1990

A microcomputer based combined machine vision and expert system for irregular object classification

Syed Azhar Saeed Zaidi
Iowa State University

Recommended Citation

Zaidi, Syed Azhar Saeed, "A microcomputer based combined machine vision and expert system for irregular object classification " (1990). *Retrospective Theses and Dissertations*. 11234.
https://lib.dr.iastate.edu/rtd/11234

U·M·I

University Microfilms International
A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
313/761-4700 800/521-0600

Order Number 9035129

A microcomputer-based combined machine vision and expert system for irregular object classification

Zaidi, Syed Azhar Saeed, Ph.D.

Iowa State University, 1990

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

A microcomputer based combined machine vision and expert system for irregular object classification

by

Syed Azhar Saeed Zaidi

A Dissertation Submitted to the

Graduate Faculty in Partial Fulfillment of the

Requirements for the Degree of

DOCTOR OF PHILOSOPHY

Department: Industrial and Manufacturing Systems Engineering
Major: Industrial Engineering

Approved:

Signature was redacted for privacy.

In Charge of Major Work

Signature was redacted for privacy.

For the Major Department

Signature was redacted for privacy.

For ~~the~~ Graduate College

Iowa State University
Ames, Iowa
1990

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1. INTRODUCTION

Need for the Research

In recent years much work has been reported on the development of microcomputer based machine vision systems [40, 68]. A substantial portion of this research assumes most of the items subjected for machine vision inspection, can be categorized by a regular geometrical shape [53, 54, 72]. A few microcomputer-based vision systems have been marketed: for example, the OPTO-SENSE MENTOR (Copperweld Robotics, Troy, Michigan, $50,000), Charger (Rhino Robots Inc., Champaign, Ill., $40,000), and the 100 Series (Unimation Inc., Danbury, Conn., $24,000) [24]. Most of the vision systems, and microcomputer-based vision systems in particular, are designed to perform singular tasks, they are niche oriented and designed to be used in an inflexible environment [77]. These systems are primarily suited for processing regular objects. The assumption of regular objects works well in the case of most mechanical parts and machine tools, but there are manufacturing situations in which the handling of irregular objects is required [81]. The non-destructive evaluation of industrial parts, and evaluation of non-metallic inclusions in steel, are a couple of many such applications. The commercial machine vision systems to handle such applications are multi-processor based and are, generally, prohibitively expensive for low budget applications. One such system is Perceptron Inc.'s MV-600 system which costs over $80,000 for the basic unit. The cost of a turn key system is several times this amount [7].

The *Iowa Conservation Commission's State Forest Nursery* (in Ames, Iowa) produces about 6 to 10 million seedlings every year. These seedlings have to be segregated according to their survival expectancy before they can be stored for final shipment. The time constraints are such that the grading process has to be completed in about eight weeks. This means that about 13,000 have to be processed every hour. As a result, several grading lines have to be arranged in parallel.

It is hoped that the following problems identified in the manual grading process would be resolved if the process can be automated through a machine vision system.

- Grading is done by minimum wage, seasonal workers, who have little or no working experience with tree seedlings.
- An initial training period is required to teach the desired quality constraints.
- Lack of working knowledge often results in acceptance of bad seedlings and rejection of good seedlings.
- Inconsistent rejection rates result in a lack of control over the supply volumes.
- The grading operations have to be performed under low temperature and high humidity, and these conditions also affect the worker performance.

The economic feasibility of an automated grading process has been established by research [18, 72]. The segregation operations were observed and the data were collected from the forest nursery in Ames by the research conducted by Mark Titus at Iowa State University [72]. Titus showed that an automated vision system would mean improved seedling quality for the customers, due to consistent grading processes, and would reduce production costs. Various options for handling of plant seedlings in an automated grading system were also studied in the research. A cable conveyor, moving around a circular path and holding the seedlings in specially designed clamps was shown to be the best option for proper seedling handler for the seedling segregation system [72].

Objectives of the Research

The irregular-object application dealing with the segregation of plant seedlings at a forest nursery is the focus and the stimulus for this research. The main objective of this research is show the functional viability of machine vision process by optimizing the processing speed and by providing means for noise tolerance, by developing an expert system to identify and isolate root and stem areas of the seedling image and to apply the grading criteria on these regions. We also want to provide a facility, for dynamic adjustments to the grading criteria, to facilitate the use of numerous species. Another important objective of the research is to make use of the cost effective computer technology such as a microcomputer-based image processing system, to keep the total system cost down. It is hoped that the development of such low cost systems would generate an interest in small industry for quality improvement through machine regulated quality control systems. Our target is to design a system with a throughput of one seedling per second. Under the current production volume, this would mean that five grading lines in parallel would be sufficient to handle the production capacity.

Scope and Impact of the Research

The research has been endorsed by the *Iowa Conservation Commission's State Forest Nursery* from its inception. There has been a continued contact with the nursery officials to get information on the criteria for quality judgements in the inspection process. The nursery provided the tree seedlings to test the prototype which was developed in a lab environment. The nursery officials believe that automating the grading process will bring improvements in the segregation process and would result in a supply of better quality products, and greater customer satisfaction.

Further applications of this research may be found in areas of detection of solder corrosion on the Printed Circuit Board surfaces, detections of risers to assure proper filling of industrial castings, quantitative evaluation of metallographic measurements, segregation of laminate and decorative tiles, and scene evaluation.

The research may also be applied in vegetable, fruit, egg grading, plus industries involved in the production of consumer products like leather, clothing, furniture and toys. It is anticipated that this research will contribute in developing applications for vision based irregular object identification and process control, where the capability of a low cost system is feasible and desired.

Outline of Subsequent Chapters

The first section of Chapter 2 provides an overview of the research in using machine vision in agriculture and food processing industry. The next section provides detailed account of the grading criteria for the nursery tree seedlings. The operation of the grading process is briefly described and various grading alternatives have been explored. The section ends with outlining the specific grading parameters used in this research. The third section reviews a project at Oklahoma State University (OSU) in which a hardware based seedling segregation system was designed. Comparisons between the OSU research and our research are made. Chapter 3 provides some fundamental information about the machine vision concepts and procedures. Various concepts dealing with expert systems and current trends are also discussed. The first section reviews various techniques used in image retrieval, restoration and processing. It also identifies various application areas for machine vision. The second section provides a brief survey of the equipment involved in a machine vision system. The section ends by describing the equipment involved in this research, and its limitations. In subsequent sections a brief introduction to expert systems, its composition, application areas, and the software characterization are presented. The problems in having a full blown expert system for the seedling grading process have also been identified.

Chapter 4 provides a detailed account of the processing decisions involved in various phases of the seedling segregation system. The chapter has been broken into three sections describing the process of clamping point identification, root area isolation and measurement, and stem area isolation and measurement. Specific references are made to the set of images acquired by the system developed for the laboratory environment.

Chapter 5 describes the function of the program and various information parameters required/supplied by the system. It also serves as a reference for the program usage. The results obtained by actual laboratory runs are provided in Chapter 6.

The conclusions and an outline of further rèsearch which might be conducted to enhance and/or expand this research, is provided in Chapter 7 and Chapter 8. Appendixes contain detailed program flow charts for various segments of the program.

CHAPTER 2. LITERATURE REVIEW

Applications of Machine Vision

The automotive industry, by far, has been the largest beneficiary of the vision technology. In the automotive industry, machine vision is used to identify parts on the conveyor belt for proper routing, it is used to continuously monitor the production processes for machining and workmanship defects, and it is used on robot arms for guidance and routing. The second large user of machine vision is the electronic industry, where vision systems are used to identify routing errors in printed circuit boards (PCB). They are used on part assembly lines to supply control information to the part placement robots for proper alignment of surface mounted parts. Vision systems are used for PCB quality control, to identify wrong or missing parts, and parts placed in the reverse orientation, resulting in wrong polarity. Finally, they are used to identify solder defects. It is estimated that the average yield in the first pass tests of PCB's is about 75%. About half of the defects are because of workmanship and missing or wrong components, and depending on the yield and supply volume, it is estimated that any where from 250,000 to several million dollars can be saved if these problems are rectified using automated part placement and inspection systems which incorporate machine vision techniques [9].

The third largest user of vision systems is the food & beverage industry. It is predicted that by 1990 the market share of vision systems in food and beverage industry would increase to 11%, from \$3.5 million in 1985 to \$50.2 million in 1990, thus, maintaining its third position [25]. Vision systems in food and beverage industry are used during production for process control and sorting, and after production for inspection and gaging [25].

A large number of applications in machine vision for processing agricultural products have been reported in the literature. Sites and Delwiche [66] describe a system to identify and estimate number of fruits on a tree using grey scale thresholding techniques. Another application in the area of agricultural botany is described by Draper [19]. Computer vision is used to identify various leaves and seed shapes. The discrimination is obtained by computing the *shape factor* and *aspect ratio* of specimens. Shape factor is defined by; $4\pi(area/perimeter^2)$. The above formula used 4π to result in a theoretical shape factor of unity for a perfectly circular object. Gunasekaran [30] reports various methods for quality evaluation of agricultural and biological materials.

Many authors have reported various pattern recognition applications in agriculture and agricultural products. Jaffe [38] reported the image analysis of plant growth. Heyne [34] worked on computerized image analysis of simulated pizza to check the regularity of the crust. Taylor and Rehkugler [70] developed an image analysis system to detect apple bruises. Sarkar and Wolfe [58] presented image processing techniques for sorting tomatoes, where the vision system made it possible to classify tomatoes by shape, size, and color. Parrish and Goksel [52] presented a pattern recognition application for apple harvesting. Wright [80] described a method to measure the size and shape of sweet potatoes. The size and area statistics of liquid chemicals used on crops were determined by Kranzler et al. [42]. The chemicals under investigation were mixed with different fluorescent dyes, which were picked up by the imaging equipment. Connectivity analysis was used for calculating the order statistics [42]. A process control application of machine vision in the food processing industry provides a continuous monitoring and detection of foreign matter in a de-greaser tank to prevent undesirable chemical reaction and possible degreaser spill [20]. De-greasing is used in food industry between batches, for cleaning process equipment and piping. Another such application uses process control in the produce industry in the development of an automated citrus packing house in Japan [36]. Ottman [51] reports on the measurement of plant roots by machine vision. A microcomputer-based prototype of an opto-electronic system has been developed by Oklahoma State University (OSU) for in-field counting of plant seedlings. The procedure is based on the interruptions recorded by infrared detectors mounted on the tractor's tool-carrier platform [16, 41].

Grading Requirements for Nursery Tree Seedlings

There are various stages where the seedling quality must be measured, in order to ensure a good chance of survival after the final plantation. The yield for plantable seedlings can be optimized by improving the storage and lifting operations and by providing deterrents to environmental variability in storage, which is potentially harmful. Seedling quality should be measured at three phases: (1) during the growing season, (2) prior to lifting, and (3) prior to field planting [64]. Environmental conditions prior to lifting, length of storage, conditions of storage, and the handling methods are of great importance for good seedling quality [10, 21]. Generally every nursery has to go through the following six steps before the seedlings leave the nursery for final planting [72, 82].

- In-field grading/inventorying
- Lifting process
- Grading process
- Storing process
- Retrieving process
- Shipping

The manual classification operations prevalent in the nurseries are highly subjective; the human operators work under low temperature and high humidity and lack the experience and vigor and, hence, are susceptible to human error. Moreover, it is not possible to have a 100% inspection in any manual grading system. Therefore, researchers have been looking at ways to automate this process.

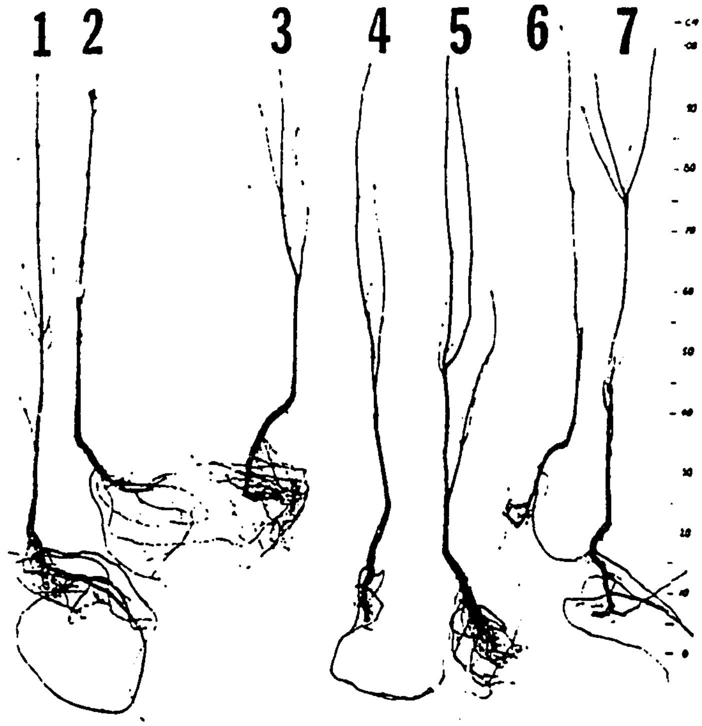
The proposed seedling segregation system would be utilized after the seedlings are lifted from the nursery beds. After this segregation process the seedlings would be ready to be stored in a cool environment in special bags to hold their moisture. The extraction of good seedlings at this point would ensure that no extra overhead would be spent for non-plantable (cull) seedlings. Morphological attributes of a seedling have been found to be important for final grading operation [75]. These attributes include the height, diameter, dry weight, shoot-to-root ratio, and the bud height [49, 56]. Stroempl [67] described root-collar diameter, stem length, and bud number of large and small seedlings to be important measures for the grading of red oak seedlings. His classification into small and large seedlings is shown in Table 2.1.

Table 2.1: Morphological classification based on seedling size

Seedling Size	Root-Collar Diameter (in.)	Stem Length (in.)
Large	0.30 to 0.34	22 to 30
Small	0.18 to 0.26	12 to 18
Reject	0.18	

The morphological conditions desirable for a plantable seedling are shown in Figure 2.1 and Figure 2.2. Stroempl [67] categorized large and small seedlings separately. Four grades were assigned to the seedlings in each category, based on various morphological attributes.

The height of seedling is a measure of its photo-synthetic capacity and transpirational area. Initial height of a seedling is also positively correlated to the future growth of the plant [49]. However, the shoot diameter is a better measure of seedling growth for estimating the probability of survival. Larger diameter ensures more structural support [49] and protection against forking to the ground [67]. A longer and thicker tap root has also been shown helpful for structural support [49]. The root-collar diameter and stem length are also found to be reliable indicators of relative growth potential, regardless of age of the stock [75]. Also, a survivable seedling must have buds on the shoot. The seedlings having buds at the apex of the shoot have a high probability of survival [67]. Buds at the lower middle of the stem and branches ensure high dormancy requirements for future plant growth.



Grade_1 (1&2): Well defined, sturdy, and straight leading shoot, with buds on most of stem. Branches relatively short.

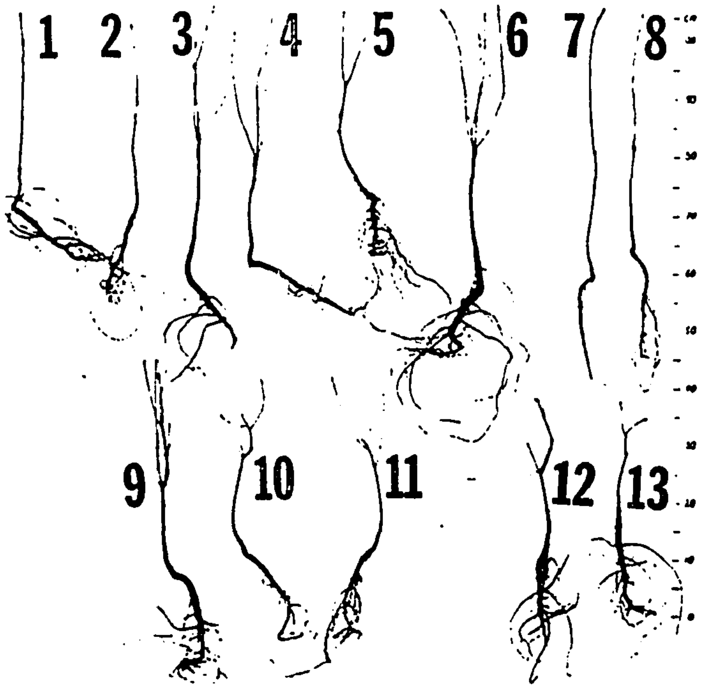
Grade_2 (3): Leading shoot less defined with fewer buds. More prominent branches.

Grade_3 (4&5): Leading shoot thin or forked. Large root–collar diameter.

Grade_4 (Reject; 6&7): Stem is very thin. Tendency to forking prominent. Fewer buds on the shoot.

Roots: Tap root length: 20 to 25 cm.
Lateral root length: 10 to 15 cm.

Figure 2.1: Morphological conditions for long seedlings



Grade_1 (1): Leading shoot sturdy and straight, with buds on most of stem. No branches.

Grade_2 (2&3): Leading shoot less sturdy. Branches sometimes too long.

Grade_3 (4 to 6): Stem with 2 to 3 leading shoots. Terminal buds should be present.

Grade_4 (Reject; 7 to 13): Stem is very thin. Tendency to forking prominent.

Roots: Tap root length: 15 cm.
Lateral root length: 10 cm.

Figure 2.2: Morphological conditions for short seedlings

Based on the above observations, it is evident that an ideal seedling segregation system should support the following features:

- Clamping point discrimination
- Seedling category (long or short)
- Shoot discrimination and measurement of length
- Branch count
- Branch length
- Caliper point identification
- Caliper diameter measurement
- Tap or lateral root discrimination
- Bud identification
- Shoot-to-root ratio
- Forking probability estimation

Use of Machine Vision in Grading of Seedlings

A dedicated machine vision system has been designed at Oklahoma State to grade southern pine seedlings [56]. The system uses the International Robomation/Intelligence (IRI) D256 machine vision development system. This system supports 256 gray levels at a resolution of 256x240 pixels. The primitive image processing functions like image convolution, run-length coding, and moments calculations are implemented in hardware. The IRI system with two cameras, a systolic array processor and a user programming environment costs over $100,000. The system computes the caliper diameter, and estimates the root area, and the length of the shoot. Two Hitachi KP-120U cameras providing a resolution of 320x244 are used for image capture. One camera is used to obtain the seedling root collar, and uses a field of view (FOV) of five inches. The other camera is used to capture the whole seedling with a FOV of 20 inches. Variable thresholding technique is used to highlight the areas of interest in the image and to suppress noise. Initially, the high FOV camera operates in a loop until it finds an image at a lower threshold. Then both cameras are used to capture the seedling using xenon strobe lighting. The whole image is run length encoded. The caliper is identified by looking for six continuous lines of minimum transitions, starting from the top of the image using a medium gray level threshold of 90. A higher threshold (140) is used to suppress stray roots and/or needles around the seedling caliper. A Laplacian edge detector is used to get the stem edges. The image is then run-length encoded and the caliper diameter computed. The root volume (projected area) is determined at a threshold of 48. The shoot height is computed from the root collar position to the top of the seedling. The system results in an average classification error rate of 5.7 percent, and does so at a speed of approximately 0.25 seconds [56].

The OSU research uses dedicated hardware for image processing. It uses grey scale images and uses thresholding as a mean to capture regions of interest. This research in contrast, relies on a microcomputer based software solution. We use binary images, and a purely stochastic approach for processing. We use edge tracking on the seedling shoot to compute its length. Furthermore, an additional process is incorporated in the root area processing by identifying the tap root and providing an estimate of the lateral root count, attached to the tap root.

CHAPTER 3. SYSTEM CONCEPTS

Machine Vision Methodology

Image acquisition and restoration

The basic components of a machine vision process are shown in Figure 3.1 [45]. The image is captured by an analog or a digital camera. If the input is analog, it is passed through an Analog to digital converter.

The image is then stored in a memory buffer by a frame grabber. The frame is then processed by a set of image processing routines by the computer according to a set criterion. The results from this routine declare the seedling to be good or bad and at that point control signals from the computer may be generated for proper disposition of the seedling. An automatic update of the inventory data base can also be incorporated at this point.

Several researchers have observed that the varying nature and complexity of the work environment such as varying light intensity, varying reflectivity of surrounding objects, pose limitations over the practicality of a machine vision system [3, 4, 6, 13, 31, 55, 79]. Some of these problems can be overcome by certain image enhancement techniques like brightness control, image stretching, and Fourier-domain processing [26, 28].

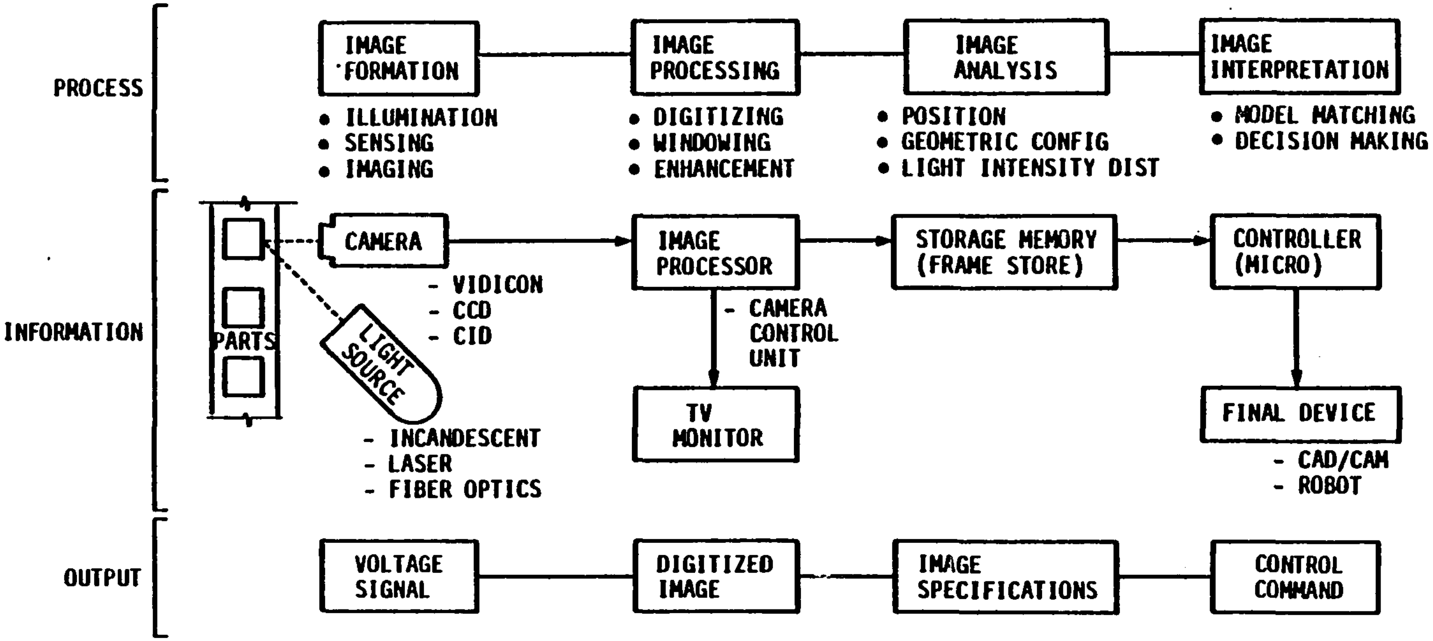


Figure 3.1: Basic components of a machine vision process

Feature extraction

Global feature method Once the image is enhanced to the required level, it has to go through the feature extraction phase. There are several approaches one can take in perceiving what is seen by the camera. One is the connectivity analysis approach, in which the binary image is broken into its connected components, from which the deterministic components and the shape features are derived. This method is best suited when there are a limited number of regular objects in the domain [57]. In the case of low contrast images, the above approach sometimes yields incomplete blobs [29]. The connectivity analysis also does not work for overlapped parts. Hence, we classify this method as a Global Feature Method. The Stanford Research Institute (SRI) vision module is based on the global feature method [27]. Specifications such as perimeter, centroid, area, and moment of inertia can be determined by algorithms based on the above method.

Local feature method A second approach is an algorithm based on what is called the Local Feature Method, which can measure the defining pieces of an object boundary, the line segments, the arc segments, and the curvature [13, 62]. This method can be successfully applied on randomly oriented regular parts. An extension to the Local Feature Method is what is called the Relational Feature Method in which the interrelationship of two local features of an object is used for part recognition. Again this method is good for a limited domain of regular objects.

Morphological method A third approach for feature extraction is the morphological approach. In this method, the images are treated as sets of points in space, as opposed to connected blobs. The operations performed on the image are set operations such as union, intersection and set difference. The set operations are carried out by using certain structuring elements. The set operations in morphology are destructive in nature. Hence, the image after such operations is essentially destroyed and backtracking from an operation is generally not possible without image reload. Therefore, it is very important to make an intelligent use of the structuring elements to assure a sufficiently high speed image analysis system by minimizing the image reloads. The structuring elements can either be regular in shape such as squares, circles, disks, lines, vectors of particular orientations or they can be irregular in shape [61]. Therefore, this method can be utilized for processing irregular objects for which a certain degree of information about the shape,size, and position, is known.

Image analysis

The next phase in image processing is the interpretation phase. The interpretation requires a domain on which the judgment can be based. Brady [6] presents two generally acceptable methods that are used for image interpretation.

Feature weighting method The feature weighting method in which we look for special features in the image, like an object having a hole in the middle, or an object having a rounded top left corner. A more precise requirement might ask for quantitative measurements of the image features. As in our application, we might like to see if the stem diameter is within acceptable limits. In the grading application like ours, there might be varying aspiration levels and combinations of features for an object to be acceptable. When there are several combinations of features required to reach a proper conclusion, then we can best code the decision rules in what is called an expert system [8, 22, 33, 35].

Template matching method The template matching method, in which representative object models are kept in an object library and the features of the image in question are matched with these models [5, 13]. This method is most commonly used where the domain of objects is fixed. Industrial parts generally fall into this classification [5]. This method is generally used in part finding/sorting applications with no occlusion or overlap.

Machine vision applications

The applications of machine vision in a production environment can be broken down into five categories.

Inspection This area of vision application deals with monitoring quality of a part on a production line. It can be subcategorized as binary and analog. In binary inspection, the part is either declared as a good part or as a bad part. Typical manufacturing applications include blank (stamping) verification and component verification. In analog inspection, the degree of deviation from the desired limits is also monitored and reported [3]. Applications with this classification include isolating objects of a particular attribute, monitoring of dimensional accuracy, hole location, surface flaws, and surface contour accuracy [45]. It is important to discriminate between a good and a bad part as early as possible in the manufacturing cycle, because as we move through the production cycle the bad component results in a bad part, which results in a bad system, and so on. Chang [12] sites a specific example: an electronic component if defective costs \$.05 and its identification and removal before assembly costs \$.50. When it is assembled on the component board, the cost to fix it increases to \$5.0, and when it becomes a part of the system, the cost of removal jumps to \$50.0. Finally, when this system is marketed and the customer requests service, the cost to repair increases to \$500. All this emphasizes that inspection is advisable at the earliest stages of the production process. The same principle applies to natural products, as the elimination of bad tree stock or produce saves the expenses of unnecessarily storing and maintaining them. Also, about 10% of the total labor cost of all manufactured durable goods is accrued in quantitative and qualitative visual inspection even when statistical sampling is used [45]. These costs can be significantly reduced if machine vision is employed. Moreover, if 100% inspection is desired, only using machine inspection makes sense.

A machine vision system to inspect the quality of glass tubing at a GE manufacturing facility has been reported [65]. The edge of the tube is identified by capturing the image with 256 gray levels, and thresholding it to make the edges as distinct as possible. The system then determines the quality of cut at the tube edges by measuring the height of the edge points. Sudden change in thickness signifies a chip from a bad cut. The system also looks at the body of the tube to identify cracks. Since the system was installed, significant reduction in machine downtime, and a significant increase in material efficiency have been reported. Another application examines and identifies baby food jars for glass particles at a rate of 10 jars/sec [43].

Pattern recognition This application differs from inspection in that the purpose is not only to accept or reject the item but to classify it in a certain category.

Pattern recognition techniques are widely used in automatic inventory control applications and in flexible manufacturing systems (FMS), especially in the field of automotive production. In a flexible manufacturing environment, there is push for a greater variety in the product range, delivery times are important, and there is a need for frequent product updates. In such case large machining centers serving a variety of parts, and subsequently, requiring a range of tooling, are needed [1]. Vision systems are used to discriminate between different parts by comparing incident parts with the part features pre-stored in a part library. These operations include part sorting, palletizing, conveyor picking, and bin picking [44].

The problems of processing satellite imagery data and identifying various components of the scene by image enhancement and segmentation would also be classified as pattern recognition. Satellite imagery is used for estimating crop yield and vegetation estimates. It is also used for remote sensing and surveillance. Our application can also be classified as a pattern recognition application as we use stochastic means to categorize the incoming seedlings into three classes; good, bad, and un-identified.

Measurement This area of vision is very similar to the analog inspection, except that rather than monitoring the deviation from the desired specifications of an object, the specifications of the object itself are monitored and reported. This idea can be utilized in applications where numerous parts are to be produced with minimal variation in size and shape [17]. Relational methods are used to identify the parts, in such applications, by identifying clusters of unique relationships among the local features in each part, and keeping this information in a part library. These ideas are also applied where the parts are partially occluded.

In manufacturing situations where the part has to satisfy rigid limits of tolerance, vision systems provide physical measurements of the parameters of interest.

