: actual caliper = 3.5
19 observations

3.1	319	32	mean
0.17	46	0.5	stdev
5.4	14.3	1.6	CV %

pev66 STATISTICS: actual caliper = 4.6
20 observations

3.8	377	32	mean
0.20	29	1.2	stdev
5.1	7.7	3.7	CV %

pev67 STATISTICS: actual caliper = 4.0
19 observations

3.6	473	33	mean
0.11	39	1.1	stdev
3.2	8.2	3.4	CV %

pev68 STATISTICS: actual caliper = 4.2
20 observations

3.7	289	33	mean
0.09	33	1.3	stdev
2.5	11.3	3.9	CV %

pev69 STATISTICS: actual caliper = 3.6
19 observations

3.2	377	24	mean
0.20	34	0.2	stdev
6.4	8.9	1.0	CV %

pev70 STATISTICS: actual caliper = 4.8
20 observations

4.4	507	34	mean
0.50	48	2.6	stdev
11.5	9.5	7.5	CV %

pev71 STATISTICS: actual caliper = 3.4
20 observations

3.1	310	29	mean
0.16	50	1.0	stdev
5.3	16.0	3.3	CV %

pev72 STATISTICS: actual caliper = 2.4
20 observations

2.3	138	25	mean
0.27	23	1.9	stdev
11.9	16.7	7.8	CV %

pev73 STATISTICS: actual caliper = 4.0
20 observations

3.5	304	30	mean
0.29	19	0.5	stdev
8.4	6.4	1.6	CV %

pev74 STATISTICS: actual caliper = 5.0
19 observations

4.4	538	35	mean
0.22	84	1.4	stdev
4.9	15.6	4.1	CV %

pev75 STATISTICS: actual caliper = 3.1
19 observations

3.0	193	29	mean
0.15	16	0.7	stdev
5.1	8.3	2.5	CV %

pev76 STATISTICS: actual caliper = 3.7
20 observations

3.2	179	36	mean
0.21	29	2.0	stdev
6.5	16.2	5.5	CV %

pev77 STATISTICS: actual caliper = 3.4
19 observations

3.1	311	28	mean
0.16	24	0.5	stdev
5.1	7.8	1.8	CV %

pev78 STATISTICS: actual caliper = 4.7
20 observations

4.2	464	34	mean
0.13	40	0.9	stdev
3.1	8.5	2.7	CV %

pev79 STATISTICS: actual caliper = 5.0
18 observations

4.0	386	33	mean
0.21	29	0.9	stdev
5.4	7.5	2.8	CV %

pev80 STATISTICS: actual caliper = 5.3
30 observations

4.7	437	36	mean
0.10	26	1.5	stdev
2.1	5.9	4.1	CV %

pev81 STATISTICS: actual caliper = 3.3
19 observations

3.1	289	26	mean
0.15	21	0.5	stdev
4.8	7.1	2.0	CV %

pev82 STATISTICS: actual caliper = 2.6
19 observations

2.6	231	25	mean
0.11	16	0.7	stdev
4.2	7.1	3.0	CV %

pev83 STATISTICS: actual caliper = 4.9
19 observations

4.5	501	34	mean
0.18	45	1.3	stdev
4.0	8.9	3.7	CV %

pev84 STATISTICS: actual caliper = 4.2
20 observations

3.8	425	35	mean
0.10	52	1.3	stdev
2.5	12.2	3.9	CV %

pev85 STATISTICS: actual caliper = 4.2
20 observations

3.7	361	34	mean
0.12	25	1.2	stdev
3.2	7.0	3.6	CV %

pev86 STATISTICS: actual caliper = 3.9
20 observations

3.6	305	33	mean
0.10	24	0.6	stdev
2.7	7.8	1.7	CV %

pev87 STATISTICS: actual caliper = 4.0
20 observations

3.7	409	30	mean
0.24	60	0.6	stdev
6.3	14.6	2.0	CV %

pev88 STATISTICS: actual caliper = 4.5
