77

BARD

FINAL REPORT
PROJECT NO. US-2199-92

Adaptive Sorting of Fresh Produce

S.W. Searcy, K. Peleg

1998

Date: **August 15, 1997**

BARD
P.O.Box 6
Bet-Dagan, ISRAEL

BARD Project No. **US-2199-92 C**

Title

Adaptive sorting of fresh produce

Investigators' Names

Stephen W. Searcy
Kalman Peleg
Victor Alchanatis
Matti Picus

Investigator's Institution

Texas A&M Research Foundation
Technion, Israel Institute of Technology
Texas A&M Research Foundation
Technion, Israel Institute of Technology

Project's starting date: **August 10, 1993**

Type of report: 1st Annual _____ 2nd Annual _____ Final X

Signatures

Principal Investigator

Stephen W. Searcy

Institution's Authorizing Official

Fred B. Furler

Table of contents

1. Detailed Description of Research Activities and Results.....	1
1.1 Adaptive Imaging	1
1.1.1 Selective wavelength image sensing	1
Sensor's specifications	2
Spectral range	2
Spectral channels	3
Light sensitivity	3
Available technologies	3
Multispectral image acquisition	5
Selection of technology for sensor implementation	9
1.1.2 Sensor construction and evaluation	9
1.1.3 High speed inspection of carrots	13
System architecture and components	13
Algorithms	15
Classification algorithms	19
Results and discussion	20
Evaluation of the features used in the first classification phase	20
Classification results	22
1.2 Adaptive Sorting by Prototype Populations	24
1.2.1 Adaptive Sorting Algorithm	25
Understanding the Adaptive Sorting Process	28
Population Classifier Design	29
1.2.2 Simulation of the Adaptive Sorting Algorithm	31
Data Creation	31
Simulation Results	32
1.2.3 Adaptive sorting of Dates	38
Data Preparation	38
Population Classifier Performance	41
Adaptive vs. Non-Adaptive Classifier Performance	42
2. Evaluation of research achievements	45
2.1 Adaptive imaging	46
2.2 Adaptive sorting	46
3. Description of cooperation	47
4. Conclusions and further research	48
5. References	48

Adaptive sorting of fresh produce

Project No. US-2199-92C; Duration: 3 Years; Budget: \$275,000

Stephen W. Searcy Texas A&M University
Kalman Peleg Technion, Israel Institute of Technology
ABSTRACT

This project includes two main parts: Development of a "Selective Wavelength Imaging Sensor" and an "Adaptive Classifiery System" for adaptive imaging and sorting of agricultural products respectively.

Three different technologies were investigated for building a selectable wavelength imaging sensor: diffraction gratings, tunable filters and linear variable filters. Each technology was analyzed and evaluated as the basis for implementing the adaptive sensor. Acousto optic tunable filters were found to be most suitable for the selective wavelength imaging sensor. Consequently, a selectable wavelength imaging sensor was constructed and tested using the selected technology. The sensor was tested and algorithms for multispectral image acquisition were developed.

A high speed inspection system for fresh-market carrots was built and tested. It was shown that a combination of efficient parallel processing of a DSP and a PC based host CPU in conjunction with a hierarchical classification system, yielded an inspection system capable of handling 2 carrots per second with a classification accuracy of more than 90%. The adaptive sorting technique was extensively investigated and conclusively demonstrated to reduce misclassification rates in comparison to conventional non-adaptive sorting. The adaptive classifier algorithm was modeled and reduced to a series of modules that can be added to any existing produce sorting machine. A simulation of the entire process was created in Matlab using a graphical user interface technique to promote the accessibility of the difficult theoretical subjects. Typical Grade classifiers based on k-Nearest Neighbor techniques and linear discriminants were implemented.

The sample histogram, estimating the cumulative distribution function (CDF), was chosen as a characterizing feature of prototype populations, whereby the Kolmogorov-Smirnov statistic was employed as a population classifier. Simulations were run on artificial data with two-dimensions, four populations and three classes. A quantitative analysis of the adaptive classifier's dependence on population separation, training set size, and stack length determined optimal values for the different parameters involved. The technique was also applied to a real produce sorting problem, e.g. an automatic machine for sorting dates by machine vision in an Israeli date packinghouse. Extensive simulations were run on actual sorting data of dates collected over a 4 month period. In all cases, the results showed a clear reduction in classification error by using the adaptive technique versus non-adaptive sorting.

1. Detailed Description of Research Activities and Results

During the three years of the project, most of the efforts in Texas were directed towards development of a "Selective Wavelength Imaging Sensor" for adaptive imaging of agricultural products. In parallel, the team in Israel concentrated on developing an "Adaptive Classifier system" for adaptive sorting of agricultural products. This section summarizes the research activities and results of the two teams

1.1 Adaptive Imaging

Many investigators have shown that important quality information about of fruits and vegetables is concentrated in specific bands of the electromagnetic spectrum. Porteous *et al.* (1981) determined that reflectance values within wavelength bands of 590-890nm and 1100-1400nm could be used to identify a wide variety of potato diseases and defects. This information was used to construct an experimental grading machine that used beam splitters, bandpass filters and eight photodetectors (Porteous and Muir, 1987). Gunasekaran *et al.* (1985) summarized many studies which found multiple wavelengths to be useful in evaluating fruits or vegetables. A wide body of literature has demonstrated that energy measurements in narrow wavelength bands are useful for quality evaluations, and that the useful wavelengths vary with fruit species and variety.

To use the appropriate spectral information, the fruit must be irradiated with energy at the appropriate wavelengths (light source), the energy must interact with the fruit and the resultant energy levels measured at the desired wavelength (sensor). A common technique (Howarth and Searcy, 1992, Miller and Delwische, 1991) is the use of a broad spectrum light source and a bandpass filter placed in front of the imaging sensor. While this technique is workable in many situations, it is not optimum for quality evaluation. Many commercial machines use this method, with the assumption that anything which does not reflect the expected level of energy within the predetermined wavelength band is to be rejected. Unfortunately, the single wavelength, or monochromatic sensing does not allow much ability to adapt for features that may change between fruit populations. Filter wheels and beam splitting arrangements have been used in experimental prototypes (Worthington, *et al.*, 1976; Porteous and Muir, 1987), but have proven too cumbersome for wide commercial application.

Recent technical developments have provided the potential for sensors which can measure energy levels at specific, narrow wavelength bands. A programmable, wavelength selectable sensor could provide the adaptive, wavelength specific image formation necessary for inspection of fruits or vegetables.

The original research proposal included a rough description of the sensor's implementation based on optical diffraction technology similar to that used in most commercial spectrographs and monochromators. From an extensive investigation conducted during the first months of the project it was found that there are two more potential technologies which can be used for the sensor implementation. Those technologies involve different conceptual and technological approach for developing the selective wavelength imaging sensor than the one described in the original proposal. Therefore three technologies were identified as potentially suitable for implementation of a selectable wavelength sensor. Those technologies involve different conceptual and technological approaches for developing the selective wavelength imaging sensor.

In contrast to the well established commercial manufacturing and use of diffraction grating spectrographs, applications based on the two additional technologies are quite new and not yet well established in the field of spectroscopy (although diffraction grating 'imaging quality' spectrographs are also still under evolution). Nevertheless, they are very promising and thus they were extensively studied and evaluated. In fact, one of the latter was chosen to be the most appropriate technology for the sensor under development.

1.1.1 Selective wavelength image sensing

This section includes:

1. a description of each one of the evaluated technologies, including the physical principle upon which the technology is based, and some considerations on the algorithms required for the selective wavelength imaging sensor in case it is constructed based on the discussed technology.
2. The construction of the sensor. The technology selected for constructing the sensor is described and the imaging sensor is evaluated.

Sensor's specifications

The selectable wavelength sensor must be able to construct a multispectral image. Two dimensions of the space are the two spatial dimensions of the inspected object. A third dimension is the spectral dimension, which depicts the spectral response of each point at the selected wavelengths. Figure 1 shows an example of a multispectral image to be constructed by the selectable wavelength sensor when λ_1 , λ_2 and λ_3 are the selected wavelengths. This multispectral image can be built in different ways, depending on the technology upon which the sensor is based.

The sensor consists of two different components. One component splits the light into individual spectral components (by absorbing or diffracting the incoming light) and the other component records the spectral response.

The sensor must be able to scan a given object at selected wavelengths. The performance characteristics of the developed sensor must be defined by the following items:

- Spectral range from which discrete wavelengths can be selected
- Minimum number of objects to be scanned per second
- Maximum number of discrete wavelengths to be scanned per object
- Light sensitivity

Spectral range

The selectable wavelength sensor is expected to be used with fresh agricultural produce. Therefore, the spectral characteristics of the inspected objects should dictate the desired spectral

characteristics of the sensor. Spectral properties of fresh agricultural produce have been widely investigated and detailed data for many crops can be found in literature. Gunasekaran *et al* (1985) in their review article include a table that summarizes 51 crops with optimal wavelengths useful measurement of specific features. From that table it can be seen that most of the features are measured in the visible and near infra-red spectrum, from 400nm to 1000nm. Some features that depend on water content, moisture, humidity other related properties use wavelengths in the near and mid infrared region, from 900 nm to 2800 nm. In order to cover most of the optimal reported wavelengths in agricultural produce, the sensor should be able to scan images within the visible and near infra-red spectrum from 400nm to 1000 nm.

It must also be kept in mind that common image acquisition devices (cameras) use CCD arrays to translate light intensity to electric signal. CCD arrays are made out of silicon and their typical spectral response is from 300 nm to 1000nm. Thus, in order to include the band of 1000nm to 2800 nm in the adaptive sensor's capabilities, special imaging arrays must be considered. In this work, only the visible and near infra-red band from 400nm to 1000nm was considered.

Spectral channels

The number of discrete wavelengths to be scanned per object is limited in practice by the number of objects that have to be scanned per second. The more the objects to be scanned per second, the less time is left for individual wavelengths. A first assumption will be that 3 different wavelengths must be scanned per object.

Light sensitivity

The sensor used must be sensitive enough to record the reflected light received from the object. Special attention must be given to this item as the light reaching one sensor's element is filtered or diffracted and thus its intensity is reduced to the relative intensity of the specific wavelength band. This is in contrast with broadband imaging where the total light intensity reaching each sensor's element is the integration of intensities in all wavelengths included in the reflected light. This limitation will result in the need for more intense illumination than would be used otherwise.

Available technologies

The available technologies for multispectral imaging derive mainly from spectroscopy applications where the spectral response of a sample is built. Specifically, the available technologies are:

- electronically tuned spectral bandpass filters. Electronic devices with no moving parts that act like conventional bandpass filters, the center wavelength of which can be changed. They are not yet well established in commercial products.
- light diffraction. Most of the commercially available spectrographs are based on diffraction gratings which diffract the light, separating it to its spectral components. Because of the moving parts of the spectrographs, accurate, and thus expensive, mechanical components are needed for focusing the desired wavelength on the output focal plane of the instrument.
- linear variable filters. Filters with spatially varying bandpass characteristics. It has no moving parts but its manufacturing process is still unreliable. They are not yet well established in commercial products.