Process control The above categories of vision application essentially deal with open loop (non-feedback) systems. The process control applications can be viewed as an analog inspection system with a feedback loop. The integration of the vision system into the feedback control system allows a direct influence over the manufacturing process [11]. Once the system recognizes a deviation from the desired specifications, it sends the signal to the processing machine for corrective action. The machine temporarily suspends the manufacturing process until the problem is resolved [17]. Industry is especially interested in vision-based process control in the areas of assembly operations in a flexible manufacturing environment and in control of chemical processes in hazardous environments. Use of a vision system in such applications ensures a reduction in departmental downtime and disruptions in production, as the process is stopped only when the vision system detects and reports irregularities in the system. The vision system also provides a continuous monitoring environment as opposed to the occasional monitoring by manual systems.

Guidance and robot control This is the most rapidly growing application area of machine vision. Guidance for autonomous vehicles, orientation devices, and industrial robots, are classified into this application area. Use of robots without a vision system requires that the objects must be presented in exact position, which is a very serious limitation in a work environment. The vision system, when associated with the robot, eliminates this problem, because the vision system monitors the positional accuracy of the incoming objects and reports the deviation to the robot operating system, which sends remedial signals to the robot [8, 74]. Shoup and Macchio [63] developed a model to evaluate the potential use of robotics and machine vision in food processing. A fairly recent approach uses vision cameras mounted directly on the robot arm which provides the robot with what is called *active vision* or *animate vision.* These robots have been programmed to play a simple version of the badminton game [78].

Study of these vision system categories emphasizes the fact that successful pattern recognition is a common denominator in all vision applications. Such a system should have a fast response time in order to keep the system running without delays. Thus, the development of a low cost, real time vision system is an active area of machine vision research. The literature survey has identified a few real time systems that utilize the computer power of mainframe and miniframe machines and which are being utilized in agriculture and other industries [2]. These systems are very expensive; for example, the hardware for the GE Optomation visual inspection system (Syracuse, N.Y.) sells for approximately $60,000.

Machine Vision Equipment

A vision system encompasses three elements: a digital camera, a frame grabber, and a digital image processor (computer). The digital cameras used for image processing applications generally use Charge Transfer Devices (CTD), which are superior to the conventional VIDICON systems as they provide higher sensitivity, operate at low power, are light weight, and provide lag free, and burn free images. The CTD cameras are either the Charge Coupled Device (CCD), or the Charge Injection Device (CID) type. the CID type cameras have the advantage over the CCD type

Table 3.1: Machine vision camera specifications

Source	*Model*	*Pixel*		*Resolution*		*Device*
		V	*H*	*V*	*H*	
Fairchild	CCAID 488	488	380	488	284	CCD
Fairchild	CCD 211	244	190	244	142	CCD
RCA	SID 52501	512	320	480	240	CCD
Hughes	HCCI 100A	100	100	100	100	CCD
Sony	SiCCD Color	492	245	350	280	CCD
Javelin	JE 7362	490	610	450		CCD
Javelin	JE 2362A	485	576	380		CCD
GE	TN 2500	244	248	244	191	CID
GE	TN 2200	128	128	128	128	CID
MicronTech	IS256	512	128	200	640	OpticRAM

cameras, as they support random access addressing, and superior anti-smearing and anti-blooming control. The CCD devices, however, are more responsive to low-light-levels [23]. Table 3.1 lists a few cameras suitable for machine vision applications.

The GE TN2500 camera listed in the table is a fast camera with a data transmission rate of up to 4.5 MHz per byte and standard video rate of 30 frames per second. It has high resolution with 488 lines of scanned video on a 525 line standard TV display in interlaced mode. It has Automatic Gain Control (AGC) capability for performance in low level lighting conditions [73]. This is a typical example of a good choice for capturing images in an industrial environment allowing for random access to the image areas which is certainly of advantage for irregular object applications. This camera and the control unit were available in the Manufacturing Automation Laboratory.

Once the image is captured by the camera, it has to be loaded on the computer memory. A frame grabber (memory buffer) is needed for synchronization between the camera (transmitting end) and the computer (receiving end). There are numerous vendors who make frame grabbers for vision applications. Most of the frame grabbers allow standard NTSC analog input. A substantial number of such boards are designed for desktop publishing and media applications, which generally are not suited for machine vision applications. One such board is the AT&T Image Capture Board which is available in our automation laboratory, supports color input, and is primarily meant for presentation graphics applications. The board is plug compatible with any IBM PC compatible microcomputer and has a fast capturing speed (30 frames per second). The only problem with the board is that it provides discrimination for 32 levels and supports a lot of features for image enhancement which are primarily meant for media applications. As a result, a single image is about 416,000 bytes in size. In our application, we are dealing with a binary image which would only be about 16,000 bytes for a 640x200 (CGA) resolution. Hence extensive pre-processing would be required if we use the AT&T board. This would add significant overhead to the overall processing time. The time constraints in our application are such that we have to keep the processing overheads to a minimum, therefore, the board is not suitable for our application.

The camera selected for this research was Micron Technology Inc.'s IDETIX IS256 digital vision system camera. The PC bus compatible frame grabber board for this camera is readily available. This camera is based on the OpticRam module placed in the camera head. The camera is relatively inexpensive because of the optic RAM as opposed to the CCD (Charge Coupled Device) or CID (Charge Injection

Device), but is less sensitive to minor variations to light. It has a slower response time, and needs relatively higher soak time to capture an image. Optic RAM cameras are strictly a binary image type device (i.e., they do not lend themselves to respond to shades of gray). However, shades of gray can be generated by capturing multiple images by varying capture/soak time, by changing the camera F-stop, or by actually varying the light on the target image. The first method is generally easier to use and provides a better control. The camera does support the strobe action, however, due to hardware limitations, this is only practical if the image has a high contrast and has fewer or preferably no shades of gray. The image at any rate, is required to have a high contrast [37]. We used this camera system, as all the system components were readily available.

Since this camera was used in our system, we will attempt to describe its features in a bit more detail [37]. The Optic RAM used by the camera is organized in four arrays of 512x128 electrically addressable elements, or 262,144 pixels. The architecture of the IS232 (the Optic RAM) is such that it has three dead zones between each array of elements. This imposes the limitation on the camera to be strictly level with respect to the target image. If not, then relative shifts in certain segments of the image are evident. Logic level of 0 is associated to a bright pixel, while, black pixel has a logic level 1. Initially, all the elements are charged to a logic level 1. When these elements are subjected to light, they are discharged. If the light intensity is high enough to discharge a particular element past the threshold, its logic level drops to zero. Another parameter which affects the logic level of an element is the amount of time the element is exposed to light. The longer the elements are exposed to light, the greater is the extent of discharge. Hence the combination of intensity and duration determines the shape of the image which is supplied by the camera. To minimize the size of the optic RAM, the image sensors are arranged in an interleaved fashion, rather than linear fashion. Hence, the array of sensors is much longer than wide, resulting in space between cells in the column direction. The above topology suggests that straight linear read of the cells will result in a scrambled image. The camera is supplied with a library to implement de-scramble logics to provide images in a variety of formats. The format used for this research was a resolution of 640x192 pixels. It was later observed that 144 rows were enough to get a complete image, hence the image in the system was clipped to 144 rows. The last 56 rows were used to display status and feedback information on the monitor.

The MicronEye image capture board features a high speed data transfer rate using Direct Memory Access (DMA), which provides data transfer without intervention from the central processor unit. The data transfer rate is 6.4μsec per data point, for an IBM/XT computer [71].

The microcomputer desired for our application was the IBM compatible personal computer. The code was designed to operate under any machine from the PC family. However, the computer having higher processing power and a higher level of processor (word size), would result in faster processing speeds. The computer used for this research was Zenith 248 computer, as this machine was readily available in the laboratory and met the requirements. The memory requirements for the final version of the program required at least 640K of Random Access Memory (RAM), which is incidentally, the higher limit supported by MSDOS, the disk operating system used by the above computer. The PC family of computers support Direct Memory Access (DMA) capability and have a large number of data Input/Output ports, which are required to interface the computer with external control devices like a frame grabber board [71].

Expert System Techniques

Expert systems fall into the broad umbrella of artificial intelligence applications, as the efforts supported by these systems require some degree of intelligence. Intelligence, as defined by Henry Bergson in early 1907 is "...the ability to create artificial objects, in particular, tools to make tools" [60]. We can see how valid this statement is by observing the enormous research activity in the present age to take the object oriented approach to computer problem solving. The domains of artificial intelligence research can be classified into the following categories:

- Knowledge-based systems
- Natural language processing (understanding)
- Intelligent robots
- Scene and signal analysis
- Information Retrieval
- Intelligent tutoring/training systems
- Automatic theorem proving
- Automatic programming

Since expert systems are bound to emulate the logic and reasoning of a human expert, based on the knowledge provided to the system, they fall in the category of the knowledge-based system. Expert systems are tools for solving problems from a specific domain of knowledge [15]. An inherent characteristic of expert systems is their ability to answer questions that are not in their knowledge-base by performing what is called *plausible reasoning*, by making judgements on what might be true, given certain assumptions about the domain [14]. If this characteristic is missing from a system, we term the system, merely, a knowledge-based system. Examples of knowledge-based systems which would not be classified as an expert system, would be an automated system for auditing an undergraduate program of study or an income tax return preparation system. Our seedling segregation system employs plausible reasoning to resolve the ambiguities incorporated by the undesirable interaction of the environment and the highly random nature of the image constructs. It is particularly used in the clamp identification and removal stage of the system. The difficulty with heuristics and intelligent defaults is that they can not be guaranteed to work, even though they may appear plausible [59].

Expert System Composition

Expert systems are the computer programs which attempt to predict, advise, and solve, problems by using the approach of the human expert in that area. Primary components of a knowledge-based expert system are composed of the knowledge base, inference mechanism, user interface, and an explanation facility. The advantage of having the knowledge-base separated from the rest of the components, is that several applications can be developed by just changing the knowledge base while keeping the rest of the components. This approach results in the availability of what are called the expert system shells. This compartmentalization approach in developing expert systems reduces the cost of developing and maintaining expert systems [35].

The expert system knowledge-base is developed by so called knowledge engineers through a knowledge acquisition procedure. The human experts are interviewed and are asked to answer a pre-designed set of questions. It is this information in the from of knowledge-base which is is used to resolve user inquiries. Therefore, the importance of having an accurate representation of the knowledge-base can not be over-emphasized. This information is commonly entered as a set of production rules, which are a set of IF-THEN constructs. These rules can either be propositional, in which case all the variables used in the expressions have their values pre-assigned, or they can be predicate compositions, which permit qualifiers like (for all..., there exists...). The later can contain un-assigned or free variables [76].

The inference engine (mechanism) is the process of applying the user query against the knowledge-base to search and reason. When a user queries the data base by requesting advice, his query is transformed into goal clauses. The inference engine resolves the queries by what is called resolution refutation, which is the process of negating the goal clauses, and adding them in the set of rules which exist in the knowledge-base.

As an example, let's consider the following facts from a rule base.

1. **John likes all kinds of food.**

2. **Anything one eats and isn't killed by, is food.**

3. **Bill eats peanuts and is still alive.**

The first step in the process of setting up a knowledge base is to represent the rules in predicate logic. The facts when properly transformed look like:

1. $\forall_x$ {**FOOD(x) $\Longrightarrow$ LIKES(John,x)**}

2. $\forall_x \forall_y$ {**EATS(x,y) $\wedge \sim$ KILLED_BY(x,y) $\Longrightarrow$ FOOD(y)**}

3. **EATS(Bill,Peanuts) $\wedge \sim$ KILLED_BY(Bill,Peanuts)**

The statements are then transformed into clausal form by applying a set of steps. These steps are outlined in reference [50]. The results of these operations, are the rules which form the knowledge base for the expert system.

The translation results in the following rules:

R1> $\sim$ **FOOD(X1) $\vee$ LIKES(John,X1)**

R2> $\sim$ **EATS(X2,Y2) $\vee$ KILLED_BY(X2,Y2) $\vee$ FOOD(Y2)**

R3> **EATS(Bill,Peanuts)**

R4> $\sim$ **KILLED_BY(Bill,Peanuts)**

Now, we can make the following query from the knowledge base:

Name a food, John likes?

In the process of resolution refutation, we add the goal clause and its conjunction with the negation of itself in the set of rules. Hence, the new rule in the rule-base after transforming the above query into clausal form and its conjunction with the negation becomes:

R5> **LIKES(John,X4) $\vee \sim$ LIKES(John,X4)**

The last step is to apply the rules and try to reach a conclusion by getting a *toutology* for John's liking. By toutology we mean a clause with no variables in it.

The process of applying these rules is shown in Figure 3.2, which shows the result that John likes Peanuts.

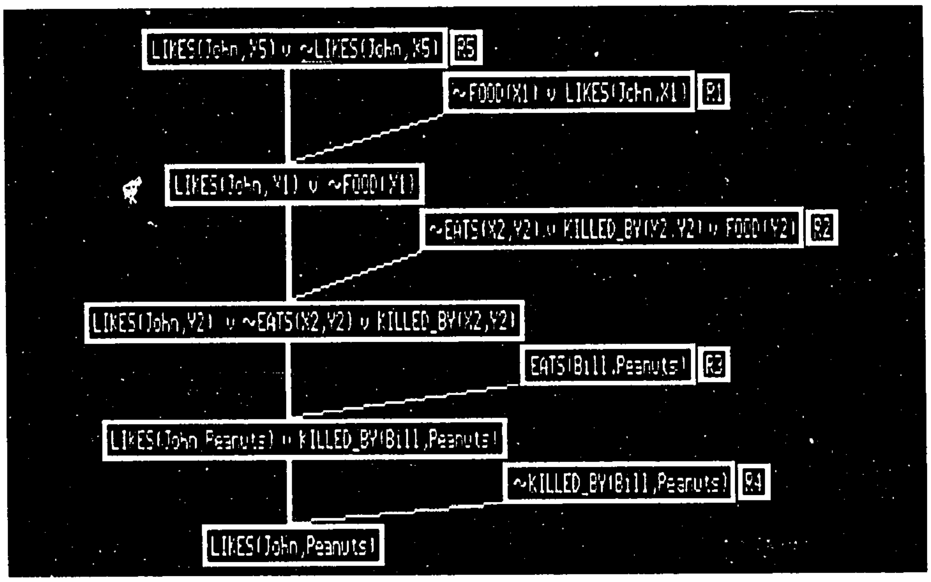

LIKES(John,X5) v ~LIKES(John,X5)
R5
~FOOD(X1) v LIKES(John,X1)
R1
LIKES(John,Y1) v ~FOOD(Y1)
~EATS(X2,Y2) v KILLED_BY(X2,Y2) v FOOD(Y2)
R2
LIKES(John,Y2) v ~EATS(X2,Y2) v KILLED_BY(X2,Y2)
EATS(Bill,Peanuts)
R3
LIKES(John,Peanuts) v KILLED_BY(Bill,Peanuts)
~KILLED_BY(Bill,Peanuts)
R4
LIKES(John,Peanuts)

Figure 3.2: Process of resolution in a rule base by refutation

The process of resolution is used to show that negation of the goal clause results in a contradiction with the knowledge base, hence, the name resolution refutation [50]. The internal (low level) representation of the knowledge-base is in the form of a decision tree. This makes the order of the rule entry irrelevant. One great advantage of expert systems is their ability to back-track from a branch of the search tree if that branch does not result in a solution. There are two common strategies: the depth first strategy, and the breadth first strategy. The former traverses one branch of the tree until it is exhausted, while the later, computes all of the first level resolvents. The depth first strategy is more efficient in most of the cases.

The program keeps track of the rules applied to reach the decision. The user can invoke the explanation facility in order to examine the validity of the results. The component by which the user interacts with the system is the user interface. It must be noted that it is not necessary for the user querying the system to be human. As in our case the system is queried by the image captured by the system and the characteristics derived from the image. The ultimate user for the grading information, however, is the human user. It is important to be aware of end user needs in determining the output format and the format of the knowledge-base. The later is important in facilitating maintenance and house-keeping of the expert system. In our system, we generate a data base of the system status. This data base is duly time stamped and reflects any change made in the system status, the specie being used, the number of good, bad, and total for that specie, and the system-wide totals of good and bad seedlings, at the time system status was changed.

In an application like ours, where it is difficult to identify the components of interest from the environmental factors or noise, we use the approach of qualitative prediction. This approach is inductive in nature as we build our identification or resolution process in terms of the causal relationships, and dependencies. The type of induction is classified as *instance-to-class* generalization, if we formulate our rules from a large set of known instances [48]. For example, in our system the rules to identify and eliminate the clamp, the rules to track the stem path, and the rules to provide special processing in the needle area (in case of Pine), we subjected numerous seedlings to our system, identified various constraints and formulated the rules from those constraints. The other type of induction involves the *part-to-whole* generalization, in which knowledge about the system is fragmented and we have to satisfy a goal based on that knowledge by a process of reconstruction [47]. The tap root identification and the identification and counting of lateral roots by our segregation system uses part-to-whole generalization, in which we use a stochastic approach to discriminate between lateral root, noise pixels, and inflecting portions of the tap root. The detailed description of the segregation methodology is presented in Chapter 4.

In order to establish baseline criteria for a broad based expert system and to institute sufficient segregation rules to accommodate the variety of soils, environments, and species in different planting regions, a greater knowledge base is needed. The nursery experts have varied opinions on this issue, therefore, it is difficult to have a generally acceptable expert system. We relied on the Ames Nursery experts for our grading rules.

Expert System Applications

As noted earlier, the availability of expert system shells and special computer environments for expert system development have greatly increased the rate at which the expert systems are being produced. Another factor in this rapid growth is the use of microcomputers in expert system development. Use of expert system shells is feasible, where the number of rules is large, and where processing time is not of primary concern. Both of these conditions do not apply to our system. The systems developed before the expert system tools were available were difficult to construct, required long lead times, and were very expensive. The classical examples from this era of expert systems are MYCIN (1974) for medical diagnosis and therapy, DENDRAL (1978) for searching chemical structures, PROSPECTOR (1979) for molecular genetics, MACSYMA (1971) for mathematical formula manipulation, and the geography teaching system SCHOLAR (1970). Often, new expert systems are developed as an extension to the previous expert systems after learning from the pitfalls and discrepancies or success from the previous generation. For example, WHY (1980) was developed following SCHOLAR which added causal reasoning to the purely factual reasoning of SCHOLAR. Generalization of MYCIN to EMYCIN (1979); GUIDON (1979) the tutorial system designed to be compatible with any EMYCIN knowledge base; and reformulation of MYCIN explanations in NEOMYCIN (1983).

Expert System Software Characterization

As discussed in the previous section, since the number of rules in our rule base is limited, and since we are designing the system for an application where processing time is of primary importance, we did not use special expert system shells, or special expert system languages. The discussion in this section about expert system shells and logic programming languages has been included for the sake of completeness. The expert system languages are specially designed to code symbolic logic. The most common expert system languages are LISP and Prolog; OPS5 being among the less common ones. LISP developed at MIT has the advantage that it is the most widely used AI language, has a high application base, and is highly modular. Among the disadvantages; it runs inefficiently on conventional hardware, and requires a high amount of real and virtual memory. Prolog's greatest strength is its automatic and full support for backtracking, a fundamental operation in AI applications, and its ability to provide a direct and natural translation of predicate logic. It too can be quite inefficient for certain applications requiring a large number of backtracking operations. Prolog was adopted as a basis for Japan's fifth generation project [39].

The first microcomputer based AI shell is reported to be the MicroExpert [60]. Other systems include the ES/P Advisor, ExpertEase, and Personal Consultant Plus. MicroExpert and ES/P Advisor are Prolog based systems, while Personal Consultant Plus is LISP based. Another major expert system shell TWAICE was introduced in Prolog in 1985 [46].

Developing expert systems, quite often, is an involved task. A small expert system project might need one scientific-year effort, while, larger projects need as long as 10 or more scientific-years and can cost as much as $70,000 in equipment [32].

A new trend in languages for writing expert systems is to use object-oriented languages, since they provide a natural way to represent expert system entities [69]. SMALLTALK-80 is the most widely known general purpose object oriented language. New expert system object oriented languages include FLAVORS, LOOPS, and KEE [69]. The ES/P is also based on objects. Modern programming languages like Pascal and C are taking an object oriented restructuring. Object oriented Pascal has been introduced and a new version of C called C++ is gaining popularity. (It will be interesting to see the impact of these new transitions in using these languages for expert system applications.)

CHAPTER 4. DESIGN METHODOLOGY

Image Processing Considerations for Processing Irregular Objects

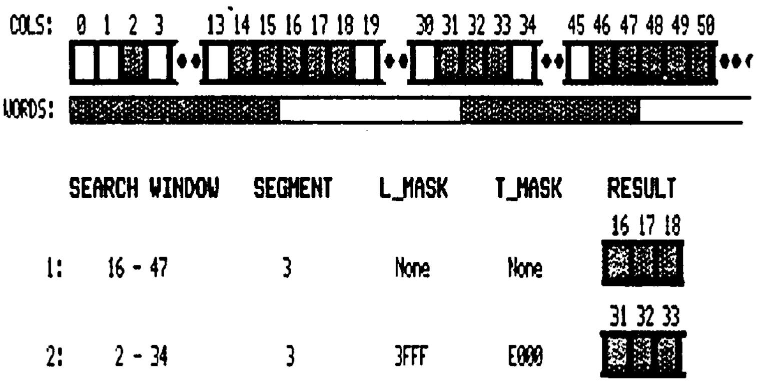
There are added restrictions when dealing with irregular objects. The image processing tools developed for processing irregular objects must accommodate such restrictions. When dealing with regular objects, noise (undesirable reflection from surrounding environment) can usually be suppressed without degrading the quality of the image of interest. This is accomplished by masking off isolated and unconnected pixels in the image, as they have no relationship to the object of interest. In the case of irregular objects, the isolated pixels, or isolated streaks of pixels, can not be ignored as they might be part of the object itself. Also, the defining elements like edges and holes in a regular objects can be discriminated from the object surface by adjusting the threshold of the gray scale. For example in a 256 level gray scale image one might specify a threshold of 100. This would mean that all the pixels which have a gray scale value of less than 100 would be treated as dark while the rest would be treated as bright. In case of irregular objects, we do not have this luxury. In case of regular objects, we can store the template of objects, or the definitions of objects, in a part library and compare the incident image with this library to identify an object. Object definition include such deterministic features as perimeter, centroid, moment of inertia, coordinate information of a particular feature, and triangulation parameters which are generally stored in a parts library. In the case of irregular objects these values are of less significance and are very inefficient to determine. With regular objects, mathematical morphology (the process of applying mathematical treatments to the image by set operations) can be applied. While, with irregular objects this is usually not possible, as generally there are no mathematical models to represent the image or image parts.

In case of irregular objects we rely on the stochastic information at hand. For example, in our application, once we have identified the starting point of the needle area, we know that in the stem area, it is unlikely for the image of the tree seedlings to have a blob of pixels as high and as wide as the clamp on which they are mounted. This emphasizes the fact that the routines to process irregular objects must have dynamic windowing features. The preceding discussion also portrays the fact that in order to have an efficient system the processing speed of the routines is very important as they have to be utilized very heavily.

Also, with regular objects the point where the image processing task should start, is of less significance. This is because the object in general has some degree of similarity around its centroid. It does not usually matter if the analysis begins from the top left corner of the screen, or the bottom right corner of the screen, for example. This, in most of the cases, is not true with irregular objects. In our application, if we are to process the stem area, it would be impractical to start from the left edge of the screen, as we know that the stem area is closer to the right edge. We will show in the next section that it is to our advantage, if we process the clamp in right to left fashion as well.

Features of Line Segment Search Routine

The Local feature method (described in Chapter 3) and the line segment tracking method, is used as the primary method for image processing in our application. This approach is opted, as we are dealing with irregular objects, and a segment search operation can be handled most efficiently by the Central Processing Unit (CPU) of the computer, specially, if the segment length is a multiple of the CPU word length. For example, the microprocessor of an 8086 or an 80286 processor would process segments which are multiple of 16 pixels, most efficiently. If a segment to be searched is not at the word (two byte) boundary, bit masking techniques are used on the leading and/or the trailing word of the search segment. The cells which are to be ignored from the leading and the trailing word of the search space are logically ANDed with a mask value. As a result, the pixels to be ignored, are always turned off, or assigned a value zero. This process is shown in Figure 4.1. A typical section of the image memory from column 0 to column 50 for a row is shown. The shaded cells are associated with bright pixels while other cells represent dark pixels. The cells which are skipped, are assumed to have a value of zero (dark). The first example listed in the figure is the process of finding a segment of 3 bright pixels in the search space from columns 16 through 47. As the start and the end of the search window fall on the word boundary of 16 pixels, the leading mask and the trailing mask are shown to be zero. The routine results in the first segment of length 3, which is at column 16 through 18. In the second example, we look for the same segment length in the search window between column 2 and 34. This time neither the leading edge nor the trailing edge fall on the word boundary. We explicitly turn off the pixels which are in column 0,1 and 35 through 47, by using the leading and trailing mask of 3FFF and E000 (in hexadecimal) respectively. It should also be noted that the first example picked columns 16 through 18 as being of length 3, because the search window was such that the preceding columns 14 and 15 were ignored. In the second example since the search window included columns 14 and 15, the segment returned by the routine was from column 31 through 33. To make the routines compatible with a variety of video formats, the decision was made to have support for a user definable search space. The user at the beginning of the image processing task defines the search space, he or she wants to work with. For example, in this research we use IBM Color Graphic Adapter (CGA) format, which is 640x200 pixels. We pass this information to the routine in the beginning and from there on our search space is defined.

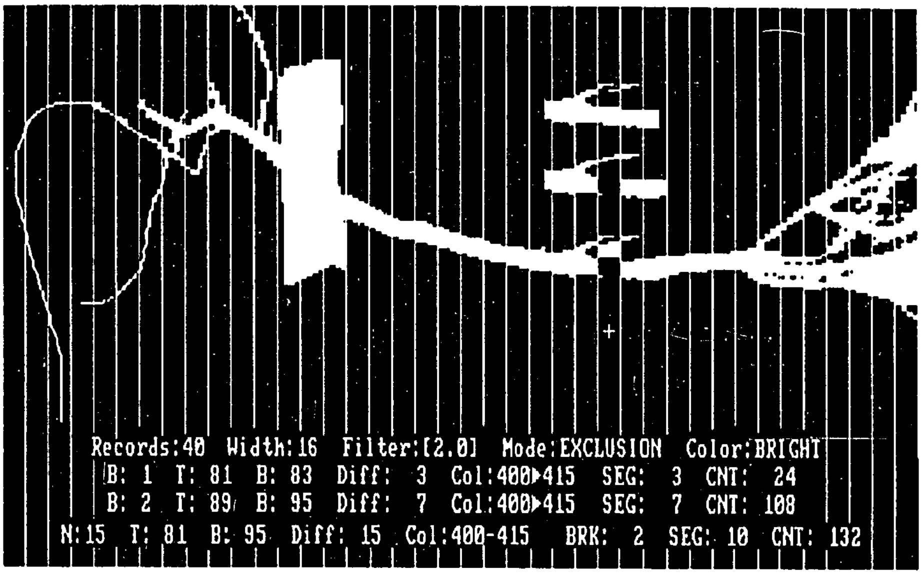

COLS: 0 1 2 3 13 14 15 16 17 18 19 30 31 32 33 34 45 46 47 48 49 50
WORDS:

	SEARCH WINDOW	SEGMENT	L_MASK	T_MASK	RESULT
1:	16 - 47	3	None	None	16 17 18
2:	2 - 34	3	3FFF	E000	31 32 33

Figure 4.1: Line segment search and masking examples

It is also important to have a feature to isolate and lock-on to specific areas of the image to make the image processing task more efficient. For example, in our application once we have isolated the root area and we are ready to process it, it is more efficient to limit the image processing activity to the root area. This is accomplished by specifying the dimensions of the search space which covers only the root area. The routine updates its global variables for the search window, and processing is automatically limited to the specified dimensions of the window. The routine monitors the integrity of the search window, and generates appropriate error codes if it is violated.

Another, advantage of having the dynamic windowing feature is the ability to process the image in small vertical panels. An example of such panel processing is shown in Figure 4.2. The interesting features of the image processing task are displayed on the screen output. The image is shown to be divided into panels of sixteen pixels. We use a filter of 2 pixels as a base value to exclude stray pixels and their association to other segments of the picture. The figure shows the results for panel 15 (N:15). It indicates that there are two blobs (or regions) in that panel (BRK:2). The regions begin at row 81 (T:81) and are 15 (Diff:15) rows high. There are 10 segments in the given range in panel 15 and the total pixel count is 132. The routine is also capable of returning the positional information about the regions, in a single call. The two cutouts of the image show the two regions as identified by the routine. The regions start at row 81, and 89, and are 3, and 7 rows wide, respectively. By using these criteria, we can track the stem area, by ignoring the regions which do not connect to the region found in the previous panel. This is evident in the two cutouts, where the first cutout from the top highlights the branch, while the next highlights the stem in the particular panel. Details of the stem area processing are given in the fifth section of this chapter.



```
  Records:40  Width:16  Filter:[2.0]  Mode:EXCLUSION  Color:BRIGHT
  B: 1  T: 81  B: 83  Diff:  3  Col:400▸415  SEG:  3  CNT:  24
  B: 2  T: 89  B: 95  Diff:  7  Col:400▸415  SEG:  7  CNT: 108
N:15  T: 81  B: 95  Diff: 15  Col:400-415  BRK:  2  SEG: 10  CNT: 132
```

Figure 4.2: Panel processing by the segment search routine

As discussed in the previous section, in the case of irregular objects, the starting point for the processing task is important. The line segment processing routine is capable of operating in top to bottom or bottom to top fashion, as well as, from left to right and vice versa. The routine has a resolution of one pixel. In other words, it can identify and isolate segments that have segment length of unity. This feature can be used for image filtering and enhancement if the stochastic information about the object of interest supports this idea.

