8

77

BARD

FINAL REPORT

PROJECT NO. US-1254-87

הספריה המרכזית
למדע וטכנולוגיה
בית-171

Robotic Systems for Selective Harvesting: Basic Concepts and Prototype Tests

G.E. Miles, G.W. Krutz, U.M. Peiper, Y. Alper,
A. Hetzorni, I. Wolf

34

630.72
BAR/MIL
2nd copy

1991

000000229753

36904

October 1, 1990

BARD
P. O. Box 6
Bet Dagan, ISRAEL

BARD Project No. US-1254-87

**ROBOTIC SYSTEMS FOR SELECTIVE HARVESTING:
BASIC CONCEPTS AND PROTOTYPE TESTS**

Investigators	Institutions
Gaines E. Miles	Purdue University
Gary W. Krutz	Purdue University
Uri M. Peiper	Agricultural Research Organization
Yekuti'el Alper	Agricultural Research Organization
Amots Hetzroni	Agricultural Research Organization
Itzhak Wolf	Agricultural Research Organization

המספרים הנרשמים
הם
11-1-11

Starting Date: October 3, 1988

Termination Date: October 2, 1990

Type of Report: Final

Gaines E. Miles
Principal Investigator

J. A. Jones
Purdue Research Foundation

630.72
BAR/MIL : 635.61 : 631.35 : 629.892
2nd copy

ACKNOWLEDGEMENTS

Many people contributed to the success of this project. Graduate students Yael Edan, Meny Benady and Martha Cardenas-Weber faithfully worked many long hours - late night and early morning. Without their willingness to learn new complex technologies then apply their knowledge and expertise to robotic harvesting, this project would not have been possible. Their dedicated efforts have succeeded where anything less would have failed.

Numerous faculty and staff have contributed their time and expertise in many ways: advising students through special problems, reviewing and editing manuscripts, and making laboratory and field facilities available. Professor Jonathan Bar-Or and his students, especially Tony Wade and Eric Johanson, developed conceptual models of the harvester and gripper. Dr. Gerry Wilcox always freely gave his knowledge of melon production practices. Don Rhine provided the technical and logistics support required to collect field data. Drs. Kamyar Haghighi, Richard Stroshine and Bernard Engel spent countless hours guiding the development of the finite element models and the intelligent controller model.

Perhaps the unsung hero has been Marilyn Haas, who has put the finishing touches on manuscripts and helped maintain order in an otherwise chaotic world.

Finally and most importantly, this work would not have been possible without the gracious financial support of the U.S. - Israel Binational Agricultural Research and Development Fund, Purdue University and The Volcani Institute.

To each and every one of you, from the depths of our hearts, Thanks ya'll.

Gaines E. Miles
Uri M. Peiper

ROBOTIC SYSTEMS FOR SELECTIVE HARVESTING: BASIC CONCEPTS AND PROTOTYPE TESTS

TABLE OF CONTENTS

Topic	Page
Abstract	1
Objectives	2
Report	3
Sensing of Melons	3
Machine Vision.....	3
Distance to Melons	26
Stereoscopic Vision	34
Triangulation	35
Ultrasonic.....	38
Laser ranging	42
Sensor fusion	51
Melon Parameters.....	67
Distribution of Melon Locations	67
Finite Element Analysis of Melons	72
Gripper Design.....	91
Finite Element Analysis of Gripper Design	91
Grippers for a Melon Harvesting Robot.....	105
Design of a Robotic Melon Harvester	113
Animated Visual Simulation	113
Effects of Robot Parameters	121
Effects of Horticultural Practices	141
Intelligent Control System.....	151
References.....	165
Description of Cooperation.....	173
Evaluation of the Research achievements	174
Sensing of Melons	174
Melon Parameters	174
Gripper.....	175
Robotic Melon Harvesting System.....	175
Publications	176

ROBOTIC SYSTEMS FOR SELECTIVE HARVESTING: BASIC CONCEPTS AND PROTOTYPE TESTS

ABSTRACT

The fundamental technologies to harvest melons robotically have been developed. Machine vision algorithms have been developed to locate each individual fruit and have proven to be able to detect 86 to 89% of the melons. To eliminate false detections and measure distance to the fruit, a projected laser line can be used with the machine vision system. Ultrasonic sensors and stereoscopic vision were unable to detect melons occluded by foliage.

A basket-type gripper has been modeled, designed, constructed and proven successful in manual tests. This device when attached to a manipulator will grasp each fruit, lift it, detach it from the vine and transport it to a packing carton or conveyor. Finite element models of melons and grippers predict the stresses and deformations imposed on the fruit are less than the magnitude required for bruising or damaging the fruit.

Dynamic, numerical simulations and animated, visual simulations of robotic harvesting have quantified the sensitivity of the machine to the type of robot, number of arms, configuration, actuator speeds and melon distribution. Two arms in tandem was the fastest configuration. The actuator along the row should be capable of moving at 200 cm/sec and the actuator across the row should move at 100 cm/sec. The frame should be approximately 2 meters, depending on spacing of the plants. Results show the machine performance is highly sensitive to melon distribution which is affected by cultural practices. Simulations using a staggered hill spacing which alternates from one side of the bed to the other show a significant improvement in machine performance can be realized under this cropping regime.

The fundamental technologies of robotic harvesting developed by this project, plus the feasibility of designs proven by the numerical and animated simulations formulate the basic concepts necessary for developing a prototype robotic melon harvester.

ROBOTIC SYSTEMS FOR SELECTIVE HARVESTING: BASIC CONCEPTS AND PROTOTYPE TESTS

OBJECTIVES

The objectives of this project were to define the fundamental technologies required to harvest melons robotically. These include:

- The machine vision techniques and image processing algorithms, and distance sensors necessary to locate melons occluded by plant leaves and vines;
- The gripping mechanisms necessary to harvest melons individually;
- The manipulator motions necessary to efficiently and effectively harvest melons; and
- The computer system architecture necessary to monitor all sensors, analyze and interpret the data, direct the manipulator to harvest each fruit detected, and control the overall systems operation of the robotic harvester.

SENSING OF MELONS

The location of melons with respect to a robot frame can be determined by machine vision to capture scenes and image processing to discriminate melons from the background and other objects. Distance to the fruit can be determined visually by triangulation or by stereo vision, or by an ultrasonic sensor.

MACHINE VISION TO LOCATE MELONS AND GUIDE ROBOTIC HARVESTING

INTRODUCTION

Typical agricultural processes, such as transplanting of young plants or harvesting of individual fruits (melons, apples), are labor intensive, tedious operations. Application of robots to agricultural processes promises to reduce the manual labor, lower the costs, and increase the overall quality of the produce. The robotic harvesting of melons is an example for an agricultural automation project which would benefit from robotic manipulation. Melons are fleshy, delicate fruits. They do not ripen at the same time. Every fruit must be separately detected, evaluated, and harvested.

The detection and evaluation of melons is a complex task because of the variability of the object features and the influences of the changing environment. The features of natural objects, such as melons, are not well defined since they vary in size, shape, texture, and color. Furthermore, the object features are dependent on the environmental influences - shadows, rain, wind, or occluding leaves - which may change or hide the absolute melon features (e.g. the perceived shape of the melon). The melon position in the field is random and its relationship with the neighboring objects (overlap, shadowing, hindrance, and occlusion) has an impact on the visible features. The environmental conditions continuously change and can not be predicted with sufficient reliability.

The dynamic nature of the robotic melon harvester further complicates the sensing process. The robotic melon harvester must be mobile to reach all melons in the harvesting area. Therefore, the melon locations must be acquired, organized, and updated simultaneously and in conjunction with the harvesting and moving operations. The melon information must be independent of the harvester motion.

The objective of this work is to develop a machine vision based concept for detecting and evaluating melons in a melon field. Machine vision was chosen as the sensing technology because of its capability to provide large amounts of scene information and its operational range covering small and large areas in a broad range of distances. This envisioned machine vision system must encompass the sensing and the processing tasks, generate the precise three dimensional melon location and preliminarily evaluate the melon ripeness.

LITERATURE REVIEW

Several research efforts have focused on harvesting fruits with robotic means and sensor support. Grand D'Esnon (1987) developed a robotic system for harvesting apples. The harvesting of grapes supported by machine vision was investigated by Sevila (1985). A sensing system for the harvesting of oranges was designed and implemented by Harrell, et al. (1988). The harvesting of tomatoes with vision was investigated by Kawamura (1987). These four robotic harvesting systems used machine vision for the two-dimensional detection of the fruits. The third dimension (distance to the fruit from the manipulator) was obtained with different techniques. Grand D'Esnon (1987) used a touch sensor, Harrell (1988) applied an ultrasonic sensor, and Kawamura (1987) used stereoscopic vision with an integrated binocular camera.

The discriminating features depend on the object of interest. Apples and tomatoes are discriminated with color filtering (Grand D'Esnon, 1987; Kawamura, 1987). Oranges are detected with filtering and color segmentation (Slaughter, 1986, 1987; Harrell, 1988). Other discriminating features are the brightness, the texture, the shape, and the size of objects. These features represent the distinguishing characteristics of the objects, or the primitive components that combine into an object. This information is used to generate hypothesis about the objects in the scene (Koch and Kashyap, 1986).

Machine vision applies image processing and analysis methods to extract and interpret the visual information in the images. A basic strategy for the two dimensional data processing and analysis (Rosenfeld, 1984) is:

1. Divide the image into contrasting regions (segmentation),
2. Extract the features of the primitive components and describe them (feature extraction),
3. Recognize the primitive components (object recognition), and
4. Describe and interpret the overall image (image interpretation).

Preprocessing methods can also be used before the basic strategy to correct for distortions from previous processes such as transmission and reception, to enhance the image, and to highlight certain image information for a specific feature extraction process.

Segmentation isolates image subareas from their background based on the similarities and discontinuities of these areas (Haralick and Shapiro, 1985). The segmentation process can be iteratively improved based on the region features. Segmentation strategies differ in the way they emphasize the desired features and in the way they balance and compromise one desired feature against another (Haralick and Shapiro, 1985).

The most common segmentation strategy is based on region analysis (Ohta, 1985). The region analysis methods can be categorized into region growing schemes, region splitting schemes (spatial clustering), and combined region growing and splitting schemes (Haralick and Shapiro, 1985). The region growing schemes start with atomic regions, such as pixels or tiny square regions, and combine them into coherent regions in a bottom up strategy. Region splitting schemes separate the image into subareas based on the spatial properties of the image. Region growing and splitting (Zucker, 1976; Kanade, 1980) combines the splitting and merging

processes of adjacent subareas whenever the local properties indicate a strong probability the areas are common.

Region splitting defines a partition in measurement clustering space and assigns a label to each pixel of the partition (Haralick and Shapiro, 1985). The image subareas are defined as the connected components of the pixels having the same label. Partition strategies for region splitting are: pixel level clustering, iterative partition rearrangement schemes, and histogram mode seeking. Pixel level clustering compares each pixel value with every other pixel value in the image. Iterative partition rearrangement schemes go through the image data set many times with or without sampling for the clustering.

Histogram mode seeking analyzes the gray level distribution over the gray scale (Rosenfeld and Kak, 1982). Histogram mode seeking assumes that homogeneous objects are clusters in the measurement space (Haralick and Shapiro, 1985). This method is particularly suitable for automatic threshold selection in situations with image characteristics varying in a broad range of intensity distributions (Gonzalez and Wintz, 1987). The valleys in the histogram separate the image into clusters based on their brightness values. A valley in a bimodal histogram separates the image into bright and dark objects.

The selection of the threshold value is critical in the segmentation process (Rosenfeld and Kak, 1982). Small variations on the selected threshold render very different segmentations whenever overlapping regions appear in the scene. Histogram smoothing is often used under this condition to locally smooth the image before selecting the threshold. The effect dampens local gray level fluctuations, and hence reduces the variability of the gray levels while preserving the mean gray level. The histogram peaks are sharpened towards their means which makes them easier to separate.

Feature extraction focuses on the analysis of the region features: the color, the brightness, the texture, the shape, and the size (Ohta, 1985). The color feature of an image pixel is represented by three values defining its red, green, and blue content. Ohta (1985) analyzed color segmentation and search strategies for a vision system. The usefulness of a color feature is greatly influenced by the structure of the color scenes, such as the characteristics of the object (curved, planar) and its relative position in the scene.

The brightness feature is utilized in many research efforts as the primary feature for detecting objects (Cardenas-Weber, 1991). Brightness is directly available in single band (monochrome) images. This brightness information is the result of two components: the reflectance of the scene objects and the illumination of the scene (Gonzalez and Wintz, 1987). The reflectance of the scene objects can be inferred from the domain knowledge. The relative motion of the robot, the large variability of object types (surface reflectance), their relationships (occlusion and hindrance), and changes in the illumination conditions (sunlight and clouds) affect the sensed image data (Hebert, et al., 1989).

Texture refers to patterns of brightness. Pratt, Faugeras and Gagalowics (1980) develop a statistical model for textures analysis; Schachter, Rosenfeld and Davis (1978) propose a random mosaic model for textures; and Rosenfeld and Lipkin (1970) and Zucker (1976b) develop a syntactic model for textures. Descriptors for texture analysis are the image correlation coefficients and the image busyness. The correlation coefficients define the gradients in different directions

and at different pixel distances. The image busyness defines the brightness gradients in the horizontal and vertical directions in the local neighborhood of one pixel (Rosenfeld and Kak, 1982).

Another feature characterizing a region or object is its shape. The most known shape analysis methods are: Fourier descriptors, Hough transform, object matching, curvature-based methods, mathematical morphology, and shape-from-x approaches. The Fourier descriptors use a series expansion of the object borders. The coefficients of the series indicate the basic shape of the image objects. The Hough transform method (Duda and Hart, 1972; Ballard, 1981) detects specific curves on an image from their functional equations. The matching method searches for regions of shape and size similar to reference objects. The curvature-based methods generate a table of primitive characteristics (and singular points) of the object borders to extract information on their shape. This method combines border following routines with shape analysis strategies, such as chain code (Ballard and Brown, 1982; Rosenfeld and Kak, 1982). The curvature of the object and its shape are estimated from the chain code of its borders.

Mathematical morphology proposes an algebra for visual information (Serra, 1982, 1988; Haralick, Sternberg and Zhuang, 1987). Morphological image analysis systems are used for feature extraction (Chu and Delp, 1989), shape decomposition (Song and Delp, 1990; Maragos and Schafer, 1986, 1987; Pitas and Venetsanopoulos, 1990), and non-linear filtering (Schonfeld and Goutsias, 1991). A major limitation of the theory is the non-linearity of the signal operations and the lack of analytical tools (Maragos and Ziff, 1990).

Horn (1986) and Kender (1988) develop complex shape detection strategies. These strategies, known as shape-from-x strategies, render three dimensional shape information. Horn (1986) pioneered the shape analysis in images with the evaluation of shape from the image shading. Kender (1988) evaluated the shape from a texture analysis of the image.

The main problem of shape analysis methods based in the object contours is the generation of the borders in noisy images with multiple objects. Under these conditions, multiple alternative object borders and noise paths occur (Sha'ashua, 1988). The Hough transform method (Duda and Hart, 1972; Ballard, 1981) and the saliency map method (Sha'ashua and Ullmann, 1988) are especially designed for applications with noise and occlusion.

Object identification matches the extracted feature information against objects patterns and reference parameters which define the objects and their environment (Koch and Kashyap, 1986). The result of this match is used to generate hypothesis about the objects in the imaged scene. The results are finally reviewed with evaluation methods to determine their certainty and reliability. These methods are knowledge based and domain dependent.

THEORY

A machine vision system is proposed for determining the three-dimensional melon location, their size, and maturity and to guide the robotic melon harvester in the picking of these melons. A sensing strategy for collecting the field images is devised and an image processing strategy for analyzing the images and extracting the information designed. The sensing strategy involves selecting the image taking strategy - acquisition frequency, desired acquisition conditions (lighting and auxiliary hardware), and the camera control parameters. The image processing strategy defines the methods for preparing the images, extracting the image features, and generating the

desired information. This information must be then converted from the image coordinates to the three-dimensional world of the robotic controller. The performance criteria are the desired accuracy in detecting all melons in the field (complete detection) and the detection correctness which defines the number of false detections and repeated detections in relation to correct detection.

Sensing Strategy

The sensing system acquires data from the area of interest and generates a digitized image frame. This sensing is performed with a stationary camera installed on a mobile platform.

The image acquisition strategy for the stationary optical sensor covers the melon bed in overlapping rectangular image frames. Three regions are defined in the image area: region A_1 , region A_2 , and region A_3 . Figure 1 displays the position of the image frames in the melon bed, the regions covered by the image frame, and their overlap. Regions A_3 and A_1 should overlap a distance of about one to one and a half melons (30 to 40 cm). This fixed overlap allows for centering of the image frame on the melon plant.

The mobile platform of the harvester advances at time $t = t_0$ from the point 0,0 along the first melon bed. At this time, the first image frame for the global sensing has been acquired and the image processing started. The image processing of this image frame provides the melon locations in image frame coordinates (X_{ic} , Y_{ic}), the acquisition time (t_i), and the melon size (R_{ic}) to the robotic controller.

Figure 1. Global acquisition strategy

The image calibration factors, the position of the field frame at the time of image acquisition, and the robot position at the start of a melon harvesting cycle are used to relate the image frame coordinates to the robot coordinates. The image coordinates (X_{ic} , Y_{ic}) and the melon size (R_{ic}) are multiplied by the calibration factors (C_x , C_y) as shown in expression [1], [2], and [3]. The calibrated coordinates (X_{cc} , Y_{cc}) and the calibrated size (R_{cc}) of the melon are generated. The calibration factors are determined with a reference object of a defined size at the beginning of the harvesting operation.

$$X_{cc} = C_x \cdot X_{ic} \quad [1]$$

$$Y_{cc} = C_y \cdot Y_{ic} \quad [2]$$

$$R_{cc} = C_y \cdot R_{ic} \quad [3]$$

The calibrated coordinates X_{cc} and Y_{cc} are still relative to the origin of the image frame. They are temporal values because the vehicle has displaced since the acquisition of the image. The current location of this origin can be computed with the vehicle forward speed and the elapsed time since the image acquisition as shown in expression [4]. This expression defines the origin of the field frame $X_o(t_i)$ at the acquisition time t_i , $X_o(t_j)$ represents the origin position at the harvesting time t_j , and v_v is the vehicle forward speed. The left edge of the row (Y_{le}) remains the reference value for the Y coordinate as shown in expression [5].

$$X_o(t_i) = X_o(t_j) - v_v \cdot (t_j - t_i) \quad [4]$$

$$Y_o(t_i) = Y_{le} \quad [5]$$

The relative position in respect to the manipulator is obtained by comparing this position with the position of the manipulator $X_r(t_j)$ at the time t_j of harvesting. The robot position ($X_r(t_j)$, $Y_r(t_j)$, $Z_r(t_j)$) is defined from the start position (X_{sr} , Y_{sr}) and the start time t_{sr} . The expressions [6] and [7] define this position where v_{rx} and v_{ry} are the X and Y components of the robot speed and v_v is the forward vehicle speed. The Z_r component is shown in the expression [8]. This third coordinate of the robotic manipulator $Z_r(t_j)$ always starts from a fixed height Z_{ref} .

$$X_r(t_j) = X_r(t_{sr}) + (v_{rx} + v_v) \cdot (t_j - t_{sr}) \quad [6]$$

$$Y_r(t_j) = Y_r(t_{sr}) + v_{ry} \cdot (t_j - t_{sr}) \quad [7]$$

$$Z_r(t_j) = Z_{ref} - (Z_r(t_{sr}) - v_{mv} \cdot (t_j - t_{sr})) \quad [8]$$

The operational time of the machine vision system t_s includes the acquisition and digitization of the frame (time t_a), the processing of the frame (time t_p), and the communication of the results to the controller (time t_c) as shown in expression [9]. An upper limit for t_s is the time in

which the robot picks all detected melons in the analyzed area (picking time). The picking time t_h includes displacing the robot arm towards the melon (time t_d , picking and detaching the melon from the vine (time t_p), and transporting it to the storage bin (time t_t) (Edan, 1990). The picking time proportionally increases with the number n_m of detected melons as shown in expression [10]. It is desired to minimize t_s below the picking time and to allow for some slack time. Other processes, such as multiple frame analysis for redundant information, can be performed in this slack time.

$$t_s = (t_d + t_p + t_c) \quad [9]$$

$$t_s < t_h = n_m(t_d + t_p + t_t) \quad [10]$$

The picking time also determines the time interval for displacing the vehicle between picking areas and, thus, the minimum vehicle speed for a fixed image frame size F_s and frame overlap F_o . The vehicle forward speed is given by the expression [11] where v_v is the vehicle forward speed, D the desired displacement between two consecutive image acquisitions, t_v the time for this displacement, F_s the frame size, and F_o the frame overlap.

$$v_v = D / t_v = (F_s - F_o) / \text{MAX}(t_s, t_h) \quad [11]$$

The vehicle must displace the distance D while harvesting the melons in the picking area to obtain the desired frame overlap F_o before the sensing system acquires a new image frame of F_s length. The entire set of ripe melons in the previous image frame must be picked during this displacement. A new frame must be acquired and processed to generate the new list of melon locations. This cycle is repeated until the melon bed has been harvested. Expression [12] is derived from [11] and represents the minimum acquisition, processing, and reporting time.

$$t_s = D / v_v = (F_s - F_o) / v_v \quad [12]$$

Image Processing Strategy

The image processing strategy defines the interaction of the sensing and the processing functions, outlines the organization of the image processing software, and defines the chosen configuration for the prototype software. The strategy is organized in two main functions: the analysis operation and the knowledge directed evaluation. The analysis provides the position of the melons in the area of interest. The knowledge directed evaluation function applies knowledge rules on the output from the analysis and verifies its correctness.

The analysis identifies the melons, their position, and their approximate size. A preprocessing function analyzes the image brightness and detects any images with illumination problems. A combined brightness and texture algorithm is used to detect the melons. The brightness algorithms use the reflectance difference of the scene objects to discriminate them from their background. The texture algorithm analyzes the vertical and horizontal brightness gradients (spatial texture) of the image. The image is then separated into regions based on the brightness and

texture information. This brightness and texture criteria may not provide sufficient information in the case of partially occluded melons. Shape analysis was considered for achieving an adequate accuracy by identifying the ellipsoidal and circular shaped melons. A low level shape analysis was used to evaluate the convexity of the objects in an image row.

The analysis is performed in three steps. Initially, the image frame is enhanced, a segmented version generated, and the image parameters extracted. In the second step, a hypothesis is generated on the image rows containing melons and, thus, on the X coordinates of melons. In the third step, the neighborhood of these X coordinates is analyzed, their features compared to the melon pattern, and the center of the matching regions defined to be the potential image coordinates of melons. These three processing steps are implemented in the preprocessing, global, and local modules of the initial prototype. These modules operate on the entire image (preprocessing and global) or on the neighborhood of an image row (local). The width and the height of the segmented region define the sizes of the potential melons. The analysis returns a melon list with the melons in the area, their (X_{ic}, Y_{ic}) image coordinates, their approximate size (R_{ic}) , the parameter values of the image from which the information was obtained (global parameters), and the parameter values from the identified melon areas (local parameters). Figures 2, 3, and 4 show the original image, the results of the segmentation process, and the graphic output of the analysis.

In this analysis, the image information has been implicitly used in the algorithm: object brightness, average object size, object area, object shape, distance between objects, and distance between plants. However, the disparities of the information (inconsistencies and contradicting values) must be recognized to avoid detection errors. A generic, algorithmic work frame lacks the control and the flexibility for such task. This algorithmic work frame is designed for multi-purpose application and cannot adapt to the domain or the specific situation.

Figure 2. Original image

Figure 3. Segmented image

Figure 4. Detected melons and preliminary size

A knowledge-based processing strategy improves the detection correctness of the initial prototype by evaluating its output with domain knowledge. Knowledge rules are applied which utilize the preprocessing, global, and local parameters to evaluate the detected melon occurrences. The knowledge rules can represent and utilize information about the application domain and the specific situation which cannot be represented and utilized in the generic image processing algorithms. The values characterizing melons under different conditions of brightness and occlusion are determined from the domain knowledge.

The objective is to organize the information in a non-redundant and consistent view of the scene. The noise and redundant detections should be identified and eliminated. The resulting information provides the basis for efficiently scheduling and executing the picking operation.

The knowledge rules implement the domain knowledge about the melons and the sensing operation. They are applied on the melon parameters to separate the melon occurrences from the noise detections. The knowledge rules assign the occurrences into their respective categories: melon, noise, and redundant occurrences. These knowledge rules are organized into three knowledge bases to take advantage of the specific knowledge type (global, merging, and local) and to reinforce the application processing of the rules in the desired sequence from global to local. The search for noise and multiple detection occurrences is narrowed down in stages by identifying noise detections, merging, and finally re-evaluating the remaining undefined detection occurrences. Figure 5 shows the structure of the software.

The global knowledge base compares the detected objects size (width, height, radius) with the reference values for maximum, minimum, and average melon size. This knowledge base identifies occurrences with a large probability of representing melons or noise based on the parameter configurations and their expected values. Large and fully visible melons are indicated by a small number of objects and a large object size. The noise occurrences are identified by the opposite criteria than the visible melons. These noise occurrences are eliminated. The occurrences that do not match the rules of the global knowledge base remain undefined occurrences.

Figure 5. Structure of the knowledge directed software

The next step is to apply merging rules that use object distance and size to eliminate multiple occurrences of the same melon. Only the undefined and the melon occurrences are subject to the merging rules. The merging eliminates the effects of a leaf or vine dividing the melon visually into two parts or of a difference in brightness in different regions of the melon. The merging is based on the vertical and horizontal separation between every pair of occurrences and a comparison to the respective length (vertical) and width (horizontal) of the pair of occurrences. The merging formula is shown in expression [13] where $|X_1 - X_2|$ is the horizontal distance and $|Y_1 - Y_2|$ the vertical distance between the center coordinates of the two occurrences. H_1 , H_2 , W_1 , and W_2 are the height and width data of the two detection occurrences. Figure 6 explains the relationships.

$$\text{Merge if } \{ |X_1 - X_2| < (H_1 + H_2)/2 \}$$

$$\text{OR. } \{ |Y_1 - Y_2| < (W_1 + W_2)/2 \} \quad [13]$$

After the merging of two or more occurrences, new global and local parameters must be calculated from the values of the merged occurrences. Most values are averaged sums as shown in expression [14] where P_{new} is the parameter value of the occurrence after merging, P_1 to P_n are the parameter values for the merged occurrences, and n the number of merged occurrences. The width and the height are the exception. They are computed to include the distance between the object centers and their sizes as shown in expressions [15] and [16] where H_{new} and W_{new} are the height and the width of the resulting occurrence, X_{max} and X_{min} the largest and the smallest horizontal center coordinate of the merged occurrences, Y_{max} and Y_{min} the largest and the smallest vertical center coordinate of the merged occurrences, W_1 to W_n and H_1 to H_n the width and height values of the merged occurrences, and n the number of merged occurrences. The merging rules are applied recursively for the set of detection occurrences in an image.

Figure 6. Schematic explanation of the merging

$$P_{new} = (P_1 + P_2 + P_n)/n \quad [14]$$

$$H_{new} = (X_{max} - X_{min}) + (H_1 + H_2 + H_n)/n \quad [15]$$

$$W_{new} = (Y_{max} - Y_{min}) + (W_1 + W_2 + W_n)/n \quad [16]$$

In the third knowledge base, the processing ratios and the local parameters are used to reevaluate the remaining unidentified detections. The merging tends to increase the significance of some parameters (width, height, and radius) and, thus, facilitates the evaluation. The processing ratios combine several parameters and represent the interdependency between these parameters. They are used to evaluate the accuracy of the segmentation and determine whether an overthresholding or underthresholding has affected the object detection. An underthresholded image is prone to noise and multiple occurrences and an overthresholded image to undetected melons. This knowledge base distinguishes between images of uniform and non-uniform reflectance before recognizing melons and noise based on the local parameter values.

PROCEDURES

Experimental Setup

The design of the system configuration focused on two aspects: the design of the acquisition system and the selection of the processing methods. Field images were acquired under different conditions. The sensing configuration was refined during the experimentation. The hardware and the software were tested with field images. The experimental setup is shown in Figure 7. It consists of the video manipulation module (1), the acquisition module (2), the processing and displaying module (3), and the support module (4). The numbers identify the modules in the Figure.

The video manipulation module consists of an RCA¹ CCD camera with auto-iris and automatic focusing and a VHS format video cassette recorder. Images of the melon beds were captured from a moving tractor and recorded in the video cassette recorder. Auxiliary hardware, such as a leaf blower fan and a generator used in the field experiment, support the acquisition. In an operational robotic melon harvester the video cassette recorder will be replaced by an on-line real-time communication line to the acquisition module.

¹Reference to a commercial product is for information purposes only, and does not imply an endorsement by the authors or Purdue University.

Figure 7. Experimental setup

The acquisition module consists of an IBM AT with the Data Translation card for digitization (DT2851), DT-Iris software, and a Sony video monitor. The images are retrieved from the video cassette recorder, displayed in the video monitor, digitized with the DT2851 card of the IBM AT and temporarily stored on disk for further processing.

A Silicon Graphics 4D Personal Iris workstation in its standard configuration was used as the processing module. The image is converted to Silicone Graphics format, processed, and displayed. In the operational melon harvester, the conversion will not be necessary because the images are digitized and immediately processed in one system.

Field Data Acquisition

The CCD camera, the video cassette recorder, a leaf blower, and a generator are mounted on a special support frame which is installed on the left side of a tractor. The lateral setup was required to avoid destroying the vines because the axial span between the tractor wheels was smaller than the width of the melon bed. Figure 8 shows the tractor with the installed hardware.

Two different series of field data were obtained. The initial image data were collected on the O'Neal farm near Lafayette, Indiana during 1988 and in the Jordan valley (Israel) during December 1988. The main image data were collected at the Purdue South-West Experimental Station in Vincennes, Indiana during the harvesting season of 1989.

The initial images show melon beds at different resolutions. The videos were taken from a mobile vehicle at a constant speed of 11.11 cm/sec (0.25 miles/hr). Different orientations and positions of the camera were used.

Figure 8. Tractor with hardware

The main data were acquired in three intervals: during the early season, during the mid-season and during the late season of the summer 1989. The data consist of images of twenty-four rows of Superstar (4) and All-star (20) melons. The videos were taped under different illumination conditions.

During the early and the mid-season data acquisition, different views of the field were tested with different field of views of the camera, with different camera angles and positions and with different fan speeds. The hardware setup and the acquisition conditions -- the camera positions and angles, the field of view, the height of the support frame, the power of the leaf blower and the power requirements -- were optimized with the image sets acquired in these periods. The final camera setup includes angles of 0 degrees pitch and ± 30 degrees yaw on three different positions of the camera. These camera positions simulate the view of three different cameras: a left side, a center and a right side camera. The field of view for these camera positions is the width of the entire row. The camera has the standard output of commercial vision systems with a 4:3 aspect ratio of the television format. The support frame was suspended at a height of 170 cm.

Image Digitization

The proposed digitization strategy of setting an automatic acquisition mechanism at equidistances was tested by analyzing the spatial distribution of the melons in the field. In the analyzed field, the melon plants were spaced at 105 cm intervals. The melons cluster in the center area around the plants. The areas between the plants and towards the sides of the bed have a low melon density. The distance between the plants determines the spatial distribution of the melons in the growing bed. Synchronizing the image digitization with the inter-plant distance would allow for centering the image region with the expected melon cluster.

The separating of the melon bed in overlapping regions was simulated by acquiring a continuous stream of images from the melon bed. The vehicle moved at a speed of 11.11 cm/sec along the bed with the camera and the video cassette recoder continuously acquiring images. These image sequences were afterwards displayed in the television monitor and digitized at fixed time intervals with the acquisition hardware.

The fixed time interval between digitizing the images simulates the acquisition at equidistances in the melon bed. The time interval is configured to obtain consecutive and overlapping images of the entire bed length. The desired 80 cm of displacement between image frames is obtained at a vehicle forward speed of 11.11 cm/sec and a time interval of 7.2 seconds between image acquisitions. Standard video rates (30 frames/sec) or higher may be used in the image acquisition which results in an inter-frame distance of 3.7 mm or less. Therefore, the time interval between digitizing images is only limited by the forward speed of the vehicle. A combined digitizing and writing speed of 1 frame every 7 seconds would provide the desired image overlap of 80 cm.

The image frames are digitized in an 8-bit format with or without neighborhood pixel averaging. This averaging process -- available in the digitization DT-Iris routine -- produces a single pixel from a given number of pixel neighbors. It smooths the differences between pixels and reduces the movement effect from the views.

The high speed oscillation of leaves created by the blowing fan resulted in blurred digital images. This effect is a consequence of the two interlaced fields in the standard video frame. Each field generates either the even or the odd lines of an image and their combined effect constitutes a video frame. The movement of the leaves is captured at a slightly different position by each of the frame fields resulting in a blurred digital image. An electronic shutter in the camera would solve this problem.

Two sets of image data were generated for the testing of the software prototype. All images were videotaped in the field and digitized with the IBM AT to a resolution of 512 by 480. The images were manually analyzed and the number of melons, their size and the XY frame coordinated were identified. This manual analysis established the information for evaluating the performance of the software. Melons too small to be mature were not considered.

Eighty images were digitized from the first image set. Three field of view levels were considered: small (covering one melon -- of approximately 60 cm^2), medium (covering only the full row width -- with approximately 120 cm^2), and large (covering areas of the field larger than 200 cm^2). The images with large field of view had a significant amount of occlusions and shadows which degraded the performance of the prototype and were therefore discarded. All digitized images contained melons since the main objective was to determine the algorithm's efficiency in detecting and measuring melons. This set was utilized to identify the best image acquisition conditions.

The second image set consists of 100 images, each image containing one or more melons. The images have a medium field of view covering the whole row width. The fan was used to expose the fruit and minimize the leaf occlusion. The fan blew on the melon plants at a 20° angle with respect to the vines. The effect was a continuous movement of the upper leaves,

making the melons visible. The lighting conditions changed from very sunny to cloudy and dark in the images. This tested the operational limits of the software prototype.

A third image set of 86 field images was digitized from the field data set acquired under improved conditions -- medium field of view, camera axis parallel to the melon bed, 170 cm height support, and the leaf blower. The image set included images with and without melons. These images represented a sample from one melon bed in the field. The melon bed contained 5 successive rows in a straight line. The digitized images were in a sequence. Eight blurred images due to strong leaf vibration were discarded since they would have been avoided if a shutter camera would have been used. The remaining 78 images represent a worst case study because the illumination varies between extreme lighting and strong shadows.

Software Testing

The software was tested with the second and the third image sets. The melons in the image set were manually counted and their image coordinates determined. These manual counts serve as the evaluation criterion for the software performance. Four types of object occurrences were defined for the evaluation of this and the following experiments: detected melons, undetected melons, multiple detections, and noise melons. The detection failures (undetected melons) were caused mainly by an inadequately selected threshold (over or underthresholded images) or by contradicting information from the processing. Noise detections (false detections) occurred whenever a strong reflection of a leaf or the sandy soil was interpreted to be a melon. Multiple detections (redundant occurrences) occurred whenever the same melon appeared in the object list more than once. This can occur if the melons were partially occluded (leaves, vines) or had small shadows).

The performance of the initial prototype was tested with different control parameter configurations. The threshold value, the global brightness, the texture row length, the local brightness, the texture column length, and the object separation were manipulated. The results were compared with the data from the manual evaluation. The parameter values were progressively increased and the impacts on the detection accuracy (number of detected melons) and detection correctness (amount of noise and redundant detections) measured. The objective was to identify a balance between the detection accuracy and the detection correctness. An initial configuration was experimentally chosen which represents a balanced configuration of the parameters for the detection accuracy and the detection correctness. The software prototype with this parameter configuration was tested in a first experiment. A relaxed parameter configuration maximizing the number of detected melons was tested in a second experiment. Both experiments were performed with the second image set.