19 observations

4.0	410	32	mean
0.28	25	1.3	stdev
7.0	6.2	3.9	CV %

pev89 STATISTICS: actual caliper = 4.0
19 observations

3.8	458	33	mean
0.13	45	1.4	stdev
3.4	9.9	4.2	CV %

pev90 STATISTICS: actual caliper = 4.8
19 observations

4.4	576	33	mean
0.26	40	1.0	stdev
5.9	7.0	3.0	CV %

pev91 STATISTICS: actual caliper = 3.4
20 observations

3.2	339	33	mean
0.24	32	1.3	stdev
7.4	9.5	4.0	CV %

pev92 STATISTICS: actual caliper = 3.3
20 observations

3.1	352	32	mean
0.18	27	0.4	stdev
5.8	7.6	1.2	CV %

pev93 STATISTICS: actual caliper = 3.2
20 observations

3.0	239	33	mean
0.16	39	0.7	stdev
5.2	16.1	2.3	CV %

pev94 STATISTICS: actual caliper = 3.4
19 observations

3.1	204	30	mean
0.25	51	2.2	stdev
8.0	24.9	7.3	CV %

pev95 STATISTICS: actual caliper = 3.6
18 observations

3.4	319	24	mean
0.14	24	0.8	stdev
4.0	7.7	3.5	CV %

pev96 STATISTICS: actual caliper = 3.4
17 observations

3.1	267	25	mean
0.18	29	1.0	stdev
5.7	10.9	3.9	CV %

pev97 STATISTICS: actual caliper = 3.0
20 observations

2.6	236	28	mean
0.15	20	2.8	stdev
5.6	8.3	10.1	CV %

pev98 STATISTICS: actual caliper = 3.5
18 observations

3.3	306	23	mean
0.14	20	0.5	stdev
4.2	6.6	2.3	CV %

```
pev99  STATISTICS:  actual caliper = 5.0
  19 observations
   3.8       415      31     mean
  0.10        37     0.0     stdev
   2.6       8.9     0.0     CV %

pev100 STATISTICS:  actual caliper = 2.9
  20 observations
   2.9       183      24     mean
  0.08        21     0.3     stdev
   2.7      11.3     1.3     CV %



  7.63     12.25    4.04     CVav %
```

SUMMARY OF GRADE CLASSIFICATIONS FOR 100 SEEDLINGS

SEEDLING NUMBER	MANUAL GRADE	NONE	CLASSIFICATION A1	A2	C1	C2	C3
1	C	0	0	0	0	6	14
2	C	2	1	0	4	1	12
3	A	0	20	0	0	0	0
4	A	0	20	0	0	0	0
5	A	4	16	0	0	0	0
6	A	0	20	0	0	0	0
7	A	0	20	0	0	0	0
8	A	0	19	0	0	1	0
9	C	0	1	0	0	14	5
10	C	1	3	0	10	6	0
11	C	0	3	0	0	17	0
12	A	0	20	0	0	0	0
13	A	0	19	1	0	0	0
14	C	0	0	0	10	0	10
15	C	3	5	0	7	1	4
16	A	4	15	0	1	0	0
17	A	0	20	0	0	0	0
18	C	0	0	0	9	5	6
19	C	0	0	0	15	1	4
20	C	0	0	0	0	0	20
21	C	0	4	0	2	0	14
22	A	0	17	0	0	3	0
23	C	0	0	0	3	8	9
24	A	0	20	0	0	0	0
25	C	2	0	0	12	3	3
26	C	0	2	0	0	18	0
27	A	0	19	0	0	1	0
28	A	0	14	0	1	4	1
29	A	0	11	0	0	1	8
30	C	0	1	0	17	1	1
31	C	0	0	0	11	0	9
32	C	0	4	0	0	16	0
33	A	0	15	1	0	4	0
34	A	0	16	0	0	4	0
35	A	0	20	0	0	0	0
36	A	0	20	0	0	0	0
37	C	0	0	0	10	5	5
38	A	0	17	0	3	0	0
39	A	3	17	0	0	0	0
40	A	0	20	0	0	0	0
41	C	0	5	0	0	15	0
42	A	1	19	0	0	0	0
43	C	1	1	0	1	16	1
44	A	0	20	0	0	0	0
45	A	0	19	0	0	1	0
46	A	1	19	0	0	0	0
47	A	0	16	0	0	4	0
48	A	0	14	0	0	6	0