The routine is capable of searching for segments of a specific length or segments in a given range. Conversely, it can also return information on segments which do not belong to the specified range. This is accomplished by supplying a base value for search, and an adjustment value, which (when added to the base value) sets the upper/lower limit for the segment search process.

Clamping Point Identification

The search window covering the image of interest is called a frame. Although, the grading system allows for multiple seedlings (or their parts) to show-up on the same image, it is assumed that the seedling to be graded does not fall on the edges of the screen. Therefore, before we do any further processing, we isolate the image of interest from other partial seedling images. The system generates appropriate informational messages if it finds that the image of interest falls on the screen edges. The first step in accomplishing the above task is to find the top and bottom bounds of a prominent seedling. The following pseudo-code outlines this task:

```
START from the middle row
WHILE not at the top row DO:
          Process a row from bottom to top
          IF found a blank row
              IF found three consecutive blank rows
                  Set frame start row and EXIT
              ELSE
                      REPEAT
              END.
          ELSE
                  REPEAT
```

```
            END.
        END.
11      FAIL
```

We start from the middle row of the screen and scan in bottom to top fashion. The objective is to look for three consecutive blank lines at the top row of the viable seedling image. Above steps give us the top row of the seedling image and isolates the image from any partial objects on the top of the image. In order to get the bottom row of the image of interest, above steps are repeated in top to bottom fashion from the middle row of the screen. Once the top and bottom rows for the frame are established, the image processing window is adjusted accordingly.

It is assumed that the image of the seedling is in the range of 200 pixels from the right of the screen. If not, the operator is advised to move the image to the right. This is to minimize blank pixels on the right of the screen, as major processing tasks for the stem and clamp area are initiated in right to left fashion. The next step is to see if the image is continuous for about 200 pixels. This is to make sure that the stem area and the starting part of the root area are captured in a single piece. If a break in continuity is found, we declare that as not being a seedling and we proceed further by setting the frame right edge to that column position, and looking for continuity for the next 200 pixels. If the continuity is not found till the left edge of the screen then a message is generated indicating that the image has not been found.

We then find the left edge of the frame. This is done by conducting a panel-wise processing from the previous step to the end of the seedling. The width of the panel is kept 16 pixels (equal to the word length of the microcomputer CPU for the sake of optimal search efficiency) which is sufficiently low in order to trap all the stray segments of the seedling image. If the end of the frame is not found, the operator is asked to move the image to the right. This completes the procedure to isolate the image of a single viable seedling.

The next step is to find the starting point of the needles (NeedleStart), on seedlings such as pine. By running several experiments, it was found that if we process an image (from here on we use the word image to mean an image of a single viable seedling) in a panel-wise fashion from its tip (right edge), the point at which the number of pixels in a panel is reduced to 180 (or less), the bulk of needle area is surpassed. However, we make sure that this is not the first panel, as the edge of the seedling close to the panel boundary might have pixels fewer than 180 or the tip of the seedling might as well have a pixel count below the threshold of 180. To save some processing time, we first find the top and the bottom limits of the seedling in each panel as we go. In case we find that the first and the second panels have fewer than 180 pixels, we conclude that the seedling is of a type where there are no needles (such as oak). In such species, the leaves are already gone before they are subjected to the grading process in the fall. If we find that the NeedleStart is less than 200 pixels from the left edge of the screen, we check that the left end of the image is within screen limits. If we find the limits within the left edge of the screen, it indicates that we are looking at an object other than a seedling, and we generate an appropriate informational message. On the other hand, if we can not find the left side limits within the screen bounds, we advise the operator to move the seedlings to the right.

The tree seedlings are mounted by a clamp. In the laboratory experimentation, a masking tape was used to serve as a clamp. It was assumed that the seedlings would be mounted at their caliper, and they would be mounted in such a way that most of the roots are exposed to the image capture system, and that the tap root would be kept horizontal as far as possible. The constraints discussed above were imposed on the segregation system to facilitate the seedling identification and grading process. Mounting the clamp at the caliper, means that portion of the image to the right of the clamp is the stem area, and that to the left of the clamp is the root area. Therefore, once the initial pre-processing of the image is completed, the next task is to identify and eliminate the clamp. The details of this process are outlined in the flowchart presented in Appendix C.

Finding the NeedleStart first, assures us that we are not looking for the clamp in the lumps of the needles. We might have a certain region in the needle area which satisfies our criteria to identify a clamp, and thus result in an identification error. The risk of mis-identification in the stem area is minimal. At the start of the program, the operator enters a range for the clamp width in pixels. He specifies a base value and an adjustment to the base. The advantage of having these values at the start of the system is to compensate for the variations in the setup for the image capture system and the impact of the lighting conditions. With our laboratory set up, we found that a value of 25 pixels for the base and an adjustment of 10 pixels worked in most of the cases.

The first step in finding the clamp is to find a valid clamp segment in the specified range from the base and the adjustment value as entered by the operator. We do this processing in top to bottom and right to left fashion. We start from NeedleStart column, which was found in the previous step. The line segment search routine operates in a line-by-line fashion in the search space. We do not know in advance where the clamp is located, and therefore, we keep our search space covering the whole image area. Now, if we look for the clamp segment (within the specified range) in the whole search space, there is a chance that a segment to the left of the clamp might satisfy the acceptable range. This way, the system would be unable to 'see' the clamp. For this reason we search for the clamp in panels of 32 pixels (2 word width). We look for a segment in the range of 16-32 pixels. Once we find a segment in this range, we expand the width of the search panel to make it wide enough to cover the clamp width.

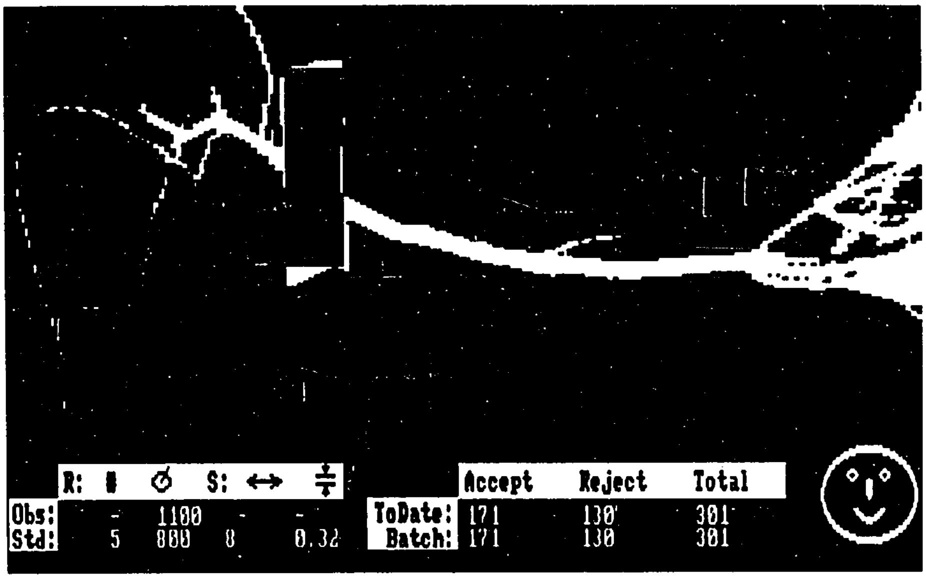
We find the viable clamp top by the above steps. The real clamp top and bottom limits are found by stretching the search window on both ends of the potential clamp segment using a pixel shift (skew) of 1/2 the base width as specified at the command line. If a segment of the desired range of clamp width is found from the top, we find a similar segment from the bottom. If no such segment exists in the expanded panel, then we move to the next panel. We then make sure that the clamp is at least 30 pixels high, it has at least 20 contiguous segments, and has an average segment length of at least half of the clamp width specified by the user. If one of the above criterion fails, we go back to the step of finding the next potential clamp top. The steps described above are listed in the following pseudo-code:

```
START at NeedleStart and proceed panels of width=32
DO:
    Find a segment in 16-32 range from top to bottom
    IF complete image processed then FAIL
```

```
5           IF segment not found then REPEAT
6           Look for the top clamp segment (with specified range)
                  in expanded search space
7           IF not found then REPEAT
8           Find similar segment from bottom
9           IF RowDiff < 30 then REPEAT
10          Find number of contiguous segments in clamp area
11          IF < 20 contiguous segments then REPEAT
12          Find total pixel count in clamp area
13          Compute average segment length
14          IF average segment length > 1/2 of clamp width then EXIT
      END.
```

Above steps give us the widest area in which the clamp might be enclosed. This area might occlude the real caliper point if it has a convex turn towards the root area. We overcome this possibility by first taking the total pixel count in the clamp area and averaging it out for a 2 pixel width panel. Then we sweep the clamp area from its root-side edge, until we get the pixel count in the panel equal to or greater than the above average. The results of processing an image by these steps are shown in Figure 4.3. The next step is to find the real top and bottom row of the clamp, as some of the segments at the tip of the clamp might not have the pixel count in the range provided by the user. This is evident in the Figure 4.3 where there are a few white segments showing on the top and bottom of the rectangular mask for the clamp. This is done by scanning vertically from the edges of the clamp area. At this point, we make sure that the clamp is not clipped at the top or bottom edges of the viewing area. If clipped, appropriate informational messages are generated.

The processing steps described above, are to find the best template for the clamp. This template is a regular rectangular area having the major body of the clamp enclosed in it. The template has been shown in Figure 4.3 to blanket major portion of the clamp. However, the figure does not show the result of the vertical scanning on the clamp.



```
      R:  ▮   ⌀    S:  ↔   ÷            Accept   Reject   Total
Obs:      -   1100      -   -  ToDate:  171      130      301
Std:      5   800   8       0.32 Batch: 171      130      301
```

Figure 4.3: Results of initial clamp identification process

Now, we have to find the top and bottom edges of the root and stem at the clamp left and right edges, respectively. We do this by conducting an on_segment processing on the edges of the clamp. By on_segment we mean; the end columns of a segment at the point of interest (focal point). On_segment is said to be uni-directional if only the following edge of the segment from the focal point is retrieved. It is bi-directional if the trailing edge is also retrieved. Figure 4.4 shows two examples of on_segment processing on a typical segment of memory. The focal point for both examples is column 16. The first example shows results for a uni-directional on_segment processing when scanned in left to right fashion. In this example a segment of 3 pixels is reported, as only the tree shaded cells to the right are processed (focal point inclusive). Second example shows the results when a bi-directional on_segment is requested. This time the result is the segment of 5 pixels, which is the length of the segment incident upon the focal point.

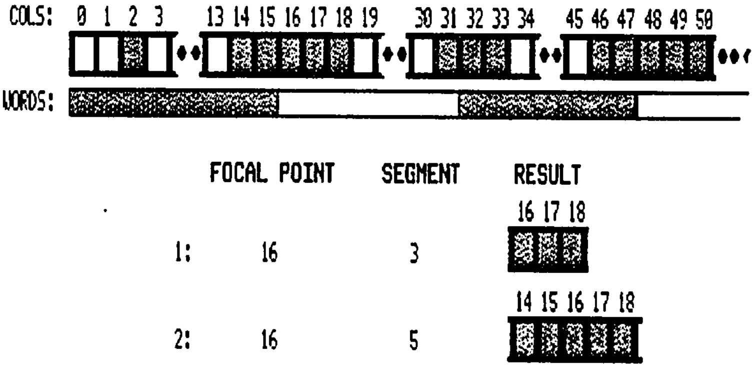


Figure 4.4: Examples of on_segment processing operations

The clamp identification process, from here on, is outlined in the pseudo code which follows this discussion. We proceed with the hypothesis that the stem and the tap root are attached to the clamp in such a manner that they at least have a single segment which is greater in length than the range of the clamp width (COUNT1 in the pseudo code). To find the Stem Clamp Top Row (SCTR), we start at the Clamp Top Row (CTR) and the Clamp Right Column (CRC). Scanning from left to right, We look for a uni-directional on_segment that has a length greater than the clamp width. If not, we stretch the CRC if the right limit on the on_segment is greater than the CRC in Steps 27 through 28 (some of these segments can be found in Figure 4.3). Otherwise, we mark the point, and we say that we are on the stem (Step 5). Now, we want to scan back in bottom to top fashion to include the stem segments which might be smaller than the one found above and still belong to the stem. We do an on_segment search until we either reach the edge of the stem (Step 9), or we find two consecutive segments which are within the limits of the clamp width (Steps 12-14). We mark the row found in this manner as the top row of the stem connected to the clamp (Step 10/Step 16). We restart the processing from the previously marked point, to find the bottom row of the stem, much the same way as we found the top row. Once the bottom row for the stem (SCBR) is found, we go on processing in top to bottom fashion, until we reach the bottom limit (CBR) of the clamp. The CBR is adjusted, if we find any segments which overhanging from the previously found clamp area. We take the maximum of the CRC from the clamp top and the CRC from the bottom to find the real clamp right column. Similar processing is done on the root side to find the top row (RCTR) and bottom row (RCBR) of the tap root and the real left edge (CLC) of the clamp.

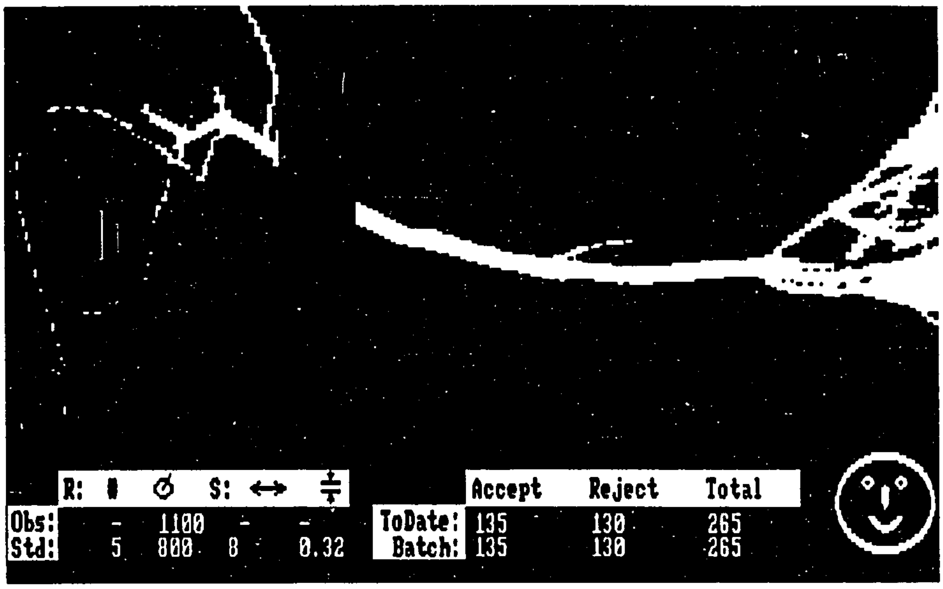
```
1    START at last found Clamp Right Column (CRC) and
               the Clamp Top Row (CTR)
```

```
Set COUNT1 as the largest possible right column according to
    the clamp range specified
WHILE the Clamp Bottom Row (CBR) not reached DO:
        IF a segment found
           IF segment beyond COUNT1
              IF not found the stem top row
                 Scan from bottom to top
                 DO:
                      IF on the stem edge
                         Update CRC and mark Stem Clamp Top Row
                                (SCTR)
                         EXIT.
                      END.
                      IF segment beyond COUNT1
                         Check for next segment beyond COUNT1
                      END.
                      IF two consecutive segments beyond COUNT1
                      OR an expanding segment length
                         Update CRC and mark SCTR
                         EXIT.
                      END.
                 END.
                 Initialize Stem Clamp Bottom Row (SCBR)
              ELSE
```

```
                DO:
                    Mark current column limit
                    IF found bottom edge of stem
                    OR found segment within specified clamp range
                    OR an expanding segment length
                        Mark SBTR
                        EXIT.
                    END.
                END.
            END.
          ELSE
          Update CRC
          END.
      END.
END.
```

Once we find the Stem Clamp Top Row (SCTR), Stem Clamp Bottom Row (SCBR), Root Clamp Top Row (RCTR), and Root Clamp Bottom Row (RCBR), we can eliminate the clamp from the image and from any further image processing considerations. The image with the clamp removed, is shown in Figure 4.5.

We can now calculate the caliper diameter for the seedling by taking the difference of RCBR and RCTR. This suggests that the axis for the caliper is parallel to the clamp. If the axis in actuality is not parallel, the error in caliper diameter would not be appreciable, as the width of the caliper would be a few rows. However, this



R:	▮	⌀	S:	↔	÷
Obs:	-	1100	-	-	
Std:	5	800	8		0.32

	Accept	Reject	Total
ToDate:	135	130	265
Batch:	135	130	265

Figure 4.5: Completed clamp identification and elimination process

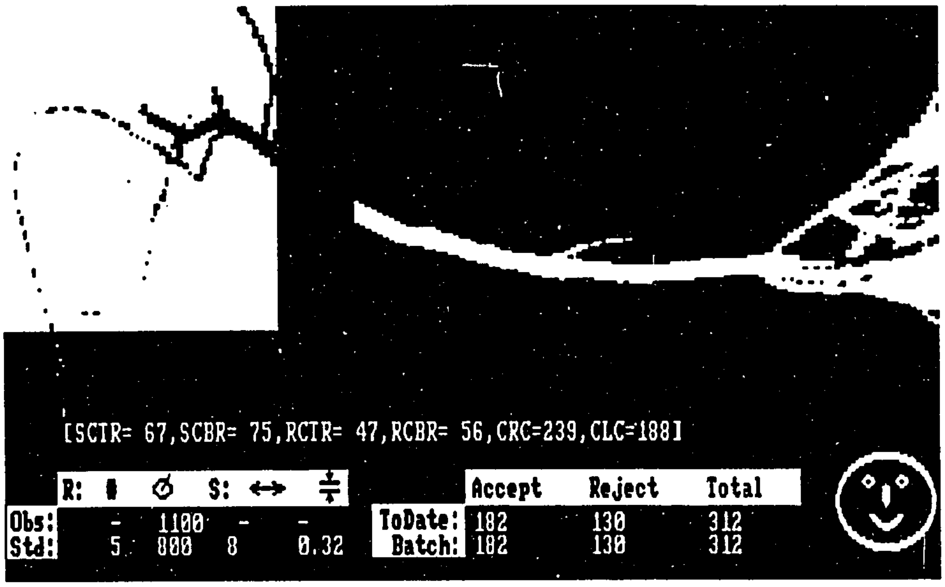
could be corrected by considering the geometry of the clamp, root edges, and their relative positioning, at the expense of increase in the processing time. At this point, however, we check for another constraint that the caliper diameter and the diameter at the stem end is between 4 and 40 vertical pixels (chosen arbitrarily). This is to account for an error if the clamp is entangled in a lot of roots, and the system can not properly discriminate the clamp. If this condition occurs, a message to that effect is generated.

Processing of Root Area

The processing described in the previous section not only provides us with the caliper diameter, but it also supplies us the limits for the root area window and the stem area window. The grading criteria which were used by the system in the case of some species suggests that a check for only the root area is sufficient, if it satisfies the specified threshold. If it does not satisfy the threshold then the image is subjected to more detailed processing. This detailed processing in the root area includes the isolation of the tap root and getting an estimate of the number of lateral roots connected to the tap root. The criteria for our experimentation were provided by the Ames Nursery experts.

It must be mentioned here that when we speak of the root area or the number of pixels representing the root area, we expect the seedlings to be mounted in such a fashion that most of their roots are exposed to the image capture system. This gives us a two dimensional representation of the three dimensional root volume. The root area threshold suggested for grading the seedlings takes this limitation into account and makes the required adjustments/compensations. Therefore, the root area is computed by just getting the pixel count in the root window. If the number of pixels are greater than the threshold of the grading criteria, and the sample belongs to the group of species which can be declared acceptable on the basis of high root mass alone, the processing stops at this point and the sample is declared as acceptable. The results of one such event are shown in Figure 4.6 where the pixel count is shown to be 1100 pixels.

On the other hand, if the root mass is below the acceptable threshold or if the seedling must also satisfy other parameters such as the number of lateral roots, caliper diameter, and the stem length, then the processing goes on. The first step in finding significant lateral roots is to identify and isolate the tap root. The hypothesis used for this purpose is that the tap root would have a cluster of long and contiguous



```
[SCTR= 67,SCBR= 75,RCTR= 47,RCBR= 56,CRC=239,CLC=188]

     R:  ‖   ⌀    S: ↔    ∓              Accept    Reject    Total
Obs:     -  1100  -       -     ToDate:  182       130       312
Std:     5  800   8       0.32   Batch:  182       130       312
```

Figure 4.6: Seedling grading based on the root mass only

segments. We examine the root area by processing the segments which are more than 16 pixels long. We also get region information for these segments. We expect the tap root to have such segments on several contiguous rows. A thin lateral root on the horizontal axis might have long segments, but, would not have more than a few contiguous rows (two in most cases) with these segments. Therefore, any regions with a thickness of less than two rows are ignored.

The rest of the segments are rank ordered according to their thickness, as the thickest region would have a greater likelihood of being a tap root. There is a possibility that the streaks of noise (reflection, sand clusters, etc.) might appear to the system as thick blobs or clusters of roots. This would make them viable candidates for the tap root. We can prevent this case by checking for each regions proximity and association with the root caliper as identified by the clamp area processing of the previous section. The processing time overhead involved in determining the connectivity of each potential tap root region with the caliper of the seedling would be quite significant. Therefore, we look for the regions which have the closest proximity with the caliper.

We examine each region from the list, to see if it satisfies the proximity constraint with the caliper at the clamp. The first region which is found closest to the caliper and has a thickness greater than two rows, is identified as the tap root. Even after having all of these considerations, if the roots are too dirty (if they have too much soil attached to them), or there are too many roots clustered together, they might form a broader region with the actual tap root, and fool the system. In the later case, in most of the instances, the seedling would be declared acceptable under root mass criterion alone. It is important to have reasonably clean seedlings with minimal soil clusters. Under certain circumstances, it might be possible to pre-treat the seedlings before they are used in an automated grading system. A good judgement in this respect can not be over-emphasized.

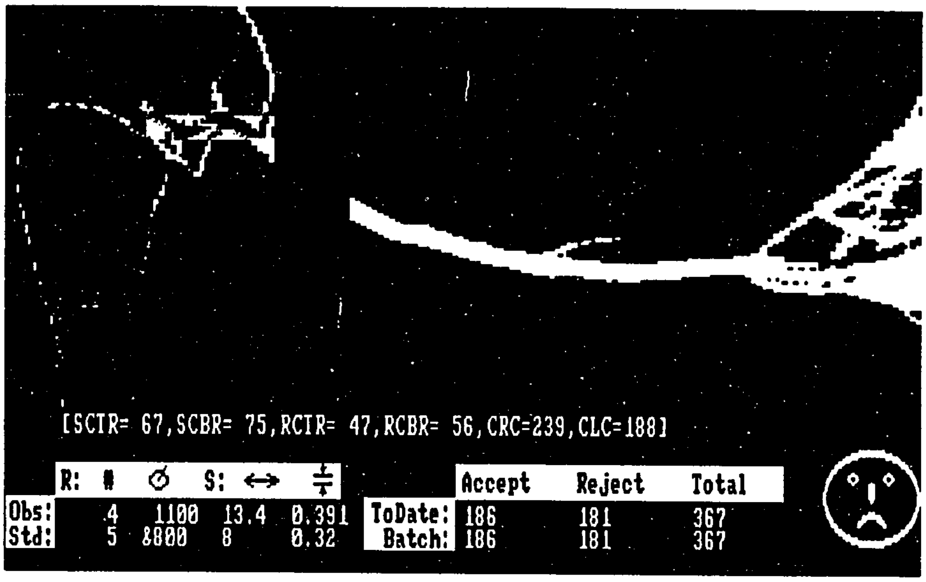
The processing steps described above only provide us with the row bounds of the tap root. Therefore, the next step is to find the right edge and the left edge of the tap root. We need this information to limit our count for the lateral roots in the tap root area, as there might be clusters of roots not attached to the tap root, and which might fall on the rows identified to have the tap root. The processing steps to achieve this goal are listed in the pseudo-code which follows.

```
Reset the search space for the tap root rows with solid segments
Start at the first tap root segment from the right on top row
```

```
of the tap root area
WHILE the present position has a bright pixel DO:
        Edge track from right to left and from top to bottom
        IF no segment at current position then EXIT
END.
Mark the right column position as POINT1
Mark the row position as POINT2
Start at the position of the last segment
DO:
    Edge track from left to right and top to bottom fashion
    IF no segment at current position then EXIT
    Store the right column position as POINT3
END.
Start at row POINT2 and column POINT1
WHILE the present position has a bright pixel DO:
        Edge track from right to left and from bottom to top
        IF no segment at current position then EXIT
END.
Mark the right column position as POINT4
Start at the first tap root segment from the right on bottom row
of the tap root area
WHILE the present position has a bright pixel DO:
        Edge track from right to left and from bottom to top
        IF no segment at current position then EXIT
        If Right column of the segment equals POINT1
            EXIT from the procedure
        END.
END.
Mark the right column position as POINT5
Set POINT6 equal to POINT5
Start at the position of the last segment
WHILE the present position has a bright pixel DO:
        Edge track from right to left and top to bottom fashion
        IF no segment at current position then EXIT
        If Right column of the segment equals POINT1
            Set POINT1 equal to POINT3
            EXIT from the procedure
        END.
END.
Mark left column of current segment as POINT6
DO:
    Edge track from left to right and from top to bottom
    IF no segment at current position then EXIT
    Mark left column of current segment as POINT6
END.
Mark the right column position as POINT4
Set Left Edge of the Tap Root Area equal to Min(POINT4,POINT5)
Set Right Edge of the Tap Root Area equal to Min(POINT3,POINT6)
```

We start by shrinking our tap root area to the rows which have no holes (dark pixels). This is done to prevent the edge tracking steps from locking on to a lateral root. We then edge track from the top row of the tap root and while maintaining connectivity, get the left most column. We do this by first edge tracking to the left and down, to get the left most limit from the top row in the downward direction. We then edge track to the right from the left edge of the segment which defines the left most column thus far. Finally, we edge track in bottom to top fashion, to find the left most edge in the upward direction. Minimum of the two left most points, defines the left edge as found from the top row of the tap root area. We do similar processing from the bottom row of the tap root, and get the left most column from the bottom. We, however, stop processing, if are on the same path which was traversed when tracking from the top row of the tap root. Finally, the minimum value of the two left most columns gives us the left edge of the tap root. An example of this processing is shown in Figure 4.7 where the tap root identified for a sample seedling and the tap root is shown as the shaded rectangular region.

The next step is to estimate a count of lateral roots which are attached to the tap root. We have to discriminate between the portions of the tap root attached to the tap root area as found by above steps, and the real lateral roots. We do this by processing each segment which is at the top and the bottom edge of the tap root area. We find a vertical region for each of these segments, and compute the average segment length based on the specific region. We assume that any segment wider than 6 pixels is not a lateral root, rather it is part of the tap root itself. Therefore, all the regions having an average pixel length less than or equal to 6 pixels are counted as



```
[SCTR= 67,SCBR= 75,RCIR= 47,RCBR= 56,CRC=239,CLC=188]

      R:  #    Ø     S: ↔    ÷              Accept    Reject    Total
Obs:      4   1100  13.4  0.391  ToDate:  186       181       367
Std:      5   8800  8     0.32    Batch:  186       181       367
```

Figure 4.7: Results of the tap root identification process

lateral roots and are compared to the lateral root criterion. The program flowchart for root area processing is presented in Appendix D.

Processing of Stem Area

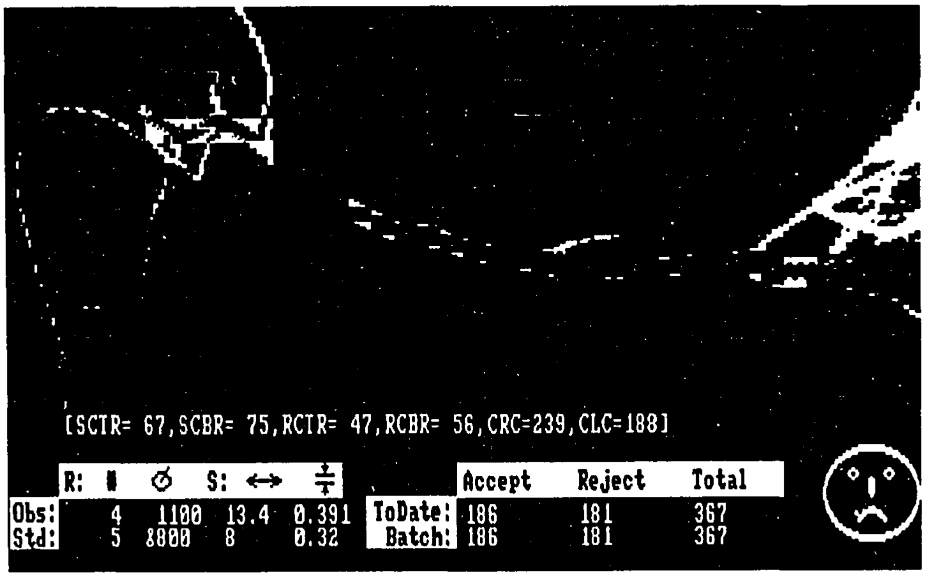
In the first section we described the computation of NeedleStart. The point where the needles for species like pine begin to develop. In the case of species which do not have needles. this point is the actual tip of the seedling. We use NeedleStart to divide the analysis of the stem area into two regions and to incorporate different set of criteria in the region with needles compared to the region without them. In the region before the needle area we are concerned with tracking the stem alone while bypassing any branches, or pixel streaks due to noise, which we might encounter. The starting point for analyzing this region, obviously, is the stem clamp point. We use Stem Clamp Top Row (SCTR) as our starting point. We do a panel-wise processing in this region. We accumulate the total number of pixels for the stem in each panel, and the horizontal and vertical components of the stem projection. We add these components for each panel and later compute the stem length from these values.