The knowledge directed prototype was then applied to the image set in a third experiment. The objective was to compare the performance of the initial prototype with the performance of a software which combines the initial prototype with a knowledge evaluation function. The three knowledge bases of the knowledge directed prototype were sequentially applied on the output that had been generated in the second experiment of the initial prototype. The first knowledge base tests the detection occurrences against predefined patterns and boundary values. The occurrences are classified in three categories: melon occurrence, noise occurrence, and undefined occurrence. The undefined occurrences could not be assigned to the melon or the

noise category because their parameters were not sufficiently significant to make a knowledge based decision. The melon occurrences and the undefined occurrences are then processed with the merging knowledge base. The occurrences that fulfill the merging criteria are merged into one occurrences and the parameters for this new occurrence are computed from the parameters of the two merged occurrences. If a melon occurrence was merged with an undefined occurrence, the resulting occurrence was classified to be a melon occurrence. If two undefined occurrences were merged, the resulting occurrence was classified to be an undefined occurrence. The remaining undefined occurrences were then processed in the third knowledge base. In a fourth experiment, the fourth image set of 100 images was tested to validate the performance of the knowledge directed prototype.

RESULTS AND EVALUATION

Statistical Measures

The prototype's image analysis accuracy is represented by a percentile statistic on correctly evaluated images. An accuracy of 100% means that all melons were correctly detected and an accuracy of 0% means that no melons were properly detected in an image with melons. In images with more than one melon, the accuracy is determined with expression [17] where N_d is the number of detected melons and N is the number of melons in the image. Detected noise occurrences were not considered in this image statistics since a noise detection would need to be represented with negative percentages.

$$\text{accuracy } [\%] = (N_d * 100)/N \quad [17]$$

A second statistic describes the detection accuracy and the detection correctness of the prototype. The detection accuracy defines the number of detected and the number of undetected melons. The statistic distinguishes three types of detections: the correctly detected melons (compared to the coordinates from the manual identification), multiple detections of melons (one melon is detected several times with different coordinates), and noise detections (objects are erroneously identified to be melons). The number of noise and multiple detections define the detection correctness. The detections are compared to the results from the manual evaluation. For images with no melons, the accuracy was set to be 100% even if noise occurrences were detected.

Initial Prototype

In the first experiment, the final version of the initial prototype was tested with an image sequence of 78 images. The image set represents one melon bed containing images with and without melons. The image set contains 91 melons of which 71 were detected representing a detection accuracy of 78%. A total of 246 occurrences were detected with the majority (120) being multiple detections of the 71 detected melons. The other 55 occurrences represent noise detections. Twenty melons were not detected at all. Tables 1 and 2 summarize the results of the first experiment. Table 1 presents the image statistics and Table 2 shows the detection statistics.

Table 1. Occurrence statistics for the first experiment

STATISTICS	TYPE	NUMBER
Detection Accuracy	Melons (manual)	91
	Undetected	20
	Detected	71
Detection Correctness	Occurrences	246
	Correct	71
	Redundant	120
	Noise	55

Table 2. Image statistics for the first experiment

No. of IMAGES	ACCURACY
62	100%
6	66%
6	50%
0	33%
4	0%
78	88.5%

The second experiment was conducted to test the adaptivity of the initial prototype with the image set from the first experiment. The impact of selecting different processing parameter values on the analysis results was investigated. The selection of the threshold has a strong effect on the melon detection. A small change in the parameter value resulted in a considerable change in the number of melon and noise detections. The brightness variability evaluates three conditions: the smoothness of the object texture, the minimum brightness of the object, and the object size. This parameter was reduced from 80% to 75%. This allows for less smooth, longer, and less bright objects to be identified as melons. The effect was less evident than the threshold effect but generated additional melon and noise detections. The influence of the object length parameter was mainly observed on the local level. Many detection errors were caused by a small variation of this parameter but the number of melon detections was not increased. The change of the neighborhood size to values below 10% resulted in a decrease of melon detections and increases above 10% had no impact on the detection.

The results of changing the brightness variability parameter also confirmed a field observation. Before relaxing the brightness variability, melons were not detected if they were separated more than the average distance from the melon cluster and had a lower brightness than the melons in the melon cluster. After relaxing the parameter, the number of detections increased. The hypothesis on the brightness differences between clustered and distant melons was confirmed in most samples. The close melons had a lower variance in brightness as compared to the melons which are separated by a larger distance.

In the final configuration, the brightness and texture constraints were changed from 80% to 75% and the threshold level for low reflectance images was reduced by 10% as compared to the initial configuration. The detection accuracy increased from 78% to 89%. Formerly undetected melons, particularly those with low brightness were now detected. A further relaxation of the parameters generated additional noise and multiple detection occurrences but did not increase the number of detected melons.

Tables 3 and 4 summarize the results of the second experiment. A total of 380 occurrences was detected and 81 of 91 melons detected. The number of noise melons increased from 55 to 118 and the number of multiple detections from 120 to 181 as compared to the first experiment. In its final parameter configuration, 89% of the melons in the image set were detected.

Table 3. Occurrence statistic for the second experiment

STATISTICS	TYPE	NUMBER
Detection Accuracy	Melons (manual)	91
	Undetected	10
	Detected	81
Detection Correctness	Occurrences	380
	Correct	81
	Redundant	181
	Noise	118

Table 4. Image statistics for the second experiment

No. of IMAGES	ACCURACY
68	100%
3	66%
4	50%
0	33%
3	0%
78	92.3%

The central problem of the initial prototype is the high number of multiple and noise detections which result in a high number of picking cycles relative to the number of picked melons. In the final configuration, 118 noise detections and 181 multiple detections were detected along with the 81 melon detections. The manipulator would frequently displace trying to reach non-existent (noise) or previously harvested melons (multiple detections). This high number of manipulator actions (380) required for harvesting a smaller number (81) of melons would not be tolerable for an operational robotic melon harvester. It was decided to apply domain knowledge and attempt a processing solution as opposed to acquiring additional images. The processing approach promises a higher efficiency as opposed to acquiring and processing an additional image.

Knowledge Directed Prototype

In the third experiment, the knowledge directed prototype was applied on the results of the initial prototype. The image set of 78 images was selected for this experiment and the output of the final configuration of the initial prototype used as input to the first knowledge base. The 118 noise occurrences from the initial prototype were reduced to 11 noise occurrences which represents a reduction of 91%. The number of multiple occurrences was reduced from 130 to 11 which represents a reduction of 91.5%. Six occurrences could not be classified as melon or noise. One was a melon occurrence (multiple detection) and five were noise occurrences.

The negative consequence was a reduction of the detection rate from 89% to 81.3% as a result of incorrectly classified melons as noise or merging different melons into one occurrence (fusion error). One melon was lost in the global knowledge base, four melons were lost by fusion error in the merging knowledge base, and two melons were lost in the local knowledge base. The melon in the global knowledge base was lost due to an overthresholded image. The melons lost during merging were small (occluded) melons in the neighborhood of visible, large melons. The losses in the local knowledge base were due to segmentation errors.

Table 5 shows the results of applying the three knowledge bases in the third experiment. The results of the manual identification, shown in the first column, are the reference values. The first data column of Table 5 presents the results for the initial prototype with relaxed parameters (second experiment). The last three data columns show the performance of the knowledge directed prototype (third experiment). Table 6 shows the image statistic for the initial prototype (second experiment) and for the knowledge directed prototype (third experiment). The fourth experiment with the knowledge directed prototype verified the results of the third experiment. The knowledge-based evaluation was validated by the results of this experiment. The results are shown in Table 7. The number of occurrences was reduced from 421 to 128 and the error rate reduced from 44% to 14% by the identification of 109 from the initial 140 noise occurrences with the knowledge directed prototype. The rate of multiple occurrences was also reduced to 2% since only 3 redundant occurrences remained from the initial 166. The negative consequence was a reduction of the detection accuracy to 83.5% (from 86.5%) because of melon misclassification as noise or fusion error. Two melons were lost in the global knowledge base, one melon was lost by fusion error in the merging knowledge base, and one melon was lost in the local knowledge base. Thirteen occurrences were not classified as melon or noise. Table 8 shows the image statistics for this data set.

Table 5. Results of the third experiment

5th Experim.	MANUAL	GLOBAL KB	MERGING KB	LOCAL KB
Melons	91 (100%)	91 (100%)	91 (100%)	91 (100%)
Detected	81 (89%)	80 (88%)	76 (83.5%)	74 (81.3%)
Undetected	10 (11%)	11 (12%)	15 (16.5%)	17 (18.7%)
Lost/Error	0 (0%)	1 (1%)	4 (4%)	2 (2%)
Occurrence	380 (100%)	380 (100%)	181 (100%)	181 (100%)
Noise	0	50 (13%)	50 (28%)	79 (44%)
Undefined	118+262	156 (41%)	89 (49%)	6 (3%)
Melon Occurr.	0	174 (46%)	42 (23%)	96 (53%)
Melon Occurr.	---	174 (100%)	42 (100%)	96 (100%)
Correct	---	37 (21%)	34 (81%)	74 (77%)
Error	---	7 (4%)	2 (5%)	11 (11.5%)
Redundant	---	130 (75%)	6 (14%)	11 (11.5%)

Table 6. Image statistics

ACCURACY	GLOBAL [# images]	KNOWLEDGE [# images]
100%	69	62
67%	3	6
50%	4	6
33%	0	0
0%	3	4
AVERAGE	92.3%	88.5%

Table 7. Results of the fourth experiment with the knowledge directed prototype

6th Experim.	MANUAL	GLOBAL KB	MERGING KB	LOCAL KB
Melons	133 (100%)	133 (100%)	133 (100%)	133 (100%)
Detected	115 (86.5%)	113 (85%)	112 (84%)	111 (83.5%)
Undetected	18 (13.5%)	20 (15%)	21 (16%)	22 (16.5%)
Error	0 (0%)	2 (1.5%)	1 (0.75%)	1 (0.75%)
Occurrence	421 (100%)	421 (100%)	250 (100%)	250 (100%)
Noise	0 (0%)	68 (16%)	68 (27%)	109 (44%)
Undefined	421 (100%)	250 (59%)	140 (56%)	13 (5%)
Melon Occurr.	0 (0%)	103 (25%)	42 (17%)	128 (51%)
Melon Occurr.	---	103 (100%)	42 (100%)	128 (100%)
Correct	---	26 (25%)	27 (62.3%)	107 (84%)
Error	---	26 (25%)	14 (33.3%)	18 (14%)
Redundant	---	51 (50%)	1 (4.4%)	3 (2%)

Table 8. Image statistics for second image set

ACCURACY	KNOWLEDGE [# images]
100%	81
67%	1
50%	7
33%	1
0%	10
85.5%	AVERAGE

The data show a considerable improvement after the application of the knowledge. With a few exceptions (2 to 3% of the initial detections) the detection occurrences can be correctly classified as being melons or noise occurrences. Most of the multiple detections of a melon can be merged into a single occurrence.

The result of the knowledge based processing is represented in the picking success rate shown in Table 9 for the first image set and Table 10 for the second image set. For the first image set, the picking success would improve from 29% to 73%. This means that the robot will harvest 74 of 91 melons in the row with 102 instead of 380 harvesting cycles. For the second image set, the number of picking cycles would be reduced from 421 to 128.

The loss of four melons in the first image set (one melon in the second image set) by fusion error can be disregarded. This error is expected to be resolved in a goal-oriented sensing and processing cycle during the harvesting process. This goal-oriented sensing and processing would be performed by an eye-in-hand camera installed in the robotic gripper. The sensing and processing of this function would guide the manipulator approach and would acquire short range images for reevaluation of the detection. This analysis should render the fact that two melons are adjacently located. The final detection rate for the first image set would then be 85.3 and 84.2% for the second image set respectively.

Table 9. Picking success rates for the first image set

EXPERIMENT	OCCURRENCE	MELON	SUCCESS RATE	FAIL RATE
3rd	246	71	29	71
4th	380	81	21	79
KB1	330	80	24	76
KB2	131	76	58	42
KB3	102	74	73	27

Table 10. Picking success rates for the second image set

EXPERIMENT	OCCURRENCE	MELON	SUCCESS RATE	FAIL RATE
Global Analysis	421	115	27	73
KB1	353	113	32	68
KB2	182	112	62	38
KB3	141	111	79	21

CONCLUSIONS

The presented work reports the development of a machine vision-based sensing system for a robotic melon harvester. A conceptual model for the sensing system was developed and applied to the robotic melon harvesting task. The focus of the work was on the image acquisition, the processing tasks, and the knowledge-based evaluation.

This initial prototype was tested with an image set of 78 images. These images represent the sequential view of a melon bedding row and were taken under varying illumination conditions. The detection accuracy of this prototype was 89% with the final parameter configuration. This high detection rate showed the feasibility of the proposed approach. The disadvantage was the high rate of multiple and false detections. This is not tolerable for a robotic system which operates in real-time.

The knowledge directed prototype was added to identify and eliminate false detections and merge multiple detections. This knowledge directed prototype provided melon information with an 85.3% detection accuracy, an error of 9.3% (noise detections), and a multiple-detection rate of again 9%. The knowledge rules eliminate about 90% of the noise and multiple detections at the cost of losing correct melon detections.

A final experiment was performed with this prototype to validate the results and test its performance with a second image set. An accuracy of 84.2% was achieved in this experiment. Only 1% of the melon detections was lost by error of the knowledge rules in this experiment.

DISTANCE TO MELONS

Distance to objects in an image can be established by comparing the change in the area of the object within the image while approaching it. If the vector of motion is co-linear with the lens axis, and the travel distance is known, the distance to the object may be calculated. In this section the error associated with this measurement, and the effects of relevant parameters are analyzed.

Assumptions

- Pinhole Lens (No lens distortion)
- No lighting based errors in edge-detection
- No computation-truncation errors
- No noise
- Objects are positioned at centre of image screen

Notation (see Figure 9)

- ϕ — View angle for lens
- S — Object size
- n — number of pixels in row/column of imaging screen
- h_i — Lens to object distance for snapshot i
- α_i — angle of view occupied by object for snapshot i
- p_i — Image size in pixels, from snapshot i

Figure 9. Nomenclature for Triangulation

Computation of Object Location

Given two snapshots of an object centered with relation to the centre of screen, and the distance between these snapshots, $\Delta h = h_1 - h_2$

We define $a_i = 2 \cdot \tan(\alpha_i/2)$

since

$$\alpha_i = 2 \cdot \tan^{-1}\left(\frac{p_i}{n} \cdot \tan(\phi/2)\right) \quad [18]$$

then

$$\alpha_i = 4 \cdot \frac{p_i}{n} \cdot \tan(\phi/2) \quad [19]$$

From the geometry of the problem we have

$$S = a_i h_i \quad [20]$$

so:

$$a_1 h_1 = a_2 h_2 \quad [21]$$

introducing

$$h_1 = \Delta h + h_2 \quad [22]$$

we obtain the objective of our computation, h_2 :

$$h_2 = \frac{\Delta h a_1}{a_2 - a_1} \quad [23]$$

or alternatively,

$$h_2 = \frac{\Delta h p_i}{p_2 - p_1} \quad [24]$$

where

$$p = \frac{S n}{2 \cdot h \cdot \tan(\phi/2)} \quad [25]$$

or

$$\frac{p}{n} = \frac{S}{2 \cdot h \cdot \tan(\phi/2)} \quad [26]$$

Error Analysis

Propagation through multiplication

$$\epsilon_A \cdot B = \epsilon_A \cdot B + \epsilon_B \cdot A + \epsilon_A \cdot \epsilon_B \approx \epsilon_A \cdot B + \epsilon_B \cdot A \quad [27]$$

Propagation through division

$$\epsilon_{A/B} = \frac{\epsilon_B \cdot A - \epsilon_A \cdot B}{(B + \epsilon_B)B} \quad [28]$$

Resolution Error

Due to limited pixel resolution any edge may be shifted one pixel in any direction.

$$p' = p + \epsilon_p = p \pm 1 \quad [29]$$

thus $|\epsilon_p| = 1$

Positioning Error

Due to lack of accuracy of vehicle/manipulator motion

$$\Delta h' = \Delta h + \epsilon_h \quad [30]$$

Error in $p_1/(p_2 - p_1)$

We denote the desired result as: $p_{1,2}$, i.e. $p_{1,2} = p_1/(p_2 - p_1)$ whereas the calculated value is denoted as $p'_{1,2}$.

therefore:

$$\epsilon_{p_{1,2}} = p'_{1,2} - p_{1,2} \quad [31]$$

Following Equation [28] we get:

$$\epsilon_{1,2} = \frac{\epsilon_{p_2 - p_1} \cdot p_1 - \epsilon_{p_1} \cdot (p_2 - p_1)}{(p_2 - p_1 + \epsilon_{p_2 - p_1}) \cdot (p_2 - p_1)} \quad [32]$$

After deriving the above expression for the four possible combinations of $\epsilon_{p_1} = \pm 1$ and $\epsilon_{p_2} = \pm 1$ we conclude that the largest error occurs when $\epsilon_{p_1} = -\epsilon_{p_2}$.

Assuming the individual errors are indeed of opposite sign:

$$\epsilon_{p_{1,2}} = \frac{p_1 + p_2}{(p_2 - p_1 - 2)(p_2 - p_1)} \quad [33]$$

Note that the singularity point at

$$p_2 - p_1 = 0 \quad [34]$$

and at

$$p_2 - p_1 = 2 = |\epsilon_{p_2} - \epsilon_{p_1}| \quad [35]$$

may actually occur in the entire surroundings of

$$|p_2 - p_1| \leq |\epsilon_{p_2} - \epsilon_{p_1}| \quad [36]$$

since the roundoff error due to limited resolution may have any value within this range.

In such cases the resulting measured distance is invalid since it is prone to extremely large errors.

Total Error in h_2

$$\epsilon_{h_2} = |\epsilon_{p_{1,2}}| \cdot \Delta h + |\epsilon_h| \cdot p_1 / (p_2 - p_1) \quad [37]$$

Numerical examples

The implications of the above analysis are demonstrated using typical values. In Table 11 the resulting errors using the following values for the problem's parameters are shown.

- $\phi = 19.6$ degrees.
- $\epsilon_h = 0.1$ metres.
- $n = 128, 256, 512$ pixels.
- $S = 0.15, 0.3$ metres.
- $h_1 = 3.0, 2.0, 1.5, 1.0$ metres.
- $\Delta h = 0.1, 0.25, 0.5, 0.75$ metres.

The table entries err1 and err2 are the first and second components in the calculation of the total error (see Equation [37]).

In cases that $p_2 - p_1$ is less than 4, the remark **err1 appears after the result.

In cases that the image size in pixels, is larger than the screen size, the remark **err2 appears after the result.

Discussion

The main conclusion from these examples is that when the difference in the number of pixels ($p_2 - p_1$) covered by the image is small, (e.g. 2-3) the resolution error is of the same order of magnitude as the measurement, causing disastrous results. This is due to the character of Equations [31] - [36].

The best results are obtained when $p_2 \gg p_1$. This may be achieved by:

- Increasing Δh
- Increasing resolution
- Decreasing ϕ (using a lens with a longer focal length)
- Increasing S (growing larger fruit)

Table 11. Accuracy of triangulation estimate of distance

n	S	h1	h2	Dh	p1	p2	p2-p1	err1	err2	error	
128	0.15	3.00	2.90	0.10	18.39	19.02	0.63	4.320	0.290	4.610	**err1
128	0.15	3.00	2.75	0.25	18.39	20.06	1.67	17.525	0.110	17.635	**err1
128	0.15	3.00	2.50	0.50	18.39	22.07	3.68	3.277	0.050	3.327	**err1
128	0.15	3.00	2.25	0.75	18.39	24.52	6.13	1.271	0.030	1.301	
128	0.15	2.00	1.90	0.10	27.59	29.04	1.45	7.116	0.190	7.306	**err1
128	0.15	2.00	1.75	0.25	27.59	31.53	3.94	1.932	0.070	2.002	**err1
128	0.15	2.00	1.50	0.50	27.59	36.78	9.20	0.486	0.030	0.516	
128	0.15	2.00	1.25	0.75	27.59	44.14	16.55	0.223	0.017	0.240	
128	0.15	1.50	1.40	0.10	36.78	39.41	2.63	4.623	0.140	4.763	**err1
128	0.15	1.50	1.25	0.25	36.78	44.14	7.36	0.513	0.050	0.563	
128	0.15	1.50	1.00	0.50	36.78	55.17	18.39	0.153	0.020	0.173	
128	0.15	1.50	0.75	0.75	36.78	73.56	36.78	0.065	0.010	0.075	
128	0.15	1.00	0.90	0.10	55.17	61.30	6.13	0.460	0.090	0.550	
128	0.15	1.00	0.75	0.25	55.17	73.56	18.39	0.107	0.030	0.137	
128	0.15	1.00	0.50	0.50	55.17	110.34	55.17	0.028	0.010	0.038	
128	0.15	1.00	0.25	0.75	55.17	220.69	165.52	0.008	0.003	0.011	**err2
128	0.30	3.00	2.90	0.10	36.78	38.05	1.27	8.064	0.290	8.354	**err1
128	0.30	3.00	2.75	0.25	36.78	40.13	3.34	4.279	0.110	4.389	**err1
128	0.30	3.00	2.50	0.50	36.78	44.14	7.36	1.027	0.050	1.077	
128	0.30	3.00	2.25	0.75	36.78	49.04	12.26	0.512	0.030	0.542	
128	0.30	2.00	1.90	0.10	55.17	58.08	2.90	4.315	0.190	4.505	**err1
128	0.30	2.00	1.75	0.25	55.17	63.05	7.88	0.638	0.070	0.708	
128	0.30	2.00	1.50	0.50	55.17	73.56	18.39	0.214	0.030	0.244	
128	0.30	2.00	1.25	0.75	55.17	88.28	33.10	0.104	0.017	0.121	
128	0.30	1.50	1.40	0.10	73.56	78.82	5.25	0.891	0.140	1.031	
128	0.30	1.50	1.25	0.25	73.56	88.28	14.71	0.216	0.050	0.266	
128	0.30	1.50	1.00	0.50	73.56	110.34	36.78	0.072	0.020	0.092	
128	0.30	1.50	0.75	0.75	73.56	147.13	73.56	0.031	0.010	0.041	**err2
128	0.30	1.00	0.90	0.10	110.34	122.61	12.26	0.185	0.090	0.275	
128	0.30	1.00	0.75	0.25	110.34	147.13	36.78	0.050	0.030	0.080	**err2
128	0.30	1.00	0.50	0.50	110.34	220.69	110.34	0.014	0.010	0.024	**err2
128	0.30	1.00	0.25	0.75	110.34	441.38	331.03	0.004	0.003	0.007	**err2

(Table 11 continued)

n	S	h1	h2	Dh	p1	p2	p2-p1	err1	err2	error	
256	0.15	3.00	2.90	0.10	36.78	38.05	1.27	8.064	0.290	8.354	**err1
256	0.15	3.00	2.75	0.25	36.78	40.13	3.34	4.279	0.110	4.389	**err1
256	0.15	3.00	2.50	0.50	36.78	44.14	7.36	1.027	0.050	1.077	
256	0.15	3.00	2.25	0.75	36.78	49.04	12.26	0.512	0.030	0.542	
256	0.15	2.00	1.90	0.10	55.17	58.08	2.90	4.315	0.190	4.505	**err1
256	0.15	2.00	1.75	0.25	55.17	63.05	7.88	0.638	0.070	0.708	
256	0.15	2.00	1.50	0.50	55.17	73.56	18.39	0.214	0.030	0.244	
256	0.15	2.00	1.25	0.75	55.17	88.28	33.10	0.104	0.017	0.121	
256	0.15	1.50	1.40	0.10	73.56	78.82	5.25	0.891	0.140	1.031	
256	0.15	1.50	1.25	0.25	73.56	88.28	14.71	0.216	0.050	0.266	
256	0.15	1.50	1.00	0.50	73.56	110.34	36.78	0.072	0.020	0.092	
256	0.15	1.50	0.75	0.75	73.56	147.13	73.56	0.031	0.010	0.041	
256	0.15	1.00	0.90	0.10	110.34	122.61	12.26	0.185	0.090	0.275	
256	0.15	1.00	0.75	0.25	110.34	147.13	36.78	0.050	0.030	0.080	
256	0.15	1.00	0.50	0.50	110.34	220.69	110.34	0.014	0.010	0.024	
256	0.15	1.00	0.25	0.75	110.34	441.38	331.03	0.004	0.003	0.007	**err2
256	0.30	3.00	2.90	0.10	73.56	76.10	2.54	10.994	0.290	11.284	**err1
256	0.30	3.00	2.75	0.25	73.56	80.25	6.69	1.227	0.110	1.337	
256	0.30	3.00	2.50	0.50	73.56	88.28	14.71	0.433	0.050	0.483	
256	0.30	3.00	2.25	0.75	73.56	98.08	24.52	0.233	0.030	0.263	
256	0.30	2.00	1.90	0.10	110.34	116.15	5.81	1.024	0.190	1.214	
256	0.30	2.00	1.75	0.25	110.34	126.11	15.76	0.272	0.070	0.342	
256	0.30	2.00	1.50	0.50	110.34	147.13	36.78	0.101	0.030	0.131	
256	0.30	2.00	1.25	0.75	110.34	176.55	66.21	0.051	0.017	0.067	
256	0.30	1.50	1.40	0.10	147.13	157.64	10.51	0.341	0.140	0.481	
256	0.30	1.50	1.25	0.25	147.13	176.55	29.43	0.100	0.050	0.150	
256	0.30	1.50	1.00	0.50	147.13	220.69	73.56	0.035	0.020	0.055	
256	0.30	1.50	0.75	0.75	147.13	294.25	147.13	0.016	0.010	0.026	**err2
256	0.30	1.00	0.90	0.10	220.69	245.21	24.52	0.084	0.090	0.174	
256	0.30	1.00	0.75	0.25	220.69	294.25	73.56	0.024	0.030	0.054	**err2
256	0.30	1.00	0.50	0.50	220.69	441.38	220.69	0.007	0.010	0.017	**err2
256	0.30	1.00	0.25	0.75	220.69	882.76	662.07	0.002	0.003	0.005	**err2

(Table 11 continued)

n	s	h1	h2	Dh	p1	p2	p2-p1	err1	err2	error	
512	0.15	3.00	2.90	0.10	73.56	76.10	2.54	10.994	0.290	11.284	**err1
512	0.15	3.00	2.75	0.25	73.56	80.25	6.69	1.227	0.110	1.337	
512	0.15	3.00	2.50	0.50	73.56	88.28	14.71	0.433	0.050	0.483	
512	0.15	3.00	2.25	0.75	73.56	98.08	24.52	0.233	0.030	0.263	
512	0.15	2.00	1.90	0.10	110.34	116.15	5.81	1.024	0.190	1.214	
512	0.15	2.00	1.75	0.25	110.34	126.11	15.76	0.272	0.070	0.342	
512	0.15	2.00	1.50	0.50	110.34	147.13	36.78	0.101	0.030	0.131	
512	0.15	2.00	1.25	0.75	110.34	176.55	66.21	0.051	0.017	0.067	
512	0.15	1.50	1.40	0.10	147.13	157.64	10.51	0.341	0.140	0.481	
512	0.15	1.50	1.25	0.25	147.13	176.55	29.43	0.100	0.050	0.150	
512	0.15	1.50	1.00	0.50	147.13	220.69	73.56	0.035	0.020	0.055	
512	0.15	1.50	0.75	0.75	147.13	294.25	147.13	0.016	0.010	0.026	
512	0.15	1.00	0.90	0.10	220.69	245.21	24.52	0.084	0.090	0.174	
512	0.15	1.00	0.75	0.25	220.69	294.25	73.56	0.024	0.030	0.054	
512	0.15	1.00	0.50	0.50	220.69	441.38	220.69	0.007	0.010	0.017	
512	0.15	1.00	0.25	0.75	220.69	882.76	662.07	0.002	0.003	0.005	**err2
512	0.30	3.00	2.90	0.10	147.13	152.20	5.07	1.920	0.290	2.210	
512	0.30	3.00	2.75	0.25	147.13	160.50	13.38	0.505	0.110	0.615	
512	0.30	3.00	2.50	0.50	147.13	176.55	29.43	0.201	0.050	0.251	
512	0.30	3.00	2.25	0.75	147.13	196.17	49.04	0.112	0.030	0.142	
512	0.30	2.00	1.90	0.10	220.69	232.30	11.62	0.406	0.190	0.596	
512	0.30	2.00	1.75	0.25	220.69	252.22	31.53	0.127	0.070	0.197	
512	0.30	2.00	1.50	0.50	220.69	294.25	73.56	0.049	0.030	0.079	
512	0.30	2.00	1.25	0.75	220.69	353.10	132.41	0.025	0.017	0.042	
512	0.30	1.50	1.40	0.10	294.25	315.27	21.02	0.152	0.140	0.292	
512	0.30	1.50	1.25	0.25	294.25	353.10	58.85	0.048	0.050	0.098	
512	0.30	1.50	1.00	0.50	294.25	441.38	147.13	0.017	0.020	0.037	
512	0.30	1.50	0.75	0.75	294.25	588.51	294.25	0.008	0.010	0.018	**err2
512	0.30	1.00	0.90	0.10	441.38	490.42	49.04	0.040	0.090	0.130	
512	0.30	1.00	0.75	0.25	441.38	588.51	147.13	0.012	0.030	0.042	**err2
512	0.30	1.00	0.50	0.50	441.38	882.76	441.38	0.003	0.010	0.013	**err2
512	0.30	1.00	0.25	0.75	441.38	1765.52	1324.14	0.001	0.003	0.004	**err2

STEREOSCOPIC VISION

Stereoscopic vision (Kak, 1985; Tanaka and Kak, 1990) computes the difference in the view alignment between two laterally displaced images to obtain the depth information. Figure 10 shows the conditions and parameters of the stereoscopic approach.

The physical locations of the cameras are represented by the vectors C_L and C_R . The image elements for the object point P are at (I_L, J_L) and (I_R, J_R) in the left and right cameras. The horizontal distance between camera positions (baseline vector B) is give by $C_L - C_R$. The distances from the object to the left and the right camera are defined by two vectors rD_L and rD_R , where r is the scalar of the distance value and D_L and D_R the direction vectors (Kak, 1985). The vectors D_L and D_R are non-unit vectors but only their directions are of interest. The scalar distance r is shown in the expression [2.18] where H_r is a calibration vector and J_r one coordinate of P on the image plane. $r = (B * H_r) / (D_L \cdot H_r \cdot J_r \cdot D_L \cdot a_r)$.

$$r = (B * H_r) / (D_L \cdot H_r \cdot J_r \cdot D_L \cdot a_r) \quad [38]$$

Equation [38] holds only when the baseline vector B is parallel to the horizontal plane and the optical axis (a_r and a_l) of the cameras are parallel to each other.

Tanaka and Kak (1990) developed an stereoscopic vision algorithm based on the Marr Poggio Grimson approach which uses dot stereograms and the images for the detection and matching of zero crossings. In this algorithm, the scene contours are matched in both images and the difference in view alignment between the images is identified using the zero crossings in both images. The zero crossings are the contours where the convolution operator passes through zero. The crossing occurrences in both images are then matched and the depth value computed from

Figure 10. Stereoscopic vision (from Kak, 1985)

that information. Tanaka and Kak (1990) use a knowledge base opportunistic framework to match the linear features in the two images. The matching is performed at different levels with different resolutions of the images. This algorithm generates a range image of the objects in the scene. This resulting range image contains only three dimensional shape information, and hence it is independent from the shadows present in the original stereo views of the scene.

A structured light beam can be projected on the objects to facilitate the matching between the object points of a stereo image pair (Kak, 1985). This reduces the number of required matches to the number of pattern components which are projected by the light beam. Different patterns can be projected, such as a single dot, a two dimensional grid, or lines. Near-real-time operation of the stereo method can be obtained with the structured lighting method. However, the implementation of this method in a non-industrial environment is not feasible with the current technology.

TRIANGULATION

A triangulation based technique was proposed by Weckler and Kranzler (1989) that relates the focusing of the camera to the object distance. The frequency content of the image is used to evaluate the level of visual detail. The optimal focus of the camera is determined whenever the high frequency content in the image is maximum and it is used to compute the object distance. The accuracy of this technique depends on the background complexity. Therefore, the complexity of the background must be kept at a minimum to achieve efficiency.

Another triangulation method compares the image sizes of the same object at two different camera distances. The axis defined by object and camera is the direction of displacement between taking the images. The object size increases as its distance to the camera decreases. The triangulation technique computes the depth value from the two dimensional information of the object size in two images.

This technique is also applied in combination with sensor hardware to compute depth (Everett, 1989). In the triangulation strategy, the object size in pixels is computed in both images (p_1 and p_2) and the camera displacement d is known. The subscript $_1$ refers to the far view image and the subscript $_2$ refers to the near view image. The depth from the near position of the camera to the object is given by Equation [39].

$$\text{depth} = \frac{P_2 * d}{P_1 - P_2} \quad [39]$$

The results of the triangulation strategy for a centered melon are shown in Tables 12 and 13. Table 12 shows the calculated distances for every pair of consecutive images at different vertical displacements of the camera support. The reference height is 120 cm for all measurements. The first column represents the vertical distance between the reference height and the camera position when acquiring the first image. The first row represents the vertical camera distance between the reference height and the camera position when acquiring the second image. The

Table 12. Triangulation results for a centered object

Distance b/images	0 cm	20 cm	40 cm	60 cm
0 cm	NA	120 cm	84 cm	54 cm
20 cm	----	NA	76 cm	50 cm
40 cm	----	----	NA	46 cm
60 cm	----	----	----	NA

Table 13. Error percentage for a centered object

Distance b/images	0 cm	20 cm	40 cm	60 cm
0 cm	NA	16.67%	5.00%	10.00%
20 cm	----	NA	5.00%	16.67%
40 cm	----	----	NA	23.33%
60 cm	----	----	----	NA

main diagonal of the table corresponds to images taken at the same distances and is, therefore, not relevant for triangulation. The entries in the lower part of the diagonal would represent the data for taking the second image taken at a larger distance from the melon than the first image. This situation is not considered in triangulation because the robotic manipulator always approaches towards the melon. The entries in the upper part of the diagonal are the distance values computed with Equation [39]. The computed distance value is compared with the camera height which is measured as the difference between 120 cm (reference height) and the position of the camera at the time of acquiring the second image.

Table 13 compares the calculated values shown in Table 12 with the measured camera heights. The camera heights are 120 cm, 100 cm, 80 cm, and 60 cm for the columns labeled 0 cm, 20 cm, 40 cm, and 60 cm, respectively. The distances to the melons are generated with an average accuracy of 14%. This represents an error of 7 cm in a measurement of 50 cm with reported values of 43 to 57 cm.

Table 14 shows the triangulation results for objects which are displaced at 5 cm from the center of the image frame and Table 15 shows the error percentage for these computations. The triangulation error in the results for the non-centered object are much higher than for the centered melon. This result shows that a high precision is desirable for determining the melon coordinates in the global analysis prototype. The problem is the calculation of the melon size from a skewed position. The perceived melon size changes unproportionally in dependency of the vertical distance to the melon if the melon is not centered. This increases the triangulation error.

Table 14. Triangulation results for a non-centered object

Distance b/images	0 cm	20 cm	40 cm	60 cm
0 cm	NA	70 cm	53 cm	48 cm
20 cm	----	NA	55 cm	53 cm
40 cm	----	----	NA	70 cm
60 cm	----	----	----	NA

Table 15. Error percentage for non-centered object

Distance b/images	0 cm	20 cm	40 cm	60 cm
0 cm	NA	41.67%	33.75%	20.00%
20 cm	----	NA	31.25%	11.67%
40 cm	----	----	NA	16.67%
60 cm	----	----	----	NA

Two conclusions can be derived from this experiment. First, the accuracy of the method is relatively low and depends on the precision of the melon location in the global analysis prototype. At each camera displacement, the centering must be corrected and a new reference frame taken. This slows the process. The triangulation process can only provide acceptable data if the melon is centered in the image. Second, the best measurements are generated with the following experimental conditions:

1. The object appears in the center of the image frame, and
2. The minimum error in the measurement is obtained with a camera displacement of 40 cm.

EVALUATION OF ULTRASONIC SENSOR FOR MELON RANGING

One of the techniques considered for measuring distance to melons is using ultrasonic measuring. An ultrasonic sensor was evaluated in the lab and in the field for ability to measure distances to fruit accurately and reliably under some of the working conditions expected for the melon harvesting robot. The experiments included comparison of the sonic reflectance of the various surfaces and objects encountered in the field, the sensitivity of the output to noise and wind blast, ranging obstructed fruit, and the available frequency of data. The results were generally positive, but a low rate of measurements (1 Hz.) was necessary to make the output stabilize under certain operating conditions.