The above technologies define three conceptually different methods of implementing the sensor. All three methods have the same final result: construction of a multispectral image that contains spatial and spectral information of the inspected object. Besides their different underlying technologies, they also differ in the manner they build the multispectral image.

In contrast to the well established commercial manufacturing and use of diffraction grating spectrographs, applications based on the two additional technologies are quite new and not yet well established in the field of spectroscopy (although diffraction grating 'imaging quality' spectrographs are also still under evolution). Nevertheless, they are very promising and thus they were extensively studied and evaluated. In fact, one of the latter was chosen to be the most appropriate technology for the developed sensor.

Electronically tuned spectral bandpass filters

Acousto-optic tunable filters (AOTF) act as electronically tuned spectral bandpass filters. An AOTF is a solid state electro-optical device with no moving parts. It consists of a crystal in which acoustic waves at radio frequencies (RF) are used to separate a single wavelength from a broadband or multi-color source. It is based on acousto-optic interaction in an anisotropic medium. Acoustic waves, which are applied to the anisotropic crystal through piezoelectric transducers, produce a periodic moving grating which diffracts portions of the incident beam. The conservation of momentum dictates that only a very narrow band of frequencies can be diffracted. The wavelength of the light selected is a function of the frequency of the RF signal applied to the crystal. Thus, by varying the frequency of the RF signal, the wavelength of the separated or filtered light can be varied. The wavelength of the diffracted light can be tuned over large optical region by simply changing the frequency of the applied RF. Technical details and applications on AOTFs can be found in Tran and Furlan (1992, 1993), Tran and Bartlet (1992), Kurtz *et al.* (1987) and Suhre *et al.* (1992).

An AOTF can very quickly switch from one wavelength to another. The time required for the switching between two arbitrary wavelengths, equals to the time needed for the acoustic wave to spread into the crystal. The speed of an acoustic wave in TeO_2 is equal to 6.17×10^2 m/s (Horowitz 1983). Depending on the physical dimensions of the crystal, the acoustic wave will typically need tens of microseconds to fill the whole crystal. When using accurate but economical digital RF generator boards, the RF signal is digitally generated by a computer, but the switching time between RF signals may be as long as 20 ms. In that case, the final scanning speed of the AOTF is dictated by the RF generator board.

Liquid crystal tunable filters (LCTF) are also electronically tuned spectral bandpass filters, based on a different principle than AOTFs. LCTFs have very low energy transmission (10%, compared to 50-85% for AOTFs) and very slow switching time (~100ms) (compared to 0.05-10ms of AOTFs). Therefore, AOTFs will be considered as the representatives of the electronically tuned spectral bandpass filters.

The AOTF can be incorporated in the adaptive sensor by placing it in the optical path between the incoming reflected light and the CCD array. Conceptually, the AOTF acts as a conventional interference bandpass filter.

Diffraction gratings

Diffraction gratings are used to disperse light; that is, to spatially separate light of different wavelengths. A typical diffraction grating consists of a substrate, usually of an 'optical material', with a large number of parallel grooves ruled or replicated in its surface and overcoated with reflecting material such as aluminum. When a parallel-beam of polychromatic light is incident on a grating, the light is dispersed so that each wavelength satisfies the grating equation. A detailed description of diffraction principle by gratings can be found in ORIEL (1994).

A spectrograph based on diffraction gratings can be used to implement the selectable wavelength imaging sensor. When a line is focused on the input slit of the spectrograph, spectral information of that line is patterned on the output focal plane of the spectrograph. If a CCD array is placed at the output focal plane, each row of the CCD will image the focused line at a different wavelength. The rows of the CCD will act as multiple filtered line scan cameras. By scanning the rows that correspond to the selected wavelengths, the sensor can build the multispectral image. Therefore, when using this technology, the sensor's capability of selecting individual wavelength is dependent on the capability of the camera to scan only certain lines of the CCD array, without having to readout the whole image and then disregard the unwanted lines. There are commercially available progressive scan cameras (e.g. manufactured by CIDTEC) with vertical windowing capabilities, which exhibit the above feature.

Linear variable filters

Linear variable filters (LVF) consist of multilayer coatings vacuum deposited on rectangular substrates. These filters offer continuous spectral coverage within the manufactured wavelength regions. A specific wavelength may be selected by simply adjusting the filter in the appropriate

linear position. Raster scanning will provide a continuous rapid scan of the total wavelength range of the filter. More details on LVFs can be found in OCLI (1993).

A linear variable filter can be mounted directly on the CCD array. With the filter mounted, each line of the CCD corresponds to a specific wavelength, according to the position of the line relative to the linear variable filter. As in the diffraction grating based sensor, the sensor's capability of selecting individual wavelengths is dependent on the capability of the camera to scan only certain lines of the CCD array.

Multispectral image acquisition

Since the sampling device is a 2-D CCD array, the multispectral image must be built by accumulating many 2-D planes. Consequently, the basic operation of the sensor is exposure and/or acquisition of a number of 2-D images. In each one of the three technologies presented above though, the axes of a single 2-D acquired image have different interpretation.

Figure 2 Multispectral image construction by diffraction gratings

Single frame interpretation

1. Spectrographs: Using spectrographs based on diffraction gratings, each column of the acquired image represents the same point on the object, viewed through many different bandpass filters. Each acquired image contains all the spectral information of one geometric line perpendicular to the direction of motion of the object. Object motion is required to form an image of a single wavelength.
2. Tunable filters: All the information in one acquired image corresponds to one single wavelength. Each image contains information of the whole object in the same spectral band.
3. Linear Variable Filters: Like with tunable filters, each image contains spatial information of the whole object but unlike them, each row of the acquired image contains information for a different wavelength. A single wavelength image must be formed a line at a time, like for the spectrographs.

When the object moves under the camera, a series of images is acquired. The relation between consecutive images and the way of constructing the whole multispectral image is discussed in what follows.

The best way to demonstrate the relationship between consecutively acquired images in the multispectral space is graphically. In each of the following figures the consecutively acquired images are mapped into the multispectral image space. Since the interpretation of a 2-D acquired image is different for each technology, a different figure is created for each one of them.

Diffraction gratings

Figure 2 depicts consecutive images, mapped into the multispectral image space. Each image is a vertical plane and contains all the spectral information of the imaged row of the inspected object. The rows corresponding to the desired spectral channels are shown. Note that in order to obtain the complete multispectral structure of the whole object, motion of the object under the camera is indispensable. Thus a sensor based on diffraction gratings technology cannot be used with stationary objects. Since motion is an inherent feature of a sorting system, the above constraint is not serious. The propagation of the object under the camera must be considered as part of the 'global sensor' (unless the inspected objects are one dimensional like in conventional spectroscopy of individual samples).

Acousto-optic tunable filter

Figure 3 depicts three consecutive images, mapped into the multispectral image space. Each image is an horizontal plane and contains only data of one wavelength. In contrast to a diffraction grating based sensor, motion of the object under the camera is not required. On the contrary, it is a side effect for which software algorithms will have to compensate.

Linear variable filter

Figure 4 depicts consecutive images, mapped into the multispectral image space. Since the points of an acquired image represent object points with different spatial and spectral features. Lines that correspond to the desired spectral channels are shown in Figure 4. Similarly to the diffraction gratings, in order to obtain the complete multispectral structure of the whole object, motion of the object under the camera is indispensable.

Figure 3 Multispectral image construction by acousto-optic tunable filter.

Video camera requirements

The camera must enable scanning the required wavelengths at the desired rate. Carrots will be used as an example for the analysis. It will be assumed that the throughput of the sorting system has to be 3-5 objects per second and that three different wavelengths will have to be scanned for each object. The average camera requirements will be estimated by considering the worst case, i.e. 3 different wavelengths must be sampled for each object at a rate of 5 objects per second. The desired spatial resolution will be assumed to be 0.5 mm per pixel. A 512x512 pixels image array will be adequate to include the image of a whole inspected carrot.

For an AOTF based sensor, each frame contains information of the whole object in one wavelength. Thus the camera's pixel resolution should meet the desired spatial resolution of the object. For each object three frames have to be acquired, therefore in one second (3 frames per object) * (5 objects per second) results 15 frames per second fps). Standard RS170 video camera with 30 fps rate is enough for this application. The time interval between frames (vertical blanking) is approximately 3ms, which is much more than the time required for the AOTF to switch from one spectral channel to another.

Figure 4 Multispectral image construction by linear variable filter

For a diffraction grating based sensor, each acquired frame would optimally consist just of the lines that correspond to the required wavelengths. Assuming that three different wavelengths have to be scanned, each frame would optimally consist of three lines. Camera's with vertical windowing capabilities can perform this task (progressive scan cameras). In terms of spatial resolution, the number of pixels in a row vertical to the direction of motion define only the resolution in one dimension. In order to get the spatial resolution of 512 pixels per object in the other dimension (similar to that in one frame acquired with the AOTF), 512 3-line frames per object are required. For a rate of 5 objects per second, the cameras frame rate must be $512 \times 5 = 2500$ frames per second.

For a linear variable filter based sensor, the situation is exactly like in the diffraction gratings case, since only the lines that correspond to the required wavelengths must be scanned. Thus the requirements from the camera are the same as described above. Nevertheless, the algorithms for processing the acquired images will be more complex since lines that are acquired simultaneously include information in different spectral channels as well as different parts of the object.

Table 1: Advantages and disadvantages of technologies for implementation of the multispectral sensor.

	Diffraction gratings spectrographs	Acousto-optic tunable filter	Linear variable filter
Motion	Required	Not needed.	Required
Camera specifications	High speed camera with vertical windowing capabilities	Standard RS170 camera	High speed camera with vertical windowing capabilities
Optics	Input optics for focusing the object line on the grating.	Optics for passing the light, through the AOTF and recombining the diffracted rays.	Standard camera lens. No special optics
Mounting	Rigid stand for holding and keeping stable the spectrograph.	Rigid stand for holding the AOTF.	After the LVF is directly mounted on the CCD array, no special mounting or handling requirements exist
Complexity of multispectral image construction	Acts like many line scan cameras that image the same line of the same object simultaneously. Average complexity	Acquired images are ready for processing by available software. Light modifications are needed (if any) for object motion during consecutive images. Low complexity	Acts like many scan line camera that scan different objects independently one from the other. High complexity
Sensitivity	Very sensitive to mechanical displacements and shocks	Sensitive to RF signal accuracy	Number and dimensions of surface defects of the coating.
Illumination	Due to the high frame rate required, the exposure time is short and thus a high intensity illumination will be needed	High intensity due to the narrow bandpass	Due to the high required frame rate the exposure time is short and thus a high intensity illumination will be needed
Cost reduction with high volume	No	Average	High

Selection of technology for sensor implementation

The above analysis shows that, in principle, the multispectral sensor can be constructed by any of the reviewed technologies. The advantages and disadvantages of each technology are presented in a form of a table that compares the impact of various features of the technologies mentioned on the sensor under development. Table 1 lists the constraints or the expected performance of a sensor based on each technology with respect to the selected features. The features are chosen in a such a manner that they help to evaluate the fitness of the technology to the multispectral sensor's specifications.