We use a panel width of 16 pixels and process segments which have a width of 9 pixels or more. This is required for filtering any isolated pixels due to environmental factors and to eliminate the registering of any branches which might have a close to vertical orientation. We get a blob information for each panel and only consider those regions which prove continuity with the region identified in the previous panel. The regions which are too close to each other and probably are separated by a single row or two, are collapsed together before they are examined for continuity. The region which is found continuous with the previous panel is used to update the total pixel count for the stem. The horizontal and the vertical components of the stem projection are computed by considering the slope of the vector which is formed by connecting the mid-point of the stem region in the present panel to the mid-point of the previous panel. These components are added to the overall horizontal and vertical vector used to compute the final stem length.

Once the region with the needles is reached, we compute the average number of pixels found in each panel which are associated with the stem. We change our threshold from 9-16 pixels to 15-16 pixels; this way we only consider regions which are almost solid. The assumption being that the stem and the needle would form a thick blob and isolated clusters of needles would have no bearing on the stem path. As before, we use the mid-point of each panel to find the projection of the stem. However, we use the average pixel count as computed above, to update the stem pixel count. After all panels to the stem tip are analyzed we compute the stem length from the resultant horizontal and vertical vectors. For the sake of completeness the total pixel count in the stem region is computed, and the average stem diameter is determined based on the total pixel count accumulated from each panel. The process is shown by the following pseudo-code:

```
Set segment search for 9-16 pixels
START in the panel at Stem Clamp Top Row and Clamp Right Column
WHILE not at panel with NeedleStart DO:
        Get total pixel count in the panel with region information
        Collapse any regions less than 3 rows apart
        Find the widest region connected to the previous panel
        IF region not found
            Apply conversion factors (pixel-to-length) to the resultant vectors
            Find stem length from horizontal and vertical components
            Update final stem pixel count
            EXIT
        ELSE
            Update horizontal component from a vector of panel mid points
            Update vertical component from a vector of panel mid points
            Update total pixel count from the region count
            Record Top and Bottom rows for next panel lookup
        END.
END.
```

The steps shown above are repeated in the region with the needles, with the exception that we do not collapse any regions, no matter how close they are. This is done in order to stick to the stem path, even if there are several blobs of needles quite close to each other. By not considering the adjacent blobs we limit the possibility of tracking a branch in the needle area as it might have a dominance and connectivity with the stem region. The results of processing the stem area are shown in Figure 4.8. The stem identified by the system, is shown by the shaded mask and the edge tracking path for the stem is shown by a series of dots. The processing steps described in this section are also presented in the form of program flowchart in Appendix E.



```
[SCTR= 67,SCBR= 75,RCTR= 47,RCBR= 56,CRC=239,CLC=188]

      R:  ▮    Ø    S:  ↔    ÷              Accept   Reject   Total
Obs:      4   1100  13.4  0.391   ToDate:  186      181      367
Std:      5   8800  8     0.32     Batch:  186      181      367
```

Figure 4.8: Results of the stem area identification process

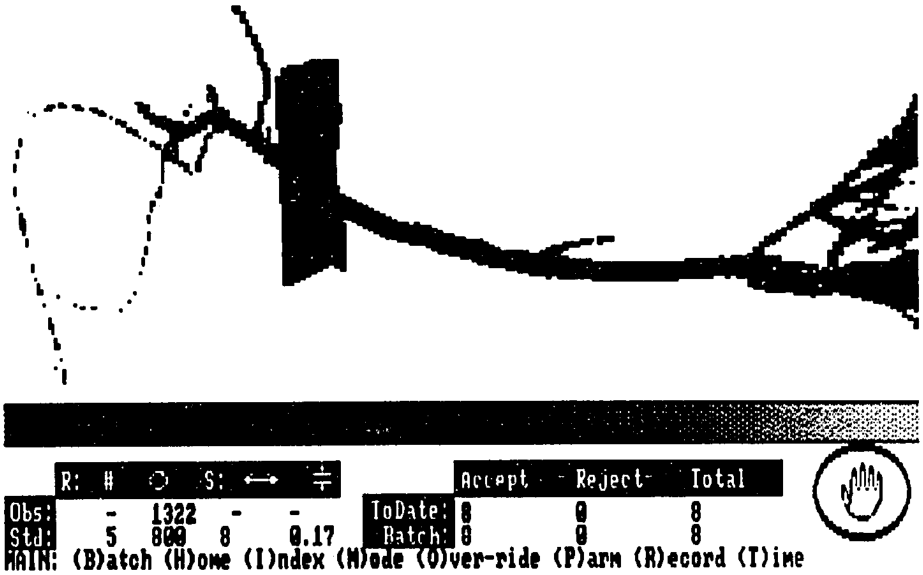
CHAPTER 5. FUNCTIONAL SPECIFICATIONS FOR THE GRADING SYSTEM

Screen Layout of the Seedling Segregation System

In this chapter we describe the seedling segregation program, its various menus, and the information which can be manipulated at run-time. We also describe the screen layout, and the structure of the information which is automatically stored by the system when specific request to record the system status is made, or when any of the system parameters are changed.

The main display of the segregation system is divided into six areas, as shown in Figure 5.1. The top most area which is 144 lines high and is full screen wide is the image display area. The image is continually captured by the image capture system. The segregation system then retrieves the image information from the image buffer and displays it.

The next area which has a width of a single line of text is the message area. All system feedback and informational messages are displayed on this line. In Figure 5.1, the message line displays the Stem Clamp Top Row (SCTR) and the Stem Clamp Bottom Row (SCBR) to be 67 and 76, respectively. The Root Clamp Top Row (RCTR) and the Root Clamp Bottom Row (RCBR) are 48 and 56, respectively, while the Clamp Right Column (CRC) and the Clamp Left Column (CLC) are 243,



```
     R:  #   ○   S:  ⟷   ÷              Accept   Reject   Total
Obs:     -  1322  -      -      ToDate: 8        0        8
Std:     5  800   8     0.17     Batch: 8        0        8
MAIN: (B)atch (H)ome (I)ndex (M)ode (O)ver-ride (P)arm (R)ecord (T)ime
```

Figure 5.1: Layout of the Seedling Segregation System display

and 192, respectively. These values are computed in pixel units. This message is generated when we are operating the system in trace mode for fine tuning and debugging the system. The next few lines of the display are divided into three sections. The left section is the grading criteria window. The desired system parameters for the current batch and the parameters returned by the system for the current image are displayed in this window. The figure shows a grading criteria of 5 lateral roots, 800 pixels in the root area, length of stem as eight inches, and the caliper diameter of 0.17 inches. The seedling processed by the system shows a root area pixel count of 1322. Hence the seedling is declared as good and the rest of the parameters are not processed.

The middle section is used to display inventory s^t^atus of the system at any given time. It displays the total number of seedlings processed by the system, the number of seedlings which were rejected, and the number of seedlings which are accepted. Similar values for the current batch are also shown in this window. The inventory status window in the figure shows that the system is started with the current batch and, thus far, 8 good seedlings have been processed. The right area at the bottom right corner of the screen is the nominal display area. This area is used to provide a nominal feedback about the image processing results. It has been designed to be symbolic in order to make it user friendly. This window can display one of four symbols at any time. When the system is interrupted to change the system parameters, then a wait symbol is shown in Figure 5.1. When the system can not identify the incident image to be a valid seedling image, a query symbol is displayed. When the seedling under process is declared as good, then a smiling face is shown. Lastly, when a seedling is declared as bad by the system, we show a frowning face. The complete set of symbols is shown in Figure 5.2.

The bottom row of the screen (see Figure 5.1) is used to show various menu options when the system is interrupted to update its parameters. The system can be interrupted at any time by pressing any key. At this point the main menu of the system is displayed at the bottom. The display of the system can be dynamically tailored to user needs. For example, the image in the image window can be turned on or off. Similarly, the grading criteria window and the inventory status window can be turned on or off. These display mode operations are mutually exclusive, and are selected from the *Mode* option of the main menu. Pressing 'M' at the main menu displays the *Mode* sub-menu.

The system can also be run in trace mode, in which case all display windows are

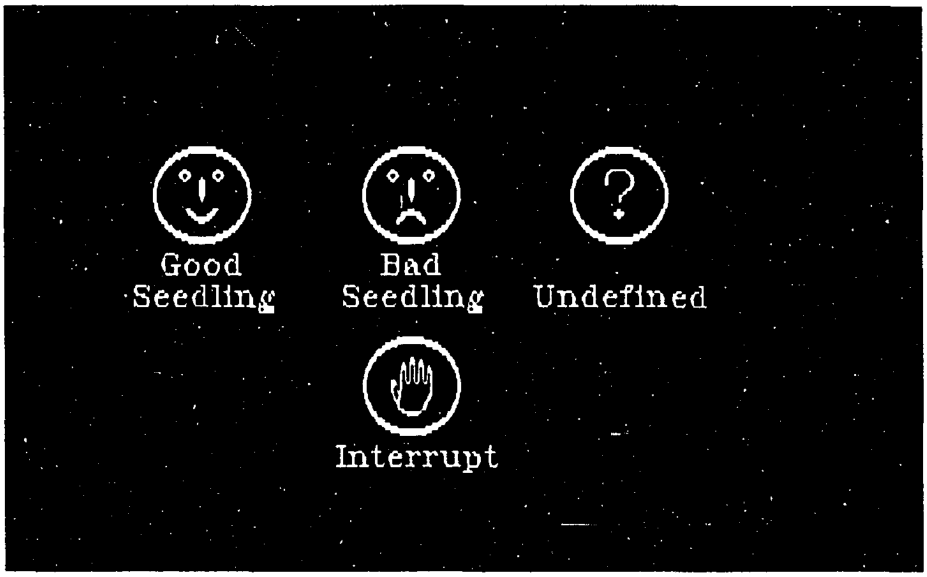


Figure 5.2: Symbols for the nominal output window

automatically turned on and the image area is appropriately highlighted at the end of each major image processing task. This mode, which is the *Trace* option in the *Mode* sub-menu, can be used to debug and fine-tune the system. It must be pointed out that each additional bit of information displayed on the screen adds to the total processing time. Therefore, the optimal performance of the system results when the system is run in the nominal output mode. The trace mode is only for debugging purposes and is usually not used in actual production.

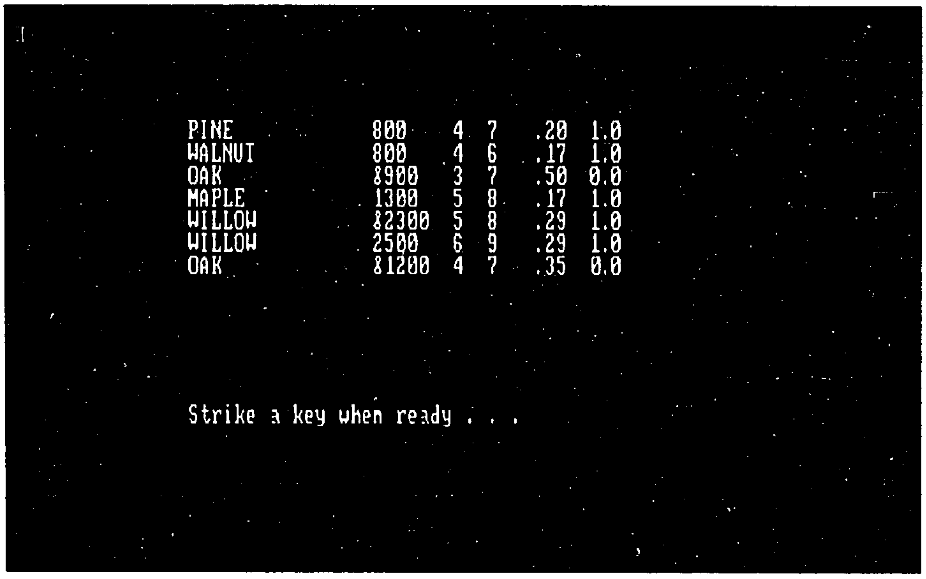
Changing Grading Parameters at Run-Time

The system can be interrupted to change the batch currently being processed by the system. This is the option *Batch* on the main menu, which is invoked by pressing 'B' on the keyboard. The user is asked to specify the batch name of the specie he wants to grade. The system retrieves grading information for that batch if it was previously defined to the system, and adjusts itself accordingly. If the specie requested by the user was used in a previous grading session then the inventory status from the last run for that batch, is also retrieved. The system is capable of remembering and restoring the status of its last run, when a new session is started. This information is stored in two separate files which are continually maintained by the system. The grading information is stored in the *BATCH.REC* file, while, the system and the inventory status information is stored in *SYSTEM.REC* file. The *SYSTEM.REC* file is duly dated and time-stamped at the recording of each event.

In the event when the grading parameters for the batch are not previously defined, the user is asked to enter these parameters. The information required by the system is the minimum acceptable number of lateral roots, the root area pixel count, minimum stem length, and the caliper diameter. The user is also asked to enter the correction length to be applied to the stem length due to the fact that the needles of seedlings like pine tend to protrude from the shoot by a full needle length. Therefore, the correction length generally is equal to the average needle length. For species with out needles, this length is zero.

Normally, a seedling is declared as acceptable if it passes on the basis of the root mass without processing it further for the stem length, caliper diameter, and the number of lateral roots attached to the tap root. However, the user can force the system to do a complete processing even if it passes on the basis of the root mass. This is done by entering an '&' symbol in front of the acceptable root mass specified by the user. The system also provides a quick way of doing this from the main way of doing this from the main menu, by toggling it through the *Over-ride* option (see Figure 5.1). The parameters supplied by the user are recorded in the batch parameter file and a record to the fact that a new batch has been defined, is recorded in the system status file. The grading criteria for our laboratory experimentation were provided by the nursery experts at the Iowa Conservation Commission Nursery in Ames.

The parameters of the current batch can be changed by the *Parm* option from the main menu. Similar information is sought for the existing batch: the batch record file and the system status file are appropriately updated. A sample of the batch record file is shown in Figure 5.3.



```
PINE           800    4  7   .20  1.0
WALNUT         800    4  6   .17  1.0
OAK            8900   3  7   .50  0.0
MAPLE          1300   5  8   .17  1.0
WILLOW         82300  5  8   .29  1.0
WILLOW         2500   6  9   .29  1.0
OAK            81200  4  7   .35  0.0



Strike a key when ready . . .
```

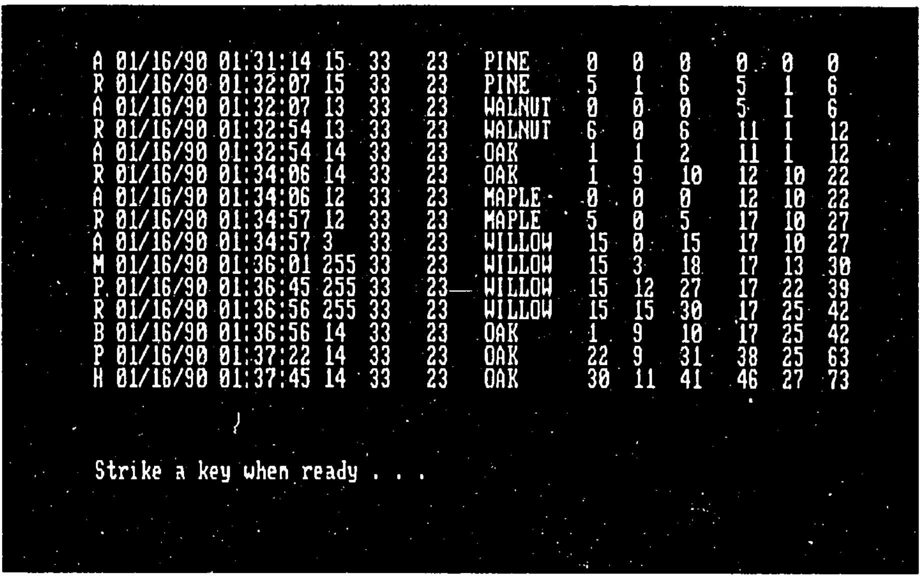
Figure 5.3: Example of a *BATCH.REC* file

Changing the System Parameters at Run Time

The system parameters are recorded in the file *SYSTEM.REC* as shown in Figure 5.4. Each record has 13 fields. The first field is a single character operation code, the date and time stamp for the record, the mode of operation for the batch, the horizontal and vertical transformation index (discussed in the next section), and the number of good, bad, and total seedlings for the batch and the overall total count.

The 1st field operation code is the single character from the main menu which invokes that specific operation. For example, the status of the system can be manually recorded at any time by selecting the *Record* option from the main menu by pressing the key 'R', as shown in the 2nd record of Figure 5.4. The mode of operation is recorded as an eight bit number (4th field in Figure 5.4), while the first 5 bits are significant. The bit 0 (the right most bit) when set, turns-on the nominal output window. The bit 1 is reserved for the comparison parameter window, while bit 3 turns on the inventory status window. The bit 2 enables/disables the image display as captured by the system. However, when the system is interrupted and the main menu displayed, the image window is automatically turned on. The bit 4 is used to turn on the trace mode. The user is offered to change the mode by the *Mode* sub-menu, which has toggle switches for each option.

When the batch to be processed by the system is changed, the system automatically retrieves the last record from the system status file and restores the status which was active for that batch. If the batch is used for the first time, the user is asked to enter the information as outlined above.



```
A 01/16/90 01:31:14 15  33   23   PINE     0   0   0    0   0   0
R 01/16/90 01:32:07 15  33   23   PINE     5   1   6    5   1   6
A 01/16/90 01:32:07 13  33   23   WALNUT   0   0   0    5   1   6
R 01/16/90 01:32:54 13  33   23   WALNUT   6   0   6    11  1   12
A 01/16/90 01:32:54 14  33   23   OAK      1   1   2    11  1   12
R 01/16/90 01:34:06 14  33   23   OAK      1   9   10   12  10  22
A 01/16/90 01:34:06 12  33   23   MAPLE    0   0   0    12  10  22
R 01/16/90 01:34:57 12  33   23   MAPLE    5   0   5    17  10  27
A 01/16/90 01:34:57 3   33   23   WILLOW   15  0   15   17  10  27
M 01/16/90 01:36:01 255 33   23   WILLOW   15  3   18   17  13  30
P 01/16/90 01:36:45 255 33   23   WILLOW   15  12  27   17  22  39
R 01/16/90 01:36:56 255 33   23   WILLOW   15  15  30   17  25  42
B 01/16/90 01:36:56 14  33   23   OAK      1   9   10   17  25  42
P 01/16/90 01:37:22 14  33   23   OAK      22  9   31   38  25  63
H 01/16/90 01:37:45 14  33   23   OAK      30  11  41   46  27  73


Strike a key when ready . . .
```

Figure 5.4: Example of a *SYSTEM.REC* file

Changing the Horizontal and Vertical Conversion Parameters

The stem length and the caliper diameter is measured by the system in terms of the number of pixels. However, these parameters supplied as the grading criteria are in length units (inches or centimeter, etc.). Therefore, it is important to provide an accurate conversion from the internal unit of measurement (pixels) to the external unit of measurement (length). This is done by providing a facility to the user to enter these conversion parameters at run time. It is important to have this feature available at run time, as the number of pixels which form the same image would significantly vary with the repositioning the camera, position of the external light source, and the general lighting conditions of the system. Hence, it is important to fine tune the system at the start of each grading session.

The conversion parameters for the system are entered by selecting the *Index* option from the main menu. The user then presents a two dimensional template of known dimensions to the system. This template is made from a thick sheet of white paper or a card. The system upon capturing the image of the template displays it in the image display area (Figure 5.5).

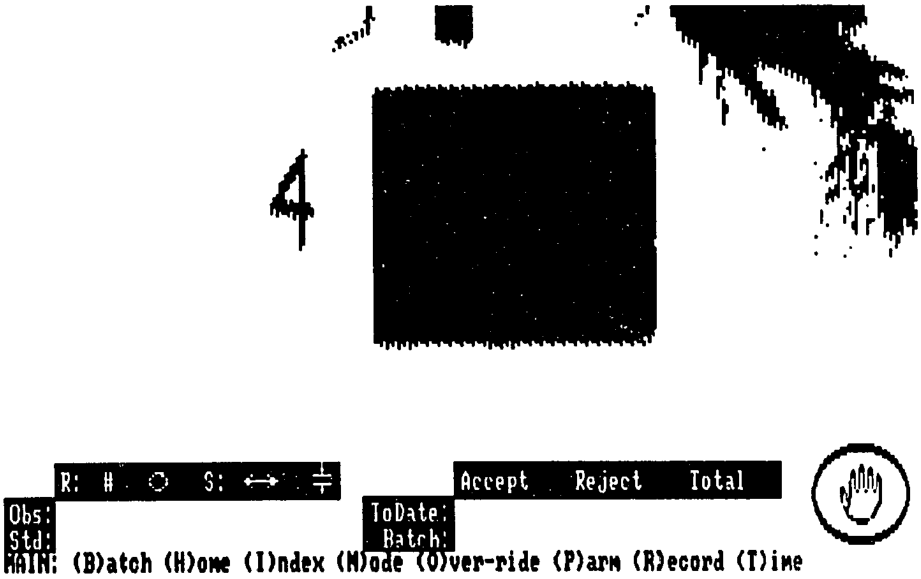
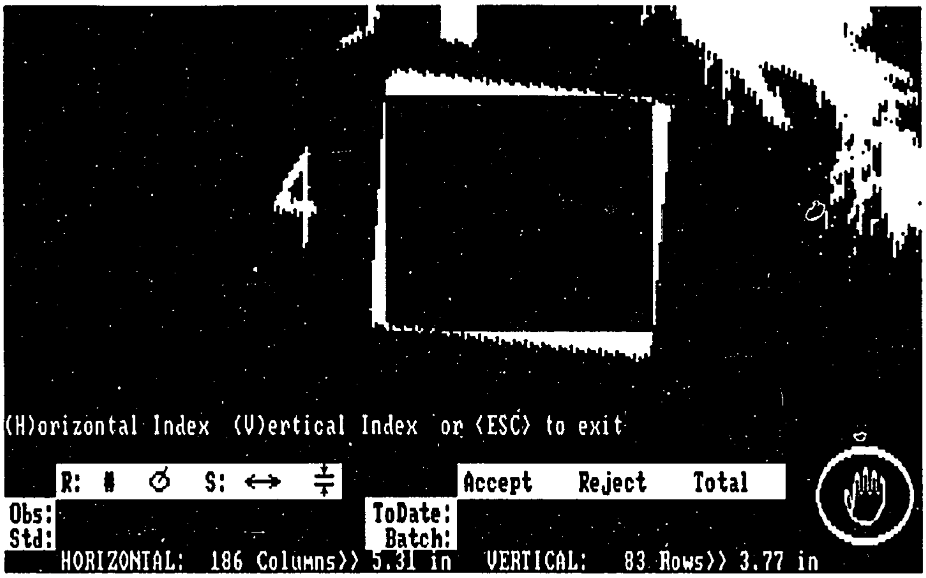

4
R: # ○ S: ↔ ÷
Accept Reject Total
Obs:
Std:
ToDate:
Batch:
MAIN: (B)atch (H)ome (I)ndex (M)ode (O)ver-ride (P)arm (R)ecord (T)ime

Figure 5.5: Display of template for setting the conversion parameters

The program then attempts to cover the template by an internal mask. The degree of masking is shown by a black cover on the card image. The system returns the horizontal and vertical dimensions of the card in units of pixels and inches using the conversion parameters based on the conversion factors in effect. The results are displayed and continuously updated at the bottom of the screen as shown in Figure 5.7.

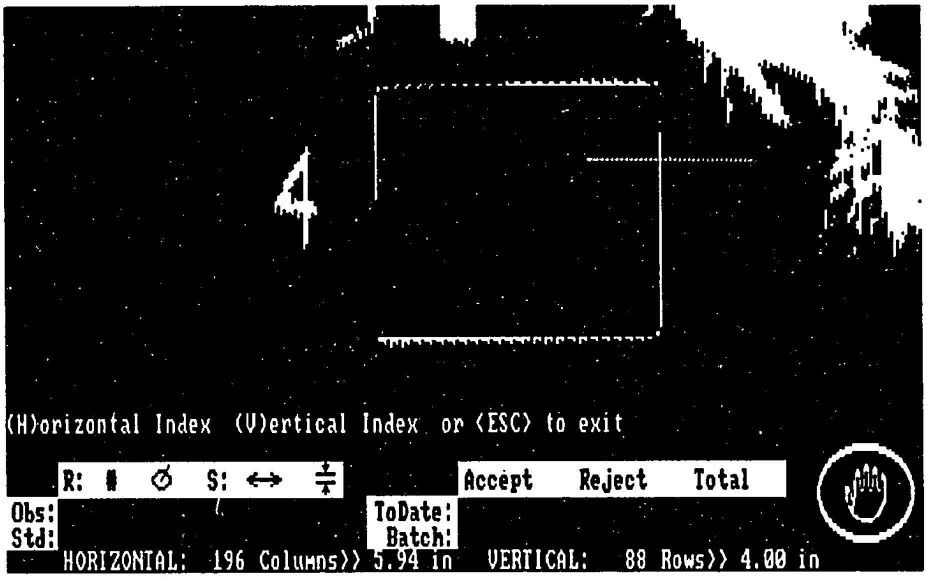


(H)orizontal Index (V)ertical Index or <ESC> to exit

HORIZONTAL: 186 Columns>> 5.31 in VERTICAL: 83 Rows>> 3.77 in

Figure 5.6: Example of a poorly placed template

The card template is rotated to make its axes parallel to the camera plane. Figure 5.6 shows a poorly positioned template which results in an imperfect mask. When the template is correctly positioned, the template is completely masked by the system (Figure 5.7), and the system returns the correct dimensions of the template. If not, the user enters the horizontal and/or vertical parameters which result in the correct dimensions of the template by the system. The system returns to normal operation when a <ESC> key is pressed. At this point a record for the new conversion parameters is written to the *SYSTEM.REC* file.



```
(H)orizontal Index  (V)ertical Index  or <ESC> to exit
R: ▮  Ø  S: ↔  ÷          Accept   Reject   Total
Obs:                  ToDate:
Std:                  Batch:
HORIZONTAL:  196 Columns>> 5.94 in   VERTICAL:   88 Rows>> 4.00 in
```

Figure 5.7: Example of a properly placed template

Examining the System Processing Time

The system processing time can be determined by selecting the option *Time* from the main menu. The system estimates the time it takes to capture the image and fill the image buffers and reports it as the image acquisition time. It also calculates the time elapsed in various image processing steps and reports them as the image processing time. The total time is also reported by the system which includes some of the overheads involved in displaying various information parameters on the screen. The processing time is minimum when the system is running in nominal output mode only. The timing information is significantly high when the system is running in trace mode, as the processing is held-up, while the system is highlighting various image processing steps on the screen.

The time measured by the system in time units, updated by the PC-BIOS (Basic Input/Output Service) routines. The time kept by the system for its house keeping and time reporting functions is updated 18.2 times a second. This number comes from the fact that the PC timer chip 8254 operates at an input clock rate of 1.19318 MHz. If we divide this number by the largest number 65535, which can be held in the 16 bit PC register, the clock rate of 18.2 seconds is returned. It is convenient to measure the time from the BIOS time stamp as it does not require any housekeeping. However, as a consequence, the minimum unit of time which is measured and returned by the timer routine is 1/18.2 (or 0.0549450549) seconds. This means that each major image processing task is measured in these units, and as a result, the processing time returned by the system is slightly higher than it actually is. This is a reporting problem and can be corrected by directly programming the timer chip before each timing event and restoring it on exit.

CHAPTER 6. SAMPLE SESSIONS AND RESULTS OF THE TEST RUNS

The details of the design methodology and the system design were presented in Chapters 4 and 5. The system was tested by using tree seedlings from the Iowa Conservation Commission Nursery. In this chapter we introduce the actual grading performance in the laboratory environment. The program was run on a Zenith 248 computer. Although, the computer was a 16 bit 80286 based machine, for compatibility reasons, the program was compiled for an eight bit, 8086 type machine. The Zenith computer had a Nortonś SI rating of 9 (i.e., 9 times faster than an IBM/XT) at a clock rate of 8 MHz. Re-compiling the code for higher word size processors (like 80386 and 80486) would improve the system performance even further. The newer generation PCs boast the clock rate all the way up to 33 MHz, which would result in a marked difference in the processing speed.

We subjected samples of White Pine and Red Oak seedlings to the system. Lighting conditions with fluorescent lamps in the lab, which were measured at about 2.5 footcandles (fc), proved to be inadequate, because of low contrast images, and low camera response time. To overcome this problem, the lighting was enhanced by using a slide projector light source at high intensity setting. This resulted in an illumination of about 8.36 footcandles. The lighting conditions at various states of illumination are presented in Table 6.1.

Table 6.1: Lighting characteristics of laboratory environment

Source	Intensity	Value (Lux)	Value (fc)
Natural	@1:00PM	2.4	.22
Natural	@5:30PM	1.8	.17
Fluorescent	Fixed	27	2.5
Slide Projector	Low	69	6.41
Slide Projector	High	90	8.36

It was found that light from the slide projector was not enough to fully capture the images, hence, the seedlings were sprayed with white paint. This improved the contrast of the images and made it possible to capture the entire seedling. The lighting conditions were affected by the hour of the day (direction of the Sun into the laboratory, cloud cover). Minor adjustments in the amount of light for image capture was possible by controlling the iris of the camera lens. The seedlings were mounted with a masking tape, serving as a clamp, on a black board (slate chalk board). The black board was used to provide an added contrast against the white seedlings. To satisfy the requirement of the system, the tape was mounted right at the start of the caliper of the seedling. The setup as configured for the tests, is shown in Figure 6.1.