INSTRUMENTATION

The system used was the M-4000 Ultrasonic Measurement and Control System manufactured by Massa Products Corporation. The analog output (0-10 Vdc) was used. The ultrasonic module tested was the M-410/215 which has a frequency of operation of 215 kHz, and a range between 3 and 24 inches. The beam angle is 10° and features no side lobes. No compensation for temperature was used in this experiment, though the M-450 compensation assembly may be added to the system.

OBJECTIVES

The main objective was to evaluate distance measurements for the different elements that will be encountered in the field, and which may have different reflectance properties. These are: Melons (Superstar and Cantaloupes), leaves, earth, mud and black plastic mulch which is used in growing practise. Distance to hard shiny surfaces were measured as a reference.

Other objectives were: To observe interference of operation by wind blasts at various angles; To evaluate the possibility of ranging errors occurring when fruit is partially obstructed by leaves; To determine possible measurement rates; and To evaluate general performance in field conditions.

PROCEDURE

Trial experiments were first held in the lab which consisted of range measurements from various heights to a polished concrete floor and two cantaloupes of different size/shape. A fan was introduced, at various angles with respect to the sonic beam, parallel to the beam, and at 45° and 75°. The frequency at which the measurement was taken was altered between 1 and 150 Hz, to evaluate the possibility of delayed signals interfering with more recent ones.

A second set of trials was held in the field, and these included measuring distance to vine with fruit from different distances and positions. In these trials the fruit was rendered unobstructed by leaves, though surrounding leaves caused turbulence. Again a fan was introduced at various angles with respect to the sonic beam, and the frequency of measurement was altered between the extreme values available. Distance to fruit directly obstructed by leaves was measured, while applying fan at the various angles.

RESULTS

1. *Linearity and consistency:* Figure 11 shows the analog output is linear with distance. For the indoor measurements linear regression of fruit and floor readings gave R^2 of 0.9998 and 0.9995 respectively (6 measurements). Linear regression of the joint data returned a R^2 of 0.9997. For the outdoor measurements of fruit combined with the other natural elements the value of $R^2 = 0.9992$ for 30 measurements (Figure 12).
2. *Disturbances due to constant wind blasts:* When the ultra-sonic module was transmitting at high repetition rates and the fan was operated the output values were extremely noisy. Nevertheless the dominant output was only slightly higher (5-10%) than the expected one. This leads to an assumption that through some filtering approximate results might be obtained. The changing of angles of wind blasts had no further noticeable effect on the outputs, though this limited experiment may not have been thorough enough to identify these effects.
3. *Surface orientations:* In certain cases no output readings could be obtained. This occurred when the sensor was aimed at the borderline of the fruit and the fan was operated too.
4. *Leaf obstructions:* Since leaves are detected as readily as fruit they certainly present an obstacle (see Figure 14). Moreover their tendency to flutter often created instability in the output readings.

CONCLUSIONS

The general conclusion is that ultrasonic measurement is a feasible technique for the environmental conditions expected for the melon harvesting robot. The audio-reflectance properties of all the elements encountered in the field are indistinguishable for our purpose. A practical problem is with the attainable frequency of measurement which seems to be quite slow for our purposes (1 Hz.). Nevertheless some intermediate value between 20-50 Hz. (which was not available on the system evaluated) might give satisfactory results. The stability problem seem surmountable since the dominant values observed by the operators had a very good fit with the expected results, and appropriate filtering of the signal is expected to render good results. Leaf obstruction may probably be overcome by changing the direction of wind blasts and using a rule-based procedure to return the distance to fruit. Another option could be to cooperate with a vision sensor which could observe obstacles.

Figure 11: Calibration of sensor for ranging to melons vs. reference surface.

Figure 12: Linear regression for measurement of natural elements found in the field.

Figure 13: Calibration of fruit ranging vs. other natural elements found in the field.

Figure 14: Ranging of fruit partially occluded by leaves.

DEPTH RANGING USING STRUCTURED LIGHT

This section describes a real-time data acquisition system for measuring height values of a cross-section of a scene, which may be used for creating elevation profiles across rows of melon plants. Based on these elevation profiles melons may be located and geometrical properties of the melon may be measured.

The method to be used is based on triangulation calculations of the spatial geometry created by a focused light source and projected on to an image plane. By extending the triangulation concept to 3-D; using a laser line projector and a two dimensional camera (as opposed to a linear scan camera) entire cross-sections will be obtained from a single frame.

OBJECTIVES

The objective of this section is to describe a data acquisition system which collects elevation profiles of cross-sections of a scene.

In order to achieve this the following steps are necessary:

- Model the spatial geometry of the problem.
- Select the necessary hardware.
- Determine the algorithms required for obtaining the pixel coordinates of the line light source projected onto the scene.
- Develop a calibration procedure for translation of values expressed in the image frame into the world frame.
- Calculate the accuracy and resolution of the system as a function of the hardware parameters and the software algorithms.

PERFORMANCE REQUIREMENTS

The performance requirements stem from the intended use of this data acquisition system. This system is to be used for locating and identifying or measuring geometrical properties of melons in a field harvesting operation. The underlying assumption is that the vision system will be mounted on to a frame travelling through the melon field at a known speed. Since obstructions and inaccuracies may occur we need to create cross-sections at distances that are several times smaller than the size of the smallest expected melon.

The notation used for the system parameters are:

V_f	: Frame speed [mm/sec]
$M_s^{\min}$	: Minimal melon size [mm]
$M_s^{\max}$	: Maximal melon size [mm]
N_s	: Minimum number of samples per fruit
D_s	: The distance between cross-section samples [mm]
T_s	: The time between cross-section samples [sec]

- Speed: The system must be able to grab a frame from the scene and create the elevation map of a cross section several times a second.

By regarding the values for N_s , V_f and M_s which result in the most stringent requirements we obtain an upper bound on the speed requirements.

If $N_s = 3$, $V_f = 0.5$, and $M_s = 90$ we have $D_s = M_s/N_s = 30$ mm. From this we can find T_s :

$$T_s = D_s/V_f = 30/500 = 0.06 \text{ sec.} \quad [40]$$

This means that we must process and analyze close to 17 images per second. This constraint is close to the limit on number of frames that can be obtained each second on standard video equipment (30 images per second), however great improvement may be achieved by using dedicated and relatively cheap hardware (Huang, et al., 1988). Commercial products such as the *MQ Laser Analog Sensor* by Aromat (Aromat Corporation, 1988) are available too, but are in the \$2,000-4,000 price range, and do not enable changing the range of operation.

- F.O.V.: A field of view of 120 cm is required if a single vision system is to scan the entire width of a bed. If two systems are used side by side then 80 cm will be sufficient to create an overlap that will enable the major part of every melon to be observed (assuming $M_x^{\max} = 300$ mm) by at least one of the systems.
- Accuracy: An initial estimate is that a resolution of 0.1 cm is sufficient for the measured height, in the application being considered. This will have to be re-evaluated on test cases with the actual hardware and algorithms used.

Density: The sufficient density of range readings along a single cross-section is estimated as every 0.5 cm but, again, re-evaluation will be necessary.

TWO-DIMENSIONAL IMPLEMENTATION

Before we describe the geometry of the 3-D ranging system we shall look at a 2-D ranging system designed and implemented in Huang, et al. (1988). Their optical distance measurement system consisted of a laser light source, a lens and a linear photodiode array. The optical arrangement is illustrated in Figure 15.

Figure 15. The optical arrangement of the 2-D laser profiling system

The laser beam is directed normally onto the plane of the surface. The image of the interception point is focused onto a photodiode array by a lens system. The optical axis of the lens is inclined by a small angle to the primary laser beam. Any change in surface elevation moves the position of the image on the diode-array. The amount of movement on the array is thus a function of the surface elevation.

One of the drawbacks of a 2-D implementation for real time acquisition of elevation profiles is that lateral motion is required to create a single cross-section. This is generally time consuming and complicates even further the mechanical design of a robotic melon harvester.

The advantage of the system described in Huang, et al. (1988) is that it is streamlined by use of dedicated hardware. This option could be considered for our 3-D system after development of algorithms and evaluation of performance using standard and more general hardware.

GEOMETRICAL DESCRIPTION OF THE PROBLEM

To obtain closed form analytical expressions describing the spatial optical arrangement we shall assume that we will be using a pinhole camera. In reality we will use a lens camera. The resulting non-linearities may be compensated for by calibration.

The basic spatial structure of the optical arrangement consists of a laser line source which is directed square to the surface. The camera is at an angle to the light source similar to the arrangement shown in Figure 15.

We now discuss how the coordinate vector of a point on an image which the light is incident on is found, given the relations between the camera and the light source and the angle of view of the camera lens. For this discussion we assume ideal optics and a pinhole camera, the main point being to describe the general feasibility of the idea, and to develop analytical tools for selecting the range which the system operates in.

We define a coordinate system O on the camera body. Figure 16 shows the projection of the geometry onto the x - y plane. The distance between a reference point on the camera to the light plane is denoted by L . This is actually the x -coordinate of the light plane relative to the coordinate system O located on the camera. From the i -coordinate of the pixel location in the image plane we can find the planar angle ω which denotes the direction of the corresponding point in the scene:

$$\tan(\omega) = \tan\left(\frac{\Phi}{2}\right) \cdot \left[\frac{1-2i}{m}\right] \quad [41]$$

where m denotes the number of pixels across the image screen, and Φ denotes the view angle associated with the lens' focal length. The view angle is generally not identical for the horizontal and vertical axis of a camera thus care should be exercised in applying the correct one.

From L and ω we can find the y-coordinate of the corresponding point:

$$\begin{aligned} y &= L \cdot \tan(\omega) \\ &= L(1 - \frac{2i}{m}) \cdot \tan(\frac{\Phi}{2}) \end{aligned} \quad [42]$$

Figure 16. Top view of 3-D ranging system

In Figure 17 we show the projection on to the x-z plane. The planar angle ψ can be found from the j -coordinate of the pixel location. For this we require also knowledge of the camera tilt angle - α .

The range of operation of the system *i.e.*, the difference in height between the extreme points that may be measured by the system is defined by L , α and Φ :

$$R = L \cdot \cotan\left(\alpha - \frac{\Phi}{2}\right) \cdot \cotan\left(\alpha + \frac{\Phi}{2}\right) \quad [43]$$

Selecting the System Hardware

The main tasks of the hardware will be to acquire the scene images, detect the peaks in image intensity and then calculate the corresponding heights based on a predetermined transformation or execute a table look-up.

In order to determine the specific hardware requirements some analysis of the problem is required. Given the general geometry of the problem which includes non-linear distortion of the scene by the depth calculation transformation, and possible occlusion problems, the relations between the light source and the camera, and the height from which measurements are taken will be defined. This, in turn, defines the focal length of the camera's lens. From the resolution of the measurement required at the output and the geometry determined above, the camera's resolution may be specified.

When selecting a specific light source the width of its projection on the surface is to be considered, since it affects the measurement resolution.

Figure 17. A side view of the 3-D ranging system

Assuming the detection of the projected light in the image is done by software, the performance required from the vision and computing hardware can be clearly defined.

In practice, though, existing computing and vision hardware will be used. The above analysis will mainly determine if it can be reasonably implemented.

A further requirement from the computing hardware will be to execute further algorithms on the data system to detect some higher-level features from the scene, but this is not within the scope of the project. A rule based approach has been proposed and implemented (Benady, 1990).

THE INPUT DEVICE

The input device to be used is an *RCA CMC020* color video camera (RCA Corporation Consumer Electronics Division, 1986) with autofocus capabilities and a zoom lens (11.5-69 *mm*). The RCA camera has a 2/3 inch format, thus, w - the width of the image sensor, is 8.8 *mm*, and h - the height, is 6.6 *mm*.

The main parameters of interest for this system are the image resolution, spectrum of sensitivity, and the field of view (FOV). No information was readily available for the parameters. Color video cameras typically have a peak response of up to 920 *nm*. The FOV for various focal lengths was found by measurement. The calculations were not precise since the location of the image plane within the camera were unknown.

We denote the maximum object dimensions (in inches) that can be seen from a given distance using a specific focal length by H and W . Measurement results for a distance of 43.75 *inches* are summarized in Table 16.

THE STRUCTURED LIGHT SOURCE

The light source to be used is the 06 *PLL 301* laser line projector (Melles Griot Corporation, 1989) manufactured by *Melles Griot*. This is a low power (3-4 mW) laser diode which uses a cylindrical lens to emit a fan of light. The system includes a low cost power supply, pre-set for the individual laser. The beam width at exit aperture is 1 *mm*. The wavelength of the light is 674.7 *nm* which is in the visible near infra-red. The field angle is 40°.

Table 16. Field of view for RCA CMC020 camera from distance of 43.75 inches

F	W	H	H/W	Φ_w	Φ_h
11.5	29.0	24.0	0.8275	36.67°	30.67°
15	21.5	18.0	0.8372	27.6°	23.25°
30	10.75	9.0	0.8372	14.0°	11.74°
69	5.0	4.2	0.84	6.5°	5.49°

IMAGING AND COMPUTING HARDWARE

The imaging hardware to be used is an Image-Pro vision system mounted on a PC-AT computer. After an image is captured vertical pixel columns must be scanned to find the peak (if any) along them. This can be achieved using the Image-Pro **Inprofcalc()** command.

This will result in a row vector containing values of pixels which the laser light is incident on. The spatial coordinates may be computed directly based on the geometry described above, or a look up table may be used. Again, this look up table may be created by a calibration procedure or by precalculating the results of the trigonometrical formulation.

EXAMPLE - SHAPE DISCRIMINATION

A melon and a flat image of a melon were placed next to each other (see Figure 18). To the eye they both look like true melons. The line of light created by the laser source was projected onto both from above (see Figure 19), resulting in a curved line for the real fruit and a straight line for the melon image. Figure 20 reveals the cause of the difference.

Figure 18. Image of a real melon and a "flat" melon

Figure 19. Laser line projected on the real and “flat” melon

Figure 20. Revealing the cause of the difference

CALIBRATION

There are two main drawbacks to a direct calculation of the elevation values from the pixel location. The first is that the calculated value relies heavily on our ability to measure the geometric relations between the light source and the camera. This is not trivial, and will have to be done whenever any small change in the desired range or other property is made. The second drawback is the difficulty in modelling the nonlinearities introduced by a lens camera.

A procedure for calibration of a structured light and camera device is described in Chen and Kak (1988). This procedure relies on the motion of a robot arm to obtain accurate displacements which are used as inputs to the calibration procedure. This requires motion in a plane and also rotation. This is not viable in the proposed robotic melon harvester and an alternative procedure relying on a template 3-D structure will have to be devised.

RESOLUTION

The resolution of this system is not identical over the image plane due to the nonlinear geometry, and also differs for the horizontal and vertical dimensions of the plane.

The average vertical resolution of the system is approximately $R/2N$, where R is the elevation range obtained from a particular geometry of the optical components, and N is the vertical resolution of the CCD device. The horizontal resolution of the system is approximately $L \tan(\frac{\Phi}{2})/2M$ where M is the horizontal resolution of the device.

SENSOR FUSION

With the maturing of the various fields of research in Robotics, the applications and demand for robots executing increasingly complex tasks may be observed. Together with widening number of domains, in which many factors and uncertainties are to be considered, a greater degree of autonomy is being required from robot systems.

In order to be capable of fulfilling these tasks, robots are being equipped with greater and more sophisticated sensing abilities. Greater on-board processing power is being added to robotic systems to reach the higher levels of intelligence required while keeping decision and execution times relatively fast.

A robot operating in an agricultural environment will be required to successfully, identify plant and fruit locations and develop efficient plans for execution of various types of tasks. The agricultural environment is unstructured, and unfavorable operating conditions may exist. Furthermore, the data is of dynamic nature and previously created mappings have a very limited validity.

The "objects" which are to be manipulated by agricultural robots are generally delicate organisms, and in many cases the results of actions made are irreversible. Improper handling may stunt future development or cause damage which may greatly affect the product quality.

In order to operate in such an uncertain environment advanced robotic applications demand a steady stream of rich and reliable information to be gleaned rapidly from sensors. Unfortunately, no single sensor can guarantee to deliver accurate information all the time. Different types of sensors have different accuracy or reliability parameters, which may depend on their underlying physical principle of operation or on environmental parameters. Also, any sensor signal is inevitably corrupted by noise.

On the other hand different sensors may have sufficiently different characteristics so that the operating limits of one sensor often do not correspond to those of another, suggesting that two or more sensors can be used profitably in combination. Even when sensors are operating within their limits, fusing their measurements can provide a more reliable reading than that provided by any one sensor, since the signal tends to be correlated between sensors, whereas the noise is uncorrelated.

Different sensors also provide different sorts of information which can be used at different levels of planning, apart from the fact that both the sensors and their interpretation programs have time complexities that vary considerably, therefore some strategy has to be formulated to decide when and where each sensor should be deployed.

In order to operate effectively, an intelligent executive controller should be capable of exploiting knowledge of sensors, actuators, environment and of the task to be performed. The executive controller should devise the plan of operation and execute it systematically while dynamically making conclusions about the environment and reevaluating the intermediate targets as more information flows in.

SENSORS AND INTELLIGENT ROBOTICS

Robots must operate in a world which is inherently uncertain. This uncertainty arises in the perception and modelling of environments, in the motion of manipulators and objects, and in the planning and execution of tasks. Robust task execution is accomplished by increasing sensor and manipulator accuracy, using high tolerance and well known objects, and supplying parts in predetermined positions and orientations. However as robot systems are increasingly required to operate in more complex domains where the environment and tasks are uncertain or unknown, then this implicit allowance for uncertainty will be insufficient to ensure efficient operation. Current robotics research has concentrated on using single sensors to extract particular features of an uncertain but known environment. These single-sensor single-algorithm systems are severely limited in their ability to resolve ambiguities, identify spurious information and to detect errors or failure. This is the result of an attempt to make global decisions based on incomplete and under-constrained information.

MULTI-SENSOR ROBOT SYSTEMS

In recent years interest in the development of multi-sensor systems has increased. This is due to a realization of the fundamental limitations of a single source of information. In order for robot systems to achieve a degree of intelligence and autonomy, they must be capable of using many different sensors in an active and dynamic manner.

A multi-sensor robot system comprises many diverse sources of information. The sensors of these systems are characterized by the variety of information they can provide and by the complexity of their operation. The measurements supplied by these sensors are uncertain, partial, occasionally incorrect and often geographically or geometrically incomparable with other sensor views. The quality of the information may depend on the current state of the robot environment and the location or state of the sensor itself. The goal of the robot system is to coordinate the sensors, direct them towards areas of interest and to integrate the resulting observations into a consistent consensus view of the environment, in order to plan and guide the execution of tasks.

We will consider as multi-sensor systems, any system which includes one or more of the following features:

1. Multiple sensor types
2. Single sensors using multiple data-extraction algorithms
3. Single-sensor and single-algorithm used from multiple vantage points

ORGANIZATION OF SENSOR SYSTEMS

A general approach to organization of sensor systems should be able to consider arbitrary sequences of observations made by diverse sensor systems, be able to integrate these observations, use them to coordinate and control sensing devices, and to provide a quantitative means of estimating system performance. To fully utilize the capabilities of many robot sensors we must be able to characterize the dynamic nature of observations; interdependencies between information sources and the effect of sensor state on the observation supplied. Three basic types of information interaction may exist:

- *Complimentary* information, when two disparate sources provide information which together provide a more global view of the environment but which do not overlap by themselves.
- *Competitive* information, when two sources provide the same piece of information and a joint consensus must be reached.
- *Cooperative* information, when one information source must rely on or cooperate with another in order to obtain observations.

MULTI-SENSOR SYSTEM MODELS

Several successful ad-hoc attempts have been made to model sensors and combine disparate data into a single representation. Flynn (1988) and Everett and Flynn (1986) have proposed a rule based approach for integrating sonar and infra-red sensor information as applied to mobile robot map-building. Sonar and infra-red sensors provide a very good complement to each others abilities; infra-red has good angular resolution properties, but poor depth-measuring ability, whereas sonar provides comparatively good depth observations but with poor angular resolution. Flynn devised rules which describe when either sonar or infra-red sensor is to be used to build the map. The two sets of information were never integrated at any single location.

A method for treatment of the systematic error produced by the low resolution data obtained when applying a wide-angle sonar sensor for measuring distance was proposed and implemented by Beckerman and Oblow (1990). The area of interest is described by a rectangular grid, and each pixel may receive one of four possible labels: Occupied, empty, unknown or conflict. By observing the area of interest from a variety of vantage points and inferring the status of each pixel through a set of rules, from different labels obtained throughout the procedure, a navigation map of the environment is obtained. Bixler and Miller (1987) proposed a sensory input system based on the fusion of simple low-resolution vision with directed high-resolution sonar. The vision system locates the direction in which an obstacle lies, and an ultra-sonic rangefinder determines the distance and tries to gain clues (e.g. surface orientation or concavity) about its shape. The latter is based on interpretation of the return signal.

The problem with these rule based approaches to sensor modelling is that they preclude any quantitative analysis of system performance and they are difficult to extend in any consistent, homogeneous fashion.

Many quantitative approaches have been proposed for combining specific types of sensory information into a single world model. Among these are: Integration of vision and touch, for object recognition (Allen, 1988) and for robotic perception (Stanfield, 1988). Ayache and Faugeras (1988) used extended Kalman filtering to combine data obtained from noisy three-dimensional descriptions. Hu and Stockman (1987) combined 3-D data obtained from structured-light range measurements and from image intensity to produce a more reliable scene representation. The main undesirable properties of the structured light approach are the sparse-ness of measurements, and the difficulty in obtaining readings for certain surface orientations. Intensity readings, on the other hand, always exist and can be variegated by changing the ambient lighting. Terzopoulos (1986) demonstrated a variational approach to surface reconstruction by integrating complimentary edge and surface-normal constraints.

None of these aforementioned techniques, though they may be successful applications, approach the general problem of sensor interaction.

LOGICAL SENSORS

Henderson and Shilcrat (1984) and Luo and Henderson (1986) presented a *multisensor kernel system* as a means for multiple sensor integration and data acquisition. Their main objectives were to develop a coherent and efficient treatment of information supplied by many sensors, and to provide a method for efficient reconfiguration of the sensor system in terms of *logical sensors* which map onto physical sensors and computation. A logical sensor specification permits an implementation independent description of the required data. In addition, alternative ways of producing the same data may be defined. Logical sensors are then a means by which to insulate the user from the peculiarities of input devices, which in the general case are physical sensors, and to create and "package" virtual sensors. For example, the kind of data produced by a physical laser range finder could also be produced by two cameras and a stereo program. The similarity of output result is more important to the user than how it was produced. The motivations for logical sensor specification are the benefits of data abstraction and the availability of smart sensors.

The abstract view of a logical sensor is shown in Figure 21. A logical sensor comprises five parts: the name, the characteristic output vector, alternate subnets, each of which is made up of a set of input sources and a computation unit over these sources, a selector whose inputs are the alternate subnets, and finally a control command interpreter. Each of the input sources to the alternate subnets may be in themselves logical sensors, or actual physical sensors. The role of the selector is to detect failure of an alternate, and switch to a different alternate. The control command interpreter takes strings from the control line and produces the required control if at a physical device, or the appropriate set of commands for the logical sensors which provide input data to the current logical sensor.

Figure 21. The organization of a logical sensor

A logical sensor can then be viewed as a network composed of subnetworks which are themselves logical sensors.

Like the rule-based approaches described previously, the logical sensor approach deals more with selection of sensory inputs than their actual integration. Nevertheless, it seems that by replacing the selector by a more sophisticated *integrator* the elegance of abstraction will not be lost. The logical sensor approach should therefore be considered as a suitable option for high-level specification of multisensor systems.

INTEGRATION OF GEOMETRIC SENSOR OBSERVATIONS

In order to make efficient use of sensory information, it is important to model the environment in a manner which explicitly accounts for inherent uncertainties encountered. The environment description should provide a homogeneous description of different observable features, allowing for manipulation and transformation of different descriptions between different sensory systems.

A systematic treatment of multiple geometric sensor observations has been developed by Durrant-Whyte (1986) and will be summarized herein, since robot perception often involves knowledge of geometry or positions.

UNCERTAIN GEOMETRY

All geometric features can be described by a parameter vector $\mathbf{p}$ and a vector function:

$$\mathbf{g}(\mathbf{x}, \mathbf{p}) = 0; \quad \mathbf{x} \subseteq R^n, \quad \mathbf{p} \in R^m \quad [44]$$

This function maps a *region* $\mathbf{x}$ in Euclidean n -space to a *point* $\mathbf{p}$ in parameter space. Each function $\mathbf{g}$ describes a particular type of geometric object; all straight lines, or the family of quadratic surfaces, for example. Each value of $\mathbf{p}$ specifies a particular instance of a geometric object modeled by $\mathbf{g}$. For example, *all* plane surfaces can be represented by the equation;

$$\mathbf{g}(\mathbf{x}, \mathbf{p}) = p_x x + p_z z + 1 = 0 \quad [45]$$

with $\mathbf{x} = [x, y, z]^T$ and $\mathbf{p} = [p_x, p_y, p_z]^T$. A specific plane can be represented as a point $\mathbf{p} \in R^3$ in the parameter space. A sensor can now be considered as observing values of $\mathbf{p}$ in the environment. The uncertainty in each geometric feature observed will be described by a probability density function (p.d.f.) $f_g(\mathbf{p})$, defined on the parameter vector.

WELL-CONDITIONED STOCHASTIC GEOMETRIES

There are generally many different functions of the form of Equation [44] which can be used to describe a single type of geometric feature. There are two conditions required from every functional representation if they are to be considered statistically well-conditioned:

1. A representation must be a one-to-one (bijective) relation between feature and parameter.

2. The parameter vector must change "smoothly" with physical feature motion. This ensures that small changes in feature do not result in abrupt changes in the describing parameter.

TRANSFORMING PROBABILITY OF UNCERTAIN GEOMETRY

The key to manipulating geometric uncertainty is to be able to transform the information or probability density function on a feature available in one form to another form of interest. This ability to transform uncertainty between coordinate systems also allows sensor observations obtained in one coordinate system to be transformed to another sensor's coordinate system so that information may be integrated. Correspondingly, if a sensor is mobile, then these transformations can be used to develop sensor strategies, making decisions as to where a sensor should be placed to best observe a given feature (Durrant-Whyte, 1987).

We need to consider what happens to the parameter vector $\mathbf{p}$ and its associated p.d.f. $f_g(\mathbf{p})$ when they are moved between coordinate systems. If

$$\mathbf{p}' = \mathbf{h}_g(\mathbf{p}) \quad [46]$$

then

$$f'_g(\mathbf{p}') = \frac{1}{|\nabla_{\mathbf{p}} \mathbf{h}|} f(\mathbf{h}^{-1}[\mathbf{p}']) \quad [47]$$

Where $|\nabla_{\mathbf{p}} \mathbf{h}|$ is the determinant of the transform Jacobian. In general if the transformation in Equation [46] is non-linear the calculation in Equation [47] can be very complex. Changing between feature descriptions increases this complexity.

Gaussian Geometry

The central limit theorem states that the continued aggregation of any independent identically distributed random variables (of finite moments) will converge to Gaussian distribution. Therefore, any reasonable policy for modelling and propagating geometric uncertainty will always tend to Gaussianity and can be approximated by such. The following results are valid for Gaussian geometry, where all geometric objects and features are parameterized by a jointly Gaussian random vector. If we assume Gaussianity, then the mean $\hat{\mathbf{p}}$ and variance $\Lambda_{\mathbf{p}}$ are sufficient information to completely define the feature density function.

Often we have partial information on the vectors $\mathbf{p}$ (infinite uncertainty in one or more degrees of freedom), whence the distribution function is improper and Λ is not strictly defined. It is sometimes easier and safer to work with the information matrix $\Sigma = \Lambda^{-1}$, where the diagonal zero elements of Σ imply that no information is available in the associated degree of freedom.

There are a number of important cases in which $\mathbf{h}$ is a linear transform. For example if the vector $\mathbf{p} = [x, y]^T$ describes the location of a mobile robot with respect to some world coordinate system (Brooks, 1984), then a linear homogeneous transformation can relocate the robot as $\mathbf{p}' = \mathbf{T}\mathbf{p}$. Which yields:

$$\hat{\mathbf{p}}' = \mathbf{T}\hat{\mathbf{p}} \quad [48]$$

$$\Lambda^{p'} = \mathbf{T}\Lambda^p\mathbf{T}^T \quad [49]$$

The same applies to all oriented vectors (edges, directions, tangents, etc.). The ease of transformation is a direct consequence of the Gaussian assumption.

In general $\mathbf{h}$ is non-linear (as in the case of relocating six-dimensional location vectors). Because of the complexity it is not viable to propagate exact distributions. An alternative is to approximate the transformed density by:

$$\hat{\mathbf{p}}' = E[\mathbf{p}'] = \mathbf{h}(E[\mathbf{p}]) = \mathbf{h}(\hat{\mathbf{p}}) \quad [50]$$

The distribution variances are transformed as differential changes:

$$\Lambda^{p'} = \mathbf{J}\Lambda^p\mathbf{J}^T \quad [51]$$

where

$$\mathbf{J} \approx \frac{\partial \mathbf{h}}{\partial \mathbf{p}} \quad [52]$$

The transformation between different coordinate systems, such as polar to Cartesian, will require use of this approximation.

If the function $\mathbf{h}$ can be described by a homogeneous transform $\mathbf{A}$ where:

$$\mathbf{A} = \begin{bmatrix} \mathbf{n} & \mathbf{o} & \mathbf{a} & \mathbf{q} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [53]$$

then the Jacobian can be found from the transformation (Paul, 1981) as:

$$\mathbf{J} = \begin{bmatrix} \mathbf{r} & \mathbf{m} \\ 0 & \mathbf{r} \end{bmatrix}, \quad \mathbf{r} = \begin{bmatrix} \mathbf{n}^T \\ \mathbf{o}^T \\ \mathbf{a}^T \end{bmatrix}, \quad \mathbf{m} = \begin{bmatrix} (\mathbf{q} \times \mathbf{n})^T \\ (\mathbf{q} \times \mathbf{o})^T \\ (\mathbf{q} \times \mathbf{a})^T \end{bmatrix} \quad [54]$$

CHANGING FEATURE DESCRIPTIONS

We often need to change the way in which a feature is described. This occurs, for example, if we wish to make inferences about one type of geometric feature from another type, or if we wish to combine sensor observations made of many types of geometric features.

Let $\mathbf{g}_i(\mathbf{x}, \mathbf{p}_i) = 0$ and $\mathbf{g}_j(\mathbf{x}, \mathbf{p}_j) = 0$ describe two physically disparate geometric features with $\mathbf{p}_i \in R^1$ and $\mathbf{p}_j \in R^m$. Suppose these two features are related so that the geometric function

$\mathbf{h}_j^i : R^1 \rightarrow R^m$ maps the parameter vector to that of the other $\mathbf{p}_j = \mathbf{h}_j^i(\mathbf{p}_i)$. In general the function $\mathbf{h}$ is not well defined in a *deterministic* sense. This is because the two related features may have different degrees of freedom $l \neq m$. For example, the observation of an edge implies the existence of two intersecting surfaces. In theory, given the surface equations, the equation for the edge could be derived. Conversely, given the equation of the edge, the equation of either surface can be not be derived. However, if the functions $\mathbf{h}$ are considered in a *stochastic* sense, then this lack of constraint can be represented as partial information and described by placing zero elements in the information matrix corresponding to the indeterminate degrees of freedom of the transformation. In Equation [51] the fact that $\mathbf{p}_j$ and $\mathbf{p}_i$ are of different dimension is accounted for by the dimension of the Jacobian. If the transformation is over constrained Λ_j will be singular. If it is under-constrained then Λ_j^{-1} must be calculated, to provide for infinite information in the unconstrained degrees of freedom.

$$\Lambda^{j-1} = J^T \Lambda_i^{-1} J^{-1} \quad [55]$$

MULTI-OBSERVATION FUSION

All geometric features in the environment are modeled by functions of form $g(x, p) = 0$. We consider a sequence of n observations $\mathbf{z} = \{z_1, \dots, z_n\}$ with $f_i(z_i | \mathbf{p}) \sim N(\mathbf{p}, \Lambda_i)$. That is, we assume that these observations are corrupted by an additive Gaussian noise, then $\pi(\mathbf{p} | \mathbf{z}_1, \dots, \mathbf{z}_n)$ (our belief about $\mathbf{p}$ after making the observations) is

$$\hat{\mathbf{p}}' = \left[\sum_{i=1}^n \Lambda_i^{-1} \right]^{-1} \left[\sum_{i=1}^n \Lambda_i^{-1} z_i \right] \quad [56]$$

and variance-covariance matrix:

$$\Lambda' = \left[\sum_{i=1}^n \Lambda_i^{-1} \right]^{-1} \quad [57]$$

where the maximum a posteriori estimate of $\mathbf{p}$ is the mean $\hat{\mathbf{p}}'$. This estimator, a discrete Kalman filter is attractive because of its computational efficiency. It is possible to extend this result to prior knowledge distribution $\pi(\mathbf{p})$ by defining $\mathbf{z}_0 \equiv \hat{\mathbf{p}}'$ and summing from $i = 0$.

This approach is bound to fail since sensors may occasionally create spurious or erroneous data due to failure in operation or other. A more realistic approach for modeling sensor observations suggested is a *contaminated Gaussian* (Durrant-Whyte, 1988):

$$f_i(z_i | \mathbf{p}) \left[(1 - \epsilon)N(\mathbf{p}, \Lambda_i^1) + \epsilon N(\mathbf{p}, \Lambda_i^2) \right] \quad [58]$$

with $0.05 < \epsilon < 0.1$ and $\Lambda_i^2 \gg \Lambda_i^1$. This indicates that very large errors can occur with a small probability.

SPARSE DATA FUSION

We again assume a sequence of observations $z = \{z_1, \dots, z_n\}$, which are derived from a sensor modeled by a contaminated Gaussian density. We will consider the most general case where both observations on more than one feature in the environment $p = \{p_1, \dots, p_k\}$ and also spurious or poor data sets are included. We need to cluster the z_i so that the outlying observations can be removed from consideration and the remaining z_i be classified and assigned to the appropriate $p_j \in p$. These can then be integrated using Equations [56] and [57].

In order to cluster the observations we need to compare each pair of observations and see if we can explain z_i given the assumedly Gaussian distribution of z_j and vice-versa. If they indeed can accommodate each other we need to establish the consensus value of $\hat{p}$.

The comparison will generally be of the form:

$$F(|z_i - z_j|) \leq T_{ij} \quad [59]$$

Where F is some monotone increasing function and T_{ij} is some positive threshold. This clustering procedure should be a consequent of the preference ordering induced by the observation model $f(\cdot | p)$ on the observation set z . We will require that the posterior distribution $f(p | z_i, z_j)$ calculated from the Bayes rule is unimodal. This can be expressed by the following condition:

$$\frac{1}{2}(z_i - z_j)^T (\Lambda_i + \Lambda_j)^{-1} (z_i - z_j) \leq 1 \quad [60]$$

In such a case we can say that z_i and z_j admit a joint maximum likelihood estimate $\hat{p}$.

The estimate $\hat{p}$ will then correspond to the *unique* maximum of $f(p | z_i, z_j)$:

$$\hat{p} = (\Lambda_i^{-1} + \Lambda_j^{-1})^{-1} (\Lambda_i^{-1} z_i + \Lambda_j^{-1} z_j) \quad [61]$$

Equation [60] can then be used for the clustering procedure. Although it is generally unlikely that we will know the variance-covariance matrices exactly, but the estimates of the Λ_i can act like thresholds; The larger the Λ_i that is used, the more disagreement will be tolerated.

These observation clusters can be built recursively. The largest clusters can be used to provide world model updates while the smaller clusters can be rejected as outlying observations.

COORDINATION AND CONTROL

So far we have concentrated on the problem of integrating diverse geometric observations from different sensors to provide a robust estimate of the location of objects in a known but uncertain environment. These techniques must be extended to the more complex problem of estimating the state of an unknown environment. We must try to make full use of the dynamic capabilities of sensor systems. These dynamic capabilities result from communication of information between different sensors, each taking limited observations of the environment.

THE TEAM DECISION PROBLEM

A multi-sensor system can be regarded as a *team* of decision makers. Each sensor in the team will be taking observations and making local decisions. Together, the sensors must coordinate their activities, guide each other to areas of interest, and ideally to come to some team-consensus view of the environment. The most important observation about a team is that the coordinated activities of the members is more effective than the sum of their individual actions.