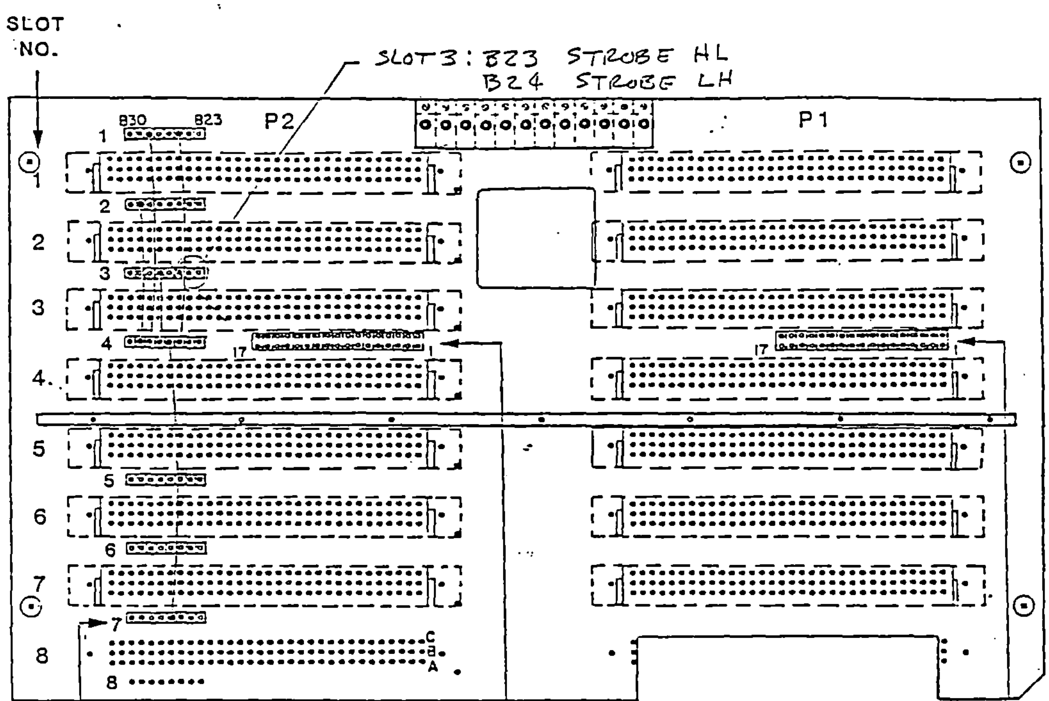
49	C	0	0	0	0	10	10
50	A	1	18	0	1	0	0
51	C	1	5	0	3	11	0
52	C	0	0	0	0	15	5
53	C	0	0	0	20	0	0
54	A	0	20	0	0	0	0
55	C	1	2	0	8	1	8
56	A	1	17	0	0	1	1
57	A	1	19	0	0	0	0
58	A	0	20	0	0	0	0
59	A	0	20	0	0	0	0
60	A	0	20	0	0	0	0
61	A	0	20	0	0	0	0
62	A	1	19	0	0	0	0
63	A	1	19	0	0	0	0
64	A	0	20	0	0	0	0
65	A	1	16	2	1	0	0
66	A	0	20	0	0	0	0
67	A	1	19	0	0	0	0
68	A	0	20	0	0	0	0
69	A	0	15	5	0	0	0
70	A	0	20	0	0	0	0
71	A	0	14	4	1	0	1
72	C	0	0	0	19	1	0
73	A	0	19	1	0	0	0
74	A	1	19	0	0	0	0
75	C	1	2	0	1	6	10
76	C	0	3	0	0	14	3
77	A	1	15	4	0	0	0
78	A	0	20	0	0	0	0
79	A	1	19	0	0	0	0
80	A	0	20	0	0	0	0
81	A	1	13	6	0	0	0
82	C	1	0	0	18	0	1
83	A	1	19	0	0	0	0
84	A	0	20	0	0	0	0
85	A	0	20	0	0	0	0
86	A	0	20	0	0	0	0
87	A	0	20	0	0	0	0
88	A	1	18	1	0	0	0
89	A	1	19	0	0	0	0
90	A	1	19	0	0	0	0
91	A	0	16	4	0	0	0
92	A	0	15	4	1	0	0
93	A	0	11	6	0	3	0
94	C	1	7	0	1	8	3
95	A	1	19	0	0	0	0
96	A	2	14	4	0	0	0
97	A	0	6	1	9	0	4
98	A	2	17	1	0	0	0
99	A	1	19	0	0	0	0
100	C	0	0	0	3	2	15

APPENDIX D

IRI/STROBOTAC INTERFACE

P256 BACKPLANE

(REAR VIEW)



SLOT 4 (CPU BOARD) P2, B1-B17

TOP ROW - GROUND
BOTTOM ROW - SEE CHART

SLOT 4 (CPU BOARD) P1, B1-B17

TOP ROW - GROUND
BOTTOM ROW - PIN 1 - +5V
PIN 2 … PIN 17 } DIG I/O 1-16

P2, B23-B30 (TYPICAL)