As the first test, the system was presented with a pre-painted White Pine seedling. At the startup of the system, the user is asked to enter the range of the clamp width in pixels, as the width of the clamp in the image varies when the system setup is changed. Under the prevailing lighting conditions, a clamp segment width

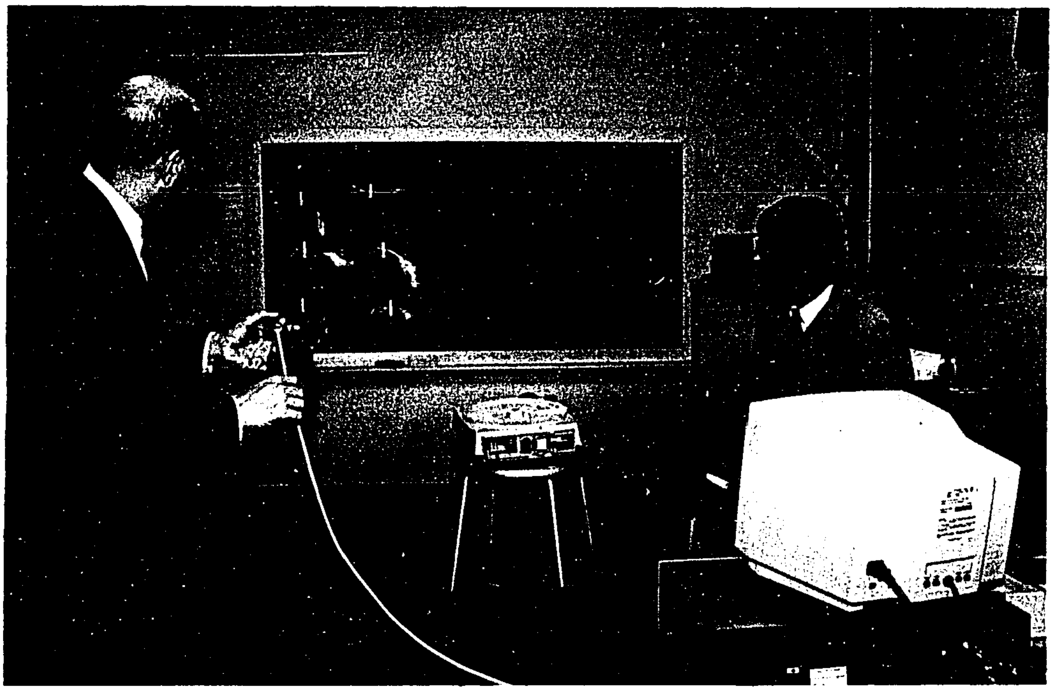


Figure 6.1: Equipment setup for seedling segregation system

between 30 and 36 pixels was required. Therefore, the command line at the program startup looked like:

```
> PROCESS 30 6
```

The first argument after the program name (PROCESS), is the base width for the clamp (30 pixels), and the second argument is the adjustment applied to the base (6 pixels). It should be noted that the system can operate on pre-stored images for initial setup and verification. In that case, the two arguments are shifted to the right with a new argument, as the filename of the pre-stored image, is added for the first argument. In order to process an image stored in file IMAGE_1, The command line looks like:

```
> PROCESS IMAGE_1 30 6
```

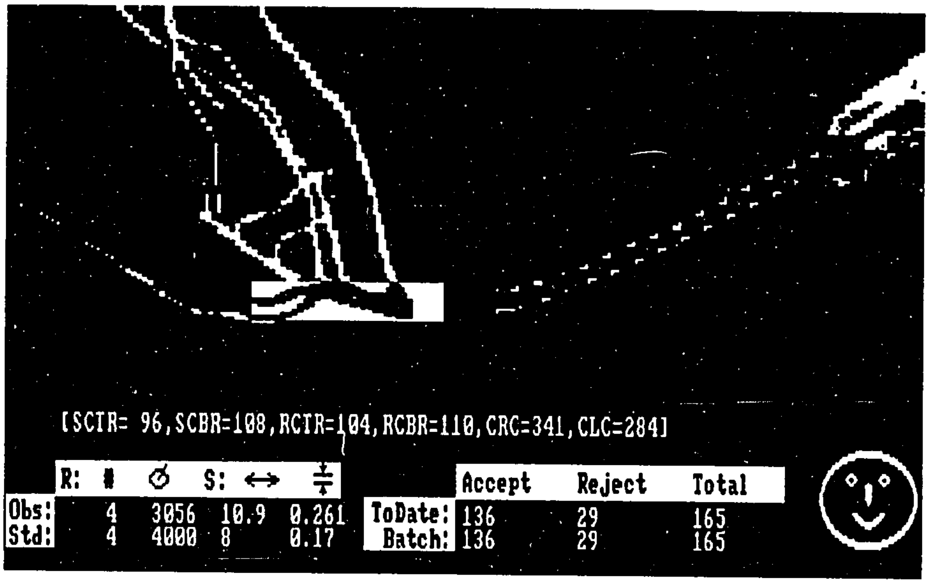
Once the above command is passed to the computer, the program title page shows up and after a few seconds the program displays the image in real time as it is captured by the camera system. The program initially operates in the pause mode, and the main menu is displayed, in case any parameters require adjustment.

Setting Conversion Parameters

Once the camera position, lighting conditions, and the camera iris are set to the desired limits, it is important to calibrate the system by choosing the *Index* option from the main menu. The process of setting the conversion parameters is outlined in the fourth section of Chapter 5. The system is presented with a rectangular template of known dimensions, and the system attempts to mask it by sensing its dimensions. Once the conversion parameters are set, they should be valid for any length of time, as long as the relative positioning of the camera with the seedlings, and lighting conditions are kept constant. The light level does vary under normal operating conditions and the parameters must be monitored regularly. The conversion parameters should be adjusted if any significant changes are noticed. Results of processing a pre-stored image are shown in Figure 6.2.

Defining a New Batch

When the system is run for the first time, the grading criteria for the current batch are undefined. Therefore, the system automatically enquires about the batch parameters. The default values are shown in square brackets which are used by the system, if the user responds by just pressing a return key. The dialogue for a sample inquiry looks like:



```
[SCTR= 96,SCBR=108,RCTR=104,RCBR=110,CRC=341,CLC=284]

      R:  #   Ø     S: ↔    ÷               Accept   Reject   Total
Obs:      4   3056  10.9    0.261   ToDate: 136      29       165
Std:      4   4000  8       0.17     Batch: 136      29       165
```

Figure 6.2: Results of processing a pre-stored image

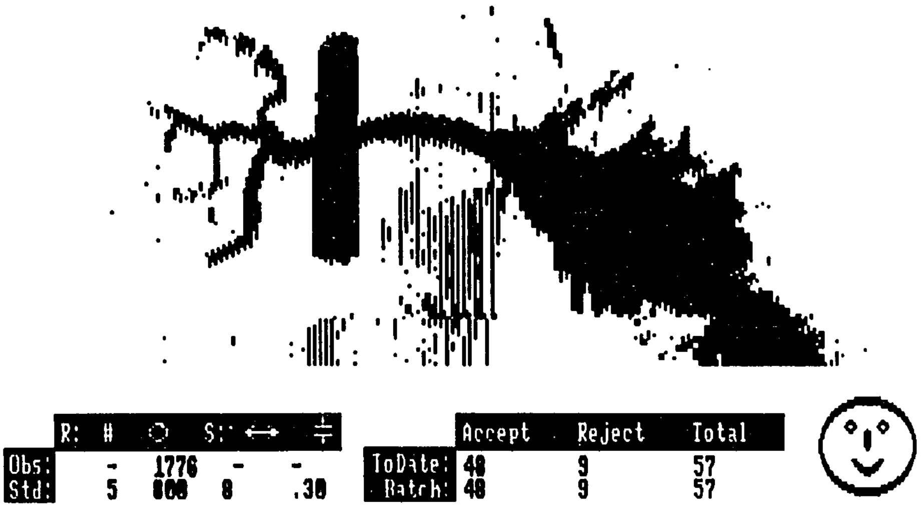
```
Specie Name for Current Batch [PINE]: <RETURN>
Minimum Acceptable Root Mass for PINE (Pix) [800]: <RETURN>
Minimum Number of Lateral Roots for PINE [5]: <RETURN>
Minimum Stem Length (in) for PINE [8]: <RETURN>
Acceptable Caliper Diameter (in) for PINE [.3]: .35 <RETURN>
End Adjustment for Stem Length (in) for PINE [1.0]: <RETURN>
```

The program first asks for the name of the current batch (which in general is the specie name in use). An alpha numeric string of up to 15 characters is supported for the batch name. In the above example, the user chose to use the default batch name PINE (by pressing return key). The user is then asked to enter the minimum acceptable root mass (represented in pixels) for the batch. The system computes the number of pixels in the root area and if the number surpasses the above threshold, the seedling is declared as acceptable, without any further processing. The system can be forced to conduct a complete processing by placing an '&' symbol in front of the acceptable root mass value in response to the above query. This is desired for some species where the root mass criteria alone, is not enough. In the above example, the default pixel count of 800 was chosen. In the next three queries, the user is asked to enter the minimum number of good seedlings, the length of the stem, and the caliper diameter for the acceptable seedling of the declared specie. The last query is for the species which require adjustment in the stem length as their needles protrude from the stem, and the system treats them as part of the stem. Generally, this length is found equal to the average needle length. Evergreen species like Pine fall into this category. However, the deciduous species like Oak where there are no needles, the adjustment value, in contrast to Pine, is zero.

The system also enquires about the inventory status of the batch and the total to-date processing inventory status. The quantities for good, bad, and total seedlings are sought as starting values. The system also enquires about the code for the mode of operation. The details for the formulation of the code are described in the third section of Chapter 5. A code of 255 can be used to run the system in trace mode. Once all the values are specified (detailed user input is shown in Appendix A), the batch record file and the system record file are created and the appropriate information is recorded. The parameters for the current batch can be updated at any time by interrupting the system (by pressing any key), and choosing the *Parm* option. Once a specie is defined to the system, the grading information is automatically restored by the system when the batch is re-introduced at a later stage.

Processing a Pine Seedling with Noise

Our first sample run deals with grading a pine seedling. The base value of 30 and adjustment of 6 was specified for the clamp. The system was running at a conversion factor of 33 and 23 pixels to an inch for horizontal and vertical conversion, respectively. The input image for the seedling is shown in Figure 6.3.



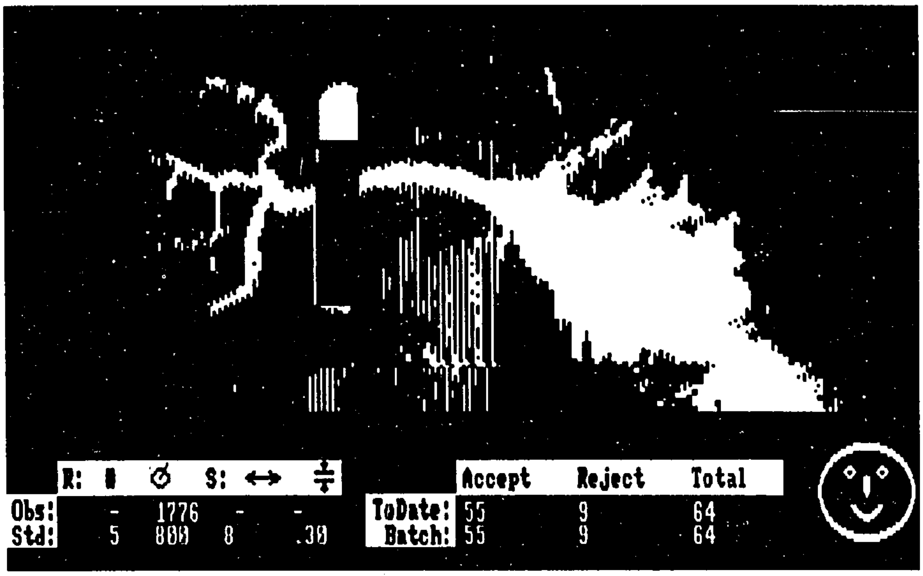
	R: #	○	S:· ↔	÷
Obs:	-	1776	-	-
Std:	5	800	8	.30

	Accept	Reject	Total
ToDate:	48	9	57
Batch:	48	9	57

Figure 6.3: Input image for the Pine sample

The grading criteria information and the results for the sample in question, are displayed in the grading parameter window at the bottom left corner of the screen. Our criterion was to force complete processing, even if the seedling passes the root area pixel count criterion of 800 pixels. We do this by using the '&' qualifier in front of the pixel count specification (as shown in Figure 6.3). The minimum number of lateral roots which are attached to the tap root was 5. The minimum acceptable stem length was 8 inches, and the minimum acceptable caliper diameter was .30 inches. In order to highlight the grading process, the system in our example was run in trace mode.

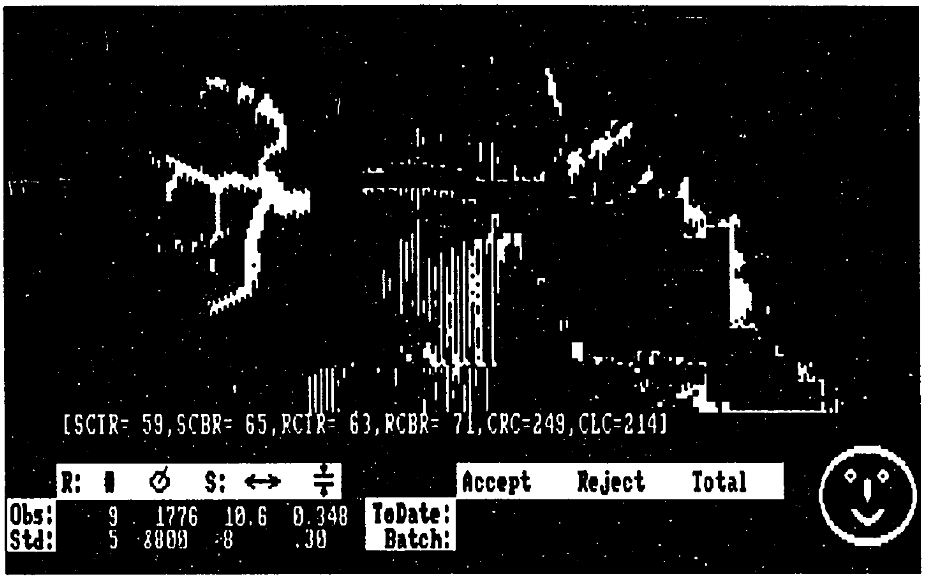
The image was captured at an illumination value of 8.36 footcandles (fc), in order to capture the thin lateral roots. This had an adverse effect of noise induction due to background reflection. This noise is specially evident in the figure, as vertical streaking in the stem area. The grading system has been designed to successfully deal with such noisy images. This can be confirmed in Figure 6.4, where the system successfully identified the clamp.



```
      R:  ▮   Ø   S:  ↔    ÷              Accept    Reject    Total
Obs:      -  1776      -    -    ToDate:  55        9         64
Std:      5   800  8      .30     Batch:  55        9         64
```

Figure 6.4: Successful clamp identification process

It can be seen in Figure 6.4 that only the lower half of the clamp satisfied the range of clamp thickness, as specified at the command line. This was because the



```
[SCIR= 59,SCBR= 65,RCIR= 63,RCBR= 71,CRC=249,CLC=214]

     R:  #    ∅     S: ↔    ÷              Accept    Reject    Total
Obs:     9   1776   10.6  0.348  ToDate:
Std:     5   8000    8     .30    Batch:
```

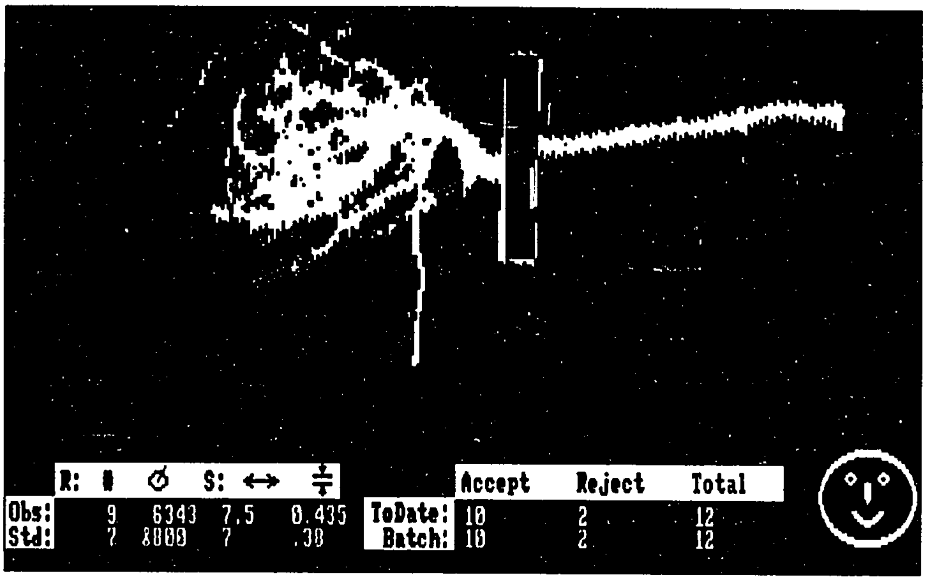
Figure 6.5: Results of complete processing of Pine sample

segments in the upper half did not satisfy the constraint for the segment length, as specified at the command input. However, the clamp was successfully identified and removed, as evident in Figure 6.5, which shows the results of the final processing. The system reported a root area pixel count of 1776 pixels, while the length of the stem was reported to be 10.6 inches. The actual physical length of the seedling was measured to be 10.7 inches. The path of the edge tracking process in the stem area is shown by the dots in the stem area. It should be noted that an adjustment of 1.0 inch was specified for the needles protruding at the tip of the stem. Hence the result of 10.6 inches was reported after subtracting the adjustment value. The caliper diameter measured by the system was .348 inches. The physical measurement of the caliper diameter reported .340 inches. The total processing reported by the system in trace mode was .91 seconds. When the system was run in nominal mode, the processing time reduced to 0.65 seconds. The processing time was fetched by selecting the *Time* option from the main menu in the trace mode, and the nominal mode, respectively.

Processing of an Oak Seedling

The next sample processed by the system was a Red Oak seedling. The seedling, as in the previous example with the Pine seedling, was sprayed in white for adequate contrast. A base of 22 and an adjustment of 12 pixels was used at the command line for the range of clamp width. As before, 8.36 footcandle of lighting was used. The system was running at a conversion factor of 33 and 23 pixels to an inch, respectively. The distinct feature about the Oak seedling is that it doesn't have needles, and its leaves are all gone when it is required to be graded in fall. This means that the adjustment in the stem area is zero. The sample seedling had a high cluster of roots, and would have passed on the root mass criterion. Complete processing was requested for illustrative purposes. The input image for the seedling with the clamp area identification is shown in Figure 6.6, while the results of complete processing are shown in Figure 6.7.

The stem area edge tracking resulted in a stem length of 7.5 inches. The physical length of the stem area was measured at 7.35 inches. The pixel count in the root area was reported at 6343 pixels. Slight variations in pixel count were experienced in the image because of the nature of the image capture system, but the change was less than 2% of the total root area pixel count. However, such variations in the image capture information can be anticipated with the natural products like tree seedlings.



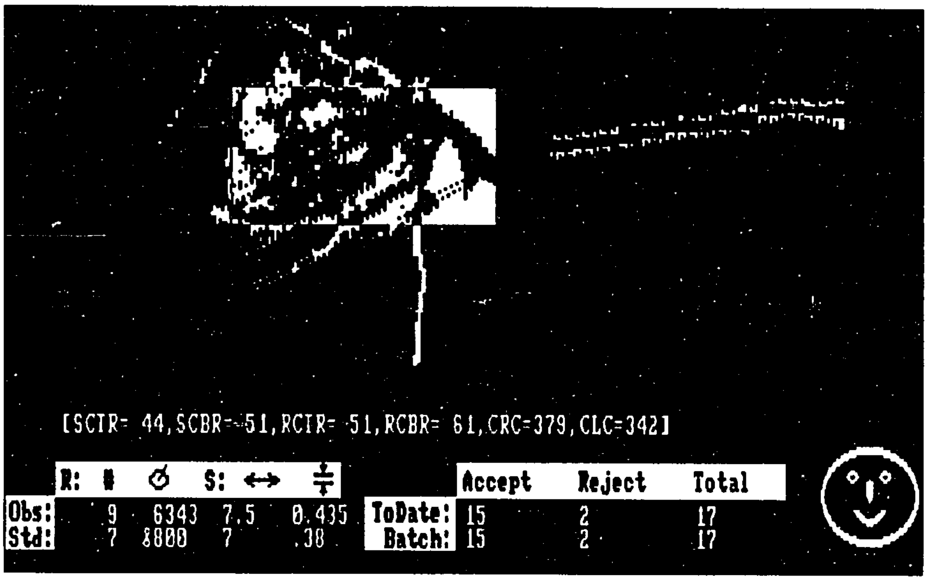
```
     R:  #   Ø    S: ↔    ∓         Accept   Reject   Total
Obs:     9  6343  7.5  0.435  ToDate: 10       2        12
Std:     7  8000  7     .38    Batch: 10       2        12
```

Figure 6.6: Results of clamp identification for the Oak sample

The caliper diameter of .435 inches was reported by the system, which satisfied the threshold of .38 inches.

As expected, the grading system included the dense root clusters as a part of the tap root. This is the problem of grading low contrast seedlings with a two dimensional image processing system. Even a system using better image processing equipment can not guarantee proper identification and isolation of the tap root. In such cases, however, the high clustering of the roots would result in an acceptable seedling on the basis of the root mass alone, and hence the dilemma of identifying the tap root would not be faced. The lateral root count of 9, as reported by the system, was irrelevant as the tap root was not identified properly.

The single line message area on top of the batch record and inventory record



```
[SCIR= 44,SCBR=-51,RCIR= 51,RCBR= 61,CRC=379,CLC=342]
      R:  #   ⌀     S: ↔    ∓              Accept    Reject    Total
Obs:      9   6343  7.5     0.435  ToDate: 15        2         17
Std:      7   8000  7       .38     Batch: 15        2         17
```

Figure 6.7: Results of complete processing of Oak sample

window displays the position of the stem at the clamping point. the caliper diameter. and the position of the clamp. The stem clamp top row. and the stem clamp bottom row of 44 and 51. respectively. are reported. The root clamp top row. and the root clamp bottom row was at 51 and 61. respectively. The clamp was identified to be between columns 379 and 342. The processing was completed in .75 seconds in trace mode. while the optimal processing in nominal mode resulted in a processing time of .32 seconds. The processing times for various samples presented in this manuscript are summarized in Table 6.2.

Table 6.2: Processing time required for various seedling samples

Type	Status	Mode	Processing Time (Sec)	Total Time (Sec)	Reference Figure(s)
Pine	Stored	Nominal	0.32	0.32	4.5,4.8
Pine	Stored	Nominal	0.38	0.38	6.2
Pine	Live	Nominal	0.65	0.75	6.3,6.5
Oak	Live	Nominal	0.32	0.43	6.6,6.7
Pine	Stored	Trace	0.75	0.86	4.8
Pine	Stored	Trace	0.65	0.75	6.2
Pine	Live	Trace	0.91	1.13	6.5
Oak	Live	Trace	0.75	0.96	6.7

CHAPTER 7. SUMMARY AND CONCLUSIONS

The effort reported in this manuscript was a result of a three year rigorous research, design, and development effort. The basic approach for dealing with two dimensional images was investigated, and the local feature method with line segment tracking features was selected as the main image processing methodology. A comprehensive image processing routine was developed for a PC based system. The main image processing routines were coded in machine language to achieve maximum processing time efficiency.

The grading system designed in this research effort was kept flexible to support increased resolution, which might result because of improved image capture system. It is capable to support high resolution video display formats, such as EGA, and VGA, in the PC environment.

The grading criteria for processing nursery tree seedlings were investigated, and a lack of unified approach in this regard was reported. Therefore, it was decided to design a grading system providing a flexibility to change the grading criteria at run time. The system was designed to support multiple batch processing, and dynamic adjustments in the grading criteria as the batches were switched. The system also supported inventory status recording.

Rule-base expert system approach was used to identify and isolate a complete seedling from the scene when it also included several partial seedlings. If-Then rules were formulated to successfully isolate the clamping point from the needles of the evergreen type species, where the clusters of the needles posed an identification problem. In all cases, the edge tracking process in the stem area was designed to bypass any branches and shoots from the specimen and lock on to the stem in order to compute its length. Special software filters were used in order to avoid processing any stray pixels in the image. The largest root region and the proximity to the caliper point were used as the criteria for identifying the tap root.

Support for running the system on pre-stored images was implemented to facilitate initial setup and tuning of the system. Trace mode operation with a closed loop feed back capability is provided for the same reasons, which applies to both the pre-stored and the live images. A nominal output mode (Good, Bad, and Undefined), was provided to minimize process time overhead, and to provide the possibility for hardware based automatic process control.

The system was shown to deal well with the noisy image capture environment. The clamp identification process worked in most of the cases. However, adjustments in the range on the clamp width were required, as the lighting conditions or the relative positioning of the image capture elements (distance between the image, the camera, and the light source) were changed. A run time capability to change the horizontal and vertical pixel-to-length conversion factors was provided.

The stem area edge tracking worked very well as the system was designed to discriminate between the stem and the branches. Few problems in tracking the stem area were evident when the seedlings with needles (Pine) were processed, and the path of the stem was well entrenched in the maze of the needles.

The pixel count in the root area was processed with minimal problems. The main problem with the system was in identifying the tap root, and subsequently, in counting the lateral roots attached to the tap root. The counting process for the lateral roots was significantly affected by the saw tooth edges in the captured images due to the OpticRam based camera. Clusters of roots also made this identification process quite difficult. However, this problem was resolved in the worst cases, as the grading criterion resulted in an acceptable seedling, on the basis of the root mass alone.

One of the primary objectives of this research was to achieve a throughput of one seedling graded per second. The system performed quite efficiently and a seedling was processed in less than one second. The processing time was comparable to the microcomputer based system designed to process regular parts, where comparatively less information is subjected to processing. Gains in efficiency were possible due to highly optimized machine code and line segment search and masking techniques.

CHAPTER 8. RECOMMENDATIONS FOR FUTURE RESEARCH

Further research in automated grading of nursery tree seedlings can be approached in the following four areas.

- Using a better image capture system and improving image capture environment
- Providing support for more rigorous processing in the root area
- Overall integration with the material handling system, and the system for final disposition of the seedling
- Actual implementation at a nursery site and operation in actual production environment
- Sensitivity analysis and comparison study with the existing manual system

The quality of the images can be improved by using better camera systems based on the Charge Transfer Device (CTD) and a better camera control unit (frame grabber board). Such a system can be operated under low lighting levels and would have better discrimination characteristics to capture the low contrast seedlings. Such systems would result in a higher resolution for the input images, which would improve the overall accuracy of the system.

A further enhancement in the discrimination characteristics of the system would be. to have a support for grey scale thresholding. This. however. would increase the overall processing time by several folds.

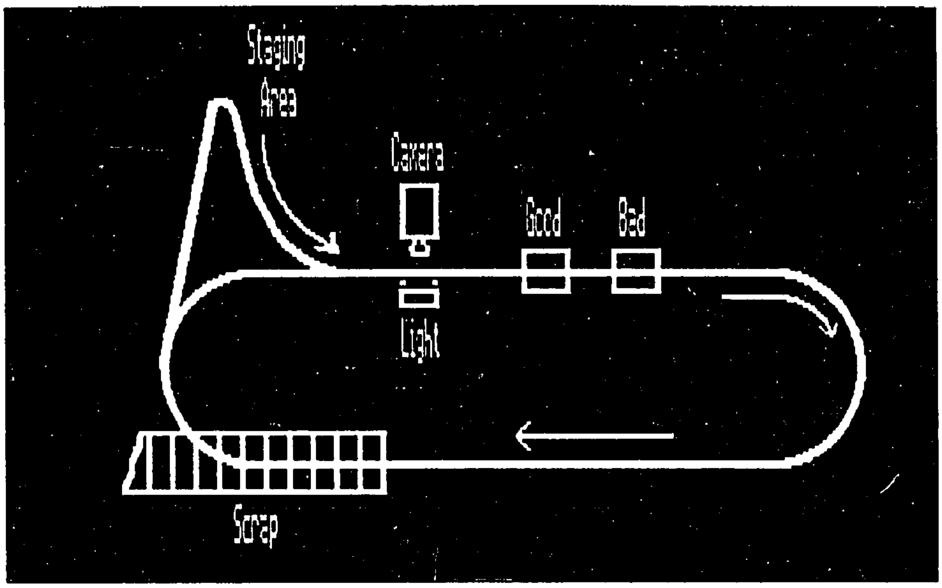


Figure 8.1: Material handling system for the seedling grading system

The grading system proposed in this research was tested by mounting the seedlings on a black board, and using a masking tape to emulate a clamp. Possibility of using strobe lighting can be studied, as that would allow for stop-action type of processing of the seedlings moving on a conveyor belt. A cable conveyor based handling system was proposed in [72], which is shown in Figure 8.1.

Further research can be conducted in integrating the image processing system with the staging and material handling system and with the product information system. The system has been designed to facilitate hardware interfacing to providing automatic ejection in the accept and reject bins by using electronic signals to the respective control devices. The software flags for nominal output can be used for this purpose. Once a working prototype in a laboratory environment is perfected, an actual on-site prototype can be developed and deployed at a nursery.

Finally, sensitivity analysis for such a grading system can be performed, and comparisons can be made with the manual grading processes.

CHAPTER 9. ACKNOWLEDGEMENTS

I am devoutly thankful to God who gave me the ability, opportunity, and tenacity, to pursue a doctoral degree.