The main elements of a multi-sensor system operating as a team are:

- How individual sensors make local decisions based on their own observations.
- How the capabilities and information provided by each sensor are described to other members of the team.
- How information obtained by one sensor is described and communicated to the other sensors.
- How local decisions made by individual sensors can be integrated to provide a team decision and action.

Since we are considering sensors as Bayesian observers, local decisions are made to find an estimate of the observed geometric feature which maximizes likelihood. The communication of information will be in the common language of uncertain geometry. In a multi-Bayesian system of observers we must have a mechanism for describing the combination of information, the differences of opinion amongst individuals, and the formation of coalitions which will guide and modify the team estimates.

The decisions made by each sensor must be evaluated with respect to some measure of preference. This preference ordering, or utility function $u_i(\mathbf{p}_i)$, $\delta_i(\mathbf{z}_i) \in R$. A sensor or team member will be considered *rational* if for each observation $\mathbf{z}_i$ of some prior feature $\mathbf{p} \in \rho_i$ it makes the decision $\delta(\mathbf{z}_i) \in \rho_i$.

Individual rationality alone is not sufficient, since if each sensor is allowed to make it's own decisions regardless of other sensor's opinions, then a group consensus will never be reached. A structure which allows for sensor opinions to be constrained towards a common team objective is required. For this a *team* utility function is defined.

$$U = U(\mathbf{p}, \delta_1, \delta_2, \dots, \delta_n) \quad [62]$$

The role of U is very important in characterizing the structure of the sensor team. If each team member makes the decision δ_i that maximizes the team utility U , regardless of personal preference we obtain what is termed an *altruistic* team structure. In the scenario of a multi-sensor system, it may not always be advisable for single sensors to agree with a team decision, particularly if the sensor has for some reason made a mistake and should not be contributing to a consensus decision.

An alternative formulation that accounts for this is to allow each team member to have a personal utility as well as an interest in the team. A team member may agree to cooperate and be

subject to the team utility, or to disagree and be subject to a personal utility. In such a case a rational team member will agree to cooperate only if it will gain by doing so, when the team utility exceeds its personal utility. This is called an *antagonistic* team. By restricting the definition of "rational" in the forthcoming manner the team decision problem is simplified; Each sensor has a preference on the decisions it can make. In order to have his estimates verified rather than disproved by the group decision, it may have to allow for its decisions to be changed by a small amount so that it can be brought to line with a consistent interpretation of all sensor observations. However, it should not allow its opinions to be changed so much that the group opinion now bears no resemblance to the sensor's original observations. This dichotomy between consensus and disagreement is fundamental to the multi-sensor decision problem.

MULTI-BAYESIAN TEAMS

Multi-Bayesian teams are static teams of estimators that make a team decision on the observed state of the environment. This is a simplification of the more general problem by interpreting group rationality as seeking only the combined group maximum likelihood estimates of the observations. The likelihood function $f_i(\cdot | \mathbf{p})$ is used as the utility function of each individual observer. The team utility is the joint posterior distribution function $F(\mathbf{p} | \mathbf{z}_1, \dots, \mathbf{z}_n)$ after each sensor has made the observation $\mathbf{z}_i$. This framework results in normalized individual and team utilities which simplifies comparisons.

Before formalizing the general multi-Bayesian system we study the simpler case of two scalar homogeneous observers: Consider two team members observing the same scalar variable $p \in \rho$ (a feature in the environment), with observation densities $f_i(\cdot | p)$. Each sensor takes a single observation z_i , which is considered independent and is derived from a Gaussian distribution with mean $\hat{p}$ and variance σ_i . We want to find some consensus estimate of state $\bar{p}$ based on the two observations z_1 and z_2 .

The individual utility function is given by the posterior likelihood;

$$u_i(\bar{p} = \delta_i(z_i), p) = f(p | z_i) \sim N(\hat{p}, \sigma_i) \quad i = 1, 2. \quad [63]$$

The team utility function is given by the joint posterior likelihood;

$$\begin{aligned} U(\bar{p}(z_1, z_2), p) &= F(p | z_1, z_2) \\ &= f_1(p | z_1) f_2(p | z_2) \end{aligned} \quad [64]$$

A team member will be considered *individually* rational if it chooses the estimate $\bar{p} \in \rho$ that maximizes its local posterior density;

$$\bar{p} = \arg \max_{p \in \rho} f_i(p | z_i) \quad i = 1, 2 \quad [65]$$

The team is considered group-rational if the team members choose the estimate $\bar{p} \in \rho$ which maximizes the joint posterior density;

$$\begin{aligned}\bar{p} &= \arg \max_{p \in \rho} F(p | z_1, z_2) \\ &= \arg \max_{p \in \rho} f_1(p | z_1) f_2(p | z_2)\end{aligned}\tag{66}$$

The estimate given by Equation [65] will simply be the unique mode of the (Gaussian) posterior likelihood function. However for the group joint posterior density, given by Equation [66], two possible results may occur;

1. $F(p | z_1, z_2)$ has a unique mode equal to the estimate $\bar{p}$ in Equation [67].
2. $F(p | z_1, z_2)$ is bimodal and no unique group-rational consensus estimate exists.

If $F(p | z_1, z_2)$ has a unique mode it satisfies;

$$\arg \max_{p \in \rho} F(p | z_1, z_2) \geq \arg \max_{p \in \rho} f_i(p | z_i) \quad i = 1, 2.\tag{67}$$

Conversely if $F(p | z_1, z_2)$ is bimodal then

$$\arg \max_{p \in \rho} F(p | z_1, z_2) < \arg \max_{p \in \rho} f_i(p | z_i) \quad i = 1, 2.\tag{68}$$

Each team member can either agree with a team estimate $\bar{p}_t$ and be subject to the team utility

$$F(p = \bar{p}_t | z_1, \cdot),\tag{69}$$

or choose its own estimate $\bar{p}_i$, and be subject to an individual utility

$$f_i(p = \bar{p}_i | z_i),\tag{70}$$

A rational team member will maximize its utility by choosing to either agree or disagree with the team consensus: if a team member's observation satisfies Equation [68] then it will not cooperate with the team estimate. An antagonistic team will thus be rational if it makes the decision;

$$\begin{aligned}\bar{p} &= \delta_i(z_i) \\ &= \arg \max_{p \in \rho} \{f_i(p | z_i), |F(p | z_1, z_2)\} \quad i = 1, 2.\end{aligned}\tag{71}$$

Whether or not individual team members arrive at a consensus team estimate will depend on some measure of how much they disagree, *i.e.*, on $|z_1 - z_2|$. If z_1 and z_2 are close enough then the joint posterior density will be unimodal and satisfy Equation [67] with the consensus estimate given by Equation [65] (see Figure 22). As $|z_1 - z_2|$ increases the joint density becomes flatter and eventually bimodal (Figure 23). At this point no consensus team decision will be reached and the members' individual estimates will satisfy Equation [68].

MAHALANOBIS DISTANCE

The point at which two antagonistic team members will disagree with each other will occur when $f(p | z_1, z_2)$ ceases to be convex in the interval $[z_1, z_2]$. This can be characterized by the following requirement on p , σ_1 and σ_2 :

$$\left[\sigma_1^{-2}(p - z_1) + \sigma_2^{-2}(p - z_2) \right]^2 \left[\sigma_1^{-2} + \sigma_2^{-2} \right]^{-1} \leq 1 \quad [72]$$

This is actually a normalized weighted sum, a scalar equivalent to the Kalman gain matrix. Accordingly, if a p which satisfies Equation [72] can be found then the consensus $\bar{p}$ which maximizes F will be given by:

$$\bar{p} = \frac{(\sigma_1^{-2} z_1 + \sigma_2^{-2} z_2)}{(\sigma_1^{-2} + \sigma_2^{-2})} \quad [73]$$

Equivalently, the existence of a consensus can be found by considering the convexity of the set;

$$f(p) = [f_1(p | z_1), f_2(p | z_2)]^T \in R^2 \quad [74]$$

By substituting Equation [73] into Equation [72], we can write;

$$(z_1 - z_2)(\sigma_1^2 + \sigma_2^2)^{-1}(z_1 - z_2) = D_{12}(z_1, z_2) \quad [75]$$

where $D_{12} \leq 1$. Figure 24 shows a plot of the set $f(p)$ for two Bayesian observers. The disagreement measure $D_{12} = D_{12}(z_1, z_2)$ is called the *Mahalanobis distance*. Figure 24 shows the relation between the concavity of the set $f(p)$ and the value of D_{12} .

Figure 22. Two Bayesian observers with joint posterior likelihood indicating agreement

Figure 23. Two Bayesian observers with joint posterior likelihood indicating disagreement

Figure 24. The space of preferences for a two-Bayesian team

GENERAL MULTI-BAYESIAN TEAMS

The concept presented above can be extended to a team of n Bayesians taking observations of the environment. Consider each taking observations of different geometric features which are hypothesized to be related to a common geometric object. Two possible approaches can be envisioned. One is a centralistic philosophy in which some sensor manager suggests hypotheses and seeks support from sensor observations. This approach would be the appropriate for use in a known environment, model-based system. The other approach is of a decentralized decision making procedure in which a sensor suggests values for the common object, the combination of which provide a consensus. This approach is more appropriate in situations where the environment is unknown.

As in the two-Bayesian system the preference order on team decisions will be defined as the joint posterior density function $F(\mathbf{p} | \cdot)$. Assuming the sensors are making independent observations we can write:

$$F(\mathbf{p} | \delta_1(\mathbf{z}_1), \dots, \delta_n(\mathbf{z}_n)) = \prod_{i=1}^n f_i(\mathbf{p} | \delta_i(\mathbf{z}_i)) \quad [76]$$

Here too the observations $\mathbf{z}_i$ will only provide a consensus estimate for $\mathbf{p}$ if $F(\mathbf{p} | \cdot)$ is convex in all its arguments. This can be verified by the Hessian of F being positive definite.

If the $n-1$ dimensional volume is concave along the j^{th} axis then the j^{th} sensor observation will be unable to agree with other sensor opinions; its individual maximum likelihood estimate is then preferred to the team decision. Thus if the volume is concave in one or more directions, then coalitions supporting different decisions will be supported.

The process of bringing together and combining information is called an opinion pool. In a multi-Bayesian setting the pooling is accomplished by using likelihood as a measure of both individual and team preferences. In order to resolve the group decision problem in cases that unanimous agreement is non-existent we need to define a team preference ordering, or team utility in which individual decision makers can compare their utilities using a common metric. The goal of the sensor system should be to provide a decision or interpretation which makes the opinion space convex in as many directions as possible. This allows individual sensors to offer opinions in a common dimensionless framework and to disagree or guide the sensor system to a correct interpretation. The way in which this will be achieved depends on the choice of the team utility. Therefore the team utility should be a function of the individual utilities.

$$U = U(u_1, \dots, u_n) = U(\mathbf{u}) \quad [77]$$

Conversely from the system's perspective the role of U is to provide incentives to get the team members to agree with each other. The following properties are thus required from U :

- Unanimity.
- No dictator.
- Indifference.

The unanimity condition ensures that the team decision moves in the same direction as individual preferences. The dictator condition requires that no one sensor can dominate the group decision at expense of all others. The indifference condition states that if a sensor has no opinion then it shouldn't affect the team decision.

The multi-Bayesian formulation of the team decision problem results in a specific case of the generalized Nash product (Durrant-Whyte, 1988).

In practice one can't guarantee that disparate sensors are indeed observing features which relate to a common object. The approach used in Durrant-Whyte (1988), is to compare pairs of hypotheses verifying disparate sensor opinions and recursively constructing estimates of the underlying geometry, and then recursively combining these groups into sets of environment hypotheses associated with objects $\mathbf{p}$.

MELON PARAMETERS

DISTRIBUTION OF MELON LOCATIONS

Introduction

The performance of a robot acting in an agricultural environment is affected by crop parameters such as spatial geometry of the crop and fruit distribution (Edan, *et al.*, 1990). When evaluating different machine designs during the development process, performance differences can be due to changes in the machine and the crop or interactions between the two. Experiments are difficult if not impossible to reproduce since crop conditions change with time and vary spatially in the field. Simulation has been used to evaluate machine performances (Miles and Tsai, 1987). Simulation of robotic systems in controlled agricultural environments have been performed to determine optimal configuration of the workcell (Kutz and Miles, 1986) and robot design parameters (Chen and Holman, 1990; Ting, *et al.*, 1990). However, since most agricultural domains are unstructured, the means to evaluate the performance for different crop conditions must be developed. Simulation can be applied to evaluate the performance for different cultural practices to provide guidelines to what kind of cultural practices should be adapted to improve performance of a robot.

Cultural Practices

Cantaloupes (*Cucumis melo* L. var. *reticulatus* Ser.) are generally planted at 75 to 125 cm intervals along the row on a 180 cm width bed. When melons are directly seeded rather than transplanted into the field the plants are thinned after emergence to one plant at 75 to 125 cm intervals. Fruit size is directly proportional to spacing: the wider the spacing the larger the fruit (O'Brien and Zahara, 1971). Increasing plant density is used commercially to maintain small uniform fruit sizes for the marketplace (Kasmire, 1981).

Lazin and Simonds (1981) found that increasing the distance between plants increased the number of fruits per plant and mean fruit weight but decreased the total number and weight of muskmelons. Holliday (1960) explained the relationship between plant population and crop yield for fruiting crops as a parabolic curve. With this type of curve, a certain population gives a maximum yield, while larger or smaller populations give lower yields. The effect of within-row plant spacings of 25, 50, 75 and 100 cm with row spacing of 270 cm on 'Burpee Hybrid' and 'Classic' melons were evaluated in field studies and indicated that as plant spacing increased, yield per plant, number of fruit per plant, fruit weight and soluble solids increased, but marketable fruit per hectare decreased (Bhella, 1984).

Melon Location Data Analysis

Since the spacing of melons along the row can effect the design and performance of a robot, the physical location of cantaloupes, cultivar 'Superstar' were collected by measuring exact X,Y coordinates of 859 melons in 29 rows at the Southwest Purdue Agricultural Center, 7 km north

of Vincennes, Indiana, 1989. A two dimensional plot of the spatial distribution of the melons in six different rows is shown in Figure 25.

The spatial distribution of the fruit was statistically analyzed using the International Mathematical and Statistical Library (IMSL Inc., 1988). The average distance between adjacent fruit along the row (X) was 34 *cm* with a standard deviation of 10 *cm*. The average distance between fruit across the row (Y) was 23.6 *cm* with a standard deviation of 14.2 *cm*. Since fruit grow in batches along the growing bed, and number of fruit and their distribution depends on plant spacing and cultural practices, the average distance along the row is not the only useful parameter for a machine design. Therefore, additional parameters such as average distance between plants, number of fruit clustered near the plant center, and the distance of the fruit from the plant center in both the X and Y direction were analyzed. Figure 26 is a histogram of the X distance to the nearest fruit.

The distance between plants was determined by a hierarchical cluster analysis using Ward's method (Andersberg, 1973). The Euclidean distance along the row (X direction) was computed as the distance criteria since this fits the growing pattern. Clusters were formed to minimize the increase in the within-cluster sums of squares. Distances are computed by weighing the distance between centroids. Thus, the minimum increase in the error sum of squares is proportional to the squared Euclidean distance between the centroids of the merged clusters. This suits the cultural practices since the melons are centered around the plants.

Once the clustering was performed gamma, logarithm and exponential distribution curves were fitted to the histograms of the distance between clusters (Figure 27) and distance from the cluster centroid in both the X and Y directions (Figures 28 and 29). The Chi-square test was used

Figure 25. Spatial distribution of Superstar melons in 6 different rows

Figure 26. Histogram of within row distance to nearest melon fruit

Figure 27. Histogram of X distance between melon clusters

Figure 28. Histogram of X distance from melon cluster centroid

Figure 29. Histogram of Y distance from melon cluster centroid

to test the goodness of fit (Walpole and Myers, 1985). The Chi-square values for each curve as derived from the routines are shown in Table 17. The best fit curve is the one with the lowest Chi-square value. The parameters for the best fit, Gamma distribution curve are shown in Table 18.

The coordinates of this best fit data curve served as input to all simulations. Similar datasets were generated by changing plant distances between 25 and 125 *cm*. The number of fruit per plant was computed according to the equations derived by Bhella (1984). For each planting distance, three different random seed numbers were used to generate the data. For each random seed number, a total of 1000 fruit coordinates were generated using the inverse transformation method (Pritsker, 1986).

Table 17. Chi-square fit of distribution curves of melon locations

<i>Distribution</i>	<i>Exponential</i>	<i>Gamma</i>	<i>Logarithm</i>
X distance between clusters	433377	0.510813	9.00486
X from cluster centroid	360323	0.316817	8.53319
Y from cluster centroid	820797	0.10644	7.37293

Table 18. Gamma distribution parameters for melon locations

$$f(x) = \frac{1}{\beta^\alpha \Gamma(\alpha)} x^{\alpha-1} e^{-x/\beta}$$

<i>Distribution</i>	α	β
X distance between clusters	13.9856	0.974217
X from cluster centroid	14.4108	0.981409
Y from cluster centroid	24.9867	1.00053

MELON MATERIAL PROPERTIES AND FINITE ELEMENT ANALYSIS OF COMPRESSION WITH APPLICATION TO ROBOT GRIPPING

INTRODUCTION

Muskmelons (*Cucumis melo reticulatus*) are a major horticultural crop in the Midwest. Bruising during handling and storage reduces quality and can lead to rejection of the melon by the consumer. In order to avoid damage, the magnitude of the handling stresses must be kept below the maximum stresses which cause bruising of the melon. Measurement of the internal stresses caused by the application of a compressive force is very difficult. An alternative approach is to estimate the stresses using finite element analysis (FEA). The FEA can model objects of non-spherical shape which have nonhomogeneous material properties (e.g. modulus of elasticity). In this study, the FEA was used to estimate both deformations and stresses.

Robotic harvesting is one application where the deformation of melons in response to a compressive force becomes important. Previous research in robotic design has concentrated on stability of gripping, actuator power requirements for a given design (Wright and Cutkosky, 1985), and optimization of harvesting efficiency and the harvesting time (Harrell *et al.*, 1988; Kawamura *et al.*, 1986). Stresses in the object being gripped have not been determined. In common applications, robot grippers handle small, light-weight components made of metal. By contrast, melons are relatively large, heavy, and soft. A gripper grasping a melon would apply a compressive force. The distribution of stresses within the melon will determine whether a portion of the flesh is bruised or cracked. The geometry of the gripper could significantly influence these internal stresses. This paper examines stresses developed in the melon rather than in the gripping mechanism. Stresses developed in two types of gripping mechanisms were analyzed in a related study (Edan, *et al.*, 1989).

Specific objectives of this study were:

1. To predict stresses on a melon loaded by compression using FEA;
2. To determine material properties and physical characteristics (modulus of elasticity, shape, size) needed in the FEA which are not available in the literature;
3. To validate the FEA by comparing the model predictions with deformations measured in a compression test;
4. To apply the model to the analysis of internal stresses developed when a melon is gripped by either a parallel plate or a v-v notch robot gripper.

LITERATURE REVIEW

The physical properties of fruit such as muskmelons vary with the stage of development, temperature, climatic conditions, and moisture content (i.e. water potential). Several authors have reported measurements of physical parameters of melon tissue. Diehl *et al.* (1979) determined the apparent modulus of elasticity of the flesh of Honeydew melons using uniaxial compression and torsion tests and calculated the Poisson's ratio of the tissue. Mizrach *et al.*

(1989) reported values of the modulus of elasticity of tissue from ripe winter melons (*Cucumis melo reticulatus* L. cv. 'Superstar'). They sampled the melon flesh at three depths: next to the seed cavity (designated "inner"), in the center of the flesh ("middle"), and next to the skin ("outer").

If the stresses and deformation of a melon can be predicted, the interpretation of the results requires a knowledge of the relationships among stresses, strains and "failure". Failure can be defined as a cracking or splitting of the fruit or bruising of the tissue. Bruising may be caused by cell wall rupturing. Several theories of bruising and failure of fleshy fruit have been proposed. Fridley *et al.* (1968) proposed that pears and peaches may be bruised under compression in the area of the maximum shear stress. Segerlind and Dal Fabbro (1978) suggested that fleshy fruits and vegetables may fail (crack or split) when normal strain reaches a critical value. Diehl *et al.* (1979) observed that relatively large strains developed before failure occurred and suggested that Segerlind and Dal Fabbro's normal strain failure criteria could apply to melon tissue. Chen and Sun (1984) found that loading rate affects failure of apple tissue. Their photograph of cracks in cylindrical samples of apple tissue showed failure along a plane of maximum shear at a fast loading rate and failure along a plane of maximum normal stress at a slow loading rate. Lin and Pitt (1986) observed the same loading rate effect for apples but not potatoes. They also demonstrated that the mode of failure in apples is influenced by water potential.

EXPERIMENTAL DETERMINATION OF MELON PARAMETERS

The modulus of elasticity and the failure strength are needed for the failure analysis of melons. The modulus of elasticity will affect the relationship between the external load and the internal stresses which develop. The failure strength, as defined in this paper, is the stress required to cause the melon flesh to fracture. This definition makes failure strength comparable to the term ultimate strength. Bruising may not be accompanied by fracture and therefore could occur at a force below that of the failure strength.

DETERMINATION OF MODULI OF ELASTICITY

The modulus of elasticity and the failure strength of melon flesh were determined using uniaxial compression of cylindrical samples. For the melon skin, uniaxial tension tests were used because compression tests were not feasible. The density, sugar content, and moisture content of the samples were also determined. Although water potential of the samples was not measured, it would be related to both the sugar and moisture contents.

Compressive and tensile tests were performed on an Instron Universal Testing machine in July of 1989. A 'Superstar' melon (*Cucumis melo reticulatus* L. cv. 'Superstar'), harvested at maturity level "full slip" from a plot in southern Indiana, was cut in half in the plane of the major axis. Samples were taken for determination of melon properties. The samples used for uniaxial compression were cut with a cylindrical borer with inside diameter of 25.4 mm (1.00 in). All samples were taken from a single melon. The bore was inserted into the melon normal to the skin surface. Measurements with a calipers verified that all samples had diameters of 25.4 mm (1.00 in). The skin and the green tissue immediately beneath the skin were cut from the core sample and the remaining portion was divided into cylinders of approximately 12 mm (0.47 in) in height. The portion adjacent to the seed cavity was designated the "inner" sample and the

portion adjacent to the skin was designated the "outer" sample. The remaining portion was designated as "middle".

Five samples each of the inner, middle and outer flesh were tested. The outer samples were lighter in color with occasional regions of light green tissue. They were comparatively hard in texture. In accordance with ASAE Standard S368.2 paragraph 8.2 (ASAE, 1990), compression speed was 25.4 mm/min (1 in/min). Many of the force-deformation curves were non-linear during the first several seconds of compression, but then became linear and remained so up to relatively large strains-- 0.40 mm/mm (in/in) or greater. Several were linear up to the point where failure occurred. Each sample was compressed until it failed. "Failure" was defined as an increase in strain with no accompanying increase in stress (i.e. the force-deformation curve became flat or the force decreased as strain continued to increase). The value of stress for a strain of 0.10 mm/mm was divided by 0.10 to give the apparent modulus of elasticity. The choice of $\epsilon = 0.10$ was appropriate because the strains in the majority of the melon flesh were less than 10% for the contact stress experiment. The stress and strain at failure were also determined and the means and standard deviations (n-1 weighting) of these values were calculated. The means and standard deviations of the moduli of elasticity, the stresses at failure, and the strains at failure are shown in Table 19.

Table 19. Moduli of Elasticity, Failure Stress, and Failure Strain for the flesh and skin of a 'Superstar' melon.

<i>Parameter</i>	<i>No. Samples</i>	<i>Measured</i>		<i>Mizrach et al., (1989)</i>
		<i>Mean</i>	<i>Std. Dev.</i>	
E_{outer} (kPa)	5	802	192	632
E_{middle} (kPa)	5	457	85	428
E_{inner} (kPa)	5	263	102	204
E_{skin} (kPa, perpendicular)	3	7,780	7,110	----
E_{skin} (kPa, parallel)	3	9,080	2,480	----
Outer Flesh, Normal Stress at Failure (kPa)	4	132	53	----
Middle Flesh, Normal Stress at Failure (kPa)	5	74.6	15	----
Inner Flesh Normal, Stress at Failure (kPa)	4	35	13.3	----
Skin (perpendicular), Normal Stress at Failure (kPa)	3	2,130	1,250	----
Skin (parallel), Normal Stress at Failure (kPa)	3	1,640	390	----
Outer Flesh, Normal Strain at Failure (mm/mm)	4	0.243	0.078	----
Middle Flesh, Normal Strain at Failure (mm/mm)	4	0.301	0.064	----
Inner Flesh, Normal Strain at Failure (mm/mm)	4	0.243	0.073	----
Skin (perpendicular), Normal Strain at Failure (mm/mm)	3	0.252	0.052	----
Skin (parallel), Normal Strain at Failure (mm/mm)	3	0.183	0.011	----

Tensile tests were performed on samples of the melon skin. A portion of the skin was cut from the melon and trimmed into a "dogbone" shape which was relatively narrow at the center and very wide at both ends. There was only a small amount of green tissue adjacent to the skin. Therefore, the green tissue was tested along with the skin. Thickness of the samples varied from 1.6 to 4.5 mm (0.06 to 0.177 in) and width at the narrow portion of the sample varied from 12.9 to 15.0 mm (0.508 to 0.591 in). The length of the narrow portion of the "dogbone" shape was assumed to be the length of the sample for purposes of calculating strain. It varied from 19.6 to 25.0 mm (0.772 to 0.984 in).

The wide ends of the sample were held in clamps connected to the Instron load cell and crosshead. Downward movement of the crosshead applied a tensile force to the melon and movement was continued until a break in the sample occurred. Crosshead speed was 25.4 mm/min (1 in/min). Thickness in the vicinity of the break was measured with a calipers. For calculations of stress, the initial cross sectional area was taken as this thickness times the width. Several samples failed in the vicinity of the clamp and the results from those tests were discarded.

A total of six samples were tested. Three of these samples were stretched in the direction parallel to the major axis of the melon and three were stretched perpendicular to the major axis. Average values and their standard deviations are shown in Table 19.

The standard deviations of the apparent moduli of elasticity for the melon skin are relatively large (Table 19). The high variability could be due in part to difficulty in measuring the thickness of the skin and difficulty in removing the same amount of underlying flesh from the skin each time. The thickness varied because of the "ribs", and in some cases it appeared that failure began in one of the depressions between adjacent "ribs". Furthermore, the green rind associated with the skin varied in thickness and it was difficult to estimate the effect of this rind on the results. Although the E_{skin} in the perpendicular direction is greater than the E_{skin} in the parallel direction, the standard deviations of the measurements are large and therefore the differences may not be significant.

Several additional measurements were taken on the samples of melon flesh. The moisture was determined by drying the samples at 70°C (158°F) in a convection oven for 24 hours. At this point, they had reached constant weight. The sugar content was measured with a refractometer. Density was calculated from sample weight and dimensions. With proximity to the seed cavity, the moisture content of the 'Superstar' melon decreased slightly while the sugar content increased significantly (Table 20). Values of the shear modulus for the flesh reported by Diehl and Hamann (1979) were used, and the Poisson's ratio was assumed to be 0.4 which is typical of similar biological materials which are relatively incompressible and have few air spaces, such as potato skin (Jindal and Techasana, 1985). These values are shown in Table 20.

In both the 'Superstar' melon samples and in the results reported by Mizrach and coworkers, the modulus of elasticity decreased with closer proximity to the seed cavity. The values reported by Mizrach for E_{inner} , E_{middle} , and E_{outer} were, respectively, 79%, 93%, and 78% of the values determined in this study. These values lie within the 95% confidence of the data reported in Table 19. The differences may have been partially the result of differences in ripeness. Their samples were much higher in sugar content (9 to 14% total soluble solids).

Table 20. Physical Properties of the Melon Flesh. Values Determined by Experiment unless a literature reference is given.

<i>Parameter</i>	<i>Value</i>
Melon weight [kg force]	3.4
Melon major diameter [mm]	143.2
Melon minor diameter [mm]	126.0
Melon cavity major diameter [mm]	90.5
Melon cavity minor diameter [mm]	60.2
Moisture content Outer Flesh [%]	95.89
Moisture content Medium Flesh [%]	95.57
Moisture content Inner Flesh [%]	94.81
Sugar content Outer Flesh [deg. Brix]	3.14
Sugar content Medium Flesh [deg. Brix]	4.18
Sugar content Inner Flesh [deg. Brix]	5.4
Shear Modulus for melon Skin [kPa]	400
Flesh Density [Kg/m^3]	0.800×10^3
Skin Density [Kg/m^3]	1.100×10^3
Shear Modulus for melon Flesh [kPa]	400 (Diehl and Hamann, 1979)
Poisson's ratio for Melon Flesh	0.380 (Diehl and Hamann, 1979)
Poisson's ratio for Melon Skin	0.400 (see text)

When a melon sample is compressed by a robot gripper, it seems likely that the melon flesh will bruise rather than fracture. Therefore, an attempt was made to determine the value of strain at which bruising would occur. Cylindrical samples from the inner, middle, and outer portions of the flesh of the 'Superstar' melon were compressed to the following strains: 10%, 15%, 20%, and 25%. After compression, the samples were kept at room temperature and observed for visible signs of bruising such as browning of the tissue. After one day, several samples were examined with a microscope under low magnification. Bruising could not be detected with the aid of the microscope. Without staining, it was difficult to identify whether individual cells had ruptured. A more sensitive test is needed to detect rupturing of individual cells. However, development of such a test was beyond the scope and resources of this study.

CONTACT STRESS EXPERIMENT

A contact stress experiment was performed to gather data for evaluation of the finite element prediction of melon deformation. The experiment was performed on the half of the melon remaining after determination of the moduli of elasticity. The melon half was placed on a sheet of plexiglass 12.5 mm thick with the seed cavity of the melon facing the plexiglass. The plexiglass sheet was in turn placed on the load cell of the Instron and the melon half was compressed with a round aluminum plate 12.6 mm (0.503 in) in thickness and 12.6 mm in diameter, which was attached to the Instron crosshead.

The compression rate was 25.4 mm/min (1.0 in/min). When the applied load reached approximately 98 N (22.0 lb), the crosshead of the Instron was reversed. A piece of filter paper was taped to the compression plate and saturated with ink. Contact between the filter paper and the melon made an ink impression on the melon which corresponded to the maximum area of contact between the plate and the melon. This maximum contact area was achieved at the maximum applied load. The major and minor diameters of the elliptical contact area were, respectively, 19 and 15 mm. Total deformation of the point of first contact between the melon and the compression plate was determined from the Instron chart. The deformation was 2.54 mm for the 98 N load and 15.2 mm for the 5215 N load (Table 21).

After the measurements of the contact area were completed, the melon was compressed a second time. This time, deformation continued until the melon failed by fracturing. When the applied force reached 535 N (120 lb), failure occurred in the vicinity of contact between the melon and the plexiglass plate about half way between the skin and the seed cavity. The fracture plane was approximately parallel to the skin surface. Deformation of the first point of contact between the plate and the melon, determined from the Instron chart, was 15.2 mm (0.598 in).

The Hertz contact stress equations (Mohsenin, 1986) were used to estimate the total deformation of the first point of contact between the melon and the compression plate and the dimensions of the elliptical contact area. Two assumptions used in deriving the equation were probably invalid for the melon half: (1) the contacting objects were not homogeneous and (2) the objects in contact probably did not behave as though they were semi-infinite (i.e. the stresses did not diminish to zero at relatively large distances from the point of contact).

The following equation was used to model the deformation δ of the first point of contact between the compression plate and the melon (Mohsenin, 1986):

Table 21. Data for the Contact Stress Experiment.

<i>Test Parameter</i>	<i>Force: 98 N</i>		<i>Force: 535 N</i>	
	<i>Predicted</i>	<i>Experimental</i>	<i>Predicted</i>	<i>Experimental</i>
Major diameter of elliptical contact area [mm]	20.21	19.00	----	----
Minor diameter of elliptical contact area [mm]	19.79	15.00	----	----
Deformation at the point of load application [mm]	6.037	2.540	18.70	15.20

$$\delta = \frac{p}{2} \left[9F^2 \frac{(k_1 + k_2)^2}{\pi^2} \left(\frac{1}{R_1} + \frac{1}{R_1'} + \frac{1}{R_2} + \frac{1}{R_2'} \right) \right]^{1/3} \quad [78]$$

with,

$$k_i = \frac{1 - \mu_i^2}{E_i} \quad i=1,2 \quad [79]$$

In Equations [78] and [79], p is a constant (in this case 1.35) calculated from the radii of curvature of the surfaces in contact, F is the applied force, R_i and R_i' are the major and minor radii of curvature of the objects at the point of contact, and μ_i and E_i are, respectively, the Poisson's ratios and the elastic moduli of the two objects.

When Equation [78] was applied it was assumed that the melon was a hemisphere with modulus of elasticity of 507 kPa (73.5 psi). This is the average of the moduli of elasticity of the inner, middle, and outer flesh and skin given in Table 19. The predictions of Equation [78] for the two applied loads, 98 N (22.0 lb) and 535 N (120 lb), are summarized in Table 21. The predicted deformation was 237% of the measured deformation for the 98 N (22.0 lb) load and 123% of the measured deformation for the 535 N (120 lb) load. The use of the average value of modulus of elasticity could be partially responsible for this discrepancy. For the 98 N load, the stresses may have been sustained by the outer, firmer flesh and the skin. Therefore use of an average value of the modulus of elasticity may have caused the predicted deformation to be too large. The effect of the choice of modulus of elasticity was evaluated by repeating the calculation using the modulus of elasticity for the outer flesh (802 kPa, 116. psi). The deformations were 175% of those observed in the experiment. The Hertz model still over-predicted the amount of deformation. A possible explanation is that the model neglected the effects of the skin and underlying green flesh, which were firmer than the outer flesh. For the 535 N load, the stresses may have penetrated through the skin into the underlying flesh and therefore the use of the average value of modulus of elasticity may have been more appropriate.

FINITE ELEMENT ANALYSIS

The finite element method (Segerlind, 1984) is a numerical procedure especially suited for solving the partial differential equations which describe stresses and strains in biological materials. This method can be used for both static and dynamic structural analyses and allows modeling of irregularly shaped objects which have non-homogeneous properties and are subjected to mixed boundary conditions.

For elastic deformation of a solid, the following system of equations describes the relationship between the applied forces $\{F\}$ and the deformations $\{\delta\}$:

$$[K] \left\{ \delta \right\} = \left\{ F \right\} \quad [80]$$

where $[K]$ is the global stiffness matrix and $\{\delta\}$ and $\{F\}$ are column vectors of global displacements and forces (Segerlind, 1984; Desalvo and Gorman, 1987). Once the values in $\{\delta\}$ have been obtained, the stresses and strains can be calculated using the stress-strain and the strain-displacement relationships.

For the analysis described in this paper, normal strains and equivalent stresses were calculated. The normal strain is the normalized deformation in a direction perpendicular to the surface of the object. The equivalent stresses are functions of the three principal stresses σ_1 , σ_2 and σ_3 . The principal stresses act on perpendicular planes which have no shear stresses. Once principal stresses are known, the equivalent stress can be calculated from the following equation:

$$\sigma_e = \frac{1}{\sqrt{2}} \left[(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2 \right]^{1/2} \quad [81]$$

where $\sigma_1 > \sigma_2 > \sigma_3$.

The concept of equivalent stress has been routinely used in failure analysis of ductile engineering materials. As far as the authors are aware, it has not been applied to analysis of failure in fruits and vegetables, although Liu (1989) and Irudayaraj (1990) have proposed that it be applied to grains and seeds. The equivalent stress analysis gives predictions which follow the Von Mises criteria for failure. The predictions of the Von Mises criteria are quite similar to those of the maximum shear stress theory, which is called the Tresca criterion for multi-stress cases. For engineering materials, the Von Mises criteria gives slightly more accurate predictions of failure. The equivalent stress analysis is also valid in the case of the modified Von Mises criteria. This is applicable to materials that fail at different stresses in tension versus compression. Raghava et al. (1973) proposed the concept for analysis of yield strength of polymers. It seems reasonable that the tensile and compressive failure stresses of melons and other fruits would differ. It also seems likely that some of the regions of a melon loaded by a robot gripper or compression plate are in a multi-stress condition. Therefore, the equivalent stress concept seems appropriate to the analysis of failure of melon tissue.