According to Table 1 the technology with the most advantages is the electronically tuned spectral bandpass filters and specifically the acousto optic tunable filters. This is a solid state device with no moving parts that does not need special camera requirements. Therefore, an imaging sensor was constructed using the AOTF technology.

1.1.2 Sensor construction and evaluation

A commercial acousto optic tunable filter was ordered from Brimrose Inc.. The AOTF system includes input optics for collecting light and passing it through the crystal, as well as output optics for focusing the filtered light on the CCD detector. The main features of the system are:

Spectral range: 500-1000nm
Spectral resolution: 5-25nm
Field of view: 10 degrees
Switching speed: 250 μ s

Using the above tunable filter, a multispectral imaging system was built. Figure 5 shows the components of the system: a Sony black and white CCD camera (model SSC-M350) was mounted at the output plane of the AOTF and an EPIX 4MEG frame grabber was used to grab the images. The center wavelength of the AOTF was selected by applying an analog voltage in the range of 0-10V on a voltage controlled oscillator (VCO), which produced the RF signal for tuning the filter at the desired wavelength. The analog voltage was generated by an D/A channel of a data acquisition board (National Instruments, Lab-PC+).

Computer programs were developed for controlling and synchronizing image acquisition and wavelength selection. Wavelength selection was synchronized to the camera's video signal and was performed during the vertical blanking period. Figure 6 shows timing details for switching wavelengths and grabbing images. Specifically, in order to grab a video field i at a desired wavelength λ_1 , the control program has to issue the command for tuning to this wavelength at the end of field $i-2$. The filter will tune to the desired wavelength during the vertical blanking period between fields $i-2$ and $i-1$. Consequently, during field $i-1$, the CCD sensor will be exposed to light that passes through the AOTF tuned at λ_1 . The information accumulated on the CCD will be read out during field i . By selecting a different wavelength at each vertical blanking period the sensor system can provide information on a different spectral channel every 33ms.

The AOTF system was tested by building spectral response charts of light sources with known spectral content. Using the procedure described above for acquiring images in different wavelengths, and increasing the voltage applied on the VCO by 0.01 V each time, spectral response charts of known LEDs lamps were constructed. The voltage applied on the VCO was translated to the expected tuned wavelength and the latter was used as the wavelength axis in the

spectral response charts. Figure 7 shows the spectral response of a near infra-red LED which is known to have a peak emission at 940 nm. The good agreement between the measured and the known peak emission wavelength of the LED indicates that the sensor operates adequately. Similar results were obtained when 8 more LEDs were tested. In some cases a small offset between the measured and the known peak emission wavelength was observed. This offset was not greater than a few nm and it is due to the VCO. VCOs can rapidly change output frequency (250 μ s), but when they cover a broad range of frequencies, the repeatability of the output frequency is only one percent. More accurate RF signal generators with similar switching speed performance can be achieved by Direct Digital Synthesis (DDS), but the cost is considerably higher. As faster DACs become available, Direct Digital Synthesizers will become more accessible to low cost applications.

The sensor was also tested with a high intensity Metal Halide lamp. Figure 8 shows the spectral response of the Metal Halide Lamp. This type of lamp produces a light which appears to be white to the eye, but all the wavelengths are not evenly represented. Actually, there are several high intensity peaks in specific wavelengths, with very low light intensity in spectral regions outside those peaks. The spectral response obtained with the developed sensor agrees with the data found in the literature. However, the peak observed in the NIR region (~820 nm) was not documented elsewhere. Data for metal halide lamps found in the literature and supplied by manufacturers of those lamps stop at 750 nm.

After testing and verifying the fast and accurate random access to spectral channels, the spectral imaging capabilities of the sensor were tested. Due to the very narrow bandpass of the AOTF, a very high intensity diffused light illumination chamber was designed and built. This was necessary in order to enable enough light pass through the AOTF and reach the camera. The illumination chamber consists of a 43cm diameter dome coated with flat white paint and six 500W tungsten-halogen lamps installed in the circumference of the bottom part of the dome. A ring installed on the edge of the dome doesn't allow for direct light from the lamps to reach the illuminated objects. The illumination system was designed to provide uniform diffused illumination for objects up to 30 cm long/wide at a minimum distance of 10cm from the edge of the dome (for avoiding direct light from the bulbs to reach the objects).

Using the high intensity diffused light and setting the gain of the video signal to high value, images of fruits were acquired. Two main problems were encountered:

1. High noise in the video signal. The high noise level is due to the high gain used to obtain bright images.
2. Image degradation caused by apparent shadows. The shadows that degraded the image quality are due to the VCO driver. VCO drivers are low cost drivers and can produce high quality signals. But when used with broad spectral range (like in our case) the RF signal quality is low. The poor quality of the RF signal affects the image quality which is expressed as shadows in the image.

The problems described above can be solved either by purchasing additional, high quality hardware components or by using software tools to compensate for the known problems.

1. Purchase a high sensitivity camera and a better RF signal generator.

A high sensitivity camera will eliminate the need for high gain of the video signal and the noise associated with it will be lower. There are several types of high sensitivity cameras, e.g. intensified cameras or cooled cameras, that can be used for this purpose. The cost associated with this type of cameras is significantly higher than the cost of the simple CCD cameras that are currently used.

A better RF signal generator will improve the image quality in terms of the apparent shadows. High quality Direct Digital Synthesizers (DDS), based on high speed DACs can produce high quality signals. This technology is constantly evolving and it is anticipated that in the future, low cost synthesizers will be available.

2. Rely on image processing to correct the known problems.

Noise reduction techniques combined with image restoration procedures can improve the image quality.

By purchasing additional hardware components, image quality can be significantly improved while maintaining the current high speed multispectral image acquisition. The cost associated in purchasing of the additional hardware components depends on the specific components. There is no cost associated with the image processing based approach, but the high speed of image acquisition is sacrificed. Noise reduction techniques require averaging of several consecutive images and image restoration techniques require computational intense operations on each pixel of the image.

It has been demonstrated that the combination of noise reduction techniques and image restoration procedures resulted images of fair quality. Specifically, noise was overcome by averaging a sequence of images that were acquired when the AOTF was tuned to the same wavelength. The resulted average image exhibited lower noise levels that were acceptable. The effect of the apparent shadows was significantly decreased by image restoration techniques. Since the shadows are a function of the driving signal, when the driving signal is constant, i.e. a specific wavelength is selected, the resulting image distortion is constant. Therefore, for each wavelength, the image distortion can be measured and quantitatively characterized. Using the distortion characteristics and applying image restoration techniques, compensation for the shadows can be accomplished and image quality can be improved. A white surface was imaged at a specific wavelength and a correction factor was calculated for each pixel. The calculation of the correction

factor was based on two assumptions: 1. The image of the white surface would be a uniform gray level image if the shadows were not present, and 2. The shadow effect is linear with light intensity.

1.1.3 High speed inspection of carrots

Fresh market carrots are a major vegetable crop in the United States as well as abroad. Carrots that reach the packing shed have a broad spectrum of shape and surface defects. Today, defective carrots are separated by visual inspection, with all the associated disadvantages, e.g. expensive labor, grading inconsistency due to human biases and fatigue, etc. An automatic inspection system could reduce handling costs as well as will increase grading consistency and accuracy.

High speed defect detection is particularly difficult with agricultural products due to their large variability in terms of appearance characteristics, shape and dimensions. No reports of high speed inspection systems were found in literature for fresh-market carrots. Some reports can be found on high speed inspection systems built for other agricultural products (e.g. Ni *et al* (1993) - corn kernels). Most of them are for products that are typically much smaller in size. Davenel, *et al*. (1988) developed a system that detects surface defects on apples. This system was based on a hardware component especially designed to make a very fast simplified approximation of the bruised area of the apple. Nevertheless, this simplification yielded to high grading errors and the system could only be used for preliminary screening of the apples from the field. Delwische, *et al*. (1993) developed a high speed inspection system for dried prunes. The system was partly based on hardware components designed specifically for that system. The image processing and classification algorithms used, were a simplified version of algorithms tested in a first prototype. This simplification provided to high grading speed but also reduced the system's performance regarding the grading errors.

Howarth (1991) has laid the basis for an automatic inspection system for fresh market carrots. Using carrot images and a relatively small data set he showed that, in principle, machine vision can be used to detect many kinds of surface as well as shape defects of fresh market carrots. Specifically, he showed that features extracted from carrot images can be used to grade carrots with curvature, fork and brokenness shape defects, as well as surface defects, such as rot and growth splits, with acceptable classification errors. Nevertheless, the grading speed of the developed system was too slow for use in packing sheds and it was only tested with a small number of carrot images.

The objective of this part of the work was to construct and test a high speed inspection system for grading fresh market carrots, based on the system developed by Howarth. In contrast to other works found in the literature, in this work no simplifications of the prototype algorithms were performed in order to sustain the low classification rates achieved with the prototype. Specifically, the high speed inspection system should have high throughput rate of 2-5 carrots per second and sustain low classification errors of less than 10%. This was accomplished by using a high speed line scan camera and a digital signal processor (DSP) that acquired and processed carrot images. Image processing algorithms were optimized and implemented on a DSP which worked in parallel with a PC based host computer to grade the carrots with high speed. Efficient communication protocols were developed that allowed both processors to perform as nearly as possible to their maximum execution speed. An important part of this work was to test the system performance with a large representative population of carrots obtained from a packing shed.

error

System architecture and components

The high speed inspection system was based on a prototype described in Howarth (1991). Specifically, the system included:

- a mechanical system for conveying the inspected carrots under the camera and
- a machine vision system for grabbing, analyzing and grading carrot images.

The developed system did not include any mechanism for sorting the carrots after they were graded. Therefore, its performance was evaluated through the decisions made by the classification system. In addition, there was no automatic feeding mechanism, therefore the carrots were manually placed on the conveyer.

Figure 9: Schematic description of the mechanical system.

The mechanical system was based on a conveying system built for a previous project. A schematic description of the mechanical system is shown in Figure 9. A flexible black belt carried the inspected carrots through an illumination chamber. The belt was powered by a 1/4HP variable speed DC motor. The illumination chamber consisted of a cylindrical white surface that diffuses the light emitted from two 200W incandescent lamps onto the area where the carrot is imaged. The camera was placed above the illumination chamber and took images of the carrots as they were conveyed through it. Detailed description of the conveying system can be found in Howarth (1990). The speed of the conveyor belt was calibrated through the motor controller, using a standard 55mm diameter disk coated with high reflective material as reference.