I gratefully acknowledge the devotion, encouragement, and guidance, of my major professor Dr. John C. Even Jr., throughout my doctoral program. His valuable suggestions most often resulted in new approaches and ideas in the problem solving and design. I also thank him for the editing of this manuscript.

I am deeply thankful to Drs. William H. Brockman, Arthur E. Oldehoeft, Patrick E. Patterson, and Keith L. McRoberts, for their interest, and advice in this research effort, and for serving on my graduate committee. Keith's contributions in the inception of this research idea and his continued support in the research effort are specially acknowledged. I would also like to thank Professor Victor M. Tamashunas for his invaluable suggestions and support throughout my educational program. He was always available for help and guidance whenever I was in need.

I am very grateful to the officials of the Iowa Conservation Commission Nursery in Ames for providing the samples of tree seedlings as and when they were needed. I very much appreciate the able guidance and valuable suggestions of Mr. Jerald Grebasch; the nursery forester, and Mr. Roger A. Jacob; the assistant nursery forester, in the development of the basic grading criteria and an insight into the present grading

operations.

I am utterly grateful to Drs. Rex A. Thomas and John Pete Boysen of the ISU computation center for their continued confidence, support, encouragement, counseling, and for providing me financial support, throughout my graduate studies. I owe them much more then I can express in words.

My gratitude and appreciation goes to my wife and my two children for their unending patience, support, and understanding, during those long nights when I was busy working in the lab or at the office.

I dedicate this work to my parents and my uncle Colonel Mehdi A. Zaidi, who instilled in me the desire and the initiative to pursue higher studies, and for the love, prayers, and the continued confidence they have vested in me. I also dedicate this work to my sisterly friend Dr. Narimah M. Kairuddin for her love and support throughout my graduate studies.

APPENDIX A. PROMPTS FROM THE USER INTERFACE

To Change a Batch Name

```
Specie Name for Current Batch [PINE]:
```

To Introduce a Batch

```
Specie Name for Current Batch [PINE]:

Minimum Acceptable Root Mass for <*> (Pix) [800]:

Minimum Number of Lateral Roots for <*> [5]:

Minimum Stem Length (in) for <*> [8]:

Acceptable Caliper Diameter (in) for <*> [.3]: .35

End Adjustment for Stem Length (in) for <*> [1.0]:
```

* Batch name as used at the first prompt

To Change Batch Parameters

```
Minimum Acceptable Root Mass for <*> (Pix) [<**>]:

Minimum Number of Lateral Roots for <*> [<**>]:

Minimum Stem Length (in) for <*> [<**>]:

Acceptable Caliper Diameter (in) for <*> [<**>]: .35

End Adjustment for Stem Length (in) for <*> [<**>]:
```

* Last batch name in effect

** Last value in effect

To Change Conversion Parameters

```
X-Axis Transformation Parameter (pix) [<**>]:

Y-Axis Transformation Parameter (pix) [<**>]:
```

** Last value in effect

To Change System Parameters

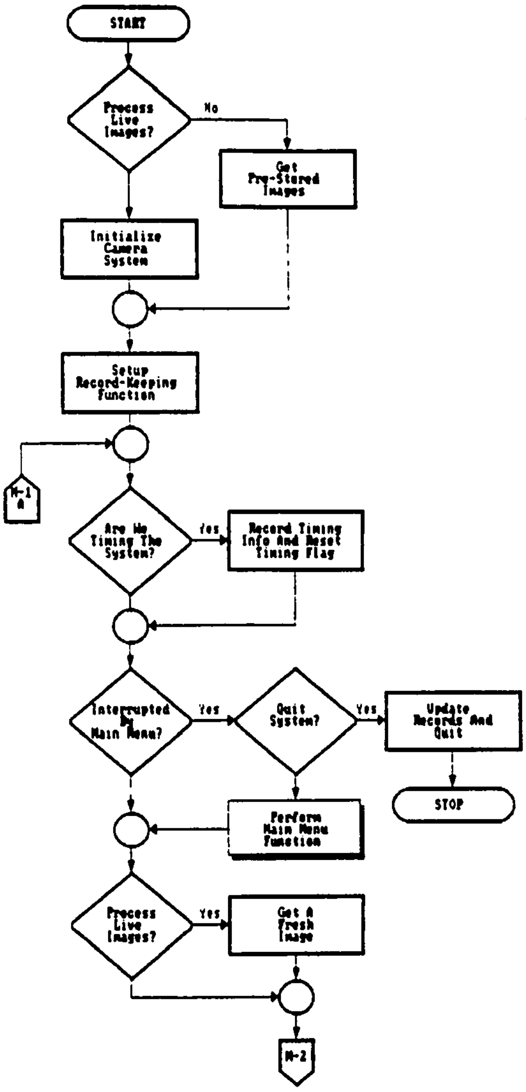
```
System Feedback Code for <*> [<**>]:

X-Axis Transformation Parameter (pix) [<**>]:

Y-Axis Transformation Parameter (pix) [<**>]:

Number of Good Seedlings for <*> [<**>]:

Number of Bad Seedlings for <*> [<**>]:

Number of Total Seedlings for <*> [<**>]:

Total Number of Good Seedlings [<**>]:

Total Number of Bad Seedlings [<**>]:

Grand Total Number of Seedlings [<**>]:
```

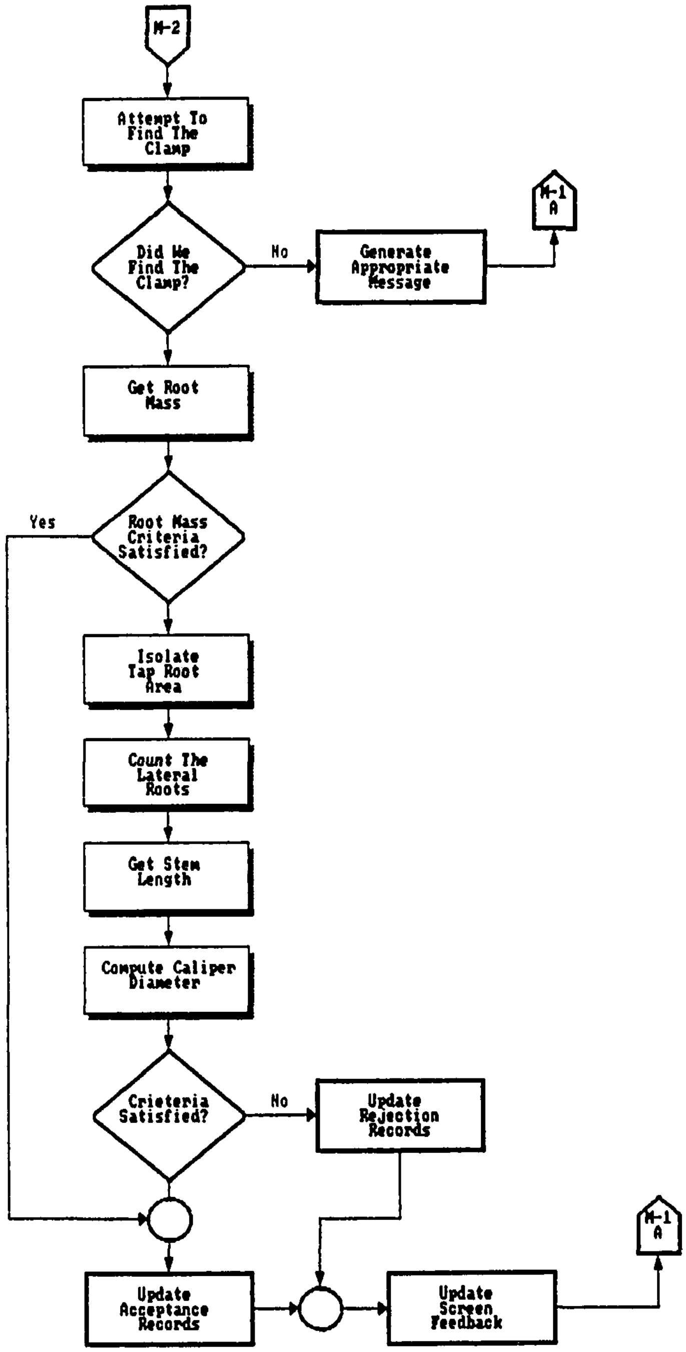
* Last batch name in effect

** Last value in effect

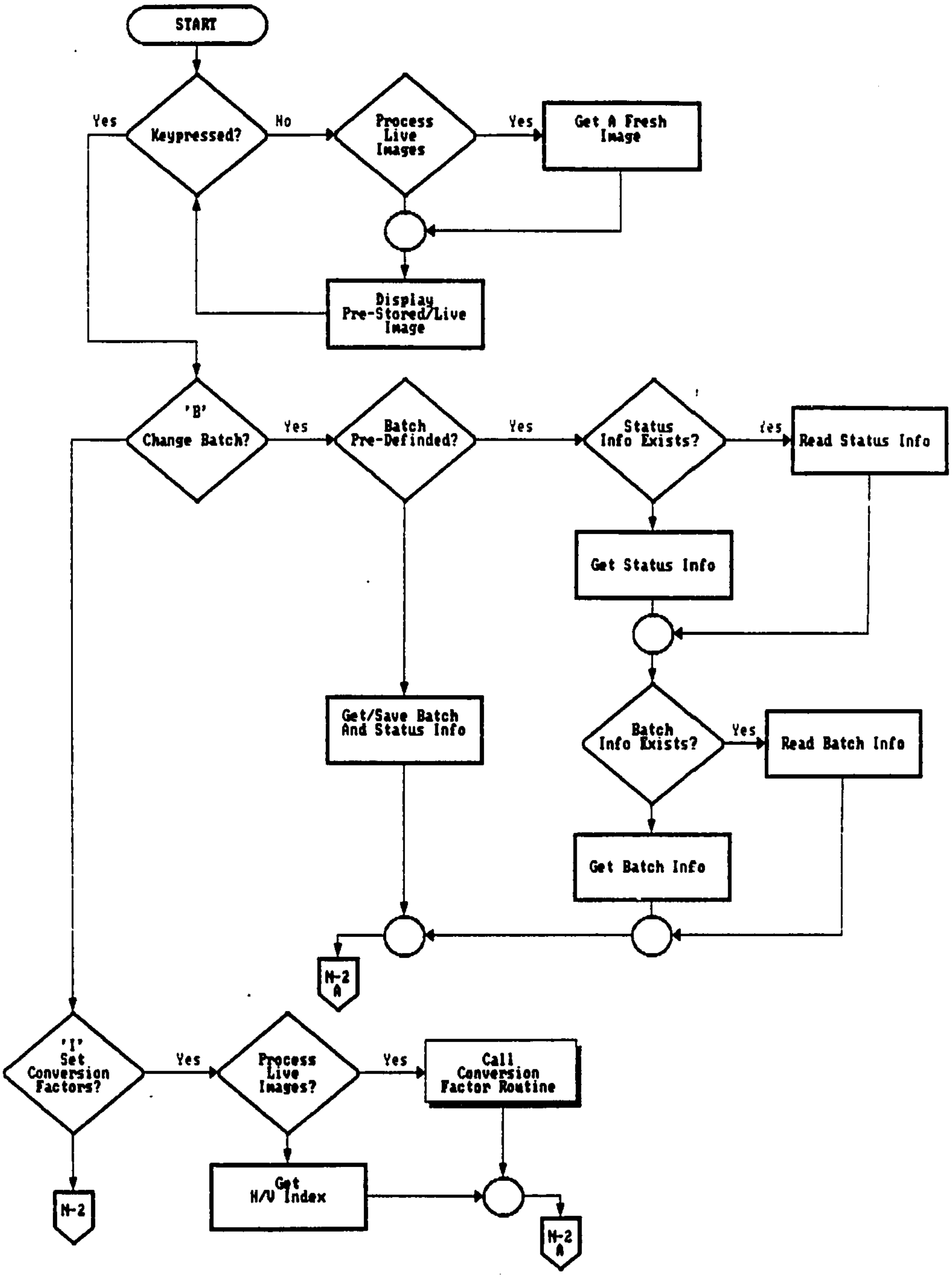
APPENDIX B. FLOW CHARTS OF THE MENU SYSTEMS

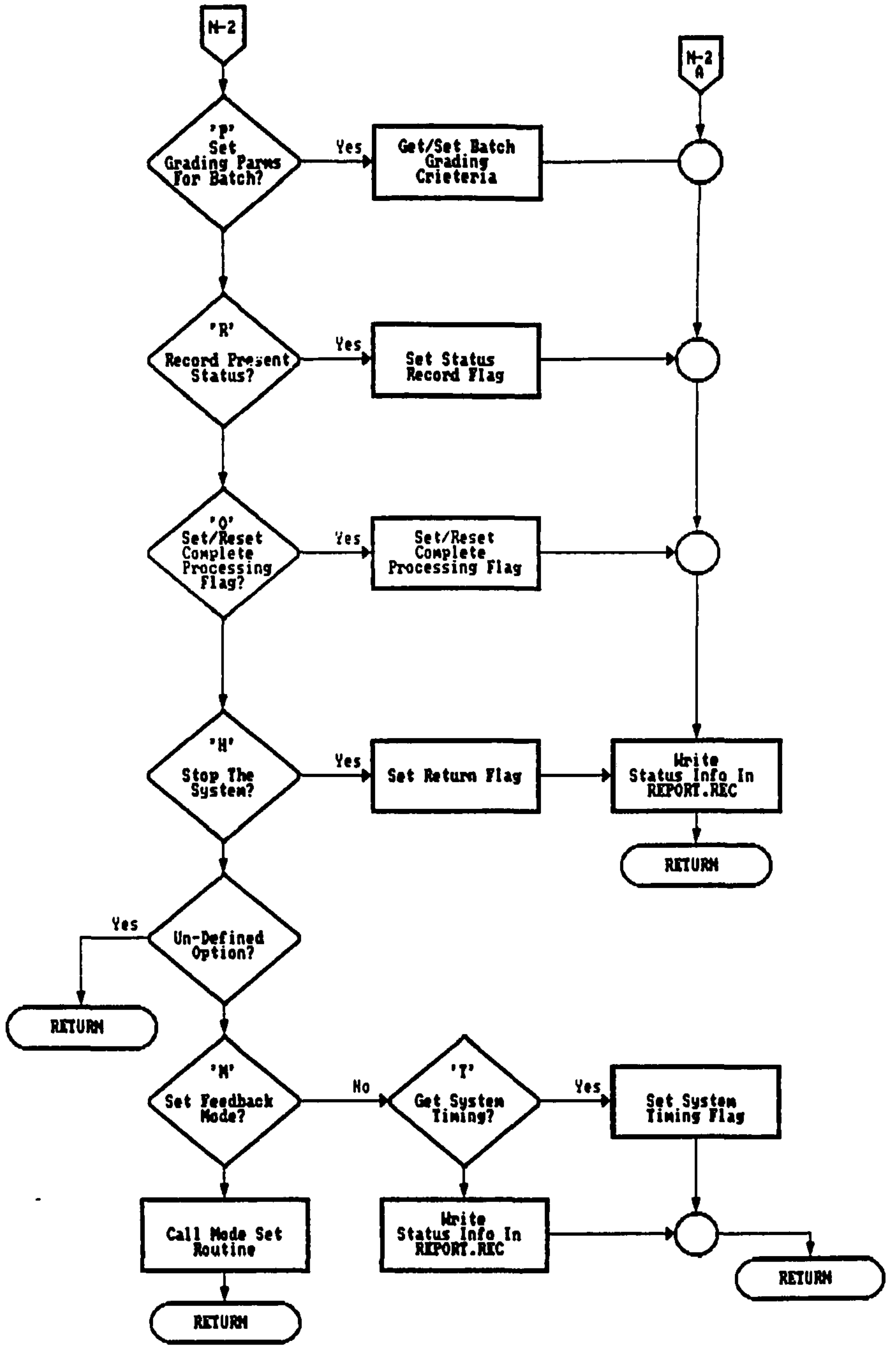
Flow Chart of The Main Program Loop



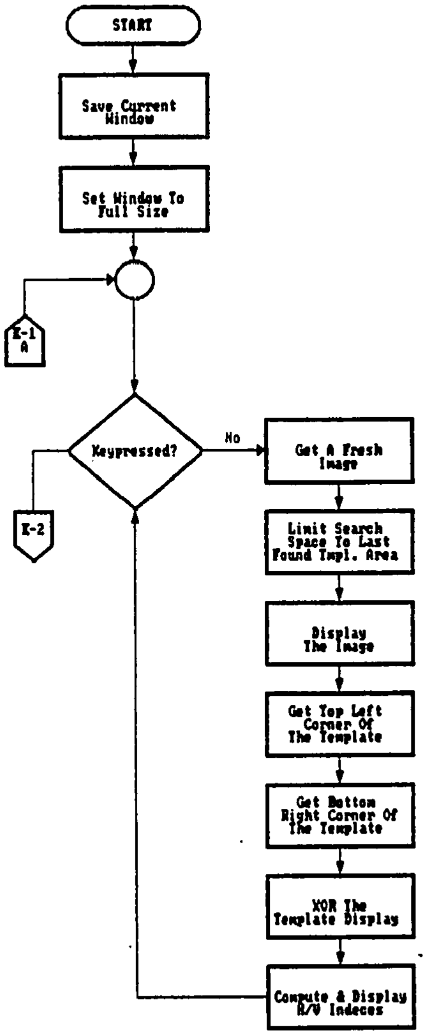

M-2
Attempt To Find The Clamp
Did We Find The Clamp?
No
Generate Appropriate Message
M-1 A
Get Root Mass
Yes
Root Mass Criteria Satisfied?
Isolate Tap Root Area
Count The Lateral Roots
Get Stem Length
Compute Caliper Diameter
Crieteria Satisfied?
No
Update Rejection Records
M-1 A
Update Acceptance Records
Update Screen Feedback

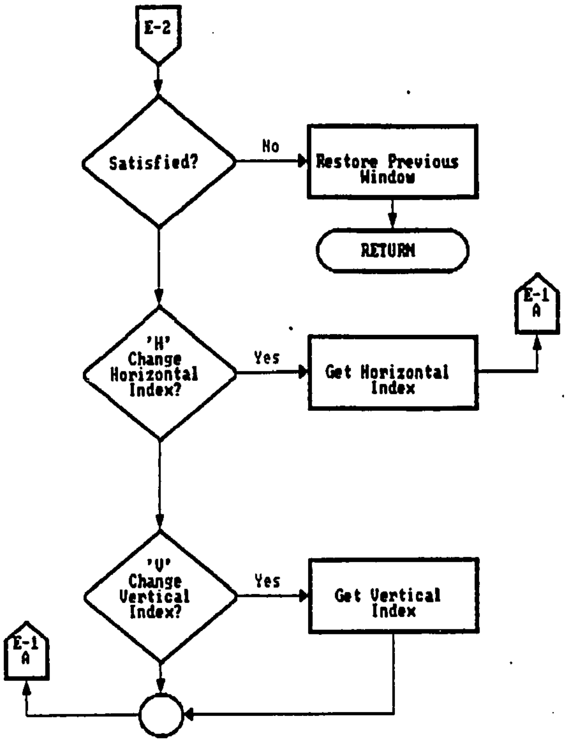
The Main Menu Flow Chart



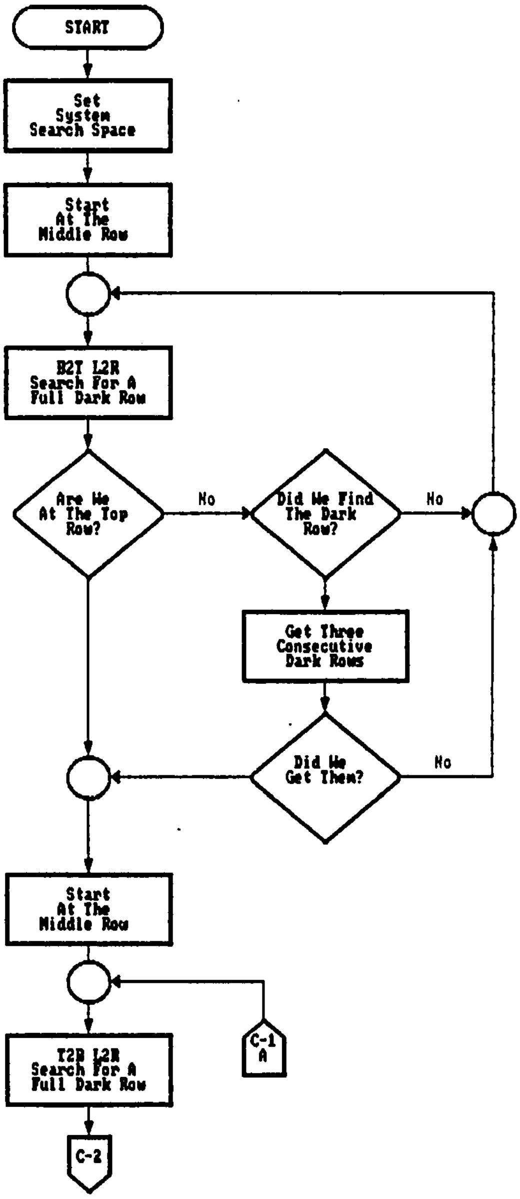

M-2
M-2 A
'P' Set Grading Parms For Batch?
Yes
Get/Set Batch Grading Crieteria
'R' Record Present Status?
Yes
Set Status Record Flag
'O' Set/Reset Complete Processing Flag?
Yes
Set/Reset Complete Processing Flag
'H' Stop The System?
Yes
Set Return Flag
Write Status Info In REPORT.REC
RETURN
Un-Defined Option?
Yes
RETURN
'M' Set Feedback Mode?
No
'T' Get System Timing?
Yes
Set System Timing Flag
Call Mode Set Routine
RETURN
Write Status Info In REPORT.REC
RETURN

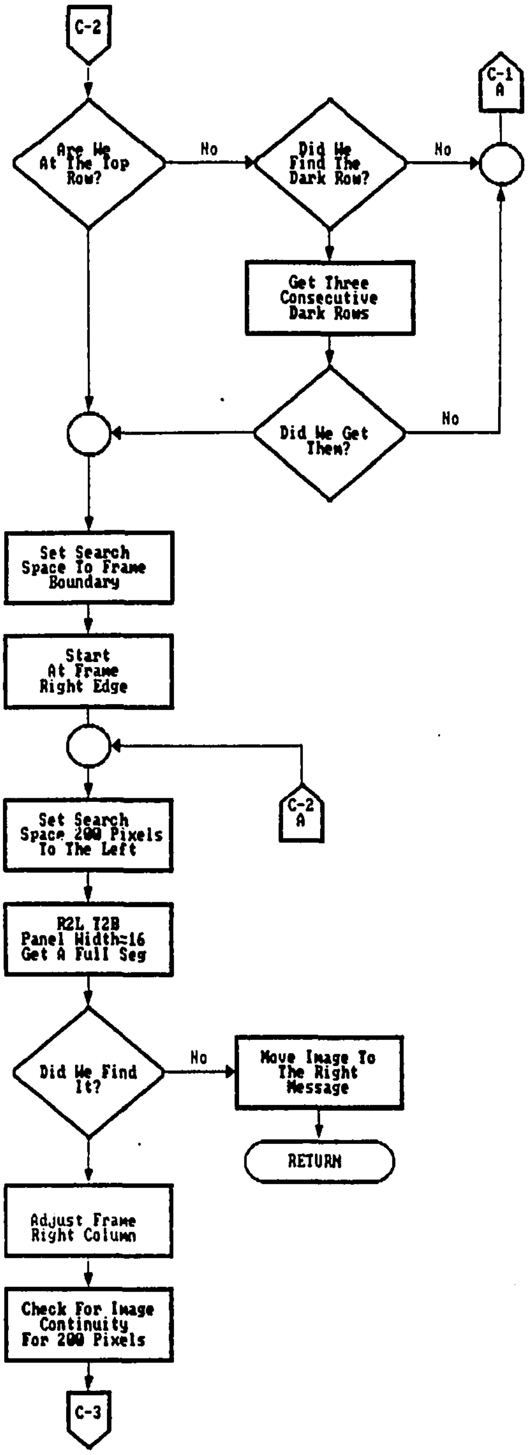
The Conversion Factor Flow Chart

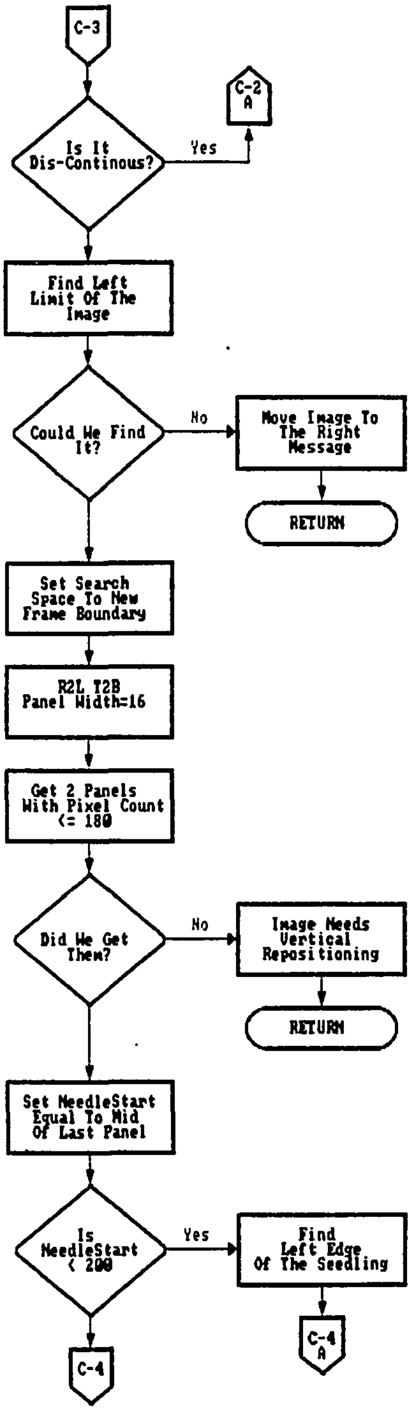

START
Save Current Window
Set Window To Full Size
K-1 A
Keypressed?
No
K-2
Get A Fresh Image
Limit Search Space To Last Found Tmpl. Area
Display The Image
Get Top Left Corner Of The Template
Get Bottom Right Corner Of The Template
XOR The Template Display
Compute & Display R/V Indeces

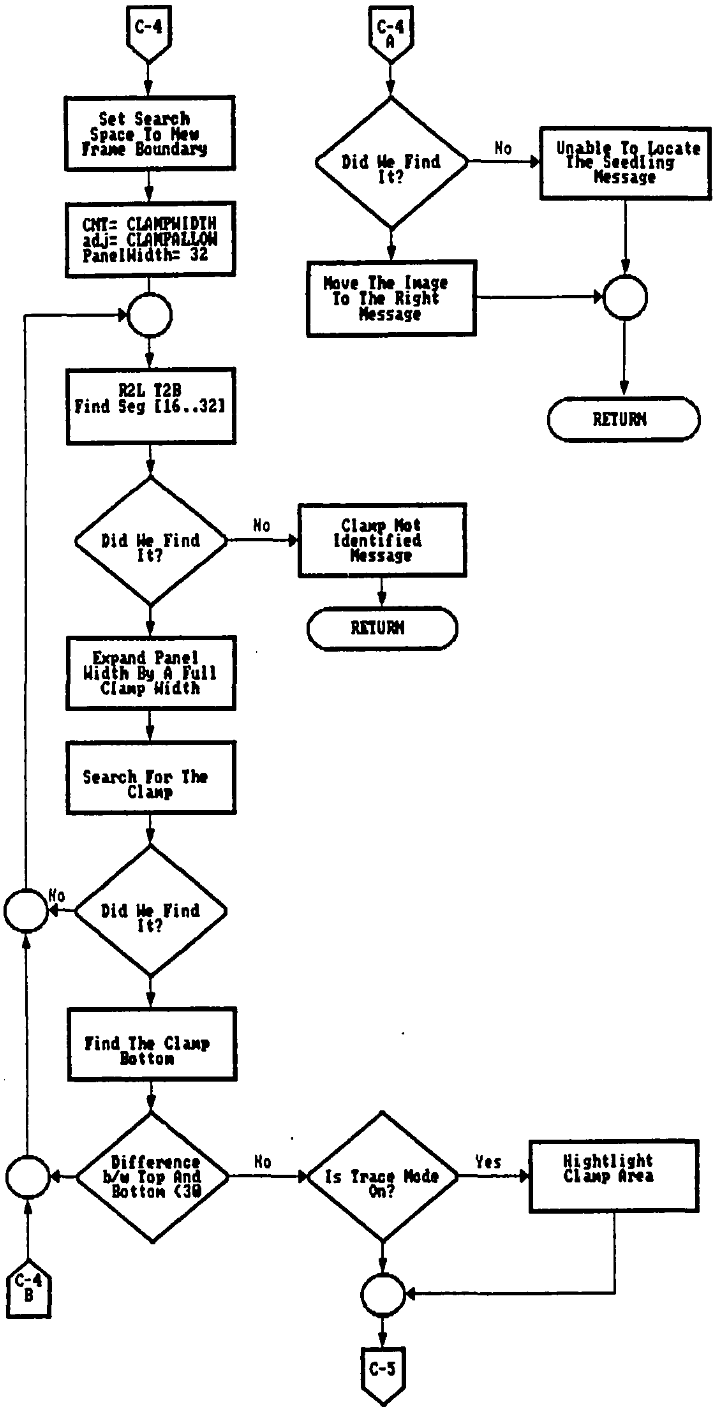

E-2
Satisfied?
No
Restore Previous Window
RETURN
'H' Change Horizontal Index?
Yes
Get Horizontal Index
E-1 A
'V' Change Vertical Index?
Yes
Get Vertical Index
E-1 A