The major and minor diameters of two 'Superstar' melons of representative size were measured and averaged. These values were used for the models of the melon handled by the grippers. The dimensions of the seed cavity were also determined. Sphericity, the geometric mean of the diameters divided by the diameter of a sphere of equivalent volume (Mohsenin, 1986), was 88%.

Two models were developed. The first was an axisymmetric model. Its cross section was a quarter of an ellipse. The major axis was oriented in the horizontal direction so as to simulate the normal orientation of a melon laying on the ground. The loads were applied axisymmetrically. A section of the model is shown in Figure 30. When the cross section is rotated 360 degrees about the y-axis it approximates a melon half. This model is appropriate for analysis of the experiment described in the section entitled "Contact Stress Experiment" and for analysis of the forces applied by the parallel plate gripper described in the section entitled "Modeling of Robot Grippers". However, for some situations, such as gripping by the v-v notch gripper (see "Modeling of Robot Grippers"), loads are not applied symmetrically. Therefore, a three-dimensional (3-D) model, was also developed.

Figure 30. The finite element grid of the axisymmetric melon model

The geometry of the 3-D model was determined using all of the available planes of symmetry. It was assumed that the gripper made contact with the melon at four points equally spaced around the diameter of the melon. This gave two planes of symmetry. Cutting the melon along these two planes gives a quarter of a melon as sketched in Figure 31. The load was assumed at the mid-point of the surface of this quarter slice and normal to the surface. This gave two more planes of symmetry. The intersection of these planes with the surface of the melon is indicated by the dashed lines in Figure 31. Cutting the quarter melon along these planes gives the melon slice which was modeled (Figure 32).

Both models describe the "thick shell" geometry created by the seed cavity. Both can accommodate non-homogeneous material properties. Different moduli of elasticity were used for the skin and the three layers of melon flesh. In the axisymmetric model (Figure 29) the skin, outer, and middle layers of flesh were each represented by single layers of elements. The inner flesh was represented by two layers. For the 3-D model (Figure 32), the skin, and the outer, middle and inner flesh were each modeled by single layers of elements.

The parameters summarized in Tables 19 and 20 were used in both models. The melon properties could easily be changed and the model could be re-run so as to investigate the effects of varying physical properties.

The models were implemented in ANSYS® a commercially available FEA program. ANSYS allows a wide variety of element types. For the axisymmetric model, the structural element STIF42 was used. This element can be used for the two-dimensional modeling of axisymmetric structures with axisymmetric loading. It is defined by four nodal points and has three degrees of freedom per node: translations in the x, y and z directions.

Figure 31. Three dimensional representation of a quarter of a melon with a load applied normal to the surface

Figure 32. The finite element grid of the three-dimensional melon model

For the 3D model, the structural element STIF45 was used for the skin while the flesh layers were modeled with STIF92. STIF45 is an isoparametric solid described by eight nodes in three dimensional space. Each node has three degrees of freedom. STIF92 is a tetrahedral solid with 10 nodes in three dimensional space. It is not isoparametric and consists of four triangles connected at their vertices. The nodes are located at the intersecting vertices, and in the mid-points of the sides of the triangles. STIF92 is the type of element suggested for use in the modeling of arbitrary volumes. It has three degrees of freedom per node. Additional details of these elements are given in the ANSYS Users Manual (DeSalvo and Gorman, 1987).

MODEL OF COMPRESSION OF THE MELON HALF

The contact stress experiment was simulated using the axisymmetric ANSYS model. The analysis was conducted using three different values of the moduli of elasticity. The "average" model used the mean values of E as indicated in Table 19. The "maximum" model was obtained by using E-values for the skin and flesh equal to the mean values plus one standard deviation. The "minimum" model used the mean values minus one standard deviation. The predicted deformation at the point of application of the load is shown in Table 22 along with the measured values and the predictions of the Hertz theory. For the 98 N (22.0 lb) load, the deformation predicted by the axisymmetric FEA model (avg E) was within 11% of the measured values. This value is considerably closer to the measured than the Hertz equation prediction (see "Contact Stress Experiment"). The FEA model error for the 535 N (120. lbf) load was only 15%.

The results of the minimum and maximum models, suggest that variations in ripeness affect melon deformation. When the maximum and minimum moduli of elasticity were used, deformations were, respectively, 87% and 120% of the deformation predicted for average moduli of elasticity.

Table 22. Comparison of Predicted and Measured Deformations for the Contact Stress Experiment.

Source	Melon Model Parameters				98 N		535 N	
	E_{skin} [Mpa]	E_{outer} [Mpa]	E_{middle} [Mpa]	E_{inner} [Mpa]	Deformation [mm]	Error [%]	Deformation [mm]	Error [%]
Experiment	----	----	----	----	2.54	----	15.20	----
Hertz prediction	0.507	0.507	0.507	0.507	6.04	138	18.70	23
FEA (max model)	11.56	0.994	0.542	0.365	2.46	-3	13.02	-14
FEA (avg model)	9.08	0.802	0.457	0.263	2.83	11	14.97	-1
FEA (min model)	6.60	0.610	0.372	0.161	3.40	34	18.00	18
FEA (avg flesh)	9.08	0.507	0.507	0.507	2.62	3	13.92	-8

The FEA simulation was also run using a single value of the modulus of elasticity for the melon flesh equal to the average value, 507 kPa (73.5 psi). This was the value used for the Hertz prediction. The average value for the modulus of elasticity (E_{avg}) of the skin was used. As demonstrated in the Section entitled "Contact Stress Experiment", the Hertz equation gives a very poor prediction of the deformation. Therefore, the discrepancy between the Hertz prediction and the FEA results is not surprising. The predicted deformations for the two loads with E_{avg} were approximately 93% of the values obtained when different moduli of elasticity were used for the skin and the three layers of flesh. The ability to incorporate non-homogeneous properties has a significant effect on the analysis.

MODELING OF ROBOT GRIPPERS: MODEL FORMULATION

Two types of human gripping strategies are often used in the design of industrial grippers (Tanie, 1985): the "three-finger" grip (similar to the action of a hand holding a pencil) and, the "wrap-around" grip (similar to action of a hand holding a hammer or a large screwdriver). The "wrap around" gripper is most suited for grasping a melon. Two common variations of "wrap-around" grippers are shown in Figure 33: (a) the parallel-plate gripper, and (b) the v-v Notch gripper. A translational mechanism moves the hands of the gripper together like two plates held parallel to each other and gradually moved closer. Thus, force is applied evenly to the object being gripped. The dimensions of the melon dictate the size of the gripper. Based on the measurements of melon diameters taken for this study, the gripper width must be 15 cm (5.90 in) to 30 cm (11.8 in), and it must be capable of supporting a load of 10 N (2.25 lb) to 30 N (6.75 lb). The gripping mechanism determines how forces are applied to the melon.

Figure 33. Two common types of gripping devices. (A) Parallel-Plate Gripper. (B) V-V Notch Gripper.

When the gripper grasps the melon, it applies normal forces N_i at the i points of contact between the gripper and the melon. If the normal forces are applied horizontally (i.e., if the gripper grasps the melon from both sides as it is lying on the ground), then the weight of the melon must be supported entirely by the vertical forces created by friction between the gripper and the melon. The coefficient of friction, k , determines the normal force needed to sustain the weight of the melon W :

$$W < k \sum_{i=1}^{i=n} |N_i| \quad [82]$$

The material from which the gripper is made and the surface texture of the melon will affect the value of the friction coefficient. The gripper design determines the number of contact points n .

In this study, the value of the friction coefficient k was assumed to be 0.95 (Serway and Faughn, 1985), the value for contact of rubber against concrete. Concrete has a rough surface texture and is therefore similar to the texture of the surface of the melon. The choice of rubber on concrete is appropriate if one assumes that the surface of the gripper would be coated with a thin layer of rubber-type material to protect the melon during handling. A rubber type material would also be chosen because it would create the maximum possible friction between the melon and the gripper surface.

The **Parallel-Plate Gripper** was modeled using the axisymmetric ANSYS model with the loading conditions shown in Figure 34a. The y-axis of the axisymmetric model (Figure 30) was taken as parallel to the direction of the applied force. The gripper makes contact with the melon at two points ($n=2$ in Equation [82]). The applied force was assumed to be $W/2$ or 16.67 N (3.75 lb). At each of the two points of contact the force $W/2$ was distributed over three nodes so as to simulate contact stress loading. The center load was 40% of $W/2$ and the loads on either side were 30% of $W/2$. Two versions of the parallel plate gripper model were run. In the first version, the dimensions of the cross section were assigned so as to simulate a melon being compressed on its ends as it is lying on the ground with its major axis horizontal (i.e., parallel to the ground). It was assumed that the robot hands would approach the melon from the sides, with movement parallel to the ground. In this case, the load is applied parallel to the major axis. For the second version, it was assumed that the melon was compressed with the major axis vertical (perpendicular to the ground). The first case would be the more usual position of a melon in the field.

The **V-V Notch Type Gripper** was modeled using the 3-D ANSYS model. The application of the load to the melon is shown in Figure 34b. The gripper makes contact with the melon at four points ($n=4$ in Equation [82]). As described in the section entitled "FEA Models", the force on each quarter-melon was assumed to be one-fourth of the total force needed to lift the melon, 8.33 N (1.87 lbf). The force on the section modeled was taken as one-fourth of this value or 2.08 N (0.468 lbf).

The stresses and deformation of the melon would be influenced by ripeness, which affects the moduli of elasticity. In order to evaluate the effects of ripeness, both gripper models were run using first the "average" moduli of elasticity and then the "maximum" and "minimum" moduli (average values in Table 19 $\pm$ one standard deviation).

Figure 34. Loading applied to the melons by the two gripping devices. (A) Parallel-Plate Gripper. (B) V-V Notch Gripper.

ROBOT GRIPPERS: RESULTS

The nodal displacements for the melon models are shown in Figures 35, 36, and 37. Figures 35 and 36 give the displacements for the two cases of the parallel plate gripper. Figure 35 is for the orientation in which the major axis was horizontal, while Figure 36 is for a vertical major axis. Note that in Figure 35 the major axis is in the y-direction while in Figure 36 it is in the x-direction. The displacements produced by the v-v notch gripper (Figure 37) are shown on a three-dimensional view of the melon slice. The greatest deformation is at the point of application of the load.

Equivalent stresses are shown in Figures 38, 39, and 40. Figure 38 shows stress contours for the parallel plate gripper with the melon oriented with its major axis parallel to the horizontal plane and Figure 39 shows the contours for the same gripper with the major axis perpendicular to the ground. Stress contours for the v-v notch gripper are shown in Figure 40. Maximum values of deformation and the maximum and minimum equivalent stresses are summarized in Table 23.

Although the parallel plate and v-v notch grippers produced nearly equal deformations, the v-v notch gripper had lower values of equivalent stresses. For the two cases of the parallel plate gripper, the one in which the major axis was horizontal produced slightly smaller deformations and equivalent stresses (approximately 97% and 99.8% of the values for the vertical orientation). Therefore, there appears to be only a slight advantage to gripping the melon at the ends of the major axis.

Figure 35. Deformation of the melon model gripped by the Parallel-Plate Gripper when the melon's major axis is oriented parallel to the horizontal plane. The deformed shape is compared to the undeformed shape. The drawing is not to scale. The deformation has been exaggerated to make it more obvious.

Figure 36. Deformation of the melon model gripped by the Parallel-Plate Gripper when the melon's major axis is oriented perpendicular to the horizontal plane. The deformed shape is compared to the undeformed shape. The drawing is not to scale. The deformation has been exaggerated to make it more obvious.

Figure 37. Deformation of the melon model gripped by the V-V Notch Gripper. The deformed shape is compared to its undeformed shape. The drawing is not to scale. The deformation has been exaggerated to make it more obvious.

Figure 38. Equivalent stresses in the melon model gripped by the Parallel-Plate Gripper when the melon's major axis is oriented parallel to the horizontal plane

Figure 39. Equivalent stresses in the melon model gripped by the Parallel-Plate Gripper when the melon's major axis is oriented perpendicular to the horizontal plane

Figure 40. Equivalent stresses in the melon model gripped by the V-V Notch Gripper

Table 23. FEA Predictions of Deformations and Stresses for the Parallel Plate and V-V Notch Robot Grippers.

Case	Maximum Deformation [mm]	Equivalent Stress [kPa]	
		Minimum	Maximum
Max E Model:			
Parallel-Plates Gripper (vertical)	0.836	0.217	64.49
Parallel-Plates Gripper (horizontal)	0.860	0.383	66.07
V-V Notch Gripper	0.890	0.0729	28.23
Avg E Model:			
Parallel-Plates Gripper (vertical)	0.962	0.209	63.70
Parallel-Plates Gripper (horizontal)	0.992	0.283	63.78
V-V Notch Gripper	0.986	0.0558	27.23
Min E Model:			
Parallel-Plates Gripper (vertical)	1.155	0.174	59.77
Parallel-Plates Gripper (horizontal)	1.199	0.194	60.58
V-V Notch Gripper	1.125	0.0439	26.80

As shown in Figures 38, 39 and 40 the maximum stresses develop near the skin in the vicinity of the point of application of the load. The minimum failure stress for the outer melon tissue is 132 ± 53 kPa (19.1 ± 7.7 psi) (Table 19). The equivalent stresses shown in Table 23 are well below this value. This indicates that there is little danger of a fracture type failure. Furthermore, the maximum strain values obtained were all below 5% (0.05 mm/mm). As reported in the section titled "Determination of Moduli of Elasticity", the stress-strain curves were linear for values of strain of 10% (0.10 mm/mm) or less. Therefore, the melon flesh probably behaves elastically for the loadings studied.

In the v-v notch gripper case, the maximum equivalent stresses are about 80% of the average failure stress for the soft, inner tissue (35 kPa or 5.08 psi, Table 19). Therefore, damage is not likely to occur. However, in the parallel plate gripper the predicted maximum equivalent stresses are, respectively, 146% and 88% of the average experimental failure stresses for the inner and middle flesh. Damage to the inner melon flesh could occur if the outer flesh and skin were unable to protect the inner, soft tissue.

A comparison of the "maximum E", "average E" and "minimum E" models (Table 23) shows that typical variations in ripeness have a relatively minor effect. For example, for the parallel plate and v-v notch grippers, the predicted maximum deformations for the "minimum E" models were, respectively, 128% and 121% of the predicted deformations for the "maximum E" models. The equivalent stresses showed the opposite trend. The predicted maximum equivalent stresses for the "minimum E" models of the parallel plate and v-v notch grippers were, respectively, 92% and 95% of the predicted values for the "maximum E" models.

SUMMARY AND CONCLUSIONS

Two finite element models, an axisymmetric and a 3-D model, were implemented on ANSYS and used to analyze compression of a melon by robot grippers. The axisymmetric model is suitable for symmetrical loading at two points on opposite sides of the melon such as would occur for flat plate compression of a melon sliced in half or in loading by a parallel plate robot gripper. The 3-D model can be used for more complicated loadings which occur when the melon is loaded with a v-v notch gripper.

The moduli of elasticity of the inner, middle, and outer regions of the melon flesh were measured using uniaxial compression tests. The modulus of elasticity of the melon skin was measured using a tensile loading test. These values were used later for the FEA model. As shown in previous studies, the modulus of elasticity of the inner flesh is much lower than the modulus of the outer flesh. Average values of the modulus of elasticity for the melon were as follows: 802 ± 192 kPa ($116. \pm 27.8$ psi) for the outer flesh, 457 ± 85 kPa (66.3 ± 12.3 psi) for the middle flesh, and 263 ± 102 kPa (38.1 ± 14 psi) for the inner flesh. The modulus of elasticity of the skin in the direction parallel to the major axis was $9,080 \pm 2,480$ kPa ($1,320 \pm 360$ psi).

The FEA predictions of the deformation for parallel plate loading of a melon sliced in half were within 11% of observed values. This indicates that the model will also give reasonable predictions of stresses developed within the melon. The ability of the FE model to vary material properties within the melon had a significant effect on the results. The FEA gave better predictions than the Hertz contact stress equations.

The models were used to predict the deformations and stresses which would develop when a melon was gripped by either a parallel plate or a v-v notch gripper. The predicted equivalent stresses for the v-v notch gripper were below the failure stresses of the skin, the middle flesh and the outer flesh. However, for the parallel plate gripper, the predicted stresses were higher than the failure stress of the inner flesh. In both cases, the maximum strains were less than 5% (0.05 mm/mm). Predicted stresses were slightly lower when the melon was oriented with its major axis parallel to the ground than when it was oriented perpendicular to the ground. Changes in "ripeness" of the melon, modeled by changing the moduli of elasticity of the melon flesh, affected the stress distribution and the deformation.

This study pointed out the need for development of criteria for predicting the bruising of melon flesh (and the flesh of other soft fruits and vegetables). If criteria were available, the models could be used to predict the maximum force which could be applied to the melon by the gripping devices without the danger of bruising. The applied force could be increased until the strains and the equivalent stresses reached their maximum allowable values. The models could also be used to study the combined effects of factors such as shape, ripeness, and rind thickness on the allowable gripping force.

GRIPPER DESIGN

FINITE ELEMENT ANALYSIS AND OPTIMIZATION OF GRIPPER DESIGN

INTRODUCTION

A robotic system can be used to selectively harvest easily-bruised fruit. Melons are one of the few fruits for which harvesting has not been mechanized (Lenker, 1984). They are delicate fruits and must be selectively harvested. The part of the robot that manipulates the melon is the gripper (also referred to as the hand or end effector). A gripper is the device mounted at the distal end of a robot arm, enabling it to pick up workpieces and hold, transfer, place and release them accurately into containers. The gripper is a critical component in any robotic workcell, and perhaps even more so in an agricultural application since the product handled is delicate and improper handling can cause damage which affects product quality. The nonuniformity of biological products (size, shape, position, orientation, stress/strain relationships) further complicates the gripper design (Simonton, 1990).

Recent advances in the field of robotics have led to the development of a wide variety of robot grippers that have been used in various handling and assembly tasks. Various grippers have been developed for agricultural applications (Trevelyan, 1985; Hoy and McClure, 1987; Kawamura, 1987; Ferrand *et al.*, 1990; Simonton, 1990; Ting *et al.*, 1990; Wolf *et al.*, 1990). The emphasis on the design of robot grippers has been devising new gripping techniques, designing new types of gripper jaws and developing new algorithms to control the gripper's actions (Pham and Heginbotham, 1986; Leaver *et al.*, 1988). However, very little work has been done in developing procedures for designing robot grippers (Baz and Vossoughi, 1986). A general procedure for design/analysis of grippers for agricultural products would be quite useful.

A design process begins with the establishment of functional requirements to satisfy a given set of needs, and ends with the creation of an entity that satisfies these functional requirements. However, since there are a large number of closely related design parameters, every design should be analyzed thoroughly and accurately. The trial and error in the design process has to be eliminated to reduce costs and improve productivity. Moreover, by optimizing these parameters at the earliest stages of conception, design quality and integrity can be maximized and design cycle time reduced significantly.

One of the first steps in gripper design is ensuring that the gripper is strong enough to withstand the specified load (melon weight) and also being as light as possible (Pham and Heginbotham, 1986; Leaver *et al.*, 1988). When evaluating strength, the first step in design/analysis of machine components is usually a static stress analysis. This validates the feasibility of the design concept. Thus, a defective design can be eliminated prior to the construction of a prototype (Imgrund and Wheeler, 1986). Baz and Vossoughi (1986) used a photoelastic modeling technique to investigate the stress fields developed in some of the common designs of robot grippers. They tested the effect of shape and location of the handled objects on the stresses generated in the gripper.

An alternative to measuring the internal stresses is to estimate them using the finite element method. The finite element method is a numerical technique for solving field problems and analyzing structural and mechanical problems (Segerlind, 1986). This method allows modeling of non-homogeneous and irregular shaped objects which have non-linear properties and are subject to mixed loading and boundary conditions. Therefore, it is a powerful tool in design and analysis of machine components (Noor *et al.*, 1990; Tseng, 1990). The design can be improved by applying optimization techniques (Kirsch, 1981 and Banichuk, 1983). Combining optimization techniques with finite element analysis is becoming one of the most popular techniques for reducing the time from concept to prototype and for improving the prototype (Solecki, 1987; Ward *et al.*, 1987; Smith *et al.*, 1989).

In this research, the finite element method was applied to evaluate the static stresses in different gripper designs for a robotic melon harvester. Based on the stress analysis, the gripper's weight was minimized. This approach, which combines optimization procedures with finite element modeling, is general and can be applied to any robot gripper design.

Two main criteria for design of a robotic melon gripper are: (1) minimization of mechanical damage to the melon caused by handling, and (2) minimization of the gripper's weight. The first of these criteria was examined in a companion study (Cardenas *et al.*, 1991). This paper deals with the second criterion. The weight criterion minimizes inertia forces (Simonton, 1990) and allows for the largest payload for a given size of motor and actuator (Rivin, 1988). The gripper has to be capable of handling melons with different radii, in a range of 8 to 20 cm (3.14 to 7.87 in) (Simon and Daniels, 1987; Wolf *et al.*, 1990). Furthermore, the design should consider the required motions needed to detach the melons from the vines.

The objective of this study was to minimize the weight of two suggested gripper designs based on the finite element analysis of the stresses developed in the gripper. This paper presents both a concept for approaching gripper design/analysis and a means for optimizing its weight. The procedure presented can be applied towards the design of any gripper with appropriate changes in design variables. When combined with a companion study of the physical properties of the product being handled, it provides a complete procedure for gripper design for agricultural robots.

DESIGN APPROACH

Geometry of Proposed Grippers

Two different grippers were studied. Both analyses were performed for steel and aluminum since these are common materials having a low stiffness-to-weight ratio. According to Rivin (1988) this ratio is an important design factor. The proposed designs were chosen because similar existing grippers are capable of handling semi-cylindrical parts with different diameters (Nof, 1985; Mason and Salisbury, 1985 and Kato and Sadamoto, 1987). They are also capable of the motions required for detaching the melons from the vine. No forces were assumed for the detaching since this operation can be achieved by a cutting mechanism (Wolf *et al.*, 1990). The gripper dimensions chosen were based on observed melon dimensions.

The first gripper retained the melons rather than grasping them. Pickup by retaining does not exert pressure on the fruit, and thus eliminates possible fruit bruising. The proposed gripping

device, designated the basket gripper (GRIPPER1), consists of two curved arms which come together to enclose the melon, as shown in Figure 41. Although this gripper does not compress the melon, bruises may be caused by movement of the melon in the basket during the handling process. Since the melon is not grasped, its motion can not be restrained.

In the basket gripper, the contact area can not be maximized because the melons vary in size. Therefore, a second gripper was also tested. Many existing designs capable of manipulating cylindrical parts of different diameters are reported in the literature (Pham and Heginbotham, 1986). The V-V type notch edge gripper (Figure 42), designated the notch edge gripper (GRIPPER2), was chosen for two reasons: First, it has four contact points with the melon - two on each side. Since the contact area is greater than for gripping at only two points, such as would occur with a parallel plate gripper, the applied forces are lower at each contact point. This decreases the potential for bruising of the fruit as was validated in a study which evaluated the stresses developed in the melon (Cardenas-Weber *et al.*, 1991). Second, according to Arai and Asada (1986) the V-V type hand is superior to others of its kind (such as the C-C type or plane hand). A major advantage is that this type of gripper can pick up the melons with less energy input than grippers such as the parallel plate type. Hence, the required force can be achieved with less energy input (*i.e.* less torque of the motors, actuators). A notch opening angle of 30° was chosen to allow grasping of different melon sizes and to conform to the design criteria specified by Arai and Asada (1986).

Figure 41. Schematic diagram of the basket gripper (GRIPPER1)

Figure 42. Schematic diagram of the V-V type notch edge gripper (GRIPPER2)

Design Evaluation

The evaluation of the gripper design involved two steps: (1) Static stress analysis and (2) Design optimization. In the static stress analysis, the static stress values under various loadings were determined using the finite element method (Segerlind, 1986). Different weights and sizes of melons were considered because they apply loads at different points on the gripper. This analysis verified that the chosen gripper dimensions were capable of withstanding the specified load (*i.e.* melon weight). Thus, the feasibility of the initial design was validated. The design optimization used the results obtained from the first step to modify the initial design. Optimization techniques were combined with finite element analysis to minimize the gripper weight. In both the finite element analysis of the static stresses and the design optimization, the finite element program ANSYS (Swanson Analysis Systems, 1987) was employed.

THEORETICAL CONSIDERATIONS

Stress Analysis

The basic problem in strength of materials is to determine the relationship between the stresses and deformations for the applied loads. These stresses and deformations depend on geometric configuration, material properties (*i.e.* elastic modulus, Poisson's ratio and elastic shear modulus) and on the magnitude and distribution of the applied forces.

In a three dimensional problem, the state of stress at a point is defined by six independent stress components as follows (Shigley and Mischke, 1989):

$$\begin{bmatrix} \sigma \end{bmatrix}^T = \begin{bmatrix} \sigma_{xx} & \sigma_{yy} & \sigma_{zz} & \sigma_{xy} & \sigma_{xz} & \sigma_{yz} \end{bmatrix} \quad [83]$$

There are corresponding components for the strains.

The maximum, intermediate and minimum normal stresses are called the principal stresses and are defined by σ_1 , σ_2 and σ_3 . The equivalent or von Mises stress (σ_e), which is typically used in design procedures, is defined as:

$$2\sigma_e^2 = \left[\sigma_1 - \sigma_2 \right]^2 + \left[\sigma_2 - \sigma_3 \right]^2 + \left[\sigma_3 - \sigma_1 \right]^2 \quad [84]$$

An initial analysis of the basket gripper was performed using the theory for curved beams. For a curved beam the stress distribution can be derived from the following equation (Shigley and Mischke, 1989):

$$\sigma = \frac{My}{Ae(R-y)} \quad [85]$$

where:

- M = bending moment;
- A = cross sectional area;
- R = curvature radius of the neutral axis;
- e = distance from centroidal axis to neutral axis;
- y = distance from the section of interest to the neutral axis.

It was assumed that the melon exerted pressure on the gripper in the manner predicted by Hertz contact stress theory. Thus, the pressure on the gripper had a semi-elliptical distribution with maximum pressure occurring at the center of the contact area.

Material Properties

The material properties used for the grippers are listed in Table 24 (Singer and Pytel, 1980). The average melon weight was assumed to be 3 kg (6.6 lb). For the optimization procedure, the maximum expected melon weight of 6 kg (13.2 lb) was considered.

Table 24. Material properties

<i>Property</i>	<i>Steel</i>	<i>Aluminum</i>
Poisson's ratio	0.25	0.33
Elastic modulus (GPa)	200	70
Elastic shear modulus (GPa)	25	84
Ultimate tensile strength (MPa)	413.4	241.15
Density ($\frac{kg}{m^3}$)	7800	2800

Design Optimization

Failure occurs when a critical state of stress is exceeded. Of the several existing failure theories (Singer and Pytel, 1980), experimental work with ductile materials shows best agreement with the distortion energy theory (von-Mises theory). Hence, this failure theory was applied. The von-Mises theory assumes that yielding occurs in a general three dimensional state of stress when the equivalent von-Mises stress, σ_e (Equation [84]), is equal to the ultimate tensile strength of the material, S_{ut} .

However, under dynamic conditions the actual maximum strength is usually below the ultimate strength and quite frequently below the yield strength (Shigley and Mischke, 1989). When stresses are applied and removed a very large number of times a fatigue failure may occur. The endurance limit of the material is given by (Shigley and Mischke, 1989):

$$S_e = K_a K_b K_c K_d K_e K_f \left[\frac{1}{2} S_{ut} \right] \quad [86]$$

where:

- S_e = corrected endurance limit of the material;
- K_a = surface factor;
- K_b = size factor;
- K_c = reliability factor;
- K_d = temperature factor;
- K_e = modifying factor for stress concentration;
- K_f = miscellaneous effects factor.

Hence, S_e instead of S_{ut} was considered as the failure criteria using the correction factors in Table 25. The corresponding endurance limits (S_e) for steel and aluminum were obtained by Equation [86] as 93.0 and 64.5 MPa (641.23 and 444.73 ksi) respectively. These values were used as the limiting values of stress in the optimization procedure.

Table 25. Correction factors considered for the fatigue analysis

<i>Modifying factor</i>	<i>Steel</i>	<i>Aluminum</i>
Surface (K_a)	0.83	0.83
Size (K_b)	0.897	0.95
Reliability (K_c)	0.897	0.897
Temperature (K_d)	1.0	1.0
Stress (K_e)	0.675	0.757
Miscellaneous (K_f)	1.0	1.0

The optimization module of ANSYS was used to determine the lightest gripper strong enough to withstand the maximum melon weight. This process is based on three key definitions: the design variable (DV), the state variable (SV) and the objective function (OBJ).

The *Design variables (DV)* are independent variables that are varied through subsequent iterations of the optimization process to obtain an improved objective function. The design variables selected in this study were geometric parameters such as the thickness of the gripper.

The *State variables (SV)* are dependent variables which represent some aspect of the design dependent on the current design iteration, which is used to constrain the problem. The state variable considered in this study was the permissible stress as derived from the failure criteria.

The *Objective function (OBJ)* is a single dependent variable representing the aspect of design to be improved. The objective in this study was to minimize the weight of the gripper. Since weight is proportional to volume, in order to decrease computing time, the objective function was defined as volume. The final weight can be obtained by multiplying optimal volume by the density.

The ANSYS algorithm approximates the objective function and state variables in terms of the design variables. A minimum number of sets or iterations is required before an approximation is first formed. This minimum number is $2N+2$, where N is the number of design variables. These initial design sets can be specified by the user, or they can be generated by ANSYS by randomly varying the design variables within their respective design spaces and looping through solution and analysis phases.

Once the required number of design sets are available and approximations have been made, the series of response surfaces are minimized by a method referred to as the sequential unconstrained minimization technique. This technique uses a series of unidirectional searches along each design variable, searching in the direction of steepest descent of the objective function. This is an iterative looping process.

After the predicted minimum has been determined, the new set of design variables, or the design vector for the next loop, is calculated. This is done by taking a step between the current best feasible design and the predicted best (minimum) design. The best design is the one which has the lowest objective function value and is feasible. Although tolerances default to 1% for

state variables, design variables and the objective function, they can also be defined by the user. Assuming there are no modeling errors, the optimization looping will stop only if the program has reached convergence or termination. The conditions for reaching convergence have been discussed by Swanson Analysis Systems (1987).

RESULTS AND DISCUSSION

Basket Gripper Analysis

The finite element analysis results were verified by comparing them to the theoretical stresses in curved beams (Equation [85]). For this procedure the basket was assumed to be two dimensional and a load was applied to the central nodes. This is similar to the stresses produced by a force applied to a hook of circular cross section. Table 26 presents the maximum stresses obtained from Equation [85] ("theoretical") and the finite element results for three different loads. The magnitude of the stresses are proportional to the load applied. The finite element results are lower than the analytical results by 7% on average. More accurate results could have been obtained by using a finer mesh. Further refinement was not attempted since it would have required a substantial increase in computing time. For this preliminary design the finite element results were considered sufficient.

The melon weight applied a force on the gripper which was modeled as forces acting on three adjacent nodes (the arrows in Figure 43). The force on each half basket was assumed to be half of the total weight of the melon. In accordance with Hertz contact theory, the forces were applied so that maximum force (1.25 N or 2.5 lbf) occurred at the center point of contact. The forces at the two adjacent nodes were one half of the maximum force (0.625 N or 1.3 lbf). The boundary conditions consisted of restriction of displacement of all the nodes along the end, where the gripper was connected to the robot (the shaded region in Figure 43). The mesh (Figure 44) was obtained using the automatic mesh generation capability of ANSYS. For the initial evaluation, the width of the gripper was 0.3 m (11.81 in) and its thickness was 0.1 m (3.94 in). Equivalent stresses obtained from the finite element analysis are plotted in Figure 44. Maximum equivalent stresses (contour E) occurred at the point of bending of the basket. They decreased in semi-elliptical contour lines as the distance from the point of maximum stress increased.

Table 26. Maximum theoretical and finite element stresses (Pa)
for different forces for a basket model

<i>Force (N)</i>	<i>Theoretical</i>	<i>Finite element</i>
15	21555	20760
24	34630	33210
30	43110	37361

Figure 43. Applied forces and boundary conditions used for the ANSYS model of GRIPPER1

Figure 44. Equivalent stresses in GRIPPER1

Notch Edge Gripper Analysis

It was assumed that the gripper would approach the melon from above. This reduces the amount of contact with the plant and prevents damage to nearby melons which may not be ready for harvest. This orientation also produces slightly lower stresses on the melon (Cardenas-Weber *et al.*, 1991).

The arrows in Figure 45 describe the forces applied to the gripper by the melon and correspond to the horizontal and vertical forces. These forces are calculated using a static analysis of forces, based on the free body diagram of the melon-gripper system. The following equations give the force components acting on the V-V type notch edge gripper:

$$F_1^x = \frac{-W \tan \theta}{4}; \quad F_1^y = \frac{-W}{4} \quad [87]$$

$$F_2^x = \frac{3W \tan \theta}{4}; \quad F_2^y = \frac{3W}{4} \quad [88]$$

Figure 45. Applied forces and boundary conditions used for the ANSYS model of GRIPPER2

where:

- F_1^x = horizontal force on lower face of the notch;
- F_1^y = vertical force on lower face of the notch;
- F_2^x = horizontal force on upper face of the notch;
- F_2^y = vertical force on upper face of the notch;
- W = melon weight;
- θ = opening angle of the notch;

The boundary conditions consist of restriction of displacement of all the nodes along the surface, where the gripper is connected to the robot (shaded in Figure 45). The mesh (Figure 46) was obtained using the automatic mesh generation capability of ANSYS. The intention was to use more elements in the notch area, since this was assumed to be the region where maximum stresses would develop. However, ANSYS would not accept triangular elements which were needed to accurately describe this area. Hence, parallel piped elements were used.

(a) stress contours for the whole gripper (b) enlarged view of the critical notch area

Figure 46. Equivalent stresses in GRIPPER2

For the initial evaluation the gripper's dimensions were: height (H) - 0.3 m (11.81 in); width (B) - 0.3 m (11.81 in); and thickness (LZ) - 0.1 m (3.94 in). Equivalent stresses obtained from the finite element analysis are plotted in Figure 46. Maximum stresses (contour I) occurred at the upper notch edge, at the point of application of the force. These stresses are located in the same region where the maximum stresses were found in the photoelasticity experiments by Arai and Asada (1986). Due to the complexity of this design no theoretical values could be obtained for comparison.

Basket Gripper Optimization

The maximum equivalent stresses obtained from the finite element analysis were 0.65 and 0.49 MPa (4.47 and 3.41 ksi) for steel and aluminum, respectively. Since these stresses were well below the endurance limit, the initial design would have an infinite life. Thus, the optimization procedure was carried out to reduce the weight. The finite element model of the basket gripper was constructed using two design variables (Figure 47a): the thickness of the gripper (LZ) and the width of the gripper (LR). The width of the gripper was constrained to be greater than 0.1 m (3.94 in) so that it would be capable of holding the melon. The optimal values of LR, LZ, volume and weight are listed in Table 27. The final stresses obtained were 92.99 and 64.49 MPa (641.16 and 444.66 ksi) for steel and aluminum, respectively, and were still below the endurance limit. The initial design weight was reduced by 80.0 percent for steel and by 69.6 percent for aluminum. Since the aluminum gripper is lighter it would probably be used to reduce power required for moving the gripper.

Notch Edge Gripper Optimization

As in the case of the basket gripper, the initial design of the notch edge gripper would have an infinite life. The maximum equivalent stresses obtained from the finite element analysis were 0.31 and 0.26 MPa (2.15 and 1.79 ksi) for steel and aluminum, respectively. Three design variables (Figure 47b) were used: thickness (LZ), height (H) and width (B). The height was constrained to be larger than the notch length (which was taken as 0.1 m (3.94 in) to allow grasp of the melon). The optimal values are listed in Table 28. The initial design weight was reduced by 84.6 percent for steel and by 82.4 percent for aluminum. The final stresses obtained were 89.78 and 62.71 MPa (619.03 and 432.38 ksi) for steel and aluminum, respectively, and were still

Table 27. Initial and final design variable values for the basket gripper optimization

<i>Optimization Parameter</i>	<i>Steel</i>			<i>Aluminum</i>		
	<i>Initial</i>	<i>Final</i>	<i>% reduction</i>	<i>Initial</i>	<i>Final</i>	<i>% reduction</i>
LR (m)	0.3	0.1	66.7	0.3	0.1	66.7
LZ (m)	0.01	0.00065	93.5	0.01	0.00095	90.5
VOL ($m^3 10^{-3}$)	0.075	0.015	80.0	0.075	0.023	69.3
Weight (kg)	5.85	0.12264	79.0	2.1	0.0639	69.6

Table 28. Initial and final design variable values for the notch edge gripper optimization

Optimization Parameter	Steel			Aluminum		
		Initial	Final	% reduction	Initial	Final
B (m)	0.3	0.1502	49.9	0.3	0.1475	50.8
H (m)	0.3	0.1999	33.4	0.3	0.2011	32.9
LZ (m)	0.01	0.0026	74.0	0.01	0.0044	56.0
VOL($m^3 10^{-3}$)	0.91	0.14	84.6	0.91	0.16	82.4
Weight (kg)	7.14	1.092	84.6	2.56	0.448	82.4

Figure 47. Design variables used for the finite element optimization

below the endurance limit. As in the previous design, the aluminum gripper would probably be used to reduce power requirements since it is considerably lighter.