The machine vision system included hardware required for acquisition, processing and grading of the inspected carrots. Figure 10 shows the architecture and connections. Specifically, the imaging system included the following components:

A line scan camera (LC1901-256 - EG&G RETICON) with a 20-70mm, f1.8 zoom lens. An interference filter, with center wavelength at 655nm and 50% bandwidth of 70nm, was mounted in front of the lens. This wavelength range is optimum for detection of most surface defects in carrots (Howarth and Searcy, 1990). The 256 pixels array could be scanned at a maximum pixel rate of 10MHz. The line scan camera was mounted so as to view the carrot as it passed lengthwise through the illumination chamber. The camera was mounted and focused to achieve a 2mm/pixel lateral resolution. The resolution in the direction of motion was determined by the conveyor velocity and the line scan rate. The system was calibrated and set to 1 pixel per mm. The output video signal and the camera control signals were connected to an interface board, described below.

An interface board for the line scan camera (RET4MEG - POYNTING Products Inc.) The interface board had three main functions:

- generate the master pixel clock.
- convert the signals between differential RS442 format used by the camera and single ended signals used by the frame grabber, and
- connect control signals from the image processing board to the camera.

The single ended video signal was connected from the interface board to the frame grabber, described below.

A frame grabber and image processing board (4MEG-10 - EPIX Inc.), that included a TMS320C25 fixed point digital signal processor (DSP) (Texas Instruments, Inc.). The frame grabber section was equipped with 1Mb of Video RAM (VRAM). The DSP resided on the board with the frame grabber and performed low level image processing. Detailed description of DSP features and

architecture can be found in Texas Instruments (1989). The main features exploited in this system are:

- 544 16-bit words of high speed on-chip RAM that could be accessed at any time, and
- the ability of the DSP to interrupt the host CPU at a predetermined IRQ level.

The high-speed on-chip RAM was exploited in conjunction with the interruption capability for parallel execution of feature extraction, image processing and digitization operations.

An AT bus host computer (DELL 486P with a 33MHz 80486DX CPU and 4M of RAM). The host computer performed higher level feature extraction and grading algorithms, as well as the operator interface,

A magneto-optic disk drive (DYNAMO128 - FUJITSU) with capacity of 128Mb per disk. Digital images were stored on magneto-optic disks for future reference and analysis

A Panasonic black and white monitor (WV-5410).

Figure 10: Schematic diagram of inspection system hardware components and architecture

Algorithms

The developed algorithms consisted of two parts: one part for image acquisition and low level image processing which was implemented on the DSP, and a second part for feature extraction and classification which was implemented on the host CPU. The combined software package was called INSPECT 2.0.

Since image acquisition was based on a line scan camera, all image processing algorithms and part of feature extraction algorithms were line oriented. Each line was acquired and processed before scanning the next line. Specifically, the following operations were performed:

1. Scan a line,
2. Adaptively threshold the line to create a binary image (Howarth, 1991),
3. Compress the binary line using Run Length Encoding (RLE),

4. Use the RLE data of the acquired line and search for information on the morphological structure of the carrot. Update the carrot's morphological representation, described by a connected components tree (Howarth 1991)
5. Proceed with scanning of the next line

After the whole image of the carrot had been formed, high level features were extracted and the carrot was classified as 'acceptable' or 'reject'

Figure 11: Flow chart of DSP main algorithm.

Figure 12: Flow chart of host CPU main algorithm

DSP algorithms

The DSP on the image processing board was responsible for synchronizing image acquisition operations and performing low level image processing operations. It controlled the digitization process, performed all image processing tasks and transmitted the results of image processing to the host CPU. A flow chart of the DSP main tasks is shown in Figure 11. During image acquisition, three basic operations were performed, as described by the inner loop of the flow chart:

1. One line was digitized,
2. A previous line was processed and
3. RLE data of the processed line were transmitted to the host CPU.

This loop was repeated until the end of the carrot was detected. At the end of the carrot, the DSP displayed the grabbed image and waited for the next carrot. Digitization and processing operations were performed in parallel, utilizing the on-chip RAM of the DSP: while the frame grabber used the image memory to store the digitized line, the DSP used the on-chip RAM to hold and process the previous line.

Details on the exact timing and synchronization between digitization and processing can be found in INSPECT 2.0 User's Manual.

Host CPU algorithms

The host CPU was responsible for system initialization, feature extraction and grading algorithms. Once the grading procedure was enabled, the host CPU initialized the DSP and let it run. Figure 12 shows a flow chart of the host CPU algorithm. It consists of two main parts. In Part A, the host CPU received image processing results from the DSP and, in parallel to image digitization, constructed the connected components tree as the carrot image was being formed. In Part B, the host CPU extracted features and classified the carrot.

The main objective of the host CPU in part A was to use the available RLE data and build the connected components tree. Since image acquisition and image processing operations are more time sensitive than feature extraction, the host CPU did not

interfere with the DSP during image acquisition and processing. Only at a DSP interrupt would the host CPU read the RLE data. Figure 13 shows the flow chart of the interrupt handler which was responsible for reading the RLE data from the DSP's program memory space. The operations performed by the interrupt handler were very simple and should be executed by the host CPU as fast as possible, since the DSP waited for completion before continuing with the next line.

Part B of the algorithm started after the carrot's connected components tree had been built. This part was executed in the time interval between two consecutive carrots. During this interval, the host CPU extracted features using the connected components tree and classified the carrot. Image acquisition was disabled during this time interval and it was re-enabled after classification was completed. More efficient algorithms for eliminating this waiting interval have been developed but not yet implemented.

Figure 14 summarizes the operations performed by all system components, including some timing details of algorithms executed for each carrot. In the first part, which corresponds to part A of the host CPU algorithm described above, the carrot image was formed and processed line by line. In the second part, the host CPU extracted features, classified the carrot while the DSP displayed the binary image. The second part was executed in the time interval between carrots. Therefore, the time interval between two consecutive carrots should be long enough to permit completion of all the necessary operations. The average time of processing by the host CPU was 300ms per carrot.

Detailed description of the operations taken place on a line basis is also shown in Figure 14. Most operations required to acquire and process one line were pipelined and executed in parallel with other operations performed on adjacent lines. Specifically, referring to Figure 14, when the DSP performed image processing operations on line N, the frame buffer digitized and stored line N+1 and the camera accumulated information of line N+2. By the time the last line of a carrot was digitized, all the image except the last line had been processed. Most of host CPU algorithms were coded in C, with few in assembly language. All DSP algorithms were coded in TMS320C25 assembly language and were optimized for execution speed.

Feature extraction algorithms

The high speed inspection system used the features developed in the prototype system (Howarth 1991) with some additional features developed to speed up the inspection process when carrots with extreme defects were involved. The original features were designed to describe shape and surface defects of the carrots. Specifically, there were features associated with shape defects as brokenness, curvature and forks, and surface defects as rot and growth splits. Detailed description of those features can be found in Howarth (1991). Features were calculated using carrot's compressed binary image (RLE) and its connected components tree that described the carrot's morphological structure.

The extraction of the features used with the prototype system could sometimes require extensive execution time, especially when the analyzed binary image was complex. Complex binary images produced complex connected components trees and consequently, required long computer time for analysis. Acceptable carrots usually had very simple binary images, consisting of only one continuous convex blob. Consequently, most of acceptable carrots had a simple connected components tree and did not require significant computing time for feature extraction. For example an average acceptable carrot whose binary image is shown in Figure 15(a), required approximately 200ms for feature extraction and classification. On the other hand, carrots with surface defects or irregular shape give concave or perforated binary images. Those carrots may require significant computing time for calculating the carrot's features from their complicated tree representation. For example the carrot whose binary image is shown in Figure 15(b) required more than 1s for feature extraction.

In a high speed inspection system, carrots that yield complex binary images must be screened out or, preferably, pre-classified, without analyzing their connected components tree. Screening out carrots with complex binary images will reduce the required processing time.

In order to detect carrots with complex binary images prior to feature extraction operations, additional simple features, defined and calculated directly from RLE data, were considered. The features were chosen to indicate the shape complexity of the binary image. All features were either directly associated with an element of the RLE or derived from the RLE data by a few simple arithmetic operations. The simplicity of those features enabled their calculation in parallel with image acquisition. Four such features are presented here as representative:

1. Average number of RLE transitions per line, T_{av} , defined as

$$T_{av} = \frac{\sum_{i=1}^N T_i}{N} \text{ where:}$$

N number of lines of the carrot's image

T_i number of RLE transitions in line i

2. Number of lines that have more than two transitions, L_2 .

3. Maximum value of four-line average of number of transitions, T_4 , defined as

$$T_4 = \max \left(\frac{\sum_{i=j}^{j+3} T_i}{4}; j = 1..N-3 \right) \text{ where } T_i \text{ and } N \text{ are as defined above.}$$

4. Maximum number of transitions in a line, T_{max} , defined as

$$T_{max} = \max(T_i; i = 1..N) \text{ where } T_i \text{ and } N \text{ are as defined above.}$$

The above features were used in conjunction with a hierarchical classification scheme described below for fast and accurate grading of the carrots.

Classification algorithms

In order to achieve high inspection speed, classification algorithms were combined with feature extraction algorithms in a two phase hierarchical decision system, shown in Figure 16. In the first phase, a simple feature was used (for example T_{av}) to classify the carrot as "potentially acceptable" or "reject". Then, if the carrot was classified as "potentially acceptable" its features were calculated and a second, more sophisticated classifier graded it as "acceptable" or "reject". The first phase

classifier was a simple threshold value and the second phase classifier was a Learning Vector Quantization artificial neural network (LVQ). LVQ is a classification network, originally suggested by Kohonen (1988), which assigns vectors to one of several classes. The neural network classifier was implemented using the commercial software package NeuralWorks (1993). The network had 18 input nodes, one for each input feature, 60 hidden nodes, and two outputs designating to which class the carrot was classified.

Figure 16: Schematic description of hierarchical classification system

Results and discussion

The high speed inspection system was tested with carrots obtained from a packing shed located in the Rio Grande Valley in South Texas. The constructed data set included carrots with a wide range of shape and surface defects. The set consisted of 1716 vectors from reject carrots and 1072 vectors from acceptable carrots. The rejects set was intentionally larger in order to account for the wide variability of the defects. In order to account for the unnatural distribution of the classes in the sets, the a priori probabilities of the acceptable and reject carrots were used. The percentage of acceptable and reject carrots in a processing batch varies from field to field. In this study it was assumed that 70% of the population is acceptable and the remaining 30% rejects, which represents an average field. In addition, it must be noted that approximately 30% of the reject carrots possessed more than one defect.

For the classification tests, two sets were composed: a training set which consisted of 1200 acceptable and 400 reject carrots drawn randomly from the original set, for a total of 1600 feature vectors and a test set that included the feature vectors of the entire data set. A similar procedure for evaluating classification schemes can be found in the literature (Peleg et al 1993)

Evaluation of the features used in the first classification phase.

The objective of the first phase classification was to increase the processing speed, nevertheless, the classification errors created by this scheme should be also considered. It was important that the number of acceptable carrots that were rejected by this scheme remained at a minimum level.