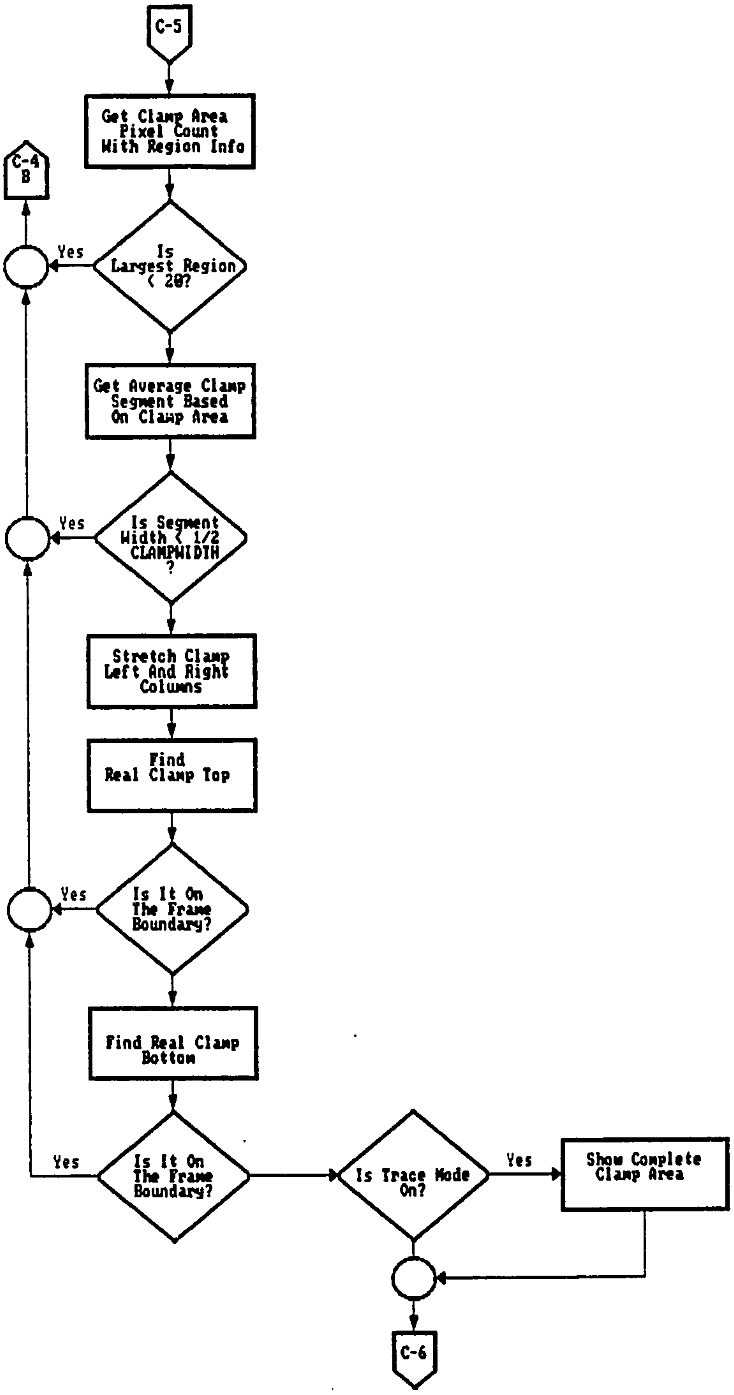
APPENDIX C. FLOW CHART OF THE CLAMPING POINT IDENTIFICATION

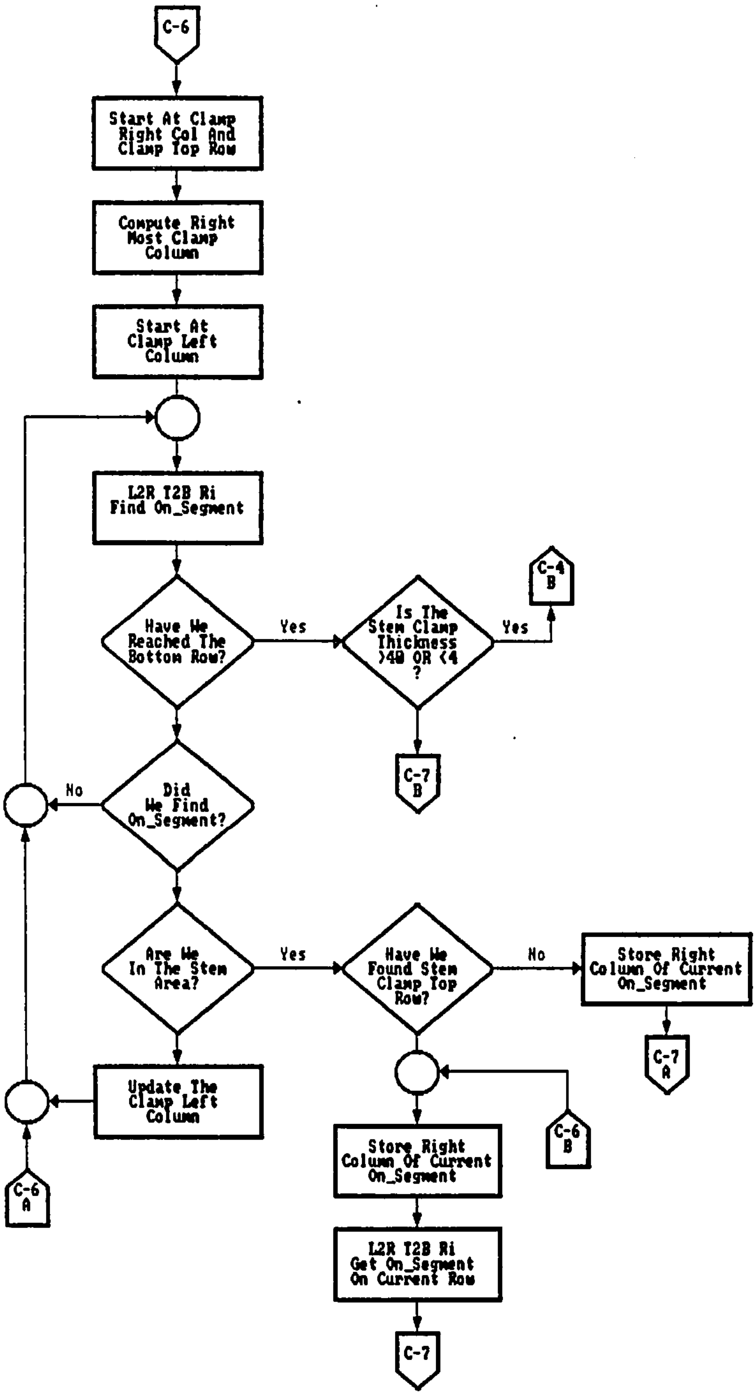


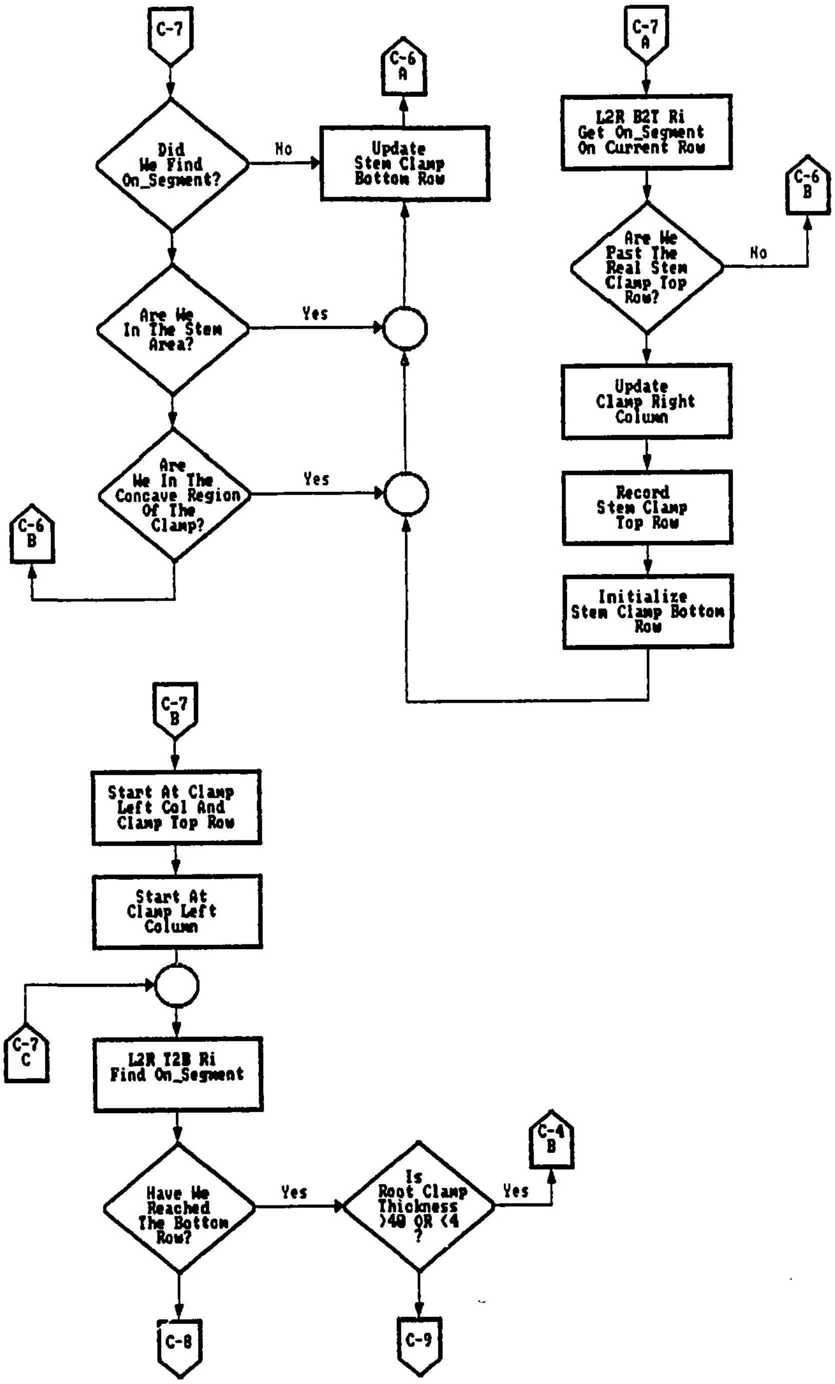

C-2
Are We At The Top Row?
No
Did We Find The Dark Row?
No
C-1 A
Get Three Consecutive Dark Rows
Did We Get Them?
No
Set Search Space To Frame Boundary
Start At Frame Right Edge
C-2 A
Set Search Space 200 Pixels To The Left
R2L T2B Panel Width=16 Get A Full Seg
Did We Find It?
No
Move Image To The Right Message
RETURN
Adjust Frame Right Column
Check For Image Continuity For 200 Pixels
C-3

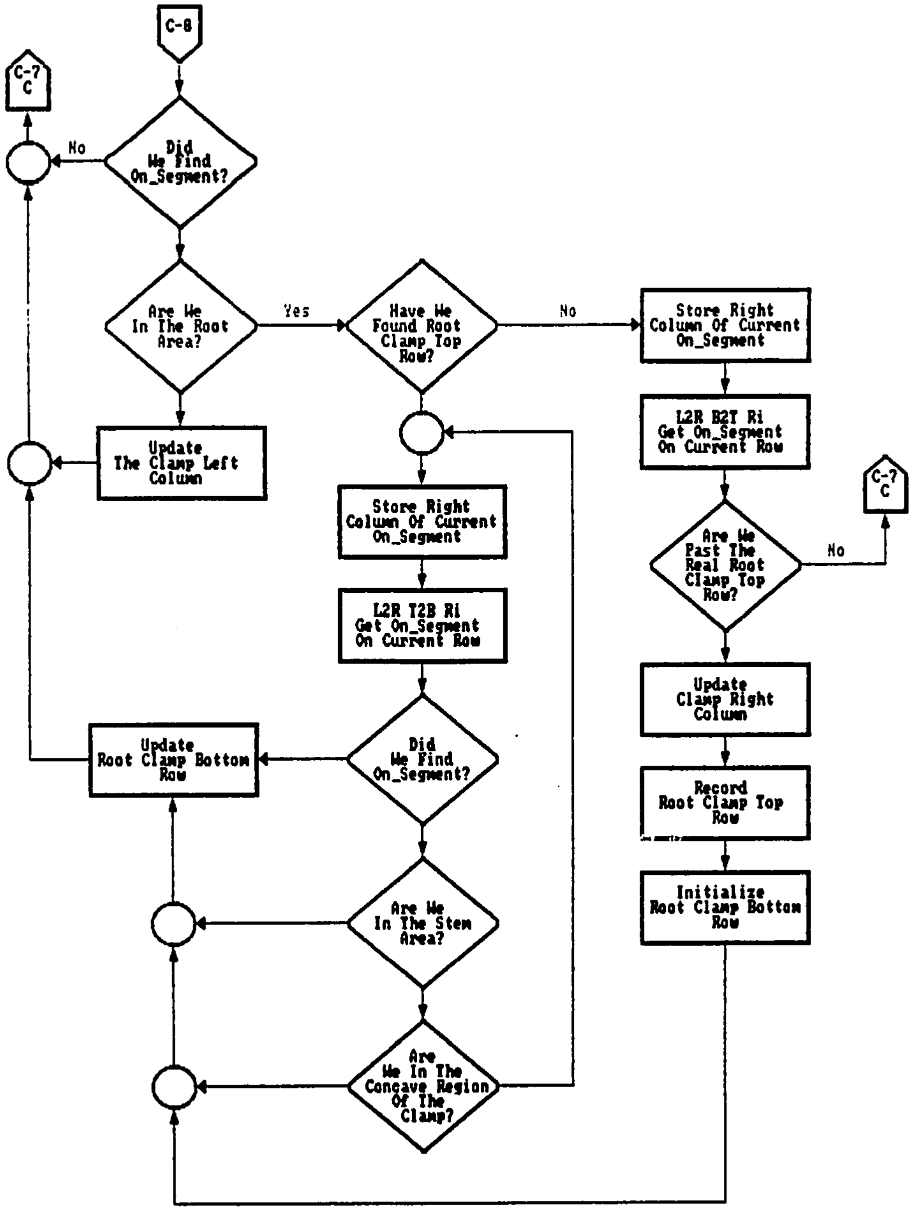

C-3
Is It Dis-Continous?
Yes
C-2 A
Find Left Limit Of The Image
Could We Find It?
No
Move Image To The Right Message
RETURN
Set Search Space To New Frame Boundary
R2L T2B Panel Width=16
Get 2 Panels With Pixel Count <= 180
Did We Get Them?
No
Image Needs Vertical Repositioning
RETURN
Set NeedleStart Equal To Mid Of Last Panel
Is NeedleStart < 200
Yes
Find Left Edge Of The Seedling
C-4 A
C-4

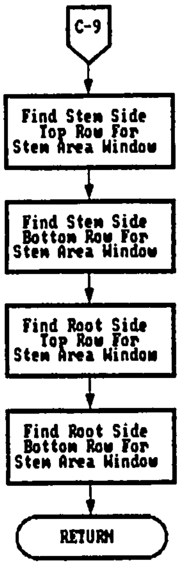

C-4
Set Search Space To New Frame Boundary
CNT= CLAMPWIDTH
adj= CLAMPALLOW
PanelWidth= 32
R2L T2B
Find Seg [16..32]
Did We Find It?
No
Clamp Not Identified Message
RETURN
Expand Panel Width By A Full Clamp Width
Search For The Clamp
Did We Find It?
No
Find The Clamp Bottom
Difference b/w Top And Bottom <30
No
Is Trace Mode On?
Yes
Hightlight Clamp Area
C-4 B
C-5
C-4 A
Did We Find It?
No
Unable To Locate The Seedling Message
Move The Image To The Right Message
RETURN


C-5
Get Clamp Area Pixel Count With Region Info
C-4 B
Yes
Is Largest Region < 20?
Get Average Clamp Segment Based On Clamp Area
Yes
Is Segment Width < 1/2 CLAMPWIDTH ?
Stretch Clamp Left And Right Columns
Find Real Clamp Top
Yes
Is It On The Frame Boundary?
Find Real Clamp Bottom
Yes
Is It On The Frame Boundary?
Is Trace Mode On?
Yes
Show Complete Clamp Area
C-6

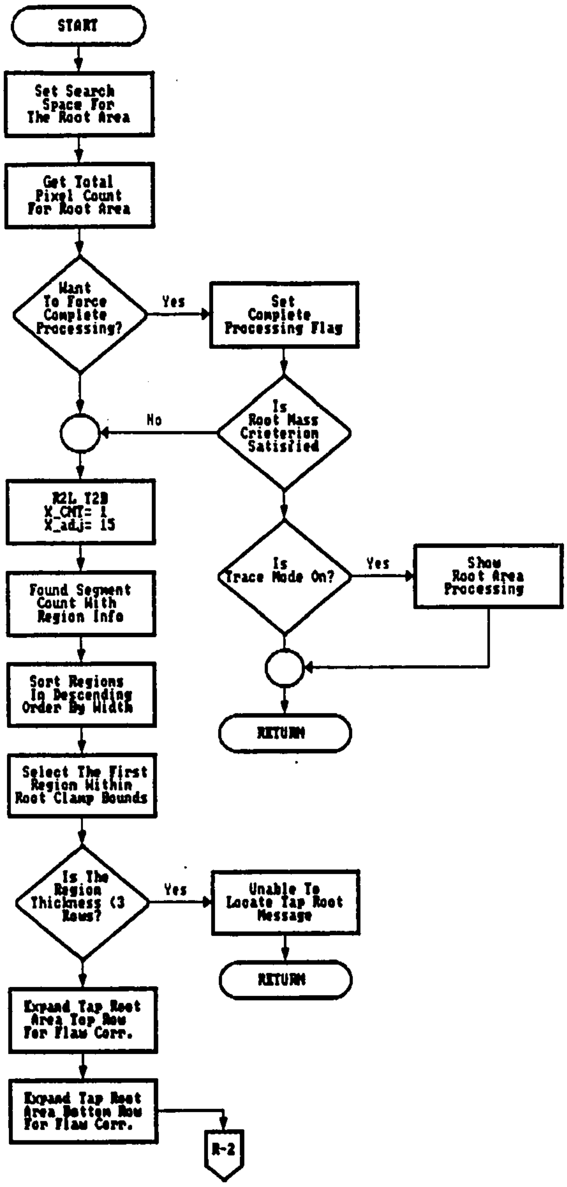

C-6
Start At Clamp Right Col And Clamp Top Row
Compute Right Most Clamp Column
Start At Clamp Left Column
L2R T2B Ri Find On_Segment
Have We Reached The Bottom Row?
Yes
Is The Stem Clamp Thickness >40 OR <4 ?
Yes
C-4 B
C-7 B
No
Did We Find On_Segment?
Are We In The Stem Area?
Yes
Have We Found Stem Clamp Top Row?
No
Store Right Column Of Current On_Segment
C-7 A
Update The Clamp Left Column
C-6 A
C-6 B
Store Right Column Of Current On_Segment
L2R T2B Ri Get On_Segment On Current Row
C-7

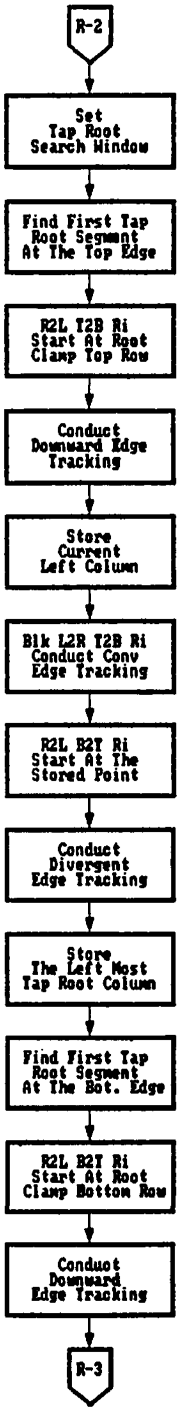

C-7
Did We Find On_Segment?
No
Update Stem Clamp Bottom Row
C-6 A
Are We In The Stem Area?
Yes
Are We In The Concave Region Of The Clamp?
Yes
C-6 B
C-7 A
L2R B2T Ri Get On_Segment On Current Row
Are We Past The Real Stem Clamp Top Row?
No
C-6 B
Update Clamp Right Column
Record Stem Clamp Top Row
Initialize Stem Clamp Bottom Row
C-7 B
Start At Clamp Left Col And Clamp Top Row
Start At Clamp Left Column
C-7 C
L2R T2B Ri Find On_Segment
Have We Reached The Bottom Row?
Yes
Is Root Clamp Thickness >40 OR <4 ?
Yes
C-4 B
C-8
C-9

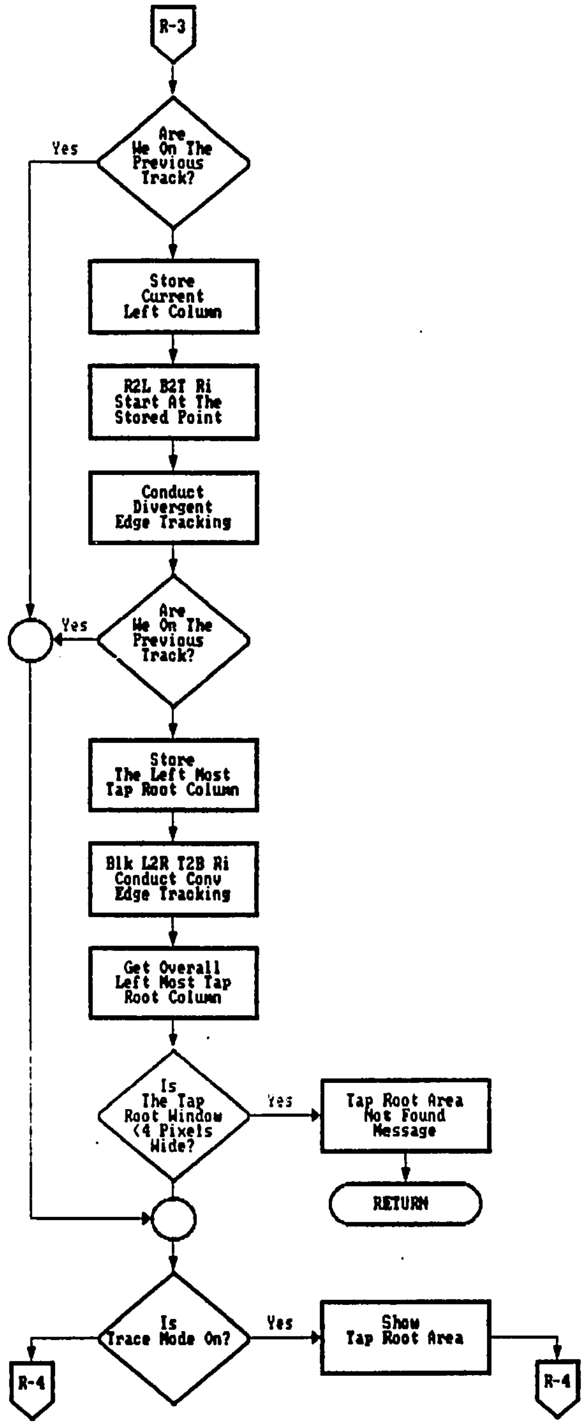

C-8
C-7
C
No
Did We Find On_Segment?
Are We In The Root Area?
Yes
Have We Found Root Clamp Top Row?
No
Store Right Column Of Current On_Segment
L2R B2T Ri Get On_Segment On Current Row
C-7
C
No
Are We Past The Real Root Clamp Top Row?
Update The Clamp Left Column
Store Right Column Of Current On_Segment
L2R T2B Ri Get On_Segment On Current Row
Update Clamp Right Column
Update Root Clamp Bottom Row
Did We Find On_Segment?
Record Root Clamp Top Row
Are We In The Stem Area?
Initialize Root Clamp Bottom Row
Are We In The Concave Region Of The Clamp?

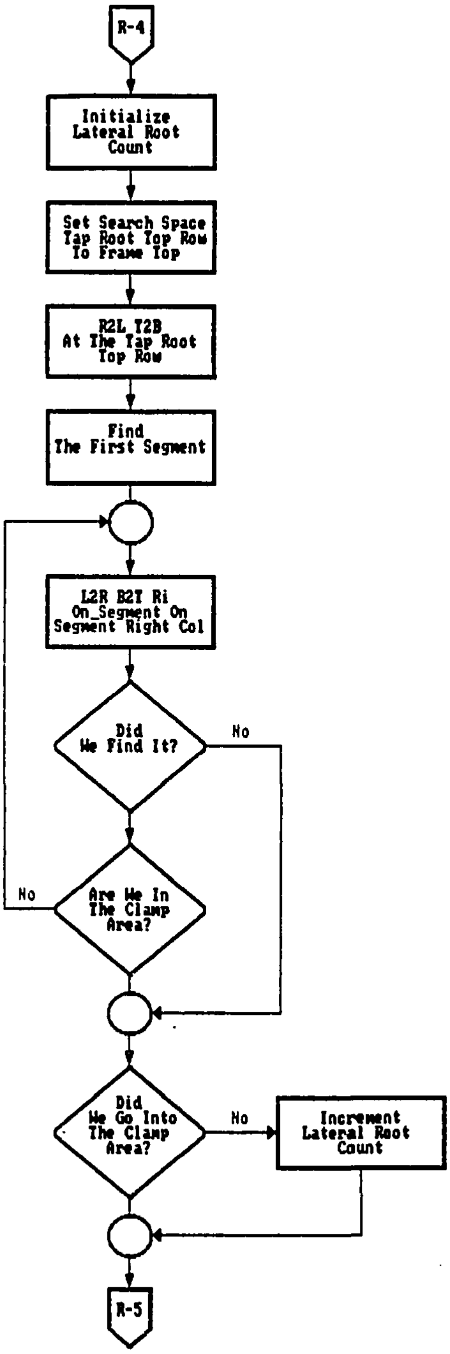

C-9
Find Stem Side Top Row For Stem Area Window
Find Stem Side Bottom Row For Stem Area Window
Find Root Side Top Row For Stem Area Window
Find Root Side Bottom Row For Stem Area Window
RETURN

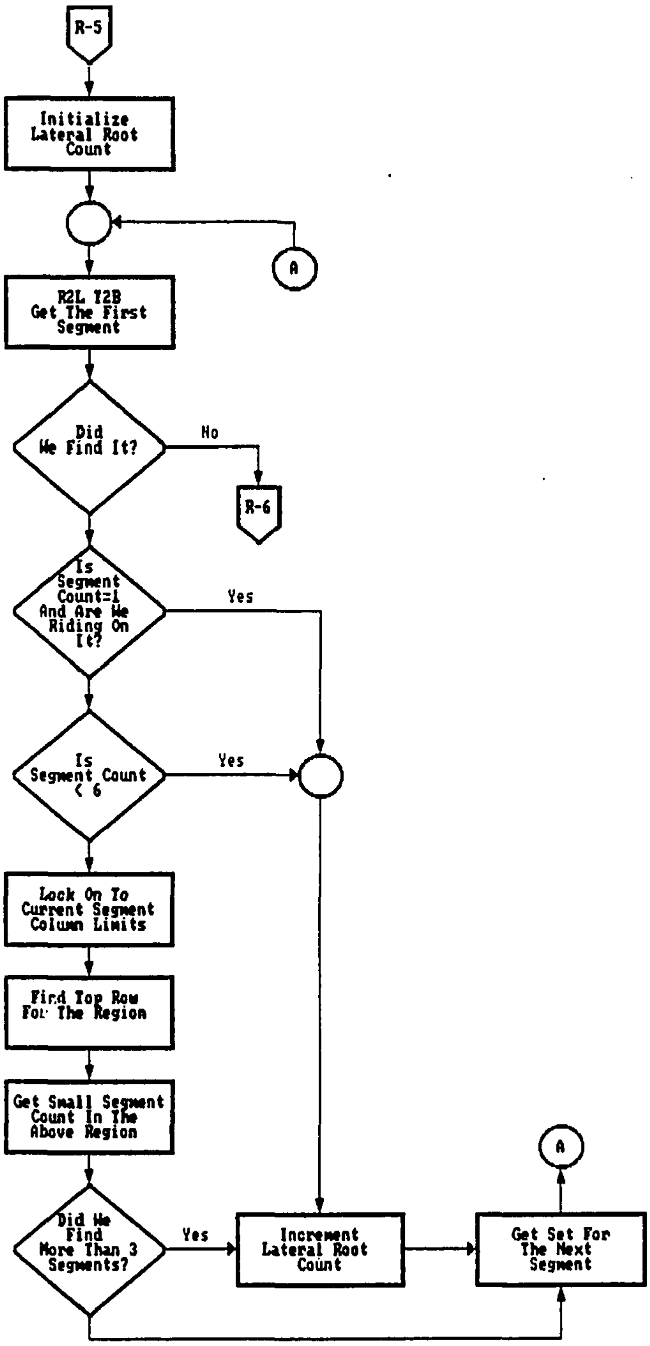
APPENDIX D. FLOW CHART OF THE ROOT AREA PROCESSING

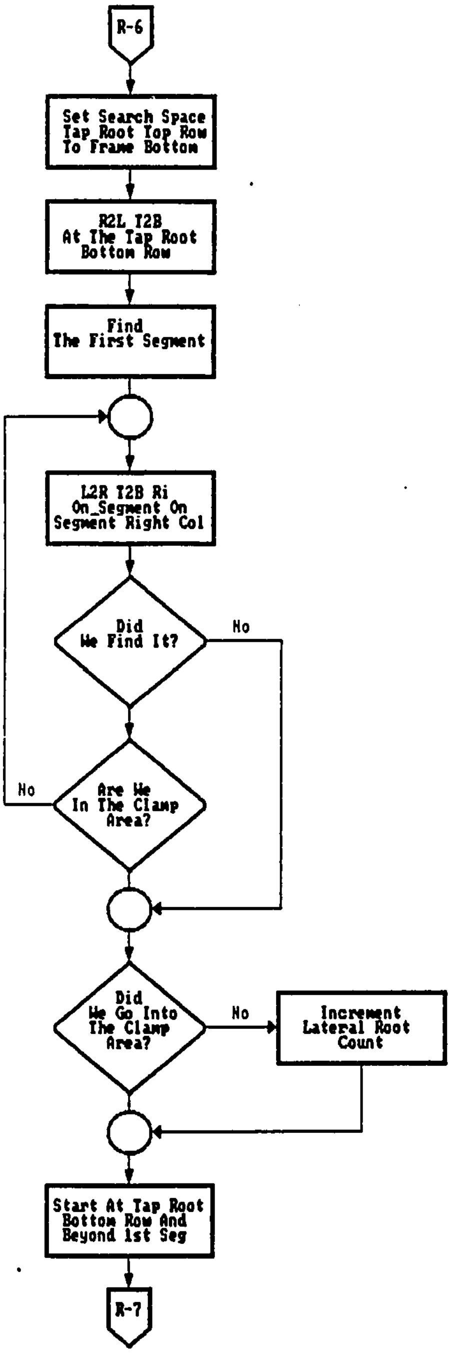

START
Set Search Space For The Root Area
Get Total Pixel Count For Root Area
Want To Force Complete Processing?
Yes
Set Complete Processing Flag
Is Root Mass Crieterion Satisfied
No
X2L Y2B X_CNT= 1 X_adj= 15
Found Segment Count With Region Info
Sort Regions In Descending Order By Width
Select The First Region Within Root Clamp Bounds
Is Trace Mode On?
Yes
Show Root Area Processing
RETURN
Is The Region Thickness (3 Rows?
Yes
Unable To Locate Tap Root Message
RETURN
Expand Tap Root Area Top Row For Flaw Corr.
Expand Tap Root Area Bottom Row For Flaw Corr.
R-2