SUMMARY AND CONCLUSIONS

A procedure for analysis and optimization of robot grippers has been developed. This procedure consisted of the following steps: (1) stress analysis of the grippers using the finite element method to validate that the grippers were strong enough to withstand the load of the melon and (2) optimization of the gripper's dimensions to minimize weight.

The procedure was applied to analyze and optimize the weight of two grippers for robotic melon harvesting. However, the procedure presented is general and could be applied in a similar manner to optimize other design parameters such as shape, materials, surface roughness, *etc.*

The basket gripper, designed to retain the melons rather than grasp them, would reduce the possibility of bruising caused by pressure applied in grasping the melon. However, it might allow the melons to move within the basket. The second gripper tested was the notch edge V-V type gripper, which grasped the melon at four contact points.

The finite element analysis of the stresses caused by the melon gave the following results:

1. The magnitudes of the stresses were proportional to the load.
2. Maximum equivalent stresses in the direction of the applied force occur at the point of bending of the basket gripper. They decreased in semi-elliptical contour lines as the distance from the point of maximum stress increased.
3. In the case of the notch edge gripper, maximum equivalent stresses occurred in the vicinity of the applied force.

The finite element analysis results were compared to analytical solutions in the case of the basket gripper and to experimental results in the case of the notch edge gripper. of curved beam theory was 7% on average and was sufficient for this preliminary design. Agreement could have been improved by applying a finer mesh but at the cost of an increase in computing time.

The stress analysis validated the feasibility of the initial design and was used to optimize the design. Both grippers were designed for infinite life. The minimum gripper weight capable of withstanding the melon load was determined. The final designs were significantly lighter than the initial ones. This reduction in weight would improve the robot design by reducing inertia forces.

The basket gripper was lighter than the V-V notch gripper. However, another important criteria for gripper design is to minimize stresses on the melon so as to prevent fruit bruising (Cardenas-Weber *et al.*, 1991). The final selection of the gripper should be based on both the gripper and melon analyses.

This procedure of gripper design/analysis combined with a parallel study of the physical properties of the agricultural product being handled could improve the performance and reduce construction costs of agricultural robots.

GRIPPERS FOR A MELON HARVESTING ROBOT

INTRODUCTION

The designed gripper is the operational end of a complex mobile robotic system for selective harvesting of melons, guided by artificial vision and ripeness sensing, which is in the early stages of development. Melons grow scattered over the beds either singly or in small groups. The plants are covered by leaves. Melons, even if of the same variety and growing in the same field, ripen in different sizes and at different times. Once the melon is identified and sensed as ripe, the gripper has to pick it up with no regard to the growing conditions. Since the melons are harvested slightly premature, the stem has to be cut off as only ripe melons can possibly detach by just lifting them from the ground. The direction of the stems could be considered as random relative to the gripper or robot. Once detached from the stalk, the gripper has to hold the melon through the transport phase and release it where required. Due to the frequency of use of the mechanism, and the environment in which it operates, it is desirable to make it as durable and simple as possible. The number of moving parts should be minimized, and they should be protected from dirt and moisture which may cause their performance to deteriorate. Any sensors or electrical components should be protected as well. The structural parts of the gripper need to be sturdy enough to be durable over time, but also must be kept as lightweight as possible to reduce the load exerted on the robot.

OBJECTIVES

To develop a gripper suitable to operate from a robot arm, able to pick up all melon sizes without bruising or being obstructed by the leaves and without damaging nearby and unripe ones. It has to cut the stem off regardless of its direction. The gripper has to hold the melon during transfer to storage and release it there undamaged. The operation time of the gripper should be minimal to reduce the cycle time of the robot.

DESIGN AND CONSTRUCTION OF THE GRIPPER

The melon gripper comprises the following three elements (Figure 48): the mounting frame, the gripping device and the cutting attachment, and is operated by pneumatic cylinders.

The mounting frame is designed to be attached vertically to the arm of the envisioned robot with the gripping device operating from underneath. This frame accommodates the activating mechanisms, including two 1.5" diameter pneumatic cylinders, one with a 1" stroke to operate the gripper and one with a 2.0" stroke to operate the cutter. The pneumatic activation was chosen over two other alternatives - electric and hydraulic due to its light weight, cleanliness even when leaking, its fast operation when required and ease of speed control.

The gripping device was designed to retain the melons rather than to grip them. It is made of two semi-spherical frames which are hinged jaw-like underneath the mounting frame, forming when closed, the outlines of a hollow sphere, 200 mm. in diameter. The semi-spherical frames are made of two rings - 200 mm. and 100 mm. in diameter spaced concentrically 60 mm. apart by means of 3 curved sections. All are made of 8.0 mm. diameter stainless steel rods. The two

Figure 48: Initial gripper - (a) front and (b) side views

gripper jaws close to within 60 mm. of each other (when the larger rings are approximately parallel). In the open position each jaw pivots 30" outward.

The gripper was designed to pick up a range of sizes of melons from less than 80 mm. to approximately 180 mm. in diameter. Pickup by retaining the fruit was considered of advantage over pickup by suction which may be obstructed by the leaves covering the melons and may pick up trash from the ground. In addition, the pickup by retaining does not exert pressure on the fruit, thus eliminating a possible cause of damage. It also enables the rotation of the melon in such a way that the vine is positioned in a downward direction, ready to be cut.

The cutting attachment is made of two arms, one fitted with a sharp blade and the other with a slotted anvil. The arms counter rotate around the gap between the semi-spherical jaws, from where the vine hangs out. Each rotates about 140° from the top downward to meet the bottom. The stalk is then cut between the blade and the slotted anvil. The cutting arms are cable pulled and spring returned to the top of the gripping jaws, thus being out of the way when the melons are picked up or released. This type of "blind" cutting device was chosen for its ability to cut the stems regardless of their location. Cutting between blade and anvil at the bottom of the gripper causes less pull on the stem by reducing the rotational movement of the melon and ensures a clear cut. Cutting by impact was required because of the unfavorable leverage of the activating system where a short lever has to operate a long arm.

In operation the open gripper is aimed to reach centrally over the melon until the jaws touch the soil surface whereupon it is closed by activating the pneumatic cylinder. The melon is then

lifted from the ground in order to stretch the stem hanging out somewhere between the semi-spheric jaws, and to prepare it for cutting. Then the cutting attachment is activated to cut the stalk leaving a piece of it attached to the melon. Finally, the gripper is opened to release the melon for further transportation.

PRELIMINARY FIELD TESTS

The gripper was tested on a melon field to evaluate both functions of gripping and cutting. The implement was connected to a compressed air supply and was operated manually (Figure 49). The gripping jaw rings closed around the melons even through a cover of leaves and did not exert pressure on the melon but slipped under it. After retaining and lifting the melon 20 to 30 cm. from the ground the cutter was activated manually and clipped the stem off. Ripe melons came off by just lifting them, or the stem detached completely from the melon by some pull exerted by the clipping action. Following the field test, the activating mechanism was redesigned to make it more compact and to fit more easily onto the robot arm. Both functions of the gripper performed well during this test. It picked up the melons within the size group for which it was designed, without any damage. The leaf cover of the melons did not present a problem. The cutting attachment detached all the melons off the plant. The melons must be lifted high enough to stretch its stem out between the gripper jaws in order to cut the stem only and not the runner. It could pick up one melon separately from another one close by, however, this may present problems when operated by a robot. Since melons often touch, it is necessary to solve this problem. In many cases the problem could be solved by guiding the robot to the exact position and orientation of the fruit. However, this requires complex image processing and control of the robotic arm. Moreover, there are several cases where this gripper will not be capable of grasping fruit without causing damage to its neighboring fruit.

Figure 49: (a) Initial design prototype, (b) Initial design field test

ALTERNATIVE GRIPPER DESIGNS

To overcome the problem of damage to nearby fruit, alternative intuitive design approaches were obtained by presenting the problem to industrial designers. A junior class of industrial design went through a preliminary stage of concept generation, based on this preliminary work, individual experience and study of existing technology (Kato, 1982 and Nof, 1985). Their initial concepts were refined by interacting continuously with the project leaders and receiving feedback from their instructor. Each proposed gripper design was related to an overall view of the robotic melon harvester design concept. Three of the generated designs are presented in the following:

Counter action gripper

The gripper consists of two semicircular (crescent shaped) hands which rotate on a track, driven by electric motors (Figure 50a). The distance between the hands is adjusted by another motor to accommodate melons of varying sizes and shapes. When a ripe melon is located by the harvesting system, the gripper is placed above its location in the open position and the hands are set at the width of the diameter of the melon. The actuator is then lowered, and the hands rotate around the melon in opposite directions to prevent rolling (Figure 50b). The hands rotate around an imaginary horizontal axis that runs through the center of the melon. After the hands are fully closed, the melon is lifted, the stem is cut, and the melon is placed on the conveyor belt for transport to the sorting and storage mechanisms. This gripper is relatively simple to implement. The possibility of damage to nearby fruit is reduced since the opening is adjustable and the rotation of the hands are closely related to the shape and size of the melon.

Figure 50: (a) Counter action gripper - front and side views, (b) operations

Double action gripper

This gripper uses a pneumatic system and a sensing device to adjust its holding size for the various kinds of melons. The open gripper moves down vertically from a position above the melon (Figure 51). As the two holders pivot around the horizontal axis of the melon, the vine is pushed to the bottom half. The holders are wedge-shaped, allowing the gripper to push away other melons which are nearby. Once the holders complete their movement, the melon is lifted off the ground, and the vine is cut by a pair of blades coming from inside the holders. Although this concept seems slightly complex, it fulfills the criteria of gripping and not damaging nearby fruit.

Mechanical fingers gripper

This gripper consists of a vacuum system combined with four moving fingers (Figure 52). The fingers have several joints which allow to close around the melon. In case of one or more melons touching the melon to be picked, the rollers on the ends of the soft fingers move the fingers around the melon to the void space. Thus, damage to nearby fruit is eliminated and all four fingers are positioned around the melon and ensure a firm grasp. The fingers are then closed around the bottom of the melon to grasp it. As the z-axis starts its retracting motion, compressed air is blown through the nodes located on the joints of the fingers. This air levitation allows the melon to rotate (to a stem down position) in preparation for the stem to be cut. After the melon is raised above the foliage, the vacuum system sucks the melon to a transfer tube. Once the melon has passed the cutting sensor, the stem cutting blades are activated. The melon continues up the lined suction tube and is deposited into a concave conveyor system. The advantage of this design is that it eliminates transition between the gripper and conveyor by using a vacuum. The concave conveyor design eliminates bruising of the melon during its travel to a storage system. This system is very complex, requires extensive air flow and suction and therefore will be difficult to implement.

Figure 51: Double action gripper - (a) front and side views, (b) operations

Figure 52: Mechanical fingers gripper

Improved gripper

Based on ideas derived from the aforementioned designs an improved model of the first gripper was designed, constructed and field tested in manual operation. The new design (Figures 53 and 54) uses only a single pair of hinged rings and two fixed rounded side plates to form the outlines of the spherical retaining shape. The rings are 200 mm. diameter and pivot only about 18° each to provide an opening of 200 mm. at the bottom of the gripper. This restricted opening (compared to the 260 mm. opening of the previous model) results in a more accurate positioning of the gripper relative to the melon, and together with the single ring jaw, provides for easier penetration between closely spaced melons. The counter rotating arms of the cutting device are mounted on opposite sides, one on each of the fixed side plates. The new design simplifies the activating mechanism and allows for a thinner construction of the arm mountings. This, in turn, improves access to closely spaced melons.

SUMMARY

A gripping device for harvesting melons has been developed. The gripper was designed to retain the melon, pick it up slightly from the ground, cut it free from the plant, and release it where required - without damage to the fruit and considering the variability in size of the melons. An initial prototype was designed and manually activated in the field to check the feasibility of the concept. This initial design performs well, even though potential problems are anticipated when operated by a robot manipulator relying on sensory inputs. To overcome the problems that were identified, alternative intuitive design approaches were obtained by presenting the problem to industrial designers. Refined versions deriving from this inter-disciplinary approach are being developed. The wide variety of solutions presented may stimulate innovative gripper designs for other field crops (Figure 55).

Figure 53: Improved gripper - front (a) and side (b) views

Figure 54: Improved design prototype in operational sequence

Figure 55: Proposed overall harvesting system appearance model

DESIGN OF A ROBOTIC MELON HARVESTER

ANIMATED, VISUAL SIMULATION OF ROBOTIC MELON HARVESTING

DESCRIPTION OF THE PROBLEM

The problem of selectively harvesting easily bruised fruit can be solved by developing an agricultural robot (Sittichareonchai *et al.*, 1989, Grand D'Esnon *et al.*, 1987, Slaughter and Harrell, 1987, Kawamura *et al.*, 1986). Melons are one of the few fruits which have not been mechanized (Lenker, 1984). They are delicate fruit which must be selectively harvested.

Selecting the appropriate robot for the melon harvesting task and defining its motions is an important factor in the overall performance of the system. Criteria for selecting a manipulator may include all or some of the following:

- Work space characteristics, including volume, shape and the manipulator's kinematic redundancy (Kumar and Waldron, 1980; Vijaykumar, *et al.*, 1987; Paden and Sastry, 1988; Yang and Lee, 1983);
- Payload;
- Mobility, *i.e.*, the number of degrees of freedom and type of independent motions (Erdman, *et al.*, 1987);
- Agility, *i.e.*, effective speeds of prescribed motions;
- Accuracy and repeatability;
- Structural stiffnesses, *i.e.*, masses, damping coefficients, natural frequencies (Vukobratovic and Potkonjak, 1982); and
- Economics, *i.e.*, purchase price, operational costs, reliability and maintainability (Harrell, 1987).

Since there are many closely related design parameters, optimal robot selection is an extremely difficult and time consuming task which is not yet automated. Rivin (1988) has suggested that robot design should be handled on the basis of statistics, with optimum conditions determined on the basis of probability of occurrence of combined loads during typical cycle performance. In the design phase of a robot, simulation has been used to evaluate robot performance (Kutz and Miles, 1986, 1987; Novak, 1985; Waterbury, 1985; Krouse, 1986). By using simulation the amount of experimental research is minimized and the best performing robot can be selected and evaluated prior to prototype testing.

An agricultural environment is complex and loosely structured. Cultural practices influence parameters such as spatial geometry of the crop and fruit distribution, which affect the design and performance of a robot. Simulation tests the effects of fruit size, shape, weight and position on machine performance. Optimum system configurations for specific crops can be determined by simulating different agricultural scenarios and robot designs and motions.

SIMULATION MODEL

The complex task of designing an agricultural robot for harvesting melons was achieved through the use of animated, visual simulation of the manipulator motions and the melon handling. The paths necessary to accomplish the desired operations were planned and programmed. Provided the CAD (Computer Aided Design) geometry and kinematics, performances of different robots were simulated and alternatives compared.

The simulations were performed on a Personal Iris workstation (Silicon Graphics, 1988) using IGRIP (Interactive Graphics Robot Instruction Program, Deneb Robotics, 1988). The IGRIP software is a user-friendly computer graphics package for workcell layout, simulation and offline programming. Parts used in the workcell are designed within the CAD system of IGRIP. Parts are joined to define devices of multiple degrees of freedom. A device has both geometric and non-geometric (*i.e.* kinematics, dynamics, velocities, accelerations) features associated with it. A workcell (Figure 56) is composed of devices positioned relative to each other. Robots can be selected from an existing extensive library of the most common industrial robots (more than 50 robots) or can be user-designed. Devices in the workcell can be programmed graphically using an interactive menu or directly using a high level programming language, GSL (Graphics Simulation Language, Deneb Robotics, 1988). GSL resembles Pascal with specific enhancements for device motion, display control and environment inquiries. Several devices can be simulated simultaneously. Input/Output (I/O) signalling permits communication between devices. Developed and proven programs are generic and independent of the selected robot and can be directly downloaded to the robot using existing translators. The 3D (three dimensional) shaded color animation provides realism that helps confirm the validity of the simulation. Predicted cycle times are computed, allowing the different scenarios to be evaluated not only for technical feasibility but for determining harvesting rates as well.

Figure 56. Simulated workcell elements

The robotic melon harvester workcell (Figure 56) consists of the following elements:

- **Robot:** Two different types of robots were simulated: a cylindrical robot and a Cartesian one. Both robots were obtained from the IGRIP library. Existing robots were modified to permit simulation of physical models with different reaches, speeds, *etc.*
- **Gripper:** The gripper was similar to an actual field-tested gripper (Miles *et al.*, 1989) and consisted of two devices: a detacher and a gripping device. Each was programmed separately and coordinated using I/O signals. The gripper was attached to the end of the selected robot and coordinated with the robot's motions and conveying system. Figure 57 shows two different CAD models of the gripper.
- **Conveying System:** Different conveying methods were simulated and evaluated. The first consisted of a box with individual cells for fruit attached to the tractor. Different placements of various containers in regard to the robot's location were examined. Figure 58 shows the initial proposed configuration.
- **Tractor:** A CAD model of a tractor was designed and the robot and conveying systems were attached.

Two different modes for picking the fruit were programmed and simulated:

- Continuous mode, and
- Step (incremental) mode.

Figure 57. CAD models of the gripper

Figure 58. Initial conveying system model

In the continuous mode the robot arm moves toward the fruit while the tractor is moving along the row. A constant tractor speed is assumed. In the step mode the tractor advances along the row and stops at each batch of fruit. Only then does the robot arm approach each fruit and pick the ripe ones. Once all ripe fruit in the batch are picked the tractor advances to the next batch. The stopping location of the tractor is determined by averaging the coordinates of all the ripe fruit in the batch.

- **Melons:** Melons were simulated by entering into the workcell spherical objects of diameters corresponding to known varietal distribution. Different colors (yellow, lime, green, dark green) simulated various stages of ripeness.

At the first stage of development, the simulation assumed that all locations of melons were known *a priori*. Since this approach is unrealistic, the program was modified so that melons coordinates from field data were read from files and melon icons were displayed in the workcell concurrently with the picking process. Thus, real-time determination of fruit locations was simulated. In these simulations, the melons appear in the workcell only after their locations are determined by the simulated image processing algorithms. Therefore, the whole row is not displayed at the beginning (although all melons exist in the field) and melons appear as the tractor advances along the growing bed.

• **Manipulator Motions:** The robot arm cycles through the following steps:

1. Move to a point located 25 cm vertically above the fruit to be picked;
2. Move to exact coordinate of fruit;
3. Wait for signal from gripper that fruit has been grasped;
4. Move away from ground 25 cm and wait until detacher has cut away stem;
5. Move towards conveyor at a distance of 25 cm from final destination;
6. Move to exact final destination; and
7. Wait until gripper has released fruit.

The sequence of motions is described in Figure 59.

Two different motions for approaching the fruit to be picked and approaching the conveyor were simulated: a straight line trajectory and a joint-interpolated trajectory (a straight line in the robot's joint space that satisfies the position and orientation constants at both end points). These two trajectories are simple, efficient, smooth and accurate and will allow for fast computation time (near real-time) for generating the sequence of control set points along the desired trajectory of the manipulator.

Figure 59. Sequence of the manipulator's motions

- **Gripper Motions:** The gripper's motions were programmed according to the actions of the field tested gripper. I/O signals were entered to synchronize between the gripping device, the detacher and the robot. The gripper cycles through the following steps as shown in Figure 60:

1. Grasp the melon, both jaws are closed around the fruit;
2. Wait until robot signals that melon has been lifted slightly from the ground;
3. Cut the stem, the cutting devices are activated;
4. Wait until robot reaches conveying system; and
5. Release melon, open jaws enclosing fruit.

Return of the cutter is done slowly but happens during transfer to the conveyor which is part of the operation cycle of the robot arm.

The following cycle times (seconds) of the gripper were estimated from field performance:

1. Grasp (after positioned by arm)	0.5
2. Lift to stretch stem	0.5
3. Cut off	0.1
4. Release	0.5

Figure 60. Sequence of the gripper's motions

DATABASE

At the first stage, the input to the simulation were data files containing randomly distributed coordinates of melons along the growing bed. Next, coordinates of melons were simulated by entering the predictable spatial distribution according to the planting conditions: 3-4 melons located 0.8-1.2 meters apart. Since cultural practices such as distribution along the row, affect the design and performance of a robot, these parameters were quantified by field measurements. Data of different varieties of melons were collected by measuring in the field exact x,y coordinates of melons along the row. Thus, the different scenarios were evaluated based on realistic input.

RESULTS

The motions required to grasp, detach and place melons in containers have been planned and programmed. Computer simulations have been conducted to determine the performance of different types of robot manipulators and suggested motions. Alternative conveying systems configurations have been designed, simulated and evaluated.

Joint-interpolated trajectories have been compared to straight line trajectories for a cylindrical and a Cartesian robot. For the cylindrical robot the joint-interpolated motion is 50% slower than the straight line motion. For the Cartesian robot joint-interpolated motion is 5% faster. The Cartesian robot is 4% faster than the cylindrical one.

Initially, the average cycle time with a Cartesian robot and a gripper (designed and field tested) was 10.8 seconds per fruit. Since this speed is considered to be slow, further simulations were conducted to improve the performance.

Speeding up the system by activating the robot in a continuous mode for the initial conveying system design, was not successful since even at a slow tractor velocity (*i.e.* less than 0.16 Km/hour), a single robot arm is not fast enough for a 1.5 meter wide bed and many melons are missed.

Different conveyor configurations which minimize the robot motions required for transferring melons were simulated and evaluated. The best results were achieved by adding an intermediate conveyor that approaches the robot arm and transfers the picked fruit to the main conveyor (Figure 61). Thus, the robotic arm approaches the fruit, picks it up, transfers the picked fruit to the nearby intermediate conveyor, places the fruit and approaches the next fruit while the intermediate conveyor conveys the fruit to the main conveying system. This is analogous to two robot arms, one having greatly simplified motions. Since both the process of picking the fruit and conveying the picked fruit are executed in parallel and the robotic arm travels less distance, the cycle time is reduced significantly.

The main conveyor consists of two moving conveyors that advances as each melon is picked. Each conveyor is at the edge of the row, where it doesn't block approach to any melon. The intermediate conveyor conveys the fruit to the closest conveyor. This configuration was tested by performing simulations on measured locations of fruit growing in the field (X,Y coordinates of 10 rows of Superstar melons grown at Purdue Southwest Experimental Station, near Vincennes, IN). The average speed of this system was 5 seconds per fruit or more than twice the speed of the initial simulated scenario.

Figure 61. Intermediate conveyor configuration

SUMMARY AND CONCLUSIONS

The motions required to grasp, detach and place melons in containers have been planned and programmed. Computer simulations have been conducted to determine the performance of different types of robot manipulators and suggested motions. The simulation permitted testing of various agricultural conditions to improve the performance and to evaluate the cycle time of the robotic harvester prior to actual prototype construction and testing.

Based on the simulation tests and the melon field data the stop and harvest mode seems to be the most appropriate approach. Fruit are located in clusters of 3-4 fruit every 0.8-1.2 meters and not distributed evenly along the row. Picking the fruit while the vehicle advances along the row requires the arm to stretch back and forth across the bed and therefore makes extended motions. This complicates the control algorithms and increases the computation time.

The design reported here consists of a Cartesian robot and an intermediate conveyor that approaches the robot arm and transfers the picked fruit to the nearby main conveyor belt. The cycle time of this design is 5 seconds per fruit.

Additional simulated scenarios to further improve the performance are being conducted. These consist of activating multiple robots which have simplified structure, changing the configuration of the conveying system so that the intermediate conveyor can be eliminated. Preliminary simulations indicate these scenarios will have considerably faster cycle times.

DESIGN OF AGRICULTURAL ROBOTS: EFFECTS OF ROBOT PARAMETERS

INTRODUCTION

The advent of intelligent machines capable of selective harvesting promises to raise the quality of fresh produce, lower production costs and reduce the drudgery of manual labor. An important factor in the overall performance of an agricultural robot is selecting the most appropriate manipulator for the specific task and defining its motions. When selecting a manipulator, the question is not only which type to use, Cartesian, cylindrical, spherical or articulated, but how many and how should they be configured to operate in concert? Is it better for all manipulators to do the same tasks in parallel or is it better for some manipulators to act as servers? The performance of a robot depends on its dynamic characteristics. However, when designing a robot, parameters such as speed and acceleration, must be determined to optimize performance. Regardless of the type of drive (*i.e.*, hydraulic, pneumatic, or electric), increasing acceleration raises the price exponentially. Higher acceleration requires larger drives which inherently have greater masses and thus require more torque. Since drive price is a nonlinear function of motor size, multiple slow arms might be less expensive than one fast arm. Moreover, by using multiple arms the complexity of the task performed by each robot may be reduced, and this can reduce the complexity of control systems, sensors and information processing that will further reduce costs. On the other hand, a single powerful arm requires only one control and sensor system, and may be less expensive than having multiple sensors, controls, and actuators. Therefore, the optimum configuration should be based on a cost/benefit ratio and determined by an economic analysis (Nof, 1985). An initial step for such an analysis is to determine the performance for different design parameters.

Since there are many closely related design parameters, optimal robot selection and design is an extremely difficult and time consuming task that is not yet automated (Rivin, 1988). The solution for similar problems in industry has been to simulate robot performance using commercial software packages that provide animated, graphical representation of the time-varying solutions (Tseng, 1988). By using simulation the amount of experimental research is reduced and the best robot can be selected and evaluated prior to prototype testing (Ting *et al.*, 1990; Chen and Holmes, 1990). For the design of an agricultural machine, simulation is essential due to the variability of the crop (Miles and Tsai, 1987). Through simulation, crop conditions can be repeated and altered to compare system performance for different machine parameters.

The objective of this research was to evaluate the performance of an agricultural robot for different design parameters: type of robot (Cartesian, cylindrical, spherical); number of arms (one, two, three); configuration (serial, tandem, parallel); and actuator speeds. The simulation models developed were applied to design a robotic melon harvester. Melons (*Cucumis melo* L. var. *reticulatus* Ser.) do not ripen uniformly and therefore must be selectively harvested. There is no current machine available for harvesting melons (Lenker, 1984). Melon harvesting is expensive and laborious (O'Brien and Zahara, 1971). The biological characteristics of melon plants can serve as an example and basis for use of robotics with other delicate and high-value products.

METHOD

Software Description

Alternative designs were compared by animated, visual simulations performed on a Silicon Graphics, Personal Iris workstation using IGRIP (Interactive Graphics Robot Instruction Program, Deneb Robotics, 1988). The IGRIP software is a computer graphics package for robotic workcell layout, simulation and offline programming. Parts used in the workcell are designed within the CAD system of IGRIP, and are joined to define devices of multiple degrees of freedom. A device has both geometric and non-geometric (*i.e.* kinematics, dynamics, velocities, accelerations) features associated with it. A workcell is composed of devices positioned relative to each other. Robots can be selected from an existing extensive library of the most common industrial robots or can be user-designed. Devices in the workcell are programmed using a high level programming language, GSL (Graphics Simulation Language, Deneb Robotics, 1988). GSL resembles Pascal with specific enhancements for device motion, display control and environment inquiries. Several devices can be simulated simultaneously and synchronized through input/output (I/O) signaling. The three dimensional (3-D) shaded color animation provides realism that helps confirm the validity of the simulation. Predicted cycle times are computed, allowing the different scenarios to be evaluated not only for technical feasibility but for determining harvesting rates as well. The paths necessary to accomplish the desired harvesting operations were planned, programmed and evaluated.

To enumerate the different design parameters based on a statistical analysis, dynamic, numerical simulation tools were developed. By applying numerical simulation, the time-consuming process of designing three dimensional graphical models and programming the various CAD models was avoided. Average cycle times were computed, and thereby the best number of arms and the preferred configuration and actuator speed were determined.

Robotic Harvester Model

In a two dimensional harvesting application, the assumption is that the robot is connected to a vehicle and the fruit is harvested while the vehicle is advancing along the row. The speed along the row consists of both the vehicle's speed (V_t) and the speed of the actuator moving along the axis of the row (V_x). The time to perform the task is constant and equal for each task. For a robotic harvester this consists of the time to grasp the fruit and cut it off from the stem (T_p). The time it takes the manipulator to reach the task point varies according to the distance between the robot and the target. For the harvester, this time consists of the time to reach the fruit and transport it to a conveyor.

For given constant vehicle speeds (V_t), actuator speeds (V_x and V_y) and a specific task time (T_p), the cycle time was calculated for each simulated case.

Database

Since distribution of the fruits along the row effects the design and performance of a robot, field coordinates of cantaloupes, cultivar 'Superstar', were collected on 859 melons in 29 rows at the Southwest Purdue Agricultural Center, 7 km north of Vincennes, Indiana, in the summer of 1989 (Edan and Miles, 1991). These data served as input to all simulations performed.

DESIGN PARAMETERS

Type of Robot

Two different types of commercially available arms obtained from the IGRIP library were simulated: a cylindrical robot (Seiko/RT300) and a Cartesian one (Amada/Optimo). Simulations performed on fruit coordinates of 10 different plants indicated that the Cartesian robot was 4% faster than the cylindrical one with an average cycle time of 10.8 seconds per fruit (Figure 62).

Different conveying systems configurations were simulated and evaluated to determine the minimum motions required to transfer melons. An intermediate conveyor that approaches the robot arm and transfers the picked fruit to a main conveyor (Figure 63) reduced the average cycle time to 6.2 seconds per fruit (Table 29). The robotic arm approaches a melon, picks it up, transfers it to the nearby intermediate conveyor, places it and approaches the next fruit while the intermediate conveyor accepts the fruit from the picker arm and conveys it to the main conveying system. The main conveyor consists of two conveyors that index when a melon is placed on it. Both main conveyors are beside the bed, where they don't interfere with the robot's motions. Since both picking and conveying the picked fruit are executed in parallel and the robotic arm travels less distance, the average cycle time was reduced significantly (by more than 30%).

To further increase productivity, pick and place motions were optimized by:

1. finding the optimum location for the robot base and the conveyors (*i.e.*, the location from which the robot picks the fruit in minimum-time);
2. finding the minimum-time task sequence (*i.e.*, the sequence to pick the fruit in minimum-time); and
3. finding the robot type that best suits the task.

Figure 62. Comparing percent time difference to pick one plant between a cylindrical and Cartesian industrial robot by animated visual simulation

Figure 63. Intermediate conveyor configuration with two side conveyors

Table 29. Comparing total times for picking 8-9 m rows
with a box and an intermediate conveyor

Row number	Number of fruit	Time (seconds)	
		Box	Intermediate conveyor
1	15	163	107
2	24	234	146.5
3	24	250	156.5
4	25	239	126
5	27	266	169.5

These parameters are interrelated, *i.e.*, the optimum sequence depends on robot and conveyor locations. The robot structure determines the optimum sequence: for the same task locations, *i.e.*, for the same robot base location and envelope area, the optimum sequence depends on the robot structural geometry (Abdel-Malek and Li, 1990). Different sequences will be optimum for different spatial distribution of the target points and this depends on the robot structure (Edan *et al.*, 1990). Algorithms for determining the optimum sequence to minimize the total time (Drezener and Nof, 1984; Dubowsky and Blubaugh, 1989; Abdel-Malek and Li, 1990; Borenstein and Koren, 1990) have been further developed in this work to compare different robots and find the robot that achieves the global minimum time, *i.e.*, for each type of robot all possible paths from all possible placements of the robot base were searched. The costs of the minimum-time path for each robot were compared. The sequence of motion was obtained by solving the traveling salesman problem (TSP, Christofides, 1975). By comparing the minimum travel times obtained for different robots, the most time-efficient robot was selected. For the melon harvesting task, comparisons were performed for a cylindrical and a Cartesian planar robot (Figure 64). All actuator speeds were assumed to be 1 cm/sec. The minimum and maximum link lengths were 30 and 120 cm. An extensive search through all possible paths and base locations of the robot was performed to determine the minimum-time sequence in which to pick the fruit. The Cartesian robot was faster for 266 of the 276 cases and therefore more points are on the positive side of the Y axis. Additional advantages of a Cartesian robot are:

Figure 64. Time difference between a cylindrical and Cartesian robot to pick one plant by applying the traveling salesman algorithm

1. Prismatic joints usually have higher resolution, equal and constant spatial resolution (*i.e.*, fixed through all axes of motion and throughout the work volume, Koren, 1985).
2. The control of such a robot is much easier since the transfer from world coordinates to joint coordinates is simple. Moreover, rotational axes usually result in a large amount of kinematic and dynamic coupling among the links with a resultant accumulation of errors. Dynamic performance is easier to achieve in a Cartesian robot (Rivin, 1988).
3. Since the required work volume of the robotic melon harvester is a rectangle, a longer horizontal link will be required in the case of the revolute joint to reach all fruit. Moreover, for 34 cases out of the 276 tested, not all fruit were reachable from all possible locations of the robot base in the fruit cluster area for a cylindrical robot. This means, that in order to pick all fruit, the cylindrical robot sometimes will need to move to different base locations. This will add to the control complexity of the system. The Cartesian robot can reach all fruit from the central base location without moving.
4. Since there are no obstacles, revolute joints are not required (that would have been necessary in case of obstacles since they are better able to maneuver around obstacles, Koren, 1985).

Number of Arms

Average cycle times were determined for up to six arms working in parallel across the row, assuming non overlapping workspaces and therefore no need to deal with collision avoidance (Figures 65-70). An additional assumption was that two or more arms can improve performance while maintaining equal utilization of each arm only if each has to harvest equal numbers of fruit. Therefore, additional data on melon locations were generated assuming uniform distributions across the row. The average cycle times were determined for different speeds of both the transverse (V_y) and longitudinal (V_x) actuators for the measured field data and uniformly distributed data.

For the uniformly distributed data, each arm harvests the same number of fruit and is utilized the same percentage of time regardless of position across the melon bed (Figure 65). The utilization of each arm decreases as the number of arms increase. For field data, the center arms are heavily utilized because fruit are clustered in the middle (Figure 66). Therefore, adding arms does not significantly increase machine capacity when each robot is restricted to harvesting the same width of the row.

The effect of number of manipulators on the average time to harvest a melon for forward speeds of 100 *cm/sec* and 25 *cm/sec* is illustrated in Figures 67 and 68. Increasing the number of manipulators decreases the time per fruit; however, the rate of decline decreases as the number of arms increase and is less than 1% for more than three arms. The influence of the transverse actuator speeds decreases as the number of arms increase. Comparing Figure 67 to 68 reveals that increasing forward speed decreases the time per fruit more significantly than increasing the transverse actuator speed.

Figure 65. Utilization of multiple arms using uniformly distributed melons

Figure 66. Utilization of multiple arms using field data

Figure 67. Effects of number of manipulators and actuator speeds on harvest cycle time for a forward speed of 100 cm/sec using field data

Figure 68. Effects of number of manipulators and actuator speeds on harvest cycle time for a forward speed of 25 cm/sec using field data

Figure 69. Effects of number of manipulators and actuator speeds on harvest cycle time for a transverse speed of 100 cm/sec using field data

Figure 70. Effects of number of manipulators and actuator speeds on harvest cycle time for a transverse speed of 25 cm/sec using field data

The effect of number of manipulators on the average time to harvest a melon for transverse actuator speeds (Y axis) of 100 *cm/sec* and 25 *cm/sec* is illustrated in Figures 69 and 70. Increasing forward speeds significantly decreases the average cycle time for harvesting. When the number of arms is greater than 3, changing the transverse actuator speed from 25 to 100 *cm/sec* speed has little influence on cycle times.