Guided by the objective of low misclassification rate of acceptable carrots, a threshold value was selected so that no more than 1% of the acceptable carrots of the training set was screened out. Using the manually assigned class of each carrot, cumulative class conditional probability curves

were constructed for each feature and a threshold value was set as described above. Figure 17 shows a sample cumulative class conditional probability curve for one of the features (T_{av}) and the associated threshold value. Most of the acceptable carrots had a low value for the selected feature, while reject carrots could obtain almost all the range of values with tendency to relatively high values

After setting the threshold, the performance of each feature was evaluated by two parameters: (i) average processing time by the host CPU of the carrots that were graded as "potentially acceptable" and (ii) percentage of reject carrots that were correctly screened out by the first phase classifier. Good features should exhibit fast processing speed and high percentage of correctly rejected carrots.

Table 2 summarizes the above defined evaluation parameters for the four simple features described previously, T_{av} , L_2 , T_4 and T_{max} . For comparison, the values of the same parameters are shown when the first phase classifier was not applied and all carrots were processed by the host CPU. Table 2 indicates that using T_{av} or L_2 in the first classification phase will meet the objective: the average processing speed of the remaining carrots was speeded up by approximately 30% and approximately 70% of the carrots with surface or fork defects were screened out. In addition, the variance of the processing time of the remaining carrots was significantly smaller (by a factor of 2.5). This means that not only the average processing time was reduced, but the number of carrots that need considerably more processing time than the average was significantly smaller. Features T_4 and T_{max} had slower processing speeds and screened out smaller portions of the carrots with defects.

A comparison between the T_{av} and L_2 features, shows that T_{av} was slightly superior, since T_{av} screened out more carrots (2.3% more) and there was no significant difference in the average processing time. However, by using L_2 , the inspection process could be terminated before full image acquisition. At the first line where L_2 exceeded the preset threshold, a classification could be assigned. This could not be done with T_{av} since all the lines had to be scanned and processed in order to compute the average value.

Carrots that were classified 'potentially acceptable' by the first phase classifier were further processed by extracting their morphological features using the connected components tree. Additional morphological features were calculated and used in the second classification phase for making a final decision. The second classification phase consisted of an artificial neural network classifier that classified the carrots according to their size, curvature, brokenness, forks and surface defects.

Table 2: Sample feature performances in screening out images with complex binary images.

FEATURE	Threshold value	Processing time [ms]		Percent screened out	
		Average	Std deviation	Acceptable	Reject
Average number of transitions per line, T_{av}	2.18	217	99	1.0	71.5
Number of lines with more than 2 trans., L_2	15	216.	97	1.0	69.2
Four-lines average number of trans., T_4	6	254	157	1.0	24.7
Maximum number of trans. per line, T_{max}	12	302.	232.	1.0	3.0
--- None ---	-	315	250	0	0

Table 3: Classification results with simple RLE features, artificial neural network and hierarchical classifier.

Feature	Total error [%]	Type 1 error [%]	Type 2 error [%]
Simple threshold with RLE feature T_{av}	17.1	3.9	48.0
Simple threshold with RLE feature L_2	17.0	3.6	48.4
Morphological features using one phase LVQ neural network classifier	9.8	4.3	22.7
Hierarchical classifier	10.0	4.8	22.2

Classification results

The components of the hierarchical classification system were first tested individually and then their combination was evaluated. The results are shown in Table 3. The total error includes classification errors in both classes, after the *a priori* probabilities have been taken into account. Type 1 error represents the percentage of the acceptable carrots that are classified as rejects, and type 2 error represents the percentage of reject carrots that are classified as acceptable.

The first phase classifier was used alone to classify the carrots using one simple RLE feature. The results for the two best features, T_{av} and L_2 , are shown. The threshold value for this evaluation was chosen such that the overall classification error was minimum. The results show that using T_{av} the minimum weighted classification error was 17.1% . Specifically, 96.1% of the acceptable carrots were correctly accepted and only 3.9% of them were misclassified as rejects. Nevertheless, as stated above, the weakness of this classifier is that many rejects were classified as acceptable too. This is reflected by the very low rate of correctly rejected carrots (52%) and very high percentage of rejects that were classified as accepted (48%). Nevertheless, since the portion of the reject carrots in the sample was assumed to be 30%, the overall weighted classification error is 17.1%. Similar results were obtained when L_2 was used to classify the entire set. The weighted

classification error was 17.0% and the specific class errors are shown in Table 3. These high classification errors were due to the fact that simple RLE features do not describe curvature defects, brokenness defects or size defects (length or width). Therefore, carrots that are actually rejects because they are curved, broken, undersized or oversized, were not detected by this simple scheme.

The second phase classifier was then tested when it was used alone to classify the carrots. In this case, the sophisticated features had to be calculated for all the carrots, and as demonstrated above, this might require significant computing time. The network was trained using the same carrot set that was used with the RLE classifier, with the appropriate feature vector components. The entire data set was used as the test set. The results are brought in Table 3. The weighted classification error of the test set was 9.8%. Specifically, 95.7% of the acceptable carrots and 77.3% of the rejects were correctly classified. The classification accuracy is acceptable but in some cases the time required for feature extraction is significantly higher than the time available in order to maintain a throughput of 2-3 carrots per second. Nevertheless, this demonstrates the feasibility of accurate automatic classification of carrots and will serve as a reference to the hierarchical classification scheme developed to speed up the feature extraction and classification processes.

Finally, a hierarchical classification system, briefly described above, was used to classify the carrots. One RLE feature was used in the first classification stage and an artificial neural network classifier was used to classify the carrots that were not rejected in the first stage. The threshold value of the first phase classifier was different than the corresponding value when used alone. The threshold value was not set by the criteria of minimum overall classification errors, but by the requirement that not more than 1% of acceptable carrots were rejected. The purpose of the first stage classifier was to create a class of rejects with almost no acceptable carrots in it. This class was rejected, while the rest of the carrots were further processed and classified by an artificial neural network classifier using more sophisticated features. The entire set was used as a test set. First it was classified by the first phase classifier using T_{av} , and then, 'potentially acceptable' carrots were classified by the LVQ neural network. The set classified by the neural network in this case is smaller than the one used when the neural network classifier was used alone, since it does not contain the carrots rejected by the first phase classifier. The *a priori* probability of the reject carrots is now lower since a large part of them had already been rejected by the first stage classifier. Table 3 shows that the overall weighted classification error is 9.8% which is practically identical to the error obtained with the neural network alone. However, the time required for feature extraction and classification was significantly reduced, as shown in Table 2.

The overall inspection time for a carrot consists of the time needed for image formation (which includes image acquisition and low level image processing) and the time required for higher level features extraction and classification. An analysis of the DSP algorithm execution speed (using a TMS320C25 software simulator) showed that in the worst case the DSP required approximately 10,000 instruction cycles in order to process one line of 256 pixels. Image processing operations consumed 90% of the time, while the remaining 10% was dedicated to overhead created by the control of digitization, control of the camera and communication with the host CPU. Consequently, with the specific DSP used in this system (TMS320C25 with a 40MHz clock and a 100ns instruction cycle) the acquisition and processing speed as dictated by the worst case scenario was 1ms per line, i.e. 1000 lines per second. In order to get a spatial resolution of 1 mm per line, the conveyor speed should be set at 1m/s. Assuming that an average carrot is 0.265m long, the developed system was capable of acquiring and processing almost 4 carrots per second. It must be noted that the image acquisition hardware is capable of much higher rates (up to approximately 32,000 lines per second with a 10MHz pixel rate instead of the present 1,000 lines per second) but the processing algorithms need more time. Higher image processing speed could be accomplished by replacing the adaptive threshold algorithm with a simple global threshold value. The expected speed up is of the order of x2. Nevertheless, previous work showed that adaptive thresholding increased significantly the system's grading accuracy (Howarth, 1991) and therefore it was optimized and incorporated into the high speed inspection system.

Feature extraction and grading of the carrot was performed by the host CPU partly in parallel with image processing and acquisition. The host CPU could perform those operations at an average rate of more than 4 carrots per second (Table 2). Since feature extraction and classification

operations were executed sequentially, with image acquisition, the average inspection speed of the developed system was 2 carrots per second. This speed can be doubled by implementing the host and DSP algorithms in such a way that they are completely independent, as opposed to the current system where the DSP waits while the host CPU grades, and the host CPU has spare time when the image is acquired. Faster DSPs and host CPUs would increase further more the systems speed.

1.2 Adaptive Sorting by Prototype Populations

Grading agricultural produce has so far eluded automation. Automated grading of fruit and vegetables differs from traditional classification tasks in three major areas:

- a) The criteria for classification are set with respect to a "reference standard," which may be either a "reference sensor" which measures some objective physical properties of the fruit, or a subjective panel of human experts. The on-line inspection system utilizes an "estimated standard" from a set of features derived from one or more "estimating sensors". The estimated standard at best correlates highly to the reference one, and at worst can differ greatly from it.
- b) The class membership feature vectors do not cluster naturally in the feature space. The boundary between classes is usually fixed by marketing or packaging concerns, and often this boundary will pass through an area of the feature space densely filled with feature points.
- c) The *a-priori* class probabilities and the class conditional distributions of the measured features change with time, growth conditions and locations, seasonal variations, storage time, etc. The classifier must reliably grade hundreds of thousands of items per day under time-varying conditions.

Many researchers have examined the first problem, and have determined estimated standards that correlate highly to the desired reference standard in agricultural produce. However the studies correlate estimated and standard sensors on relatively small training sets, in comparison to the millions of items that pass through a typical produce grading line. The typical study's inherent assumption of a constant time-invariant feature space influences the data collection procedure and precludes analysis of the time- dependent variations in the feature space.

Using "estimator sensors" to divide a feature space into arbitrary categories will lead to classification errors. The errors will be concentrated at the category borders. The density of the feature space at the category borders will determine the misclassification rates. These errors can be minimized using the Bayes rule for fixing the class discriminant boundaries, but the errors will be finite and measurable. Any estimate of Bayesian discriminant functions requires estimation of the *a-priori* class probabilities ("priors") and the class-conditional probability densities. The time-varying nature of the priors and the probability densities together with the finite sensor errors have, up to now, yielded in unsatisfactory classifier performance in automated agricultural produce sorting.

Various approaches to adjusting the classifier decision surface to time-varying features exist. Previous classifier decisions can be incorporated into the classifier itself in the "decision directed" approach. The previous decisions are used to continually adjust the classifier to slowly varying feature distributions in a feedback arrangement. The decision directed approach is particularly suitable for "template-matching" based classification tasks, where the reference standard is a noiseless signal or a "correct image". In particular, the template must somehow be clearly distinguished from the noise. In produce classification there are no "correct" templates that can serve as a reference for the adaptive classification process, and as mentioned in the second statement above there is no inherent difference between the classes. The decision-directed adaptive classification approach can be unstable under these conditions, and therefore this approach is not suitable for adaptive sorting of fruit and vegetables.

A more plausible solution to improving classifier performance through time can be developed through a more global view of the classification task. Adaptation generally refers to adjustment of the classifier by inspection of the items over time. For instance, signal or plant parameters can be determined dynamically in the Kalman recursive filter for slow time-varying tasks. The form of the signal or plant estimated must be known *a-priori*. This approach is not applicable to traditional

produce sorting as each item must be classified independently of the previous decisions. However, the produce stream itself can be considered as a "signal" (in statistical terms) to be classified into a finite number of prototype streams or populations. Then for each unique prototype population a separate, optimal "grade classifier" can be designed for sorting individual fruits. This is the algorithm proposed by Peleg and Ben-Hanan (1993) and called "Adaptive Classification by Prototype Populations".