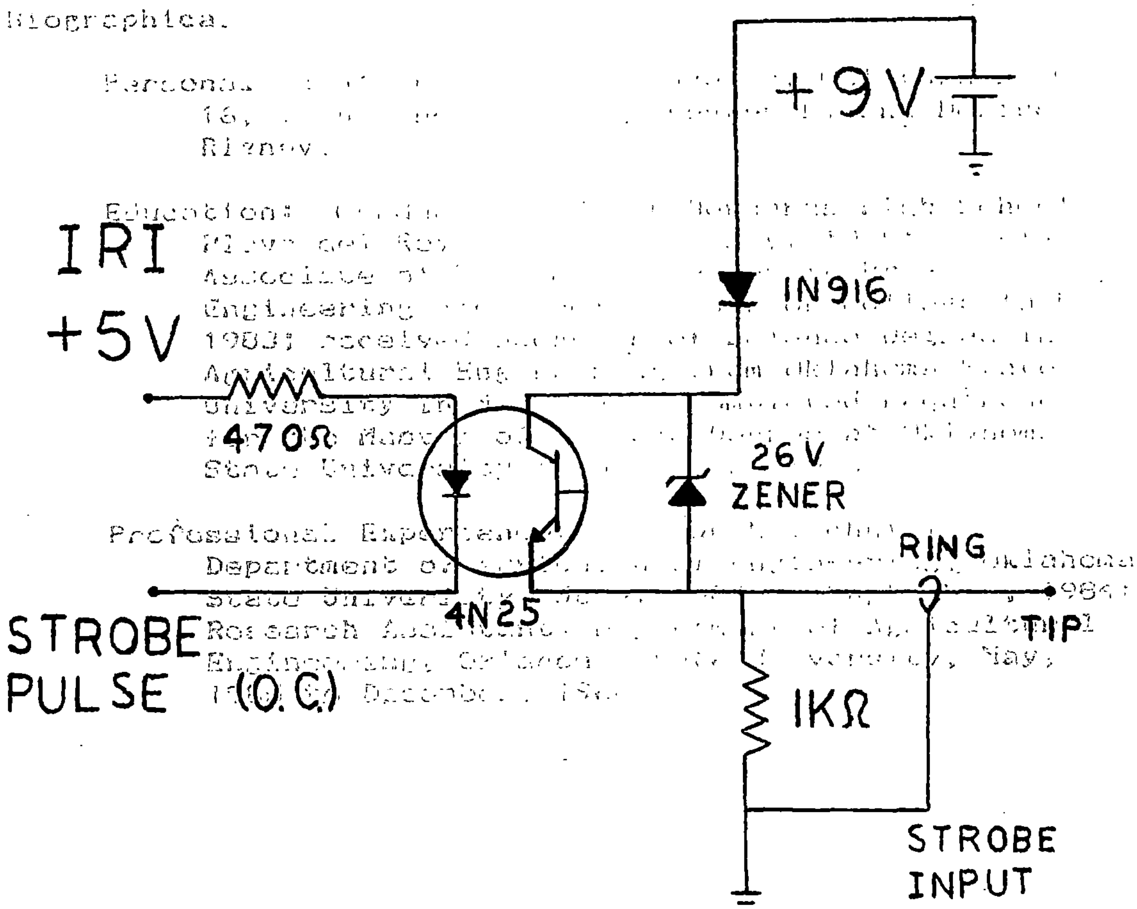
SLOT 4 (CPU BOARD):
INTERRUPT INPUTS 0-7

ALL OTHER BOARDS:
GENERAL PURPOSE OR
INTERRUPT OUTPUTS

BOARD POSITIONS

SLOT 1 - COPROCESSOR
SLOT 2 - COLOR/GRAPHICS BOARD
SLOT 3 - IBB
SLOT 4 - SPC
SLOT 5 - PERIPHERAL/MEMORY EXPANSION
SLOT 6 - PERIPHERAL/MEMORY EXPANSION
SLOT 7 - PCI (DISKS)
SLOT 8 - (NOT USED)

Michael Patrick Rigney



2

VITA

Michael Patrick Rigney

Candidate for the Degree of

Master of Science

Thesis: MACHINE VISION FOR THE GRADING OF PINE SEEDLINGS

Major Field: Agricultural Engineering

Biographical:

Personal Data: Born in Los Angeles, California, May 16, 1960, the son of Sylvester L. and Denise Rigney.

Education: Graduate of Saint Bernards High School, Playa del Rey, California, May, 1978; received Associate of Technology Degree in Pre-Engineering from Seminole Junior College in May, 1983; received Bachelor of Science Degree in Agricultural Engineering from Oklahoma State University in May, 1985; completed requirements for the Master of Science Degree at Oklahoma State University in December, 1986.

Professional Experience: Research Technician, Department of Agricultural Engineering, Oklahoma State University, June, 1984 to September, 1984; Research Assistant, Department of Agricultural Engineering, Oklahoma State University, May, 1985 to December, 1986.