Configuration, Actuator Speeds, Picking Time and Tractor Speed

The configurations considered were:

- Single: one arm harvesting all the fruit growing across the bed (Figure 71a).
- Parallel: two identical arms doing the same task, each harvesting the fruit on half a bed and conveying the fruit to a nearby conveyor (Figure 71b).
- Sequential: two arms, the first arm harvests the fruit and transfers it to the nearby second arm that transfers the picked fruit to the main conveyor (Figure 71c).
- Tandem: two identical arms acting in a tandem mode. One is following the other. Each arm is responsible for picking and conveying the fruit (Figure 71d).

Figure 71. Schematic diagram of the different configurations: (a) 1 arm (b) 2 parallel arms (c) 2 sequential arms (d) 2 tandem arms

To increase productivity and minimize the travel time between consecutive task points, the dynamic capabilities of the robot must be fully utilized, *i.e.*, move at the maximum possible velocity. In a two dimensional field application (*e.g.*, harvesting of fruit, vegetables; transplanting; controlling weeds) the robot is connected to a vehicle and tasks are performed while the vehicle advances along the row. The combined forward speed (V_f) along the row (direction X) consists of both the vehicle's speed V_t and the speed of the longitudinal actuator (V_x):

$$V_f = V_t + V_x \quad [89]$$

The forward speed can be increased by increasing the speed of the X actuator (*i.e.*, the actuator moving along the row) and allowing the actuator to move in the direction of motion a distance D_b (*i.e.*, while the vehicle advances the arm can perform tasks that are up to D_b from the current base position). When tasks are grouped in clusters, the robot can perform some before the vehicle reaches the task center, some while the vehicle is over the center, and some after the vehicle has passed. While the vehicle advances along a distance D_b , the arm must perform all tasks in this area or it is less than 100% efficient. The time the tractor advances along this distance is $\frac{D_b}{V_t}$, assuming a constant velocity of the vehicle. The time to perform the task, *i.e.*, the task time is denoted as T_p and is constant, *i.e.*, equal for each task. This time does not include the time to travel towards the task. For a melon harvester, this task time consists of the time to pick the fruit and cut it off from the stem. The time to perform all tasks is T_p times N where N are the number of tasks. The time it takes the arm to reach one task point is T_i . This time varies according to the distance between the robot and the target. For the harvester, this time consists of the time to reach the fruit and transport it to a conveyor. The total time it takes the arm to reach all task points is T_t defined as:

$$T_t = \sum_{i=0}^N T_i \quad [90]$$

Since the distances between tasks along the row are not distributed uniformly, this additional degree of freedom (D_b) permits the robot to perform a task simultaneously with advancing along the row. This can be summarized in the following equation:

$$\frac{D_b}{V_t} \geq NT_p + T_t \quad [91]$$

The number of task points within the frame (D_b) is denoted as N and can be estimated from the average distance between neighbor tasks as $\frac{D_b}{D_x}$ where D_x is the average distance between tasks along the row. Therefore, the average forward speed of the vehicle can be estimated:

$$V_t = \frac{D_x}{T_p + T_t} \quad [92]$$

and is limited:

$$V_t < \frac{D_x}{T_p} \quad [93]$$

To utilize both actuators fully, the speed should be determined so that the motions in both X and Y directions would be completed at the same time. That is:

$$\frac{V_y}{D_y} = \frac{V_f}{D_x} \quad \text{or} \quad V_y = \frac{D_y}{D_x} V_f \quad [94]$$

If this condition does not exist then one actuator will be idle while the other is still moving; that is, it is over designed. For a given constant tractor speed (V_t), actuator speeds (V_x and V_y), a specific task time (T_p), and a given frame size (D_b), the average cycle time can be calculated.

The tandem configuration was the fastest followed by the parallel, sequential mode and a single arm respectively for different longitudinal actuator speeds evaluated (Figures 72 and 73). Figure 72 illustrates the relationships for a transverse actuator speed (V_y) of 11.8 cm/sec and Figure 73 illustrates this for a speed of 100 cm/sec. For all cases, increasing the longitudinal actuator speed decreases the cycle time. For V_x greater than 200 cm/sec, the cycle time declines gradually. As V_y increases the difference between the configurations becomes smaller.

Figure 72. Effects of configuration and longitudinal actuator speeds on harvest cycle time for a transverse speed of 11.8 cm/sec using field data

Figure 73. Effects of configuration and longitudinal actuator speeds on harvest cycle time for a transverse speed of 100.0 cm/sec using field data

The tandem configuration was the fastest for different transverse actuators speeds (V_y) evaluated for forward speeds of 12.5 and 62.5 cm/sec (Figures 74 and 75). For all cases, increasing the V_y actuator speed reduces the cycle time. As V_f increases the difference between the configurations becomes smaller. Transverse actuator speeds exceeding 50 cm/sec have little effect on cycle times.

Transverse actuator speeds exceeding 100 cm/sec have little effect on cycle times for the tandem configuration (Figure 76). Increasing V_y from 100 cm/sec to 200 cm/sec reduced the time per fruit by 0.74% for V_x of 25 cm/sec and by 3.2% for a V_x of 100 cm/sec. The time per fruit declines linearly as the longitudinal actuator speed increases.

An increase in longitudinal speeds (V_x) beyond 400 cm/sec results in small changes in the cycle time for the tandem configuration (Figure 77). Increasing V_x from 100 cm/sec to 200 cm/sec decreases the time per fruit by 30%. Comparing Figures 76 and 77 indicates that increasing the longitudinal actuator speed, decreases the required cycle time more significantly than increasing the transverse actuator speed.

Task Times, Accelerations and Frame Sizes

To select a motor, the torque needs to be specified and therefore accelerations and masses need to be determined. Masses were determined from system requirements (such as payload). Accelerations were derived from the performance/velocity analysis, assuming a trapezoidal or triangular velocity profile depending on distance traveled (Edan, 1990). The effects of frame

Figure 74. Effects of configuration and transverse actuator speeds on cycle time for a longitudinal speed of 12.5 cm/sec using field data

Figure 75. Effects of configuration and transverse actuator speeds on cycle time for a longitudinal speed of 62.5 cm/sec using field data

Figure 76. Effects of transverse and longitudinal actuator speeds on cycle times for the tandem configuration

Figure 77. Effects of longitudinal and transverse actuator speeds on cycle times for the tandem configuration

sizes and accelerations on percentage of fruit picked is described in Figure 78. As acceleration increases, the percentage of fruit successfully picked increases. As the frame size increases until 1 m, the percentage of fruit picked increases. For frame sizes between 1 and 2 m the percentage of fruit successfully picked does not change. For frame sizes larger than 2 m, the percentage of fruit successfully picked declines gradually for all accelerations. The slight decline at larger frame sizes is because more time is spent on conveying each fruit due to the larger distances that the robot has to travel to the discharge point.

Effects of frame size and tractor speeds on percentage of fruit picked is illustrated in Figure 79. For the fastest tractor speed simulated (66 cm/sec), the percentage of fruit picked decreases as the frame size increases. This is because the total percentage of fruit successfully picked is low. For a tractor speed of 33 cm/sec the percentage of fruit picked for frame sizes up to 2 m remains constant and beyond that, decreases. For slow tractor speeds of 11 and 22 cm/sec the percentage of fruit picked increases for frame sizes up to 100 cm, then stays constant for frame sizes from 1 to 2 m and then decreases for frame sizes greater than 2 m.

The effect of picking times and frame sizes on percentage of fruit picked is illustrated in Figure 80. As the frame size increases until 0.8 m, the percentage of fruit picked increases. For frame sizes between 0.8 and 2 m the percentage of fruit picked does not change, except for a slight increase for picking times of 0.5 seconds and 2 seconds at a frame size of 1 m. For frame sizes larger than 2 m there is no increase in the fruit percentage successfully picked for the different picking times, tractor speeds and accelerations evaluated.

Figure 78. Effects of frame sizes and accelerations on percentage of fruit picked

Figure 79. Effects of frame sizes and tractor speeds on percentage of fruit picked

Figure 80. Effects of frame sizes and picking times on percentage of fruit picked

Additional simulations (Figure 81) indicated that less than 65 percent of the fruit is picked when tractor speed exceeds 22 cm/sec (0.8 km/hr) and only one arm with a 1.5 second picking time is used, even if its acceleration is as high as 30 m/sec^2 . For a slower tractor speed of 11 cm/sec (0.4 km/hr), acceleration of 10 m/sec^2 and picking time of up to 2 seconds, more than 80 percent of the fruit can be successfully picked with one arm. Two arms are necessary if the tractor speed is to be increased. Two tandem arms pick 90 percent of the fruit with a task time of 1.5 seconds, acceleration of 10 m/sec^2 and tractor speed of 22 cm/sec (0.8 km/hr ; Figure 82). If task time is increased to 2 seconds only 80 percent of the fruit are picked for the same conditions.

SUMMARY AND CONCLUSIONS

The performance of robotic melon harvesting has been simulated to quantify effects of type of robot, number of arms, configuration (parallel, tandem, serial), and actuator speeds. Animated, visual simulation initiated the design process and permitted assessment of alternative concepts and designs in a timely manner. The motions required to grasp, detach and place melons in containers were planned, programmed, simulated and evaluated. From the animated visual simulation, the following conclusions were reached:

Figure 81. Effects of accelerations, tractor speeds on percentage of fruit picked for a picking time of 1.5 seconds and frame length of 2 m

Figure 82. Effects of accelerations, configurations and picking times on percentage of fruit picked for a tractor speed of 22 cm/sec and 2 m length frame

1. The Cartesian robot evaluated was faster than the cylindrical.
2. More than one arm will be necessary if the fruit are picked while the tractor advances along the row.
3. Picking the fruit while the tractor is advancing along the row can only be enabled for very slow forward speeds and even then, many fruit are missed. Therefore, to succeed in picking while advancing along the row, algorithms need to be developed to optimize the arm's motions.

Animated visual simulation was utilized to simulate the real-world, verify all robot paths and motions, reduce cycle times and evaluate alternative designs in a timely manner. To optimize performance based on statistical and sensitivity analysis, numerical simulation tools were developed and applied.

Dynamic, numerical simulation on measured field data indicated that increasing the number of arms beyond two does not reduce the cycle time significantly. This is because the fruit are clustered around the center of the plant and arms harvesting the sides of the row were underutilized. Therefore, activating two arms in tandem was the fastest configuration. Even for uniformly distributed melons, adding more than three arms does not reduce time per fruit significantly.

Based on the simulations performed, the actuator speeds should be: V_f equal to 200 *cm/sec*; and V_y equal to 100 *cm/sec*. Higher speeds will only slightly reduce the cycle time. The picking time has a large impact on the magnitude of accelerations required to harvest the fruit and therefore should be reduced to a minimum. Two arms are necessary to increase the tractor speed. Two tandem arms can succeed in picking 90 percent of the fruit with a picking time of 1.5 seconds, acceleration of $10m/sec^2$ and tractor speed of 22 *cm/sec* (0.8 *km/hr*). The frame size should be between 1 and 2 *m*. Larger frames will result in decrease of percentage of fruit successfully picked, since the arm will have to travel longer distances. Smaller frame sizes cause many fruit to be missed.

DESIGN OF AGRICULTURAL ROBOTS: EFFECTS OF HORTICULTURAL PRACTICES

ROBOTIC MELON HARVESTING

The model for the robotic melon harvester consists of a robot connected to a frame attached to a vehicle that advances along the row. This vehicle can be an autonomous mobile one or manually operated. Initially, the robotic melon harvester will be attached to a frame that is connected to a tractor. Fruit are picked while the vehicle advances along the row. Different robot designs and motions have been planned, modeled, simulated and evaluated (Edan and Miles, 1989; Edan, 1990). System performance was evaluated for the following robot design parameters: number of arms, configuration (*i.e.*, serial, parallel, tandem), actuator speeds and picking times (Edan and Miles, 1991). Average cycle times were determined for up to six arms working in parallel across the row, assuming non overlapping workspaces and therefore no need to deal with collision avoidance. The configurations considered were:

- Single: one arm harvesting all the fruit growing across the bed.
- Parallel: two identical arms doing the same task, each harvesting the fruit on half a bed and conveying the fruit to a nearby conveyor.
- Sequential: two arms, the first arm harvests the fruit and transfers it to the nearby second arm that transfers the picked fruit to the main conveyor.
- Tandem: two identical arms acting in a tandem mode. One is following the other. Each arm is responsible for picking and conveying the fruit.

The combined forward speed (V_f) along the row (direction X) consists of both the vehicle's speed V_t and the speed of the actuator moving in the axis of the row (V_x). Different forward, longitudinal (V_x), transverse (V_y) actuator speeds and picking times (T_p) were simulated and evaluated.

PERFORMANCE RESULTS

Cycle times were evaluated for the actual coordinates of the measured data (859 fruit) and the stochastically generated data. For each planting distance, the computed cycle times were an average of the three different seed numbers (1000 fruit each).

Frame sizes, planting distances, picking times and tractor speeds each significantly effect percentage of fruit picked (Figures 83 and 84). For the same forward speed and picking time, more fruit are successfully picked for the larger planting interval. For the same planting interval and forward speed, less fruit are picked for the larger picking time (Figure 83). Increasing picking times increases the difference between the percentage of fruit successfully picked for different forward speeds (Figure 84). The higher the forward speed the larger the gradient of change for frame sizes larger than 2 m. Maximum harvest efficiency is achieved for the slowest forward speed (11 cm/sec), fastest picking time (0.5 sec) and largest planting distance (125 cm).

Figure 83. Effect of frame size on percentage of fruit picked for frame sizes of 25 and 50 cm; picking times of 1.0 and 1.5 seconds; and forward speeds of 22 and 33 cm/sec

Figure 84. Effect of frame size on percentage of fruit picked for planting distances of 75 and 125 cm; picking times of 2.0 and 0.5 seconds; and forward speeds of 11 and 22 cm/sec

Figures 85 and 86 illustrate effects of planting distances, configuration and transverse actuator speeds on cycle times for V_f scaled so that both the transverse and longitudinal motions are completed at the same time. The cycle time decreases as the distance between plants increase.

Effects of planting distances, forward speeds for V_y scaled so that both transverse and longitudinal motions are completed at the same time, and different configurations is illustrated in Figures 87 and 88. Increasing forward speed decreases the cycle time required to harvest the fruit. For a given forward speed, planting distance has little effect on cycle time.

EXPERIMENTAL CULTURAL PRACTICE

Simulation

Since melons are concentrated in the center of the row, adding arms decreases the cycle time, but not significantly because the outer arms are underutilized (Edan and Miles, 1991). Adding multiple arms operating along the row center would add complexity since collisions between arms would have to be avoided. The tandem configuration was the most time-efficient configuration for the conventional melon pattern growth (Edan and Miles, 1991). Therefore, activating two arms along the center, where most of the fruit are located, is faster than activating two parallel arms across the row. An assumption is that improved performance could be achieved by two arms operating in parallel, if the number of melons could be distributed uniformly across the row. Each arm would operate on half a row and therefore pick the same number of fruit. The best performance was achieved for a planting distance of 125 cm. By

Figure 85. Effects of planting distance and configuration on cycle times for a transverse actuator speed of 11.8 cm/sec

Figure 86. Effects of planting distance and configuration on cycle times for a transverse actuator speed of 100 cm/sec

Figure 87. Effects of planting distance and configuration on cycle times for a longitudinal actuator speed of 11.8 cm/sec

Figure 88. Effects of planting distance and configuration on cycle times for a longitudinal actuator speed of 62.5 cm/sec

increasing the distance between plants on each half row, the overall forward speed can be increased. However, this has to be achieved without decreasing the total number of plants or fruit. These studies lead to the following suggested growth pattern which maintains the same growing area (22500 cm^2) for each plant, but in a different format: instead of planting one plant across the row every 125 cm (Figure 89a), 2 plants are planted every 250 cm in an alternating sequence (Figure 89b): the distance between consecutive plants remain 125 cm, however the distance between each plant on one half of the row is increased to 250 cm. By alternating plants across the row, each robot is responsible for picking the same number of fruit. Since the distribution pattern is such that fruit are clustered around the center, fewer fruit are located along the center of the row and therefore the common workspace problem is reduced. To evaluate system performance data that corresponded to this proposed growing pattern was generated for various planting distances.

Simulation results for two arms picking simultaneously, each on half a row, on this proposed cultural design (Figure 90) indicate that the average cycle time is 1.31 seconds per fruit. Since 2 fruit are picked concurrently, the total time that accumulates is the maximum time between the two systems and the time per fruit is the total time divided by the number of fruit. This is a 56.3 percent reduction from the tandem configuration on the conventional cultural practice, for a picking time of 1.5 seconds. This alternative spacing favors two parallel arms since now each arm has less transverse distances to travel.

Figure 89. Schematic diagram of the revised alternating planting pattern

Figure 90. Effect of planting distance on cycle times for the tandem configuration and for 2 arms acting on conventional and experimental planting distances

Experimental Field Data

To evaluate the effect of the suggested cultural practice on the melon growth patterns, a field experiment was performed. Three rows of cantaloupes, cultivar 'Superstar', were planted at a spacing of 895 *cm* (Figure 89a) and three rows were planted at a distance of 125 *cm* between plants in an alternating sequence (Figure 89b), both at the O'Neal Vegetable Memorial Research Farm, Purdue University, Lafayette, Indiana, 1990.

A two dimensional plot of the planar distribution of melon locations is illustrated in Figure 90. The first three rows (0-600 *cm*) indicate the proposed experimental cultural practice while the last three rows (600-1200 *cm*) indicate the conventional practices. The second plant in the second row and the last plants in the first and second rows, did not develop and therefore are missing. In the first row the vines were left as is and not disturbed. In the second and third rows the vines were manually spread out at 1 week intervals in order to increase the distance between melons.

The number of melons in the experimental planting distance decreased, as compared to the conventional planting practice, however, this is probably because of the missing plants that did not develop. The average distance *X* increases and therefore the forward speed can be increased for a fixed longitudinal actuator speed (Table 30). The average distance *Y* of the experimental planting distance also increases as compared to the conventional planting distance (Table 30). However, this does not impose a problem if two arms are operating across the row since the average distance *Y* for each arm remains similar. The *X* and *Y* distribution from the cluster centroid decreases. The *X* distance to the nearest fruit remains the same. The distribution curves are similar as can be seen in the histogram of average distance between clusters, *X* and *Y* distances from cluster centroid as displayed in Figures 92, 93, 94, and 95 accordingly.

Table 30. Statistics of melon locations (experimental data)

<i>Statistics</i>	<i>Average</i>		<i>Standard deviation</i>	
	<i>Conventional</i>	<i>New</i>	<i>Conventional</i>	<i>New</i>
Number of melons	85	71	-	-
Average distance <i>X</i>	38.0	43.3	65	8.3
Average distance <i>Y</i>		0.1	38.0	5.5
Average distance <i>X</i> between cluster centers	132.0	158.0	1645.0	878.0
Average <i>Y</i> of cluster centers	39.3	15.0	17.1	154.0
<i>X</i> distance to nearest fruit	10.0	10.1	62.3	36.4
<i>X</i> distance from cluster centroid	12.3	10.9	49.6	26.1
<i>Y</i> distance from cluster centroid	5.0	0.0	43.0	166.0
Average <i>X</i> distance between clusters	14.1	14.8	156.7	182.5

Figure 91. Spatial distribution of melons in 6 experimental rows

Figure 92. Histogram of X distance between clusters for experimental melon data

Figure 93. Histogram of X distance to nearest fruit for experimental melon data

Figure 94. Histogram of X distance from cluster centroid for experimental field data

Figure 95. Histogram of Y distance from cluster centroid for experimental melon data

SUMMARY AND CONCLUSIONS

Simulation was used to evaluate the effect of fruit distribution on the performance of a robotic melon harvester. The simulations indicated that horticultural practices highly effect machine performance and therefore should be considered throughout the design phase of an agricultural robot.

To increase the forward speeds along the row and reduce the acceleration requirements on the robot's actuators the planting distance should be increased as large as permitted by the melon cultural practice (*i.e.*, so that yield is not reduced). An experimental cultural practice that consisted of one plant on each half row in an alternating sequence was simulated and field tested. The planting interval is twice the conventional one, thus allowing each plant the same growth area. The distance along the row increases while maintaining an equal number of fruit on both sides of the row. Therefore, activating two arms in parallel improves the cycle time. This cultural practice decreased the cycle time by 56.3% and achieved a cycle time of 1.31 seconds per fruit with actuator speeds of 100cm/sec and a 1.5 seconds picking time as compared to 3.84 seconds per fruit for the tandem configuration with similar parameters.

INTELLIGENT CONTROL SYSTEM SIMULATION OF AN AGRICULTURAL ROBOT

INTRODUCTION

The advent of intelligent machines in agriculture has the potential to raise the quality of fresh produce, lower production costs and reduce the drudgery of manual labor. Robots are perceptive machines that can be programmed to perform a variety of agricultural tasks such as spraying, cultivating, transplanting and harvesting. However, a robot acting in an agricultural environment must contend with an uncertain and ill-structured environment that requires development of technologies to solve difficult problems such as:

1. Mobile operation in a three-dimensional, changing track;
2. Random location of targets, *i.e.*, fruit;
3. Variability in fruit size and shape;
4. Delicate products; and
5. Hostile environmental conditions like dust, dirt and extreme temperature and humidity.

Thus, an agricultural robot must be equipped with sensing, scheduling and adaptive planning capabilities to perform a number of semi-independent activities (Nagdhay, *et al.*, 1988; Cox and Gehani, 1989). Examples are:

1. Control of different actuators and sensors;
2. Monitoring and interpretation of data from multiple sensors;
3. Representation, coordination and communication of sensory information;
4. Planning of activities that must be performed in parallel;
5. Asynchronous interaction between subsystems to adequately respond to dynamic changes in environmental conditions;
6. Synchronization of the robot with external events; and
7. Dynamic planning to adopt, retain and discard control algorithms in response to changes in the surroundings.

To achieve fast and robust operation, an efficient control system must be developed (Kak, *et al.*, 1986; Ish-Shalom, 1987; Waldon, *et al.*, 1987). The major architectures for real-time control of complex systems are: hierarchical control, message based and blackboard systems (Saridis, 1983, 1988; Brooks, 1986; Elfes, 1986; Nii, 1987; Meystel, 1988). The controller structure should define how to organize the sensory data, how information is coordinated and communicated, and how sensed data should be synchronized with motion planning and task execution. Once designed and modelled, this control system must be verified. Verification of the control architecture must confirm that algorithms and computer programs execute as intended before implementing it on a robot performing in the real-world.

The control architecture which is a qualitative model cannot be verified using numerical algorithms since it involves functional and logical flow of data, that cannot be evaluated analytically. Graphic simulators have been developed to design robotic systems and recently to serve as part of their control system (Nof, 1991). Using computer graphics, operations are programmed, animated and observed through a dynamic, realistic, three dimensional visualization

of the robot mechanisms. In the agricultural environment, simulation is essential because of the variability of crop conditions (Miles and Tsai, 1987). Experiments are difficult to reproduce since crop conditions change with time and vary spatially in the field. Performance differences caused by changes in the control algorithms are often impossible to distinguish from those caused by changes in crop conditions.

The objective of this work was to develop an intelligent control system for an agricultural robot. Specific objectives were to:

1. define how to organize sensory data;
2. define how information is coordinated and communicated;
3. develop methods for on-line integration of sensed data with motion planning and execution;
4. develop methods for verifying the control model and motion control algorithms before implementation in a real-world; and
5. evaluate motion control algorithms for a robotic melon harvester.

This paper structures an intelligent controller for agricultural robots and implements this structure for robotic melon harvesting. Melons (*Cucumis melo* L.) do not ripen uniformly and therefore must be selectively harvested. There is no current machine available for harvesting melons (Lenker, 1984). Melon harvesting is expensive and laborious (O'Brien and Zahara, 1971). The biological characteristics of melon plants can serve as an example and basis for use of robotics with other delicate and high-value products.

Evaluation of motion control algorithms at the design phase reduces development time and ensures improved performance at the implementation phase. The successful development of an efficient and intelligent controller is an essential prerequisite to development of a mobile, autonomous field robotic harvester. An agricultural robot will increase product efficiency and improve overall fruit quality (Sistler, 1987).

CONTROL SYSTEM ARCHITECTURE

A robot performing in the uncertain and ill-structured agricultural domain must be equipped with sensors. Multiple and different types of sensors are required to extract accurate information on the robot's surroundings because of the variability and disturbances in the agricultural environment and limitations intrinsic to any kind of sensor. This multitude of data must be sensed and processed in real-time to form information and knowledge such as existence and location of the fruit. Based on this knowledge, tasks and motions must be planned, executed and modified in real-time corresponding to changing conditions. The blackboard model was selected since it provides a powerful mechanism for structuring various types of sensory data and for integrating diverse information since both qualitative (rule-base, heuristic) and quantitative methods (numerical solutions) can be coupled (Engelmore and Morgan, 1988). For each sub-problem and at each stage of the solution formation, the best knowledge representation and solution strategy can be selected. This is important for robust operation in the dynamic agricultural environment where for example, changes in light conditions because of clouds, or shades require different image processing routines; and fruit distribution variations imply activating different motion control algorithms. The blackboard approach is useful for complex and ill-structured

problems (Nii, 1987; Barrett and Jones, 1989). In the blackboard model, opportunistic reasoning can be used and the knowledge module applied can be determined dynamically, one step at a time, rather than in a fixed and systematic sequence (Engel, 1990). Since the blackboard architecture is modular, modifications can be easily made.

To reduce complexity, the controller was divided into a high-level planning function and a low-level control function hierarchy (Figure 96): Sensing, task planning, and action are distributed between independent modules that communicate asynchronously over globally accessible blackboards at the highest possible level, *i.e.*, each module transfers only essential information. The planning level is loosely coupled, and may be implemented by either connecting the subsystems through a communication network or via shared memory. Since time is critical for real-time control, the control level must be tightly coupled, and implemented by direct communication through shared memory.

Figure 96. Intelligent control system for an agricultural robot

Vision Module

Vision provides the most important source of information about the robot's environment. Because of complexity, vision is independent and distinguished from other sensory systems. In general, vision provides navigation and object recognition and classification capabilities that involve two distinct processes: global path planning and local guidance. A hierarchy of sensors is introduced for the planning (far-vision) and control (near-vision) levels. Each system updates a blackboard and is divided into two levels: the sensor control (*i.e.*, the data acquisition algorithms, the executable programs that obtain the data), and the processing level (*i.e.*, the image processing algorithms that transform the data into information). The input to these modules are the acquired images, the output is the extracted information.

Far-vision: The machine vision sensor scans scenes ahead of the robot's current position. For autonomous mobile vehicles, it provides guidance directions and information about obstacles. For robotic harvesters, this system recognizes and classifies object information such as position, orientation, visual texture, color, shape and size of fruit. A certainty factor indicates the degree of confidence in the validity of the information.

Near-vision: The machine vision sensor is mounted on the end effector of the robot and provides visual feedback for motion control and updates of exact position and orientation of the target.

Task Planner

The task planner defines the tasks to be performed, divides them into subtasks and determines the appropriate sequence. It contains global strategies that optimize the planned sequence of tasks to minimize performance criteria such as the time to pick all the fruit or mechanical damage to the fruit. The input to this module is the information supplied by the sensory systems such as locations of fruit to be picked, orientation and position of the robot end-effector, *etc.* The output is the sequence of tasks to be executed.

Transfer Controller

This module transfers information from the planning blackboard to the control blackboard. In addition, it assigns task priorities based on local strategies that may modify the plan according to new information obtained from real-time processing of sensed data. The transfer controller provides error recovery and replanning in case of system failure, unexpected events or alarm occurrences. The input to this module is the sequence of tasks to be performed and sensory data. The output is the next task to be performed.

Trajectory Planner

This planner generates trajectories to move the robot arm from its current position to the intended target. Strategies to determine where the arm should move, when it should stop and wait for new sensory data, and how it should revise its original plans have been developed. This level is horizontally decomposed into concurrent processes for planning the reaching and wrist/grasping motions. Reach planning provides the coarse coordinates of the trajectory. Near-vision and sensor data provide information to plan wrist orientation and grasping motions.

The input is the robot's current position and the location of the next target. The output are the positions, velocities and accelerations of the robot's joints that form the trajectory that the robot's arm should follow.

Trajectory Controller

The trajectory controller senses the position of each actuator and signals each motor to comply with the planned path. Deviations and position data are communicated to the trajectory controller. The inputs to this module are the current location of the robot and the target. The outputs are control signals to the robots actuators.

Gripper Controller

The gripper controller modifies the wrist and grasp motions based on proximity information derived from near vision and other sensors. It communicates fine corrections to the trajectory controller to accurately reach the target with the required orientation. The inputs to this module are the current location and orientation of the gripper and the target. The outputs are control signals to the robot and gripper's actuators.

Sensors and Actuators Modules

These modules control the necessary sensors and actuators. Each module consists of two closely-coupled processes that communicate directly through messages: a master and a slave. The master retrieves necessary information from the blackboard, distributes it to its slave and updates relevant information provided by the slave. The slave monitors the sensors and controls the actuators. Since time is critical this is a stand-alone process that may reside in a different processor or include special purpose electronic circuits or mechanics. The input for the sensor modules are the environmental data and conditions on when to execute the data acquisition programs such as time constraints and interrupts. The output is the derived information such as speed, position, steering angle of the vehicle or location and ripeness of fruit. The input for the actuator modules are the necessary control commands, *e.g.*, move to location X,Y and the conditions on when to move. The outputs are control commands, *e.g.*, move the robot arm and steer the vehicle.

Peripheral Controllers

These modules control complex high-level devices such as additional robots performing similar tasks in parallel or performing secondary tasks.

ROBOTIC MELON HARVESTING

Model

The robotic melon harvester consists of a robot connected to a tractor that advances along the row in two different modes: step mode and continuous mode. In the step mode, the tractor advances along the row and stops at each grouping of fruit. Then the robot arm approaches each fruit and picks the ripe ones. Once all ripe fruit are picked, the tractor advances. In the

continuous mode, the robot arm picks the fruit while the tractor is moving. The far-vision system is mounted ahead of the current position of the vehicle and robot and determines melon locations. A fan blows away the foliage to expose the melons that maybe hidden by the canopy. The near-vision system is mounted on the robot end effector robot and provides local guidance of the arm to the next fruit to be picked.

Software Description

Animated motions of the robot and gripper were displayed using IGRIP (Interactive Graphics Robot Instruction Program version 1.6, Deneb Robotics, IGRIP, 1988) on a Silicon Graphics, Personal Iris 4D/20 workstation (Silicon Graphics, 1988). The IGRIP software is a computer graphics based package for robotic workcell layout, simulation and offline programming. A workcell is composed of devices and robots, positioned relative to each other. Devices in the workcell are programmed using the high level GSL language (Graphics Simulation Language, Deneb Robotics, GSL, 1988). GSL provides the ability to construct programs for individual devices in a simulation to govern their actions and behavior. Several devices can be simulated simultaneously and are synchronized through Input/Output signaling.

Database

To evaluate the performance of the control structure based on realistic input, exact field coordinates of *Cucumis melo*, L. var. *reticulatus* Ser., cultivar 'Superstar', were collected on 859 melons in 29 rows at the Southwest Purdue Agricultural Center, 7 km north of Vincennes, Indiana, in summer of 1989 (Edan and Miles, 1991). These data served as the input to all simulations performed. The average distance between adjacent fruit along the row (X) was 34 cm with a standard deviation of 10 cm. The average distance between fruit across the row (Y) was 23.6 cm with a standard deviation of 14.2 cm. The histogram of distance between fruit clusters best-fits the Gamma distribution curve (Edan, 1990). This distribution was used to generate similar datasets corresponding to common horticultural practices. Melon locations were obtained by changing plant distances between 25 and 125 cm (Edan, 1990). The number of fruit per plant were computed using the equations derived by Bhella (1984). For each planting distance, three different random seed numbers were used to generate the data using the inverse transformation method (Pritsker, 1986).

Implementation

Each module of the intelligent controller was connected to a different device in the IGRIP workcell and had a separate GSL program (Figure 97). All programs were executed concurrently. All information is exchanged through blackboards that are implemented by global accessible data files. Lists of observations from the sensory systems arrive at the blackboard asynchronously. All references to the blackboard are part of a transaction. Each module keeps track of its current status by storing time flags in a dedicated control file. Data obtained at different times are saved in sequentially numbered files. To ensure access of relevant data, each module checks the time flag in its own control file and the control file of the module responsible for deriving the required data before accessing data.

Figure 97. Description of the IGRIP simulated modules

The far-vision system was implemented by simulating a process that provides at distinct time intervals fruit locations, size, ripeness indicator (unripe or ripe) and certainty factor of detection. Fruit locations are read from data files that contain the measured melon coordinates. Once the *simulated* vision program is completed, melons are displayed into the workcell:

- a green melon indicates that it is unripe (small or green as detected by the far-vision system);
- an orange melon indicates that it is ripe (and has to be further investigated by the near-vision system to determine exact ripeness stage).

Based on these locations, the task planner determines the stopping point of the tractor and the sequence of fruit to be picked. The transfer module transfers information from the planning blackboard to the control blackboard. Based on this list of fruit and new fruit detected by the near-vision system, it defines the next fruit to be picked. The robot controller guides the arm to this fruit. Once the robot arm is located above a fruit to be picked, it starts moving and is guided by the near-vision system that also determines the ripeness stage. Based on local guidance heuristics the arm is positioned above the fruit and waits for a signal from the near-vision system. Once the fruit has been detected by the near-vision system and depending on its ripeness stage the robot either picks the fruit or restarts the picking cycle with the next fruit. The near-vision system is implemented by simulating a process that generates various random conditions to simulate the following dynamic conditions:

- Detecting the fruit at the location indicated by the far-vision system.
- Random changes in the fruit locations (simulates inaccuracies in the far-vision system detection).
- Displaying *new* fruit into the workcell that were not identified by the far-vision system (indicated by changing the melon's color to white).
- Deleting existing fruit from the workcell (that were falsely detected by the far-vision system - this is displayed by changing the melon's color to purple and then its display to invisible).

After the arm reaches the target, a gripper is activated and picks the fruit up from the ground, detaches it from the vine and transfers it to a conveying system. This time is denoted as the picking time. Once all the fruit in the current group have been picked, the tractor advances to the median of the next cluster of fruit that have been detected. This process continues until all fruit along the row have been picked. The sequence of animated motions shown in Figure 98 is:

1. the robot arm approaches the fruit, picks it up, transfers the picked fruit to the nearby intermediate conveyor;
2. the robot arm places the fruit on the intermediate conveyor;
3. the robot arm approaches the next fruit while the intermediate conveyor conveys the fruit to the main conveying system; and
4. the robot arm picks the next fruit.

Figure 98. A sequence of the robot's motions

Performance Evaluation

The logical flow of the data was ensured for various scenarios by varying the following parameters:

1. maximum velocity and acceleration of the robot actuators (varied by changing robot parameters in the IGRIP workcell: velocities of 500, 1000 and 2000 *cm/sec* and accelerations of 0.5, 1.0 and 2.0 *g* were simulated);
2. sensor execution time (adding a delay between 1 and 500 *msec* in the vision program);
3. planning execution time (could only be increased; a delay loop was added into the task planner program);
4. number of ripe fruit;
5. distribution of fruit; and
6. detection certainty factors.

To produce a different test case for each simulation, number of ripe fruit and detection certainty factors were varied by assigning to them random numbers (generated from a uniform distribution). Percentage of ripe fruit versus unripe fruit were varied by assigning different thresholds to the random numbers that were assigned to simulate these parameters. Fruit that received a number larger than the setup threshold were simulated to be ripe, others unripe. Distribution of fruit was varied by changing the input data to consist both of measured field data and stochastically generated data (corresponding to different planting distances). Thus, for each simulation test, a different row of melon coordinates was simulated. An additional random number was assigned to each fruit at the ripeness detection stage and simulated dynamic changes in the sensory information provided by the "near" vision system (*e.g.*, changes in fruit location, and detection of fruit). Thus, the adaptive planning algorithms were evaluated for simulated real-time changes in the sensory data.

Results

The control architecture was verified by observing the motions of the robot on the screen and ensuring the following criteria:

1. All ripe fruit were picked and conveyed to the correct conveyor according to their ripeness stage; and
2. All unripe fruit were left in the field.

Through an interactive process of simulating and visualizing the motions, all algorithms were verified and it was ascertained that each submodule has the necessary information at the right time for all conditions and the programs performed correctly. This ensured that all program codes were tested before being implemented in the real-world robot. Since the performance criteria were met for all simulated parameters, for all field conditions tested and for dynamic changes in the sensory data, the implemented control architecture is considered verified.