A global "population classifier" decides to which prototype population the current stream of produce belongs, and switches to the appropriate grade classifier. The population classifier utilizes a finite-length stack of features continuously updated from the most recently sorted produce. The statistical features of the stack are analyzed to determine which population is currently passing through the system. When the population classifier determines that the stack contents have originated from a different prototype population, it changes the active grade classifier to the most appropriate one for the current population.

The adaptive classification by prototype population scheme may use some or all of the features used by the grade classifiers, as well as "metafeatures." Examples of metafeatures of a data set are the mean vector of the current stack values, the class-dependent covariance matrices calculated from the stack, or the sample of the cumulative distribution function (CDF) represented by the data in the stack. Use of the CDF in theory and practice will be described, as it is the most appropriate of the metafeatures. The metafeatures may be treated as any other feature in the prototype population classification task. This attempt to use the metafeatures in classification tasks assumes that a finite number of prototype populations exist in the data with different meta-features. The motivation for using only a finite number of prototype populations is the stability problem mentioned above.

1.2.1 Adaptive Sorting Algorithm

This section outlines the overall structure of the adaptive sorting algorithm, with reference to Figure 18. The upper actions are actually external to the current work and involve finding the feature vectors for the data. A discrete process $X = \{x_1, x_2, \dots, x_N\}$ where $x = \mathcal{R}P$ is sampled, and the feature vector x (with p elements) is calculated for each item in the sample $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$. The current report will not describe the actual feature extraction, but will assume that previous work has been done to insure that among the p features there exist $p \leq p$ optimal features for discrimination between the ω_i grades ($i=1..C$). The sampling process can be modeled as:

$$\mathcal{X} = \{x_1, x_2, \dots, x_N\} \subseteq X \text{ where } x = \mathcal{R}P \text{ is a single item of } X \quad (1)$$

Non-Adaptive Classification

The traditional non-adaptive classification scheme is shown with a white background. Parallel to feature acquisition, the sampled items are classified by a reference sensor and grade labels ω_i attached to each. Using any supervised classification techniques, a grade classifier $\omega = g(x)$ is designed for the n vectors in the data set $\mathcal{X}$. In the on-line process, this grade classifier is used to classify the items $x \in X$. The design of the grade classifier is dependent on the training set used. Any variations in the actual data X that are not present in the training set $\mathcal{X}$, if they are even noticed while the classifier is being used, require an adjustment of the classifier to compensate for the new data. While solutions to this problem in one- or two-dimensions exist (see for example Gutman et.al. (1994)), modern sorting machines with multiple sensors overwhelm the human operator.

Figure 18. Adaptive Sorting algorithm flowchart including traditional sorting operation □, prototype population classifier ■, and dynamically updated population classifier ■

Adaptive Classification

The adaptive sorting process is shown with a light gray background. The dataset is partitioned into K *a-priori* known, sequential populations $\mathcal{X}_k$, $k=1,2,\dots,K$ each of length n_k :

$$\mathcal{X}_k = \{x_{m+1}, x_{m+2}, \dots, x_{m+n_k}\}, k=1,2,\dots,K, m = \sum_{j=1}^{k-1} n_j, \mathbf{X} = \bigcup_k \mathcal{X}_k, \quad (2)$$

For each source population, a grade classifier $\omega = g_k(\mathbf{x})$ is designed:

$$\omega = g_k(\mathbf{x}) \quad \forall \mathbf{x} \in \mathcal{X}_k \quad (3)$$

The parameters for each classifier are stored. A population classifier $k = G(\mathcal{X})$ is also designed. Usually, the feature vectors themselves will not suffice to discriminate between prototype populations, and some function of the feature set $f(\mathcal{X})$ must be calculated. The population classifier then becomes:

$$k = G(f(\mathcal{X})) \quad k=1,2,\dots,K \text{ where} \quad (4)$$

$f(\mathcal{X})$ is a function of the set or subset of features in $\mathcal{X}$

We estimate the sample's single-dimensional-feature cumulative probability distribution function and compare it to the prototype population's feature cumulative distribution. Other approaches to the function $f(\mathcal{X})$ have been proposed, but were less successful..

In the on-line operation of the sorter, the past q data are stored in a First In First Out (FIFO) stack, shown as a cylinder in Figure 18. The stack can be modeled as:

$$\mathcal{X} = \{x_m, x_{m+1}, \dots, x_{m+q-1}\} \text{ where } q \text{ is the length of the stack and} \quad (5)$$

$m = 1,2,\dots,N-q$ is the index of the items to be sorted.

After a certain number of new items have been classified, the features of the stack are calculated and the population classifier is used to determine the current population. The appropriate grade classifier $g_k(\mathbf{x})$ is then used for subsequent grade classification. We demonstrate the advantage of this technique by comparing the actual grade label ω_r to a vector of labels obtained by applying the three types of classifier functions discussed:

$$\omega = [\omega_a, \omega_p, \omega_t, \omega_1, \omega_2, \dots, \omega_K]^T \text{ where} \quad (6)$$

$\omega_a = g_k(\mathbf{x})$: $\mathbf{x} \in \mathcal{X}_k$ (the adaptive classifier with an erring population classifier),

$\omega_p = g_k(\mathbf{x})$: $\mathbf{x} \in \mathcal{X}_k$ (the adaptive classifier using a flawless population classifier),

$\omega_t = g(\mathbf{x})$: $\mathbf{x} \in \mathcal{X}$ (the non-adaptive classifier), and

$\omega_k = g_k(\mathbf{x})$: $\mathbf{x} \in \mathcal{X}$, $k=1,2,\dots,K$ (the single population-classifiers used on all the data)

The "flawless" adaptive classifier uses the data point's known population label, and thus flawlessly identifies the proper population. It can be used only in laboratory conditions. The simulations described in the following section show a clear reduction in misclassification rate for ω_a compared to the other strategies.

The adaptive classification scheme must also recognize new prototype populations. The dark gray sections of Figure 18 outside the initial classifier design utilizes a statistical hypothesis test to determine the current prototype population. When the population classifier returns a "not-recognized" label for the data sample $\mathcal{X}$

$$k = G(\mathcal{X}), k \notin \{1,2,\dots,K\} \quad (7)$$

the machine will know that currently in the stack there is an unknown population. Either the stack is in transition between two populations, or indeed the process has encountered a distinctly new population. The Kolmogorov-Smirnov two-sample test, described in Press (1992), is one example of a non-parametric statistical test particularly suited to the task, and is described below.

Given a Bayesian grade classifier, the new population may possibly be modeled as a change in *a-priori* class probabilities. In this case the grade classifier parameters can be automatically modified to more accurately assign grade labels. If adjustment of the priors cannot sufficiently explain the stack contents, a new unique population $k=K+1$ has been encountered. This rare occurrence will require reference grade labels for a new training set to train the new grade classifier.

The optimization box in the classifier design stage is an expression of the need to examine the K *a-priori* populations. While the populations may be distinguishable, they may not require unique grade classifiers. Reduction in the number of prototype populations and subsequent simplification of the population classifier may improve the system's robustness and reduce overall misclassification rates. This can be accomplished by comparing the grade classifier's performance on the "wrong" populations:

$$\omega_p = g_k(\mathbf{x}): \mathbf{x} \in \mathcal{X}_j, j \neq k \quad (8)$$

where the symbols are as in (6), and using the misclassification rates as an indication of the "distance" between the populations. Those populations where the misclassification rates are not significantly different can be combined. An example of this will be published in a paper and in the doctoral thesis supported by this project.

Optimization of the stack length q was also studied. The section in the extreme right of Figure 18 shows a possible alternative to fixed stack length. A shorter stack length will lead to less "lag" in the decision to change the grade classifier, resulting in fewer items belonging to the new population classified by the grade classifier of the previous population. However, too short a stack will not enable the population classifier to confidently identify the actual prototype population. Rather than using a fixed length q for all populations, q can be allowed to vary dynamically, according to the current confidence level of the population classifier decision.

Understanding the Adaptive Sorting Process

The following sections will describe the algorithm in depth. The requirements from the data collection system will be defined. At the core of the adaptive sorting system is the population classifier. A discussion of the ideal stack length will come next, followed by a short section on identification of source populations.

Data Collection Requirements

Small modifications must be made in the way machine manufacturers and packing house staff collect information to be fed into the sorting line. The basic premise is that there is underlying structure in the time-dependent variations in the features used to sort agricultural produce. These stem from the conditions mentioned above, namely

- The conditions under which the produce grow vary across the space of the orchard or field. Most produce are harvested in such a way that these variations are preserved in the bins which are transported to the packing house. The bins are presented to the sorting line consecutively, converting the space-dependent variations to time-dependent variation in the features measured by the machine.
- The produce are harvested at different times. Each field may be harvested under totally different weather conditions or time of day. Again, the harvest order is preserved in the bins which are transported to the packing house.
- Storage times and conditions vary during the year. Most large packing houses strive to be productive 12 months a year, and store the produce at harvest time for later sorting. The produce continues to breathe and loses moisture during storage, changes that are reflected as variations in the measured features on the sorting line.

- Variations in the sorting equipment itself on a daily, weekly, or seasonal basis also can influence the features measured by the sensors in the machine.

To successfully develop an adaptive approach to sorting, these factors of change must be documented. The identification of prototype populations is much easier if the conditions of growth, harvest, storage, and feature acquisition are known. Much of this information is available at the time of feature acquisition in the packing house. Large operations track the grower's produce by lot, and must be accountable to the grower for the final sorted produce. However current machine sorting technology does not record this information. Cooperation is necessary with the machine manufacturer and the packing house to insure that this data is preserved.

Population Classifier Design

The requirements of the population classifier are:

- It must accept as inputs the current contents of the stack $\mathcal{X} = \{x_m, x_{m+1}, \dots, x_{m+q-1}\}$ where q is the length of the stack and $m+q-1$ is the current item x being processed.
- It must return as outputs the current stack population $k = G(\mathcal{X})$ and a confidence measure α , $0 \leq \alpha \leq 1$.
- It must be sensitive to a new stack population even before the stack completely empties of the previous population.
- In order to allow the dynamic adjustment of the stack length, the population classifier's confidence measure α must be insensitive to, or at least compensate for, changes in the stack length.

Stack Parameters

The population classifier can err in two cases. As a new population enters the stack, an optimal population classifier would immediately recognize the data as belonging to a new population and change the grade classifier accordingly. However there is some latency in the identification of a new population. The machine continues to classify the grade by the old population, until it notices that something has changed. Thus the first few items from the new population are sorted by the old grade classifier. Subsequently, the stack continues to fill with the second population. The population classifier may classify the mix as a third population, and sort the items according to its decision. As the first population empties out of the stack, the population classifier should settle on the correct identification of the new population.