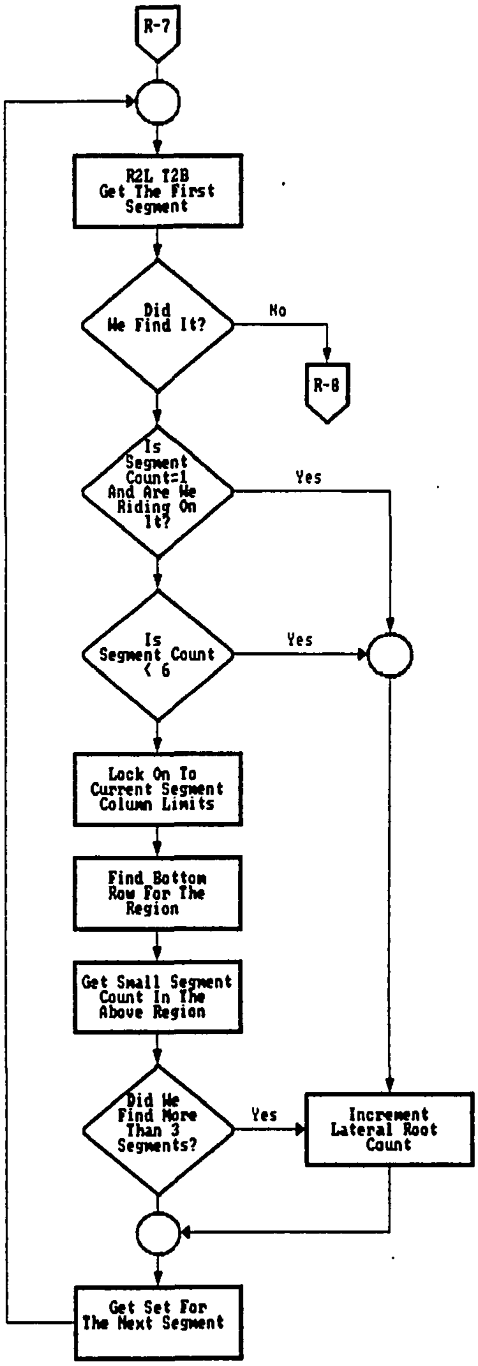

R-2
Set Tap Root Search Window
Find First Tap Root Segment At The Top Edge
R2L T2B Ri Start At Root Clamp Top Row
Conduct Downward Edge Tracking
Store Current Left Column
Blk L2R T2B Ri Conduct Conv Edge Tracking
R2L B2T Ri Start At The Stored Point
Conduct Divergent Edge Tracking
Store The Left Most Tap Root Column
Find First Tap Root Segment At The Bot. Edge
R2L B2T Ri Start At Root Clamp Bottom Row
Conduot Downward Edge Tracking
R-3

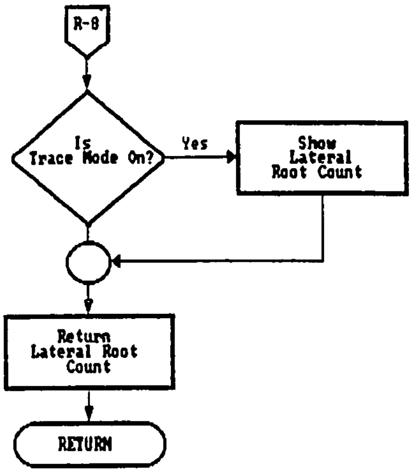

R-3
Are We On The Previous Track?
Yes
Store Current Left Column
R2L B2T Ri Start At The Stored Point
Conduct Divergent Edge Tracking
Are We On The Previous Track?
Yes
Store The Left Most Tap Root Column
Blk L2R T2B Ri Conduct Conv Edge Tracking
Get Overall Left Most Tap Root Column
Is The Tap Root Window <4 Pixels Wide?
Yes
Tap Root Area Not Found Message
RETURN
Is Trace Mode On?
Yes
Show Tap Root Area
R-4
R-4


R-4
Initialize Lateral Root Count
Set Search Space Tap Root Top Row To Frame Top
R2L T2B At The Tap Root Top Row
Find The First Segment
L2R B2T Ri On_Segment On Segment Right Col
Did We Find It?
No
Are We In The Clamp Area?
No
Did We Go Into The Clamp Area?
No
Increment Lateral Root Count
R-5

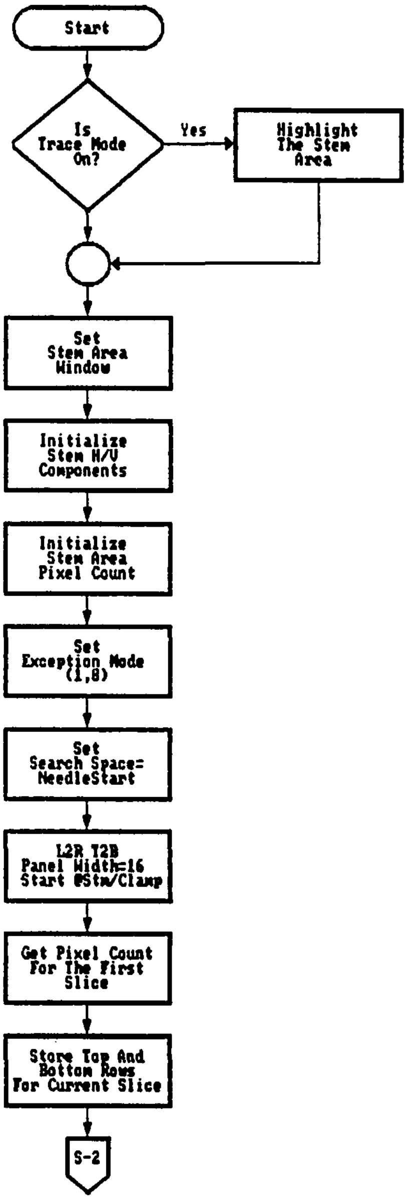

R-5
Initialize Lateral Root Count
A
R2L T2B Get The First Segment
Did We Find It?
No
R-6
Is Segment Count=1 And Are We Riding On It?
Yes
Is Segment Count < 6
Yes
Lock On To Current Segment Column Limits
Find Top Row For The Region
Get Small Segment Count In The Above Region
Did We Find More Than 3 Segments?
Yes
Increment Lateral Root Count
Get Set For The Next Segment
A

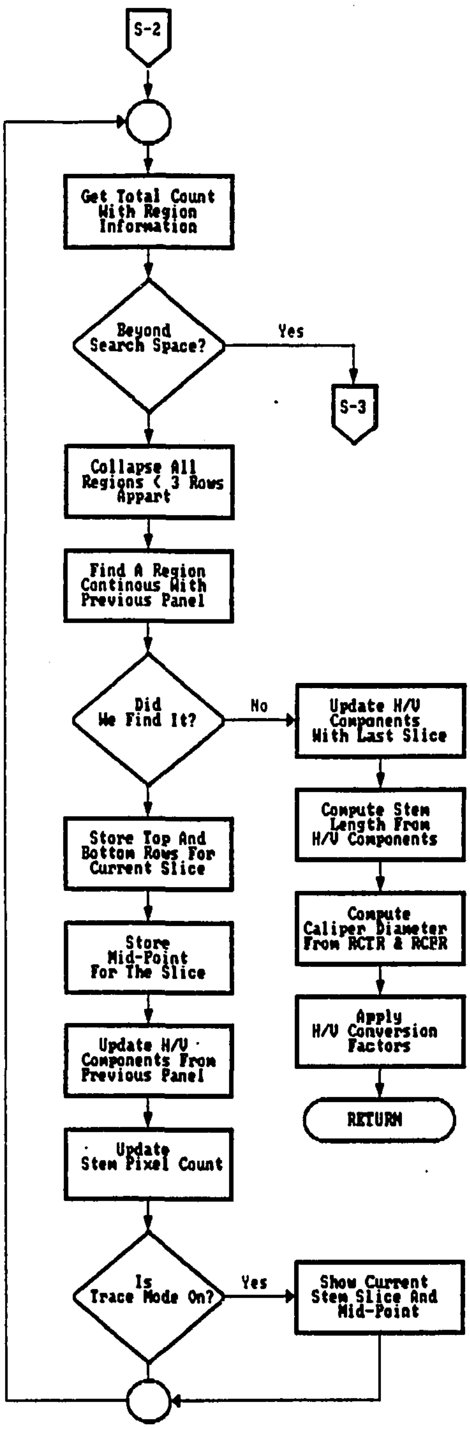

R-6
Set Search Space Tap Root Top Row To Frame Bottom
R2L T2B At The Tap Root Bottom Row
Find The First Segment
L2R T2B Ri On_Segment On Segment Right Col
Did We Find It?
No
Are We In The Clamp Area?
No
Did We Go Into The Clamp Area?
No
Increment Lateral Root Count
Start At Tap Root Bottom Row And Beyond 1st Seg
R-7

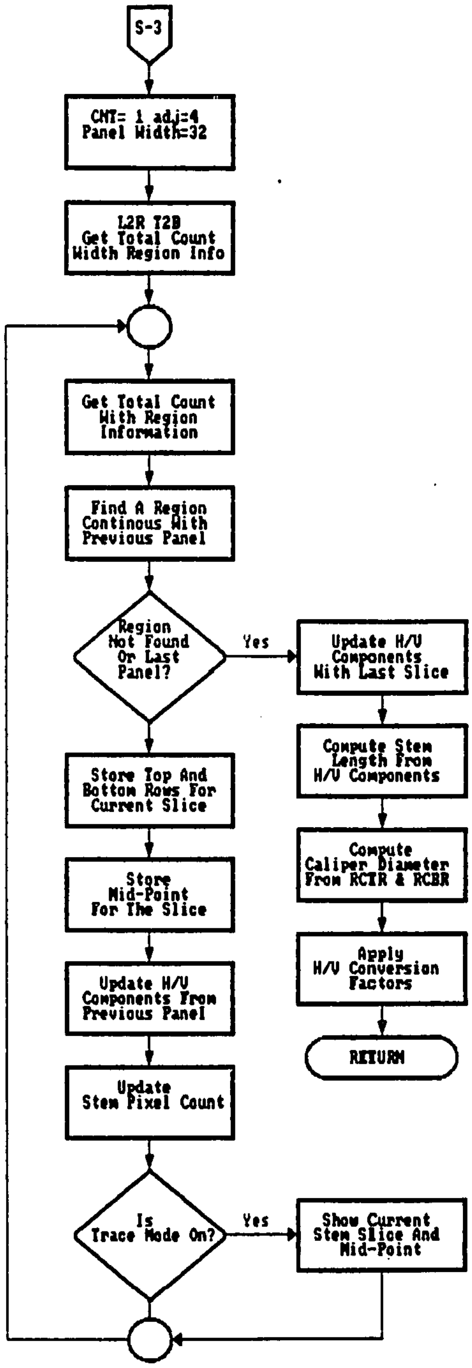

R-7
R2L T2B
Get The First
Segment
Did
We Find It?
No
R-8
Is
Segment
Count=1
And Are We
Riding On
It?
Yes
Is
Segment Count
< 6
Yes
Lock On To
Current Segment
Column Limits
Find Bottom
Row For The
Region
Get Small Segment
Count In The
Above Region
Did We
Find More
Than 3
Segments?
Yes
Increment
Lateral Root
Count
Get Set For
The Next Segment


R-8
Is Trace Mode On?
Yes
Show Lateral Root Count
Return Lateral Root Count
RETURN

APPENDIX E. FLOW CHART OF THE STEM AREA PROCESSING


Start
Is Trace Mode On?
Yes
Highlight The Stem Area
Set Stem Area Window
Initialize Stem H/V Components
Initialize Stem Area Pixel Count
Set Exception Mode (1,8)
Set Search Space= NeedleStart
L2R T2B Panel Width=16 Start @Stm/Clamp
Get Pixel Count For The First Slice
Store Top And Bottom Rows For Current Slice
S-2


S-2
Get Total Count With Region Information
Beyond Search Space?
Yes
S-3
Collapse All Regions < 3 Rows Appart
Find A Region Continous With Previous Panel
Did We Find It?
No
Update H/V Components With Last Slice
Compute Stem Length From H/V Components
Compute Caliper Diameter From RCTR & RCBR
Apply H/V Conversion Factors
RETURN
Store Top And Bottom Rows For Current Slice
Store Mid-Point For The Slice
Update H/V Components From Previous Panel
Update Stem Pixel Count
Is Trace Mode On?
Yes
Show Current Stem Slice And Mid-Point


S-3
CNT= 1 adj=4
Panel Width=32
L2R T2B
Get Total Count
Width Region Info
Get Total Count
With Region
Information
Find A Region
Continous With
Previous Panel
Region
Not Found
Or Last
Panel?
Yes
Update H/V
Components
With Last Slice
Compute Stem
Length From
H/V Components
Compute
Caliper Diameter
From RCTR & RCBR
Apply
H/V Conversion
Factors
RETURN
Store Top And
Bottom Rows For
Current Slice
Store
Mid-Point
For The Slice
Update H/V
Components From
Previous Panel
Update
Stem Pixel Count
Is
Trace Mode On?
Yes
Show Current
Stem Slice And
Mid-Point

BIBLIOGRAPHY

[1] G. H. Abdou. Availability Analysis of Flexible Manufacturing System. Ph.D. Dissertation. Iowa State University, Ames, Iowa, 1987.

[2] G. J. Agin. Computer vision systems for industrial inspection and assembly. *Computer* 13 (May 1980):11–20.

[3] J. W. Artley. Automated inspection: The new technology. *Design News* 38 (July 1982):94–96.

[4] B. G. Batchelor. Automatic visual inspection in industry. *The Industrial Robot* 5 (Dec 1978):174–176.

[5] T. O. Binford. Survey of model-based image analysis systems. *International Journal of Robotics Research 1* 1(1)(1982):18–63.

[6] Michael Brady. Computational approaches to image understanding. *ACM Computing Surveys* 14(1)(1982):3–71.

[7] Don Braggins and Jack Hollingum. *The Machine Vision Sourcebook.* Bedford, U.K.:IFS(Publications) Ltd., 1986.

[8] D. R. Brown, D. V. Fowler, W. T. Park, and A. E. Robinson. Applications of artificial intelligence/robotics. In *Robotics and Industrial Inspection*, pages 50–57. SPIE, Washington D. C., August 24–27, 1982.

[9] N. R. Brunelle and F. P. Higgins. Line scan vision system. *AT&T Technical Journal*, 65(July/August 1986):58–65.

[10] A. N. Burdett. Lifting, grading, packaging, and sorting. *Forest Nursery Manual: Production of Bare Root Seedlings*, pages 227–234. The Hague:M. Nijhoff/Dr. W. Junk Publishers, 1984.

[11] C. A. Chang, M. Leonard, O. Miller, and Y. Lee. A quality control issue in using computer vision inspection systems. In *Vision '86*, pages 4-57 - 4-74. Machine Vision Association of SME, Dearborn, Mich., 1986.

[12] Ning San Chang. Overview of machine vision: Current applications. In *Machine Vision: Breakthroughs and Challenges in Object Recognition and Architectures*, pages Chang 1 - Chang 24. IEEE, Piscataway, N.J., Nov 29, 1989.

[13] R. T. Chin and C. R. Dyer. Model-based recognition in robot vision. *Computing Surveys* 18(1)(March 1986):67–108.

[14] W. J. Clancey. Intelligent tutoring systems: A tutorial survey. In *Current Issues in Expert Systems*, chapter 3, pages 39–77. New York:Academic Press, 1987.

[15] R. Davis and D. Lenat Knowledge-based systems in artificial intelligence. New York:McGraw-Hill, 1980.

[16] D. R. DeVoe and G. A. Kranzler. Image processing to inventory tree seedlings in nursery beds. In *Proceedings of the Agri-Mation 1 Conference and Exposition*, pages 215–222. ASAE, Chicago, Feb 25–28, 1985.

[17] A. R. J. Doerr. Industrial vision application. In *Proceedings of the 2nd International Conference on Robot Vision and Sensory Controls*, pages 47–52. IFS, Stuttgart, Germany, Nov. 2–4 1982.

[18] J. H. Dooley and R. B. Fridley. Equipment development for forest nurseries. In *Proceedings of the American Society of Agricultural Engineers Symposium of Engineering Systems for Forest Regeneration*, pages 30–37. ASAE, Raleigh, N.C., March 1981.

[19] S. R. Draper and Travis A. J. Preliminary observations with a computer based system for analysis of the shape of seeds and vegetative structures. *Journal of the National Institute of Agricultural Botany (Great Britain)* 16(3)(1984):387–395.

[20] A. G. Dust. Machine vision inspection for hazardous environments. In *Vision '86 Conference Proceedings*, pages 9-9 - 9-18. Machine Vision Association of SME, Michigan, 1986.

[21] J. W. Edgren. Nursery storage to planting hole: A seedling's hazardous journey. In *Forest Nursery Manual: Production of Bare Root Seedlings*, pages 235–242, The Hague:M. Nijhoff/Dr. W. Junk Publishers, 1984.

[22] E. A. Feigenbaum. The art of artificial intelligence: Themes and case studies of knowledge engineering. In *International Joint Conference on AI*, pages 1014–1029. IJCAI, Massachusetts, 1977.

[23] D. G. Fink and D. Christiansen, editors. *Electronics Engineers' Handbook*, chapter 11. New York:McGraw-Hill Book Company, 1982.

[24] Ed. Flora, Philip C. *International Robotic Industry Directory.* 4th ed. Conroe, Texas:Technical DataBase Corporation, 1984.

[25] Food and beverage products increase use of machine vision in an effort to protect product purity, *Chilton's Food Engineering* 58 (November 1986):14.

[26] M. L. Gasperi. Introduction to morphological image processing. In *Vision '86, Conference Proceedings*, pages 5–63 – 5–84. Machine Vision Association of SME, Detroit, Mich., June 3-5, 1986.

[27] G. J. Gleason and G. J. Agin. A modular system for sensor controlled manipulation and inspection. In *Proceedings of 9th International Symposium on Robots*, pages 57–70. SME, Washington, D.C., 1979.

[28] R. C. Gonzalez and R. Safabakhsh. Computer vision techniques for industrial applications and robot control. *Computer* 15 (Dec. 1982):17–32.

[29] R. C. Gonzalez and P. Wintz. *Digital Image Processing.* London:Addison-Wesley, 1977.

[30] S. Gunasekaran. Optical methods for n.d. quality evaluation of agricultural and biological materials. *Journal of Agricultural Engineering Research* 32 (Oct 1985):209–241.

[31] D. E. Guyer, G. E. Miles, and M. M. Schreiber. Potential for computer vision as a spray controller. In *Proceedings of the Agri-Mation 1 Conference and Exposition*, pages 156–164. ASAE, Chicago, Feb. 25–28 1985.

[32] R. H. Heil and Michael Cole. Quantitative materials evaluation and inspection with the image analyzing computer. In *Imaging Techniques for Testing and Inspection*, pages 131–143. SPIE, L.A., Calif., Feb 1972.

[33] C. Hewitt. How to use what you know. In *International Joint Conference on AI*, pages 189–198. IJCAI, Tbilisi, Georgian SSR, 1975.

[34] L. Heyne. Computerized image analysis and protein quality of simulated pizza crusts. *Journal of Canadian Institute of Food Science and Technology* 18(2)(May 1985):168–173.

[35] L. F. Huggins, J. R. Barrett, and D. D. Jones. Expert systems: Concepts and opportunities. *Agricultural Engineering*, 67 (Jan/Feb 1986):21–23.

[36] M. Hwamoto. Recent studies on development in an automated citrus packing house facility in japan. In *Proceedings of International Society of Citriculture*, pages 831–834. Intl. Citrus Congress, Tokyo, 1982.

[37] IDETIX Digital Vision System Operator's Manual. *Micron Technology, Inc.*, Boise, Idaho, August 1986.

[38] M. J. Jaffe. Computer-assisted image analysis of plant growth, thigmomorphogenesis, and gravitropism. *Plant Physiology* 77 (March 1985):722–770.

[39] Harry Katzan. Artificial intelligence: State of the art. In *Expert Systems in the Organization: An Introduction for Decision-Makers*, chapter 3, pages 76–86. Chichester, London:Ellis Horwood Limited, 1988.

[40] Robert E. Keil. Survey of off-the-shelf imaging systems. In *Third Annual Applied Machine Vision Conference Proceedings*, pages 1–10. Robotics International of SME, Dearborn, Mich., 1984.

[41] G. A. Kranzler and D. R. DeVoe. Opto-electronic inventorying of tree seedlings in nursery beds. In *Proceedings of the AgriMation 1 Conference and Exposition*, pages 223–230. Chicago, Feb. 25–28, 1985.

[42] G. A. Kranzler D. R. CeVoe, H. W. Downs, and M. L. Stone. Computer vision for evaluation of agricultural chemical application. In *Proceedings of the AgriMation 1 Conference and Exposition*, pages 136–141. Chicago, Feb. 25–28, 1985.

[43] A. J. Laduzinsky. Object identification takes form in many shapes and stripes. *Control Engineering*, 33 (Oct 1986):81–83.

[44] J. A. Losty and P. R. Watkins. Computer vision for industrial applications. *Software and Micro Systems*, 2 (Oct 1983):130–138.

[45] Machine Vision Systems: A Summary and Forecast. *Tech Tran Corporation*, Naperville, Ill., 1983.

[46] Gero Mensel. Case study 1: A commercial application. In *Expert Systems in the Organization: An Introduction for Decision-Makers*, chapter 6, pages 101–115. Chichester, London:Ellis Horwood Limited, 1988.

[47] R. Michalski, H. Ko, and K. Chen. Qualitative prediction: The sparc/g methodology for inductively describing and predicting discrete processes. In *Current Issues in Expert Systems*, chapter 5, pages 125–157. San Diego, Calif.:Academic Press, 1987.

[48] R. S. Michalski. A theory and methodology of inductive learning. In *Machine Learning: An Artificial Intelligence Approach*, pages 83–134. Palo Alto, California:Tioga Publishing Co., 1983.

[49] K. R. Munson. Principles, procedures, and availability of seedling quality tests. U.S. Forestry Service, Rocky Mt. Forestry Range Exp. Stn., Technical Report 125, Feb. 1986.

[50] N. J. Nilson. *Principles of Artificial Intelligence.* Palo Alto, California:Tioga Publishing Company, 1980.

[51] M. J. Ottman. Measurement of viable plant roots with image analyzing computer. *Agronomy Journal*, 76 (Nov/Dec 1984):1018–1020.

[52] E. A. Parrish and A. K. Goksel. Pictorial pattern recognition applied to fruit harvesting. *Transactions of the ASAE* 20 (5)(1977):822–827.

[53] Volker P. Petersen. A microcomputer based binary vision system. Master's thesis. Iowa State University, Ames, Iowa, 1985.

[54] Volker P. Petersen and J. Even. A Microcomputer based binary vision system. *International Journal of Applied Engineering Education* 1 (6)(1985):405–414.

[55] G. B. Porter and J. L. Munday. Visual inspection system design. *Computer* 13 (5)(1980):40–48.

[56] M. P. Rigney and G. A. Kranzler. Machine vision for grading southern pine seedlings. ASAE, Technical paper 86–1597, 1986.

[57] C. Ronse and P. A. Devijver. *Connected Components in Binary Images: The Detection Problem.* Letchworth, Hertfordshor, England:Research Studies Press Ltd., 1984.

[58] N. Sarkar and R. R. Wolfe. Feature extraction techniques for sorting tomatoes by computer vision. ASAE, Technical Paper 84-6018, 1984.

[59] S. E. Savory. Expert systems: How can they be of use to your company. In *Expert Systems in the Organization: An Introduction for Decision Makers*, chapter 1, pages 21–33. Chichester, London:Ellis Horwood Limited, 1988.

[60] S. E. Savory. Tools for building expert systems. In *Expert Systems in the Organization: An Introduction for Decision-Makers*, chapter 4, pages 87–94. Chichester, London:Ellis Horwood Limited, 1988.

[61] J. Serra. *Image Analysis and Mathematical Morphology.* New York:Academic Press, 1982.

[62] Y. Shirai. Edge finding, segmentation of edges and recognition of complex objects. In *Proceedings of 4th International Joint Conference on AI*, pages 674–681. IJCAI, Tbilisi, Georgian SSR, 1975.

[63] W. D. Shoup and V. R. Macchio. A model for establishing opportunity values of robotics/automation in food processing. In *Proceedings of the Agri-Mation 1 Conference and Exposition*, pages 69–80. ASAE, Chicago, Feb. 25–28, 1985.

[64] D. G. Simpson. When to measure seedling quality in bare root nurseries. U.S. Forestry Service, Rocky Mt. Forestry Range Exp. Stn., Technical Report 125, Feb. 1986.

[65] Vivian Singer and M. A. Haase. Vision system tackles glass .inspection. *I&CS* 62 (February 1989):70–80.

[66] P.W. Sites and M.J. Delwiche. Computer vision to locate fruit on a tree. ASAE, Technical Paper 85–3039, 1985.

[67] G. Stroempl. Grading northern red oak planting stock. *Tree Planter' Notes 1*, U.S. Forestry Service 36(1)(1985):15–18.

[68] S. L. Supernault. PC-based image processing. *Robotics Age*, 7(3)(1985):20–22.

[69] Norihisa Suzuki. Knowledge-engineering workstations. In *Current Issues in Expert Systems*, chapter 8, pages 191–229. San Diego, Calif:Academic Press, 1987.

[70] R. W. Taylor and G. E. Rehkugler. Development of a system for automated determination of apple bruises. In *Proceedings of the Agri-Mation 1 Conference and Exposition*, pages 53–62. ASAE, Chicago, Feb. 25–28, 1985.

[71] Technical Reference: Personal Computer AT. *International Business Machines*, Boca Raton, Fla., March 1984.

[72] Mark A. Titus. A production system analysis: Seedling sorting and material handling process. Master's thesis, Iowa State University, Ames, Iowa, 1985.

[73] TN2500 Solid State Video/Digital Camera Operating Manual. *General Electric*, Syracuse, New York, April 1984.

[74] L. L. Toepperwein, M. T. Blackman, W. T. Park, W. R. Tanner, and W. D. Adolfson. *Robotics Applications for Industry: A Practical Guide.* Park Ridge, N.J.:Noyes Data Corporation, 1983.

[75] F. W. Von Althen. Proposed planting stock grades for hardwoods planted in ontario. Inf. Rep. 0-X-106, Canadian Forestry Service, Sault St. Marie, Ont., 1970.

[76] D. A. Waterman. A guide to expert systems. Reading, Mass:Addison-Wesley, 1985.

[77] Larry Werth. Machine vision implementation in agri–applications: The general purpose approach. In *Proceedings of the Agri-Mation 2 Conference and Exposition*, pages 63–77. ASAE, Chicago, Mar 3-5, 1986.

[78] D. L. Wheeler. Computer scientists try a new line of research toward development of robots that can 'see'. *The Chronicle of Higher Education*, 36 (Oct 11, 1989):A4.

[79] S. H. Witlwer. Automation in agriculture: Programming the future. In *Proceedings of the Agri-Mation 1 Conference and Exposition*, pages 1–14. ASAE, Chicago, 1985.

[80] M. E. Wright. Measuring sweet potato size and shape with a computer. *Louisiana Agricultural Experiment Station Bulletin*, 28(1)(1984):12–13.

[81] H. W. Yankee. *Manufacturing Processes.* Englewood Cliffs, N.J.:Prentice-Hall, Inc., 1979.

[82] F. Zensen. Lift and pack procedures at the J. Herbert stone nursery. *USDA Forest Service Research Paper North Cent. Forestry Exp. Stn.*, 185 (6)(1985):30–32.