PLANNING ALGORITHMS

Sequence Planning Algorithm

The minimum-time sequence to pick all the fruit was obtained by solving the Traveling Salesman Problem (Edan, 1990) and performing an extensive search through all possible sequences. The cost function for the TSP was calculated as in Drezner and Nof (1984) to be the distance to move the empty arm from the place on the conveyor to the next fruit. Whenever the robot arm was ready for the next target, the search was halted and the current best solution defined the next sequence of fruit to be harvested. Thus, at each point in time the best solution was obtained. The algorithm was evaluated by simulations performed for a Cartesian robot with actuator speeds of 1 *cm/sec* on locations of measured melon data. Figure 99 shows the difference between the minimum-time traveling salesman solution (optimum) and a solution received from the first evaluated sequence (regular). The total time was reduced by 49.44%.

Motion Planning Algorithms

Step mode: To prove the flexibility of the blackboard structure to integrate both qualitative and quantitative methods the following heuristics and optimization algorithms were implemented to define the tractor's motions along the growing bed in the step mode: For each fruit cluster, the median location is calculated and the tractor is stopped to align the robot's base with the median of the fruit. If the distance between sequential fruit is less than 100 *cm* and if the number of ripe fruit is less than 5 the stopping point is the median of these fruit. Otherwise, a cluster analysis was performed on all available data to determine the number of fruit clusters. The cluster analysis algorithm (Pal and Duda, 1987) was implemented by iterative optimization to minimize the sum-of-squared-error (Edan, 1990). The median is derived for each cluster.

Figure 99. Percent cycle time difference between optimum algorithm and non optimal TSP

The heuristic planning algorithms were implemented by rules in a CLIPS (CLIPS, 1989) program that is called from the GSL program. CLIPS is a rule-based, forward-chaining expert system shell and was selected since it is highly portable, easily integrated with external software and easily embedded into other systems. Optimization algorithms were implemented by integrating CLIPS with external functions written in the C programming language.

Continuous mode: To improve the robot harvester's performance the fruit should be picked while the tractor advances along the row. However, if the robot is slow and the tractor's forward speed is high and constant, the harvest efficiency is reduced, since many fruit are missed by the robot. The continuous mode algorithm was evaluated for a Cartesian robot with actuator speeds of 100 *cm/sec* on the measured field data. As the forward speed and picking time increase, fewer fruit are picked (Figure 100). Increasing the forward speed from 22 to 66 *cm/sec* decreases the percentage of fruit picked from 100 to 30 for a picking time of 0.5 seconds and from 45 to 1 percent for a picking time of 2.0 seconds.

To increase field capacity (*i.e.*, acres/hour) and attain a high percentage of picked fruit, an algorithm was developed to vary the forward speed so that the robot has just enough time to pick all the fruit. According to the number of fruit in the next cluster, the tractor slows down or speeds up: when fruit density is low the tractor speed is increased and when many fruit are clustered together the tractor slows down and waits for the robot.

Figure 101 presents the effect of picking time on the total time to harvest a row for constant and variable velocities of 11 and 66 *cm/sec*. For constant velocities, the total time decreases as the picking time increases since fewer fruit are picked. For variable velocities all fruit are successfully picked and thus, as the picking time increases the total time increases. Figure 102 shows the effect of the picking time on cycle times for constant and variable velocities of 11 and 66 *cm/sec*. The time per fruit increases as the forward speed decreases and the picking time increases. The variable velocity mode is faster than the constant velocity mode by 3.3%.

Figure 100. Effects of picking time and forward velocity on percentage of fruit picked

Figure 101. Effect of picking time on total time for variable and constant velocities

Figure 102. Effect of picking time on cycle times for variable and constant velocities

Local Guidance Algorithms

The flowchart of the robots motions is given in Figure 103: The robot arm moves down to a point 35 cm above the next fruit to be picked and waits for information from the near-vision system. The two modules are closely coupled and all communications occur directly through the control blackboard since time is critical. If the far-vision system indicated detection with high accuracy the robot arm moves down 30 cm and waits for ripeness information from the near-vision system. If lower certainty has been indicated, the arm moves down only 10 cm, to a distance of 25 cm from the fruit, to provide the near-vision system with a larger field of view. If the fruit *disappears* while the arm moves towards the fruit and the accuracy of detection was high, the arm stops and moves up and asks the near vision system to recheck, and waits for an acknowledge signal for continuation. If the fruit *disappears* while the arm moves towards it and the accuracy of detection was low, the arm proceeds to pick the next one. The execution of these modified plans are visualized on the screen, *i.e.*, actions of the robot arm change according to the simulated dynamic environmental conditions.

Figure 103. Flowchart of the robot's motions

SUMMARY AND CONCLUSIONS

An intelligent controller capable of interfacing sensors and control devices has been designed, simulated and verified for a robot acting in the ill-structured and uncertain agricultural environment. A hierarchical control system that activates through multiple blackboards was developed. Each level consists of independent modules that communicate through their corresponding blackboard. The blackboard controller was implemented by sharing data through files. The blackboard provided a flexible and modular structure that facilitated modifications and addition of modules and coupling of different problem-solving methods. This is essential for the complex agricultural domain. All stages of the data flow have been simulated: starting with input from the sensors, through transformation to information and plans, and to final execution of tasks and control. Dynamic changes of sensory data have been simulated, and algorithms, heuristics and optimization methods incorporated into the planning and execution of the motions for the robotic melon harvester. This intelligent control structure is essential for efficient and robust operation of an agricultural robot.

The simulations on measured melon locations verified data consistency and correct logical flow of the data. By observing the robot's motions for various sets of system parameters and field conditions the control architecture was verified for a variety of conditions prior to implementation in the real-world robot. Animated, visual simulation provided an important and powerful tool for verifying the control model, debugging all sensor, planning and robot control computer programs and evaluating motion control algorithms. Simulations of dynamic changes of sensory data is important for ensuring robust operation.

Evaluation of planning algorithms on measured melon locations indicated that the traveling salesman algorithm decreases the cycle times required to pick the fruit by finding a minimum-time sequence by 49% and therefore should be implemented. A continuous mode of operation decreases cycle times. Variable speed control improves performance significantly as compared to a constant velocity mode where many fruit are missed at high forward speeds. By controlling the velocity, all fruit are successfully picked and the overall speed is increased. The evaluation of these algorithms prior to prototype testing ensures improved performance and reduces the amount of experimental research.

REFERENCES

1. ASAE. 1990. ASAE Standard S368.2. Compression Test of Food Materials of Convex Shape. In: Standards 1990, 37th Edition of Standards, Engineering Practices and Data. ASAE, St. Joseph, Michigan.
2. Abdel-Malek, L.L. and Z. Li. 1990. The application of inverse kinematics in the optimum sequencing of robot tasks. *International Journal of Production Research*, 28(1):75-90.
3. Allen, P.K. 1988. Integrating vision and touch for object recognition tasks. *International Journal of Robotics Research* 7(6).
4. Andersberg, M.R. 1973. Cluster analysis for applications. Academic Press, New York.
5. Arai, T. and M. Asada. 1986. Properties of rotational hands with versatility. In: *Robot grippers*, Springer Verlag, Berlin.
6. Arai, T. and M. Asada. 1988. Robot grippers. In: D.T. Pham and W.B. Heginbotham (eds.), IFS, UK.
7. Aromat Corporation. 1988. Laser-based linear displacement photoelectric sensor. *Sensors*, July.
8. Ayach, N. and O.D. Faugeras. 1988. Building, registering and fusing noisy visual maps. *International Journal of Robotics Research* 7(6).
9. Banichuk, N.V. 1983. *Problems and methods of optimal structural design*. Plenum Press, New York.
10. Barrett, J.R. and D.D. Jones. 1989. Knowledge engineering in Agriculture. An ASAE Monograph No. 8, ASAE. St. Joseph, Michigan 49085.
11. Baz, A. and J. Vossoughi. 1986. Stress consideration in robot grippers. In: *Robot grippers*, Springer Verlag, Berlin.
12. Beckerman, M. and E.M. Oblow. 1990. Treatment of systematic errors in the processing of wide-angle sonar sensor data for robotic navigation. *IEEE Transactions on Robotics and Automation* 6(2).
13. Benady, M. 1990. A blackboard based approach to geometric sensor fusion using clips. AGEN 565 Term Project, Purdue University, W. Lafayette, IN.
14. Bhella, H.S. 1984. Response of muskmelon to within-row plant spacing. *Proceedings of the Indiana Academy of Science* 94:99-104.
15. Bixler, J.P. and D.P. Miller. 1987. A sensory input system for autonomous robots. In: M. Kaufmann (ed.), *Proceedings of the Workshop on Spatial Reasoning and Multi-Sensor Fusion*. American Association for Artificial Intelligence.
16. Borenstein, J. and Y. Koren. 1990. Task-level plan generation for mobile robots. *IEEE Transactions on Systems, Man and Cybernetics* 20(4):938-943.

17. Brooks, R.A. 1984. Aspects of mobile robot visual map making. Second Int. Symp. Robotics Research, p. 287.
18. Brooks, R.A. 1986. A robust layered control system for a mobile robot, *IEEE Journal of Robotics and Automation* 2(1):14-23. CLIPS. 1989. Reference Manual, Version 4.3, Artificial Intelligence Section, L.B. Johnson Space Center.
19. Cardenas-Weber, M., R.L. Stroshine, K. Haghighi and Y. Edan. 1989. Finite element analysis of melons handled by different robot gripper designs. Paper No. 89-1602. American Society of Agricultural Engineers, St. Joseph, MI.
20. Cardenas-Weber, M., R.L. Stroshine, K. Haghighi and Y. Edan. 1991. Melon material properties and finite element analysis of compression with applications to robot gripping. *Transactions of the ASAE*. In press.
21. Chen, C. and R.G. Holmes. 1990. Simulation and optimal design of a robot arm for greenhouse/nursery. ASAE Paper No. 90-1537, ASAE, St. Joseph, MI 49085.
22. Chen, C.H. and A.C. Kak. 1988. 3d-poly: A robot vision system for recognizing objects in occluded environments. Technical Report TR-EE 88-48, School of Electrical Engineering, Purdue University, W. Lafayette, IN.
23. Chen, P. and Z. Sun. 1984. Critical strain failure criterion: pros and cons. *Transactions of ASAE* 27(1):278-281.
24. Christofides, N. 1975. The Traveling Salesman Problem. In: *Graph Theory - an Algorithmic Approach*, pp. 236-281. Editor: W. Rheinbolt. Academic Press, New York.
25. Cox, I.J. and N.H. Gehanni. 1989. Concurrent programming in robotics. *The International Journal of Robotics Research*, 8(2):3-16.
26. DeSalvo, G.J. and R.W. Gorman. 1987. *ANSYS Engineering Analysis System Users Manual*, Volumes I and II. Swanson Analysis Systems, Inc., Houston, PA.
27. Deneb Robotics Inc. 1988. *IGRIP Simulation System*, Troy, MI.
28. Deneb Robotics Inc. 1988. *GSL Reference Manual*, Troy, MI.
29. Diehl, K.C., D.D. Hamann and J.K. Whitfield. 1979. Structural failure in selected raw fruits and vegetables. *Journal of Texture Studies* 10:371-340.
30. Drezner, Z. and S.Y. Nof. 1984. On optimizing bin picking and insertion plans for assembly robots. *IIE Transactions* 16(3):262-270.
31. Dubowsky, S. and T.D. Blubaugh. 1989. Planning time-optimal robotic manipulator motions and work places for point-to-point tasks. *IEEE Transactions on robotics and Automation* 5(3):377-381.
32. Durrant-Whyte, H.F. 1986. Integration of disparate sensor observations. In *Proceedings IEEE International Conference on Robotics and Automation*, p. 1464.
33. Durrant-Whyte, H.F. 1987. Uncertain geometry. In *Proceedings IEEE International Conference on Robotics and Automation*, p. 851.
34. Durrant-Whyte, H.F. 1988. *Integration, Coordination and Control of Multi-Sensor Robot Systems*. Kluwer Academic Publishers, Boston.

35. Edan, Y. 1990. Control and design of an intelligent agricultural robot. Ph.D. Thesis. Purdue University, West Lafayette, IN 47907.
36. Edan, Y. and G.E. Miles. 1989. Animated, visual simulation of robotic melon harvesting. ASAE Paper No. 89-7612, ASAE, St. Joseph, MI 49085.
37. Edan, Y. and G.E. Miles. 1991. Design of a robotic melon harvester. ASAE Paper No. 91-7029, ASAE, St. Joseph, MI 49085.
38. Edan, Y., K. Haghighi, R. Strohine and M. Cardenas-Weber. 1989. Finite element analysis and optimization of a robot gripper design. Paper 89-7537. American Society of Agricultural Engineering, St. Joseph, MI 49085.
39. Edan, Y., T. Flash, I. Shmulevich, Y. Sarig and U.M. Peiper. 1990. An algorithm defining the motions of a citrus picking robot. *Journal of Agricultural Engineering Research* 46:259-273.
40. Elfes, A. 1986. A distributed control architecture for an autonomous mobile robot. *The International Journal for Artificial Intelligence* 1(2):99-108.
41. Engel, B.A., D.B. Beasley and J.R. Barrett. 1990. Integrating multiple knowledge sources. *Transactions of the ASAE*. 33(4):1371-1376.
42. Englemore, R. and T. Morgan. 1988. *Blackboard systems*. Addison Wesley, Wokingham, England.
43. Erdman, A.G., T. Thompson and D.R. Riley. 1987. Type selection of robot and gripper kinematic topology using expert systems. IN: J.M. McCarthy (ed.), *Kinematics of Robot Manipulators*. MIT Press. pp 202-208.
44. Everett, H.R. and A.M. Flynn. A programmable near-infrared proximity detector for robot navigation. *SPIE Mobile Robots* 727:221-230.
45. Ferrand, C., P. Baylou and G. Grenier. 1990. A non-damaging effector for high speed transplanting robots. ASAE Paper No. 90-7501, ASAE, St. Joseph, MI 49085.
46. Flynn, A.M. 1988. Combining sonar and infrared sensors for mobile robot navigation. *International Journal of Robotics Research* 7(6).
47. Fridley, R.B., R.A. Bradley, J.W. Rumsey and P.A. Adrian. 1968. Some aspects of elastic behavior of selected fruits. *Transactions of ASAE* 11(1):46-49.
48. *GSL Reference Manual*. 1988. Deneb Robotics Inc., Troy, MI.
49. Grand D'Esnon, A., R. Pellenc, G. Rabatel, A. Journeau and M. Aldon. 1987. MAGALI: a self propelled robot to pick apples. ASAE Paper No. 87-037.
50. Harrell, R. 1987. Economic analysis of a robotic citrus harvester in Florida. *Transactions of the ASAE* 30(2):298-304.
51. Harrell, R.C., P.D. Adsit, T.A. Pool and R. Hoffman. 1988. The Florida robotic Grove-Lab. Paper 88-1578. American Society of Agricultural Engineers, St. Joseph, MI 49085.
52. Henderson, T. and E. Shilcrat. 1984. Logical sensor systems. *Journal of Robotic Systems* 1(2):169-193.

53. Henderson, T., C. Hansen and B. Bhanu. 1988. The specification of distributed sensing and control. *Journal of Robotic systems* 2(4):387-396.
54. Holliday, R. 1960. Plant population and crop yield. *Nature* 186 (4718):22-24.
55. Hoy, R.M. and W.F. McClure. 1987. A unique hollowed finger gripper with tactile feedback. ASAE Paper No. 87-3055, ASAE, St. Joseph, MI 49085.
56. Hu, G. and G. Stockman. 1987. 3-d scene analysis via fusion of light striped and intensity image. In: M. Kaufmann (ed.), *Proceedings of the Workshop on Spatial Reasoning and Multi-Sensor Fusion*.
57. Huang, C., I. White, G. Thwaite and A. Bendeli. 1988. A noncontact laser system for measuring soil surface topography. *American Journal of Soil Sciences* 52:350-355.
58. IEEE International Conference on Robotics and Automation: 975-977.
59. IGRIP Simulation System Ver 1.6. 1988. Deneb Robotics Inc., Troy, MI.
60. Imgrund, M.C. and M.J. Wheeler. 1986. Reducing design costs by integrating finite element and optimization techniques, Spring National Design Engineering Show.
61. Irudayaraj, J. 1990. Finite Element Simulation of Viscoelastic Biomaterial Behavior During Drying. Ph.D. Thesis. Agricultural Engineering Department, Purdue University, West Lafayette, IN.
62. Ish-Shalom, J. 1987. Task level specification of a robot control. *IEEE International Symposium on Intelligent Control*: 196-201.
63. Jindal, V.K. and O. Techasena. 1985. Compression tests for measuring the firmness of potatoes. Paper 85-1072. American Society of Agricultural Engineers, St. Joseph, MI 49085.
64. Kak, A.C., K.L. Boyer, C.H. Chen, R.J. Safanek and H.S. Yang. 1986. A knowledge-based robotic assembly cell. *IEEE Expert*, Spring: 63-83.
65. Kasmire, R.E. 1981. Muskmelon production in California. Division of Agricultural Sciences, University of California, Leaflet 2761.
66. Kato, I. 1982. *Mechanical Hands Illustrated*. Kogyo Chosakai Publishing Co. Ltd., Japan.
67. Kato, I. and K. Sadamoto. 1987. *Mechanical hands illustrated*. Springer Verlag, New York.
68. Kawamura, N., K. Namikawa, T. Fujiura and M. Ura. 1986. Study on agricultural robot. *Memoirs of the College of Agriculture, Kyoto University*, No. 129:29-46.
69. Kawamura, N., K. Namikawa, T. Fujiura, M. Ura and Y. Ogawa. 1987. Hand of a fruit harvesting robot. *Research Report on Agricultural Machinery* No. 17:1-7. Laboratory of Agricultural Machine, Kyoto University, Kyoto, Japan.
70. Kirsch, U. 1981. *Optimum structural design, concepts, methods and applications*. McGraw Hill Book company, N.Y.
71. Koren, Y. 1985. *Robotics for Engineers*. McGraw Hill Book Company, New York.
72. Krouse, J.K. 1986. Engineering without Paper, *High Technology* 6(3):38-43, 45-46.

73. Kumar, P. and K.J. Waldron. 1980. The dextrous workspace. Design Engineering Tech. Conf. ASME Paper No. 80-DET-108.
74. Kutz, L.J. and G.E. Miles. 1986. CAD simulation speeds robotic workcell development. Agriculture Engineering, May/June:20-22.
75. Kutz, L.J., G.E. Miles, P.A. Hammer and G.W. Krutz. 1987. Robotic transplanting of bedding plants. Transactions of the ASAE 30 (3):586-590.
76. Lazin, M.B. and S.C. Simonds. 1981. Influence on planting method, fertilizer rate and within row plant spacing on production of two cultivars of honeydew melons. Proceedings of Florida State Horticultural Society 94:180-182.
77. Leaver, S.O., J.M. McCarthy and J.E. Bobrow. 1988. The design and control of a robot finger for tactile sensing. Journal of Robotic Systems 5(6):567-581.
78. Lenker, D.H. 1984. Factors limiting the harvest mechanization of some major vegetable crops in the U.S.A. in Fruit, Nut Vegetable Harvesting Mechanization Symposium: 29-38.
79. Lin, T. and R.E. Pitt. 1986. Rheology of apple and potato tissue as affected by cell turgor pressure. Journal of Texture Studies 17:291-313.
80. Liu, Minzhu. 1989. Mechanical Properties of Soybeans during Processing. M.S. Thesis. Agricultural Engineering Department, Purdue University, West Lafayette, IN.
81. Luo, R.C. and T.C. Henderson. 1986. A servo controlled robot gripper with multiple sensors and its logical specification. Journal of Robotic Systems 3(4):409-420.
82. Mason, M.T. and J.K. Salisbury. 1985. *Robot hands and the mechanics of manipulation*. MIT Press, Cambridge, MA.
83. Melles Griot Corporation. 1989. Diode Lasers and Accessories. Melles Griot Corporation, 1770 Kettering St., Irvine, CA 92714.
84. Meystel, A. 1988. Intelligent control in robotics. Journal of Robotic Systems 5(4):269-308.
85. Miles, G.E. and Y. Tsai. 1987. Combine systems engineering by simulation. Transactions of the ASAE 30(5):1277-1281.
86. Miles, G.E., Y. Alper, U.M. Peiper, I. Wolf, Y. Edan and M. Cardenas-Weber. 1989. Robotic systems for selective harvesting: basic concepts. BARD US-1254-87, year 1 report.
87. Mizrach, A., N. Galili, D. Teitel and G. Rosenhouse. 1989. Acoustical, mechanical and quality parameters of melon tissue. Paper 89-3051. American Society of Agricultural Engineers, St. Joseph, MI 49085.
88. Mohsenin, N.N. 1986. Physical Properties of Plant and Animal Materials. Gordon and Breach Science Publishers, New York.
89. Nagdhy, F., C.K. Wai and G. Nagdhy. 1988. Multiprocessing control of robotic systems.
90. Nii, H.P. 1987. Blackboard systems: the blackboard model of problem solving and the evolution of blackboard architectures. The AI Magazine 7(3):38-53.

91. Nof, S.Y. 1985. Handbook of Industrial Robotics. John Wiley and Sons, New York.
92. Nof, S.Y. 1991. Industrial Robotics. Handbook of IE (Ed: G. Salvendy), Chapter 16, John Wiley and Sons, New York.
93. Noor, A.K., K.O. Kim and J.A. Tanner. 1990. Analysis of aircraft tires via semianalytic finite elements. The International Journal of Applied Finite Elements and Computer Aided Engineering 6(3):217-233.
94. Novak B. 1985. Robotic simulation facilitates assembly line design. Simulation 43(6):298-299.
95. O'Brien, M. and M. Zahara. 1971. Mechanical harvest of melons. Transactions of the ASAE 14(5):883-885.
96. Paden, B. and S. Sastry. 1988. Optimal kinematic design of 6R manipulators. International Journal of Robotics Research 7(2):43-59.
97. Pal, S.K. and M.D.K. Duda. 1987. Fuzzy mathematical approach to pattern recognition. John Wiley and Sons, New Delhi, India.
98. Paul, R.P. 1981. Robot Manipulators. MIT Press.
99. Pham, D.T. and W.B. Heginbotham. 1986. *Robot grippers*. Springer Verlag, New York.
100. Pritsker, A.B. 1986. Introduction to simulation and Slam II. John Wiley and Sons, New York.
101. RCA Corporation Consumer Electronics Division. 1986. CMCO20 Color Video Camera - Owner's Manual. RCA Corporation Consumer Electronics Division, 600 N. Sherman Dr., P.O. Box 1976, Indianapolis, IN 46206.
102. Raghava, R.S., R.M. Caddell and G.S.Y. Yeh. 1973. Yield locus of oriented polycarbonates. Journal of Materials Science 8¹:225-232.
103. Rivin, E.I. 1988. Mechanical Design of Robots. McGraw Hill Book Company, New York.
104. Rumsey, T.R. and R.B. Fridley. 1977. Analysis of viscoelastic contact stresses in agricultural products using a finite element method. Transactions of ASAE 20(1):162-167.
105. Saridis, G.N. 1983. Intelligent robotic control. IEEE Transactions on Automatic Control 28(5):547-557.
106. Saridis, G.N. 1988. Knowledge implementation: structures of intelligent control systems. Journal of Robotic Systems 5(4):254-268
107. Segerlind, L.J. 1984. Applied Finite Element Analysis, Second Edition. John Wiley and Sons, New York.
108. Segerlind, L.J. 1986. *Applied finite element analysis*. John Wiley and Sons, New York.
109. Segerlind, L.J. and I.D. Dal Fabbro. 1978. A failure criterion for apple flesh. Paper 78-3556. American Society of Agricultural Engineers, St. Joseph, MI 49085.
110. Serway, R.A. and J.S. Faughn. 1985. College Physics. Saunders College Publishing, The Dryden Press. Philadelphia, PA 19105.

111. Shigley, J.E. and C.R. Mischke. 1989. *Mechanical engineering design*. Fifth edition. McGraw Hill Book Company, New York, NY.
112. Silicon Graphics Reference Manual. 1988. Silicon Graphics.
113. Simon, J.E. and D.D. Daniels. 1987. Honeydew cultivar trials for southern Indiana, 1986. Station bulletin No. 528, Dept of Horticulture, Agriculture Experiment Station, Purdue University, W. Lafayette, IN.
114. Simonton, W. 1990. Robotic end effector for handling greenhouse plant material. ASAE Paper No. 90-7503, ASAE, St. Joseph, MI 49085.
115. Singer, F.L. and A. Pytel. 1980. *Strength of materials*. Harpow and Row Publishers, New York.
116. Sistler, F.E. 1987. Robotics and intelligent machines in agriculture. IEEE Journal of Robotics and Automation 3(1):3-6.
117. Sittichareonchai, A., F. Sevil, J-M. Fatou, A. Constans, A. Brons and A. Davenel. 1989. A robot to harvest grape: a feasibility test. ASAE Paper No. 89-7084.
118. Slaughter, D.C. and R.C. Harrell. 1987. Color vision in robotic fruit harvesting. Transactions of the ASAE 30(4):1144-1148.
119. Smith, M.G., K. Haghighi, G.W. Krutz, L. Bennett and J. Buchanan. 1989. Finite element modeling and design of a diesel engine chassis mount. Applied Engineering in Agriculture 5(4):527-531.
120. Solecki, J.S. 1987. Design optimization series part 1 - minimum weight of a truss bridge. ANSYS News. Fifth Issue:7-9.
121. Stanfield, S.A. 1988. A robot perceptual system utilizing passive vision and active touch. International Journal of Robotics Research 7(6).
122. Swanson Analysis Systems, Inc. 1987. Design optimization seminar notes for revision 4.3., Houston, PA.
123. Tanie, K. 1985. Design of robot hands. In: Y. Nof Shimon (ed), Handbook of Industrial Robotics. John Wiley and Sons, New York.
124. Terzopoulos, D. 1986. Integrating visual information from multiple sources. In: A.P. Pentland (ed.), From Pixels to Predicates, Ablex Press.
125. Ting, K.C., G.A. Giacomelli and S.J. Chen. 1990. Robot workcell for transplanting of seedlings. Part II. Transactions of the ASAE 33(3):1013-1018.
126. Ting, K.C., Y. Yang and W. Fang. 1990. Stochastic modeling of robotic workcell for seedling plug transplanting. ASAE Paper No. 90-1539. ASAE, St. Joseph, MI 49085.
127. Trevelyan, J.P. 1985. Skills for a shearing robot: dexterity and sensing. In: The Second International Symposium on Robotics Research. MIT Press, Cambridge, MA: 273-280.
128. Tseng, A.A. 1988. Assessment of robotic simulation software. Proceedings of the 1988 ASME Computers in Engineering Conference 2:217-226.

129. Tseng, A.A. 1990. Material characterization and finite element simulation for forming miniature metal parts. *The International Journal of Applied Finite Elements and Computer Aided Engineering* 6(3):251-265.
130. Vijaykumar, R., K.J. Waldron and M.J. Tsai. 1987. Geometric optimization of serial chain manipulator structures for working volume and dexterity. IN: J.M. McCarthy (ed.), *Kinematics of Robot Manipulators*, MIT Press. pp. 99-111.
131. Vukobratovic, M. and V. Potkonjak. 1982. *Dynamics of manipulation robots - theory and application*. Springer-Verlag, New York.
132. Waldon, S., D. Gaw and A. Meystel. 1987. Updating and organizing world knowledge for an autonomous control system. *IEEE International Symposium on Intelligent Control*: 423-430.
133. Walpole, R.E. and R.H. Meyers. 1985. *Probability and statistics for engineers and scientists*. McMillan Publishing Co., New York.
134. Ward, P.A., R.W. Wakelling and R. Russel. 1987. Design of an excavator arm using optimization techniques. SAE Paper No. 870814. Society of Automotive Engineers, Warrendale, PA.
135. Waterbury, R. 1985. Software animates assembly workcells. *Assembly Engineering*.
136. Wolf, I., J. Bar-Or, Y. Edan and U.M. Peiper. 1990. Developing grippers for a melon harvesting robot. ASAE Paper No. 90-7504, ASAE, St. Joseph, MI 49085.
137. Wright P.K. and M.R. Cutkosky. 1985. Design of Grippers. In: Y. Nof Shimon (ed), *Handbook of Industrial Robotics*. John Wiley and Sons, New York.
138. Yang, D.C.H. and T.W. Lee. 1983. On the workspace of mechanical manipulators. *ASME Journal Mechanisms, Transmissions and Automation in Design* 105(1):62-70.

DESCRIPTION OF COOPERATION

The amount and depth of cooperation between the U.S. and Israeli scientists and engineers has been outstanding and has been instrumental in the successes realized by this project. Voice communication between Purdue University and the Volcani Institute was minimized in favor of FAX and especially electronic (computer) mail. This modern day telegraph service provided seemingly instant communication on a daily basis between colleagues oceans apart. When required, remote logins via internet enabled researchers at Purdue to access data and computers in Israel.

Uri Peiper and Yekutiel Alper visited the United States in 1989 near the end of the first year to evaluate progress with the simulation models and, discuss and approve future plans for this work. Itzhak Wolf designed, constructed and tested a pneumatically activated gripper which was shipped to Purdue for further study. A videotape of its operation in the field was made in Israel. Personnel at the Volcani Institute also provided videotapes of Israeli melons grown in the Jordan Valley which were digitized and analyzed by image processing algorithms at Purdue.

The Volcani Institute also contributed significantly to this project by helping Purdue recruit two outstanding young people, Yael Edan and Meny Benady, who have made major contributions while working on advanced degrees.

Several faculty and staff not officially listed with this project have cooperated and contributed their own unique expertise in invaluable ways. Drs. Bernard Engel, Richard Stroshine, Kamyar Haghighi and Jonathan Bar-Or have advised students on projects which culminated in publications. Without Don Rhine's contributions, field data acquisition at Purdue would not have occurred. Dr. Gerald Wilcox (Professor of Horticulture) on numerous occasions provided his vast expertise on melon cultural practices and contributed field plots of the sensors studies.

EVALUATION OF THE RESEARCH ACHIEVEMENTS

SENSING OF MELONS

The machine vision algorithms developed by this project can successfully locate better than 80% of the melons in occluded, noisy scenes characteristic of field conditions in the U.S. and in Israel, but the almost 20% undetected and more importantly the large number of false detections (even with the knowledge rules included), which slow the effective harvest rate of the robot, leads to the inevitable conclusion that supplementary sensing is required. Brightness alone was able to detect 86.5% to 89% of the melons, but with numerous false detections. Lowering the brightness threshold increased the proportion detected, but substantially increased false detections. Adding knowledge rules requiring detected melons to meet constraints on size, shape, proximity to other melons, and uniformity of brightness reduced the number of false detections significantly, but the additional algorithms require additional processing time, which either raises the cost of machine vision hardware or slows the harvest rate. Initial studies with structured light and ultrasonics were performed, and it appears that laser line projection in combination with machine vision has the potential to substantially reduce the number of false detections.

Stereoscopic vision proved unable to measure distance to the fruit because the images contain noise due to leaf and camera movements. An ultrasonic sensor failed to distinguish melons from foliage on the soil surface. Triangulation, that is the use of multiple images to calculate the distance to the fruit, was shown to be an accurate method, but it requires the camera be mounted on the manipulator, eye-in-hand fashion. Triangulation is easily accomplished with a projected laser line and machine vision system. Thus we recommend this technology for further study.

Sensor fusion will be necessary to enable the robot controller to deal with uncertainties associated with conflicting information derived from multiple sensors.

MELON PARAMETERS

The distribution of *Cucumis melo*, L. var *reticulatus* Ser., cultivar 'Superstar' melons grown in Indiana was best modeled with Gamma functions. The average distance between adjacent fruit in a cluster was 34 cm within the row and 23.6 cm across the row.

Finite element models of melons were developed and used to calculate the deformations and stresses imposed on the fruit by robot grippers. The moduli of elasticity of melon skin and flesh were measured and used by the finite element models. Experimental observations of melon deformation by a flat-plate type of gripper were within 10% of the values predicted by the finite element model. The maximum deformation of a ripe melon was 1.32 times that of a green melon. The maximum stress developed in a ripe melon was 93% of the stress of a green melon. The predicted equivalent stresses were well below the failure stresses.

GRIPPER DESIGN

Stresses which develop in basket and notched-V types of grippers designed for harvesting melons were calculated using finite element models. Maximum stresses occurred at the boundaries close to the applied force. The magnitude of the stresses were proportional to the load. Experimental results with a notch-edge gripper were within 8% of the predictions using curved beam theory.

A pneumatically activated basket-type gripper was designed, constructed and manually tested. This design constrains the fruit with minimum force, thereby keeping bruising to a minimum. A cutting device was included to detach the fruit from the vine.

DESIGN OF A ROBOTIC MELON HARVESTER

Animated, visual simulation of robotic melon harvesting revealed the optimum design is sensitive to both robot parameters and cultural practices for growing melons.

The performance of robotic melon harvesting has been simulated to quantify effects of type of robot, number of arms, configuration (parallel, tandem, serial), and actuator speeds. To quantify performance sensitivity to design parameters, numerical simulation tools were developed and simulated on measured melon data. The traveling salesman algorithm was applied to select the best robot for the melon harvesting task. This algorithm was applied to a set of field data and indicated that the Cartesian robot best suits the melon harvesting task. However, one arm proved to be insufficient.

Two arms are necessary to increase the tractor speed. Increasing the number of arms beyond two does not reduce the cycle time significantly. This is because the fruit are clustered around the center of the plant and arms harvesting the sides of the row were underutilized. Therefore, activating two arms in tandem was the fastest configuration. Even for uniformly distributed melons, adding more than three arms does not reduce time per fruit significantly.

Based on the simulations performed, the actuator speeds should be: V_f equal to 200 *cm/sec* and V_y equal to 100 *cm/sec*. Higher speeds will only slightly reduce the cycle time. The frame size should be between 1 and 2 *m*. Larger frames will result in decrease of percentage of fruit successfully picked, since the arm will have to travel longer distances. Smaller frame sizes cause many fruit to be missed. Picking time has a large impact on the magnitude of accelerations required to harvest the fruit and therefore should be reduced to a minimum.

PUBLICATIONS

- Cardenas-Weber, Martha, Richard L. Stroshine, Kamyar Haghighi and Yael Edan. 1989. Finite Element Analysis of Melon Handled by Different Robot Gripper Designs. Paper No. 89-1602. The ASAE, St. Joseph, MI 49085-9659. 20p.
- Cardenas-Weber, Martha, Amots Hetzroni and Gaines E. Miles. 1991. Machine Vision to Locate Melons and Guide Robotic Harvesting. Paper No. 91-7006. The ASAE, St. Joseph, MI 49085-9659. 21p.
- Cardenas-Weber, Martha, Richard Stroshine, Kamyar Haghighi and Yael Edan. 1991. Melon Material Properties and Finite Element Analysis of Melon Compression with Application to Robot Gripping. Transactions of ASAE 34(3):920-929.
- Edan, Yael and Gaines E. Miles. 1989. Animated Visual Simulation of Robotic Melon Harvesting. Paper No. 89-7612. The ASAE, St. Joseph, MI 49085-9659. 9p.
- Edan, Yael, Kamyar Haghighi, Richard L. Stroshine and Martha Cardenas-Weber. 1989. Finite Element Analysis and Optimization of a Robot Gripper Design. Paper No. 89-7537. The ASAE, St. Joseph, MI 49085-9659. 14p.
- Edan, Yael, Bernard A. Engel, Gaines E. Miles and Uri M. Peiper. 1991. Intelligent Control of a Robotic Melon Harvester. Proc. Second France Israel Symposium on Robotics: Advances in Non-Manufacturing Robotics. April 8-10, Saclay, France. Institut National des Sciences et Techniques Nucleares. pp 6.33-6.47.
- Edan, Yael and Gaines E. Miles. 1991. Design of an Agricultural Robot for Harvesting Melons. Paper No. 91-7029. The ASAE, St. Joseph, MI 49085-9659. 23p.
- Wolf, Itzhak, Jonathan Bar-Or, Yael Edan and Uri M. Peiper. 1990. Developing Grippers for a Melon Harvesting Robot. Paper No. 90-7504. The ASAE, St. Joseph, MI 49085-9659. 9p.