The actual asymptotic modeling of the stack length as the number of items sorted becomes large is covered in detail in the doctorate sponsored by this project, to be published around the end of 1997.

Using CDFs to Identify Prototype Populations.

The Cumulative Distribution Function (CDF) incorporates within it the class-dependent probability distributions and the *a-priori* probability of that class in the data. It therefore reflects the parameters needed to calculate the *a-posterior* probabilities. If a classifier can be designed that accurately estimates the *a-posterior* class-dependent probabilities, it will approach the optimal Bayes classifier (Duda and Hart, 1973). Any changes in the data that mandate a change in the Bayes classifier must be expressed in the *a-posterior* probabilities, and will change the CDF. Therefore the CDF is a good starting point for a population classifier statistic.

The sample-estimate of the CDF (created by sampling the random variable and for each value calculating the proportion of data with that value or less) is also much less noisy than its derivative, the probability density function (PDF) most often estimated with histogram techniques. It must be a monotonically increasing function, and its steps are in increments of $1/N$ where N is the number of data points in the sample. The PDF estimate, however, is dependent on the difference between consecutively ordered values of the sample, and usually requires some signal-processing type of smoothing to be useful.

The CDF is visually less satisfying, since it is not obvious from the S-shaped curve whether the distribution is unimodal, skewed, or contains outliers. But a wide variety of theoretical non-parametric techniques for analyzing differences between CDFs are known in the literature (Fukunaga 1990, Conover 1980).

Peleg and Ben-Hanan (1993) trained a neural network to identify the population's CDF. They tested a two-population data set collected on apples, with limited success. The neural network was built in such a fashion that it could return not only the distance metric but a measure of the confidence in its decision, as described in the previous paragraph. A different metric was formed by calculating the root mean square (RMS) difference between the CDFs at a constant number of fixed values. The number of measures and their values are chosen at training time to ignore the extreme tails of the data while weighting the center of the CDF more heavily. This metric was used in the simulation described below. No expression of the confidence measure for this metric was found.

The Kolmogorov-Smirnov goodness-of-fit test was found to be particularly useful, and will be described in detail below.

Population Classification by the Kolmogorov Smirnov Statistic

The Kolmogorov-Smirnov two-sided statistic T , as described in Press (1992) was proposed by Kolmogorov and tabulated by Smirnov. It is defined as the maximum difference between two cumulative distribution functions:

$$T = \max |S_1(x) - S_2(x)| \text{ where } S_n(x) \text{ is the sample cumulative distribution function for a sample drawn from a random variable } X_n \text{ with a distribution of } P_n(x). \quad (9)$$

Under the hypothesis H_0 that $P_1(x) = P_2(x)$, the distribution of T can be tabulated as a function of the number of samples in X_1 and X_2 . An analytical expression to calculate the significance of the T value was proposed by Stephens (1963). The probability that the distributions characterized by two CDFs are NOT different is:

$$\text{Probability}(H_0: P_1(x) = P_2(x) | T) = f_{KS} \left(T \cdot \left[\sqrt{N_e} + 0.12 + \frac{0.11}{\sqrt{N_e}} \right] \right) \quad (10)$$

where N_e is the effective number of points: $N_e = \frac{N_1 * N_2}{N_1 + N_2}$, N_m = number of points in

$$\text{sample } X_m \text{ and } f_{KS}(\lambda) = 2 \sum_{j=1}^{\infty} (-1)^{j-1} e^{-2j^2 \lambda^2}, \quad 0 \leq f_{KS}(\lambda) \leq 1$$

The hypothesis test can act as a population classifier $k = G(X)$:

$G(X) = n: (T_n < T_m \forall m \neq n)$, where

$$T_n = \max |S_n(x) - S_{STACK}(x)|, \text{ where} \quad (11)$$

$S_n(x)$ is the sample cumulative distribution function from the population n in the training set, and S_{STACK} is the current stack's CDF.

In addition $\alpha = f_{KS}(T_n * N_s)$ where $N_s = \frac{N_{STACK} * N_n}{N_{STACK} + N_n}$ gives the probability that the CDFs are, indeed, equal. Thus the KS test can answer the first requirement for the population classifier stated above.

Sensitivity to a New Population Entering the Stack

The KS test is not sensitive to the order of the data in the stack. At its core it reorders the data to find its CDF. It is not feasible to somehow emphasize the first few datum in the stack to improve the sensitivity of the classifier to a new population, as would be possible with a different family of

classifiers. However the CDF itself is very sensitive to changes in the stack composition, and the entrance of a new population to the stack rapidly changes the measure α described above.

Ability to Compensate for Changing Stack Length

The measure N_s takes into account the stack length N_{STACK} . The classifier accuracy depends on both the size of the training set and the size of the stack, but its performance depends greatly on enough data present in the stack to estimate the CDF.

1.2.2 Simulation of the Adaptive Sorting Algorithm

Adoption of the adaptive sorting algorithm makes new demands on machine manufacturers. A user friendly proof of the advantages of the adaptive techniques is necessary to convince commercial interests. The possible reduction in classification errors must be clearly demonstrated in actual packing house conditions. A software tool was developed for this purpose that permit use of the adaptive algorithm while using a unified data input and presentation interface. Both synthetic data and actual data may be used with the program.

The simulation was built in Matlab (Mathworks 1994), using Graphical User Interface (GUI) capabilities to display the data and significant results while allowing the user to easily change the simulation environment. As the simulation runs, it displays a series of sorting matrices. These compare the actual grade labels w_r as determined by a reference sensor to those determined by classifiers as described in Equation (6) above.

After processing the entire test data, the program displays information about the performance of the strategies, including the number of items processed and the number of errors made by the population classifier used in the adaptive technique, as well as the overall misclassification rate expressed as a "Weighted Grade Contamination Index" C_ω , described in Peleg (1985):

$$C_\omega = 1 - P_\omega = 1 - \sum_i P_{gi} W_i \quad (12)$$

The quantity P_ω is the opposite of C_ω and is termed "Weighted Grade Purity Index", while P_{gi} is the pure product fraction of grade i , resulting from the sorting operation.

$W_i = \frac{K_i P_i}{\sum_i K_i P_i}$ is a weighting function composed of the relative costs or penalties K_i

for misclassifying grade i while the *a-priori* grade probabilities P_i are determined by the proportion of grade i in the sorted produce stream. ...

In practice, larger penalties K_i are associated with the higher priced grades but in the context of this project we assumed equal penalties for all grades, $K_i = 1$.

Simulations were run on three aspects of the stated problem.

- The size of the training set used for the various classifiers was altered while all other parameters were held constant.
- The size of the stack used in the population classifier and the size of the data set were varied independently while all other parameters were held constant.
- After processing the entire test data, the program displays information about the performance of the strategies, including the number of items processed and the number of errors made by the population classifier used in the adaptive technique.

Data Creation

The data design philosophy was based on the three difficulties in classifying agricultural produce: an erring sensor is used to measure features for classification originally performed by a reference sensor; the grades do not cluster naturally in the feature space; and the prior grade probabilities

and the class-conditional distribution functions are not constant throughout the classification task. Two-dimensional feature vectors $[x_1, x_2]$ ($p=2$) were generated for each population. They were distributed normally $N(\mu, \Sigma)$ where the covariance matrix Σ for each population was the unit matrix multiplied by an integer, and all had the same mean μ . The vectors were labeled with grades using a constant "reference" function. The function determined whether the point was inside either of two, two-dimensional, concentric boxes as shown in Figure 19. If the point was inside the inner box, it was labeled as belonging to grade ω_1 . The points between the two boxes were labeled ω_2 . All points outside the outer box received the label ω_3 . The labeling function is similar to a two-feature hierarchical classifier where each feature has an optimal value. For instance tomatoes must be neither too small nor too large, and red but not too red.

Subsequent to labeling, random Gaussian error of the form $N(\mu, \Sigma)$ was added to each data point. The noise function mean μ was different for each population while the covariance matrix was held constant. The error mean was equal in magnitude across the populations, but its direction vector differed. The error modeled the difference between the reference sensor and the on-line erring sensor. For instance, the reference color of the tomato could be measured by an integrating sphere colorimeter, while the on-line classifier would use a color value determined by one view of the fruit alone. The sensor error would depend on the inclusion of the calyx or stem area in the image. This error would in turn depend on the fruit geometry, which could vary from grower to grower or field to field.

In Figure 19 a schematic representation of the data clouds is shown. The position of the common mean of the original population data clouds preceding error addition is marked with an asterisk (*). Note that the error mean moved each cloud an equal distance from the original center position, but in different directions. The grade labels for each point in the population were determined by the location of the point relative to the gray squares before the addition of the sensor error. After each point was labeled, the error was added to shift the mean for the population to the point marked with a number: 1 for population $\mathcal{X}_1$ and so on. The differing means represent the change in the sensor error functions between populations. The magnitude of the error bias, and the covariance matrix for its Gaussian distribution were equal for all the populations. The size of the circle represents the covariance matrix of the population's generating function. The data creation parameters for the grades and populations are summarized in Table 6.

The data set created contained two parts: a training set and a test set. Each population was equally represented in both the training and test sets.

Simulation Results

A k-NN grade classifier with $k=1$ was trained for each population using a portion of the data set for the training set. The default training set size was 100 items from each population. The entire training set was used to train the non-adaptive classifier. The grade classifiers (essentially the training data for this type of classifier) were stored for each population.

Two metrics were used to classify the populations. The K-S metric T (described above) was compared to a root mean square (RMS) distance between the two CDFs at 20 constant values. Feature 1 was used for this calculation. The grade classifier belonging to the population with the minimum distance metric was used to determine grade labels. The default stack size was 50 data points.

A test set of data was also created, using the same parameters as the training set. By default it consisted of 500 data points from each population, arranged consecutively by population. A simulation run consisted of data set creation, classifier training, and classifier testing using the test set.

Since the populations $\mathcal{X}_1$, $\mathcal{X}_2$, $\mathcal{X}_3$, and $\mathcal{X}_4$ appeared consecutively in the test data, with equal prior probabilities, there were three population transitions in the default test set.

Each of the simulation runs resulted in sorting matrices for each classifier strategy. Typical sorting matrices are presented in Table 7, along with the total misclassification rates for each classifier. In each sorting matrix the actual class labels are compared to the classifier labels, and totals for each grade are given. The correct decisions appear on the diagonals and are highlighted. Below each

sorting matrix the number of misclassified items is shown. The final table shows the misclassification rate for each simulated classifier.

It can be seen that the adaptive classifier succeeded in improving the classification error rate over all the non-adaptive classifiers trained on the entire data set. Even when the population classifier error is finite, the adaptive scheme succeeded in improving the misclassification rate. Note that the simulation assumes all the populations will be available in the training set, and that the training data were labeled with no error. In actual agricultural sorting tasks, the total space of populations is not known at any point in time. The next harvest always brings with it some surprises. While in both the adaptive and the traditional global classification techniques a new classifier must be trained on the new data, the adaptive classifier provides a framework for automatic detection of changed data parameters. The user of the global classifier will only detect the changes after the produce has passed through the sorting machine and quality control rejects the sort.

Table 6. Simulation Data Characteristics: Four populations created from two-dimensional Gaussian distributions with added error.

Popu- lation	Population Generation: Normal Distribution with Parameters		Added Error: Normal Distribution with Parameters	
	Mean	Covariance matrix $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} * M$	Mean $\mu = \mu_m * 0.75$	Covariance matrix $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} * M$
$\mathcal{X}_1$	$\mu = [10, 10]$	$M = 4$	$\mu_m = [-1, 1]$	$M = 0.5$
$\mathcal{X}_2$	$\mu = [10, 10]$	$M = 4$	$\mu_m = [1, 1]$	$M = 0.5$
$\mathcal{X}_3$	$\mu = [10, 10]$	$M = 2$	$\mu_m = [1, -1]$	$M = 0.5$
$\mathcal{X}_4$	$\mu = [10, 10]$	$M = 2$	$\mu_m = [-1, -1]$	$M = 0.5$

Table 7. Simulation Results. The 2000 data points were classified by 6 strategies, as in equation (6). The population was classified by the Kalmogorov-Smimov statistic T . The final table consolidates the contamination indices.

Optimal Non-Adaptive Classifier					Adaptive Classifier				
Actual		ω_1	ω_2	ω_3	Total	ω_1	ω_2	ω_3	Total
	ω_1	56	113	111	280	172	104	4	280
	ω_2	122	301	256	768	106	563	99	768
	ω_3	138	306	508	952	4	153	795	952
	Total	316	720	964	2000	282	820	898	2000
	Error	260	419	456		110	257	103	

All data by χ_1 Classifier					All data by χ_2 Classifier				
Actual		ω_1	ω_2	ω_3	Total	ω_1	ω_2	ω_3	Total
	ω_1	138	131	11	280	153	119	8	280
	ω_2	164	499	105	768	218	458	92	768
	ω_3	5	175	772	952	18	222	712	952
	Total	307	805	888	2000	389	799	812	2000
	Error	169	306	116		236	341	100	

All data by χ_3 Classifier					All data by χ_4 Classifier				
Actual		ω_1	ω_2	ω_3	Total	ω_1	ω_2	ω_3	Total
	ω_1	144	132	4	280	121	154	5	280
	ω_2	119	513	136	768	87	455	226	768
	ω_3	4	193	755	952	2	164	786	952
	Total	267	838	895	2000	210	773	1017	2000
	Error	123	325	140		89	318	231	

Classifier	Misclassification Rate
Optimal Non-Adaptive	30.7%
Optimal Adaptive	23.0%
Adaptive w/Erring Pop. Class	23.5%
χ_1	29.5%
χ_2	33.9%
χ_3	29.4%
χ_4	31.9%

Misclassification as a Function of Population Dissimilarities

In the first simulation, the effect of varying the similarity of the populations (the magnitude of error added to each population after reference sensor labeling) was explored. Figure 20 shows the misclassification rate determined by the classifier, as a function of the magnitude of the added sensor error for all seven classifier strategies. The process of data creation using a new data set, classifier training, and simulation run was repeated four times at each error magnitude for each classification strategy. Other than the error magnitude, all other data parameters were as described in Table 6. While running the simulation, the stack size was constant at 100 points. The test data set comprised 250 data points from each population ordered consecutively.

The two lower solid lines result from use of one of the adaptive strategies, while the dashed lines show the results of the classifiers trained for a single population. The upper solid line between the dashed lines in the non-adaptive global classifier's result.

The added error magnitude has an impact on the misclassification rate of the non-adaptive strategies while the adaptive ones remain almost insensitive to the changes. This demonstrates the adaptive strategy's ability to overcome changes between populations. The differences between the two adaptive strategies to the others become statistically significant at the 0.95 level for added error magnitudes larger than 0.6.

Misclassification as a Function of Training Set Size

A second simulation was run to explore the effect of the training set size on the misclassification error. For each training set size, four data sets were created using the default values in table 6. After training the classifiers, a test set of data consisting of 250 data points from each of the four populations was classified. The results are shown in Figure 21.

The misclassification rate diminishes with increasing size of the training set, as expected. The adaptive strategies' misclassification rates are (statistically) significantly lower than the non-adaptive ones for training set sizes above 30 with confidence at the 0.95 level. The misclassification rates for the small training sets show little difference between the strategies. This phenomenon is due to the poor single-population classifier performance for such small training sets.

Misclassification as a Function of Stack Size

A third simulation was run to explore the influence of the stack size on the population classifier. The fulfillment of the adaptive strategy's potential is dependent on the ability to correctly identify the current population. As stated above, the population classifier error can be modeled as the sum of the errors during transition between two populations, and errors in the classifier itself. In Figure 22 the effect of different stack length on the population classifier misclassification rate is shown. For each stack size, new data sets were created with the default values of table 6. Each data set was run through the procedure of classifier train and test. The training set size was fixed at 100 points from each population. Four sets of train and test data were created for each size of 1500, 2000, 2500 and 3000 points. In each, the order of appearance of the populations was $X_1, X_2, X_3, X_4, X_3, X_2, X_1, X_4$. A minimum is obvious in each curve.

To further explore the results, a model of the population classifier error was developed. The classification errors caused by more than one population being present in the stack are termed the "transitional errors" E_T , the probability of this error occurring given that the stack contains more than one population is $P(E_T)$, and the proportion of items classified under these conditions is termed $P(\text{transition})$ or $P(T)$. If the stack is composed of a single population and the population is identified incorrectly, the classifier itself has erred. These errors are termed classifier errors E_G . The proportion of items classified under these conditions is $1-P(T)$.

The overall probability of population classifier error is the total of these two errors:

$$P(G(X)=k | x \in X_k) = P(E_T) \cdot P(T) + P(E_G) \cdot (1-P(T)) \text{ where} \quad (13)$$

$P(\bullet)$ is either the theoretical probability of $\bullet$ or the proportion of $\bullet$ in the total sample,

$G(X)$ is the population classifier decision and

x is the feature vector of the item currently being sorted.

$P(T)$ is the number of transitions between populations in the classified data multiplied by the stack length. Given a reasonable choice of $G(X)$, $P(E_T)$ will be less than $P(E_G)$. But $G(X)$ is dependent on the stack length: a longer stack should improve the power of $G(X)$. Thus a graph of equation (13) as a function of stack length will have a minimum at the optimal stack length. Moreover, for a given number of transitions between populations, increasing the size of the test set will reduce $P(T)$. Thus a reduction in classifier error should become apparent as the test set size increases as is demonstrated in Figure 22.

To further analyze the minimum-error stack size, the stack sizes were normalized by the size of the test data set, neutralizing differences in $P(T)$. The curves in Figure 23 show that for given data parameters the optimal stack size is around 2.5 percent of the total test data set.

In conclusion, it can be seen that in all the cases the adaptive classifier succeeded in improving the classification error rate over all the non-adaptive classifiers trained on the entire data set. Even when the population classifier error is finite, the adaptive scheme succeeded in improving the misclassification rate.

Comparison of the K-S Metric to the RMS Metric

Many different metrics for measuring distances between CDFs can be developed. A comparison of the strength of the K-S metric to a RMS type metric was made. Data sets were created with the default parameters, modifying stack length and test set size. In each case the test set size was set

to 40 times the stack length, mimicking the optimal behavior from Figure 23. Table 8 shows that the K-S metric performed better as long as the stack length was greater than 30.

Table 8 - Comparison of metrics for measuring the differences between CDFs. In each run the data creation parameters were left at default values, except stack length and test set size, which was set to 40 times the stack length.			
Population Classifier Errors	Stack Length	Metric	
		T (K-S statistic)	RMS
	20	0.32	0.32
	30	0.28	0.28
	40	0.18	0.34
	50	0.21	0.39
	60	0.16	0.30
	70	0.24	0.38

1.2.3 Adaptive sorting of Dates

The above adaptive sorting experiments with artificial datasets were useful for demonstrating the structure and operation of the adaptive sorting scheme by prototype populations. They were furthermore very useful for exploring sensitivities to the system's parameter changes.

The purpose of experiments with real data from a commercial sorting operation was to confirm and demonstrate a basic premise of the project, i.e. that global feature spaces of commercial produce grading systems can be beneficially subdivided into prototype populations. Furthermore, we wanted to confirm and demonstrate that these prototype populations are sufficiently dissimilar, whereby a population classifier can be designed to distinguish between them and by using separate grade classifiers for each, the overall misclassification rate can be significantly reduced.

According to the original research plan, we wanted to conduct adaptive sorting simulations with carrot data collected in the US and date data collected in Israel.

To do so, the data must be organized as the artificial data was, namely for each datum vector with p features, its population and reference label must be supplied. Unfortunately this precluded the use of the carrot data collected by the US collaborators. The data there was collected from a variety of growers and conditions, but the produce lost its identity as it was transferred from the growers through the packing house to the researchers. Consequently, the adaptive sorting experiments were run only on date data.

The date data was collected from the Ardom Regional Packing House, in the Southern Arava region of Israel, near Eilat. This packing house recently installed, (with the supervision of one of the researchers), a newly developed date sorting machine. Using a set of 25 features obtained by machine vision and a hierarchical classifier, it can sort about 20 dates per second. The manufacturers of the machine agreed to modify the software so that the machine saves the features of the last 24 dates sorted once every five minutes. The packing house tracks the source of the date lots through the packing process to properly compensate the growers, thus an *a-priori* population label is available and was stored as well as the time the sorting took place and the output of the sorting machine.

For commercial reasons, the manufacturer requested that the actual physical meaning of the features and the exact structure of the grade classifier remain in confidence. This was not a problem since the adaptive sorting simulation program was designed to work with any type of grade classifier and accept various feature structures, as may be used by various commercial sorting machines, which employ conventional non-adaptive sorting algorithms.

Data Preparation

The raw data were supplied as a number of ASCII files. Since the processing platform of choice is Matlab, a number of conversions were necessary to input the data. The data were parsed into a matrix form, with each row representing a data vector. Unreasonable data were rejected. All data where the output grade label reflected the inability of the machine to reach a decision were rejected. All clear outlier data on any feature was rejected.

The result was a data set of ~20,000 vectors. The first element represents the time the date passed through the machine, the second was the shipment ID (every incoming shipment of dates receives a unique label which is used to track its progress through the packing house and to compensate the grower) The third was the machine that sorted the date (there are 2 machines working in parallel) and elements 4-28 are features.

A unique demi-population label was constructed from the following:

- Time and date sorted
- Shipment ID
- Machine sorted on (1 or 2)

Feature selection and date grade classifier design

The general structure of the hierarchical classifier used by the machine is depicted in figure 24. In classification phase 1, a first subset of the 25 features set is used to compute a primary score termed F1 (an integer 1-5). In the second classification phase a linear discriminant classifier uses a second subset of the 25 features to compute a secondary score F2 (an integer 1-4). In the third phase the scores F1 and F2 are combined in various ways to yield quality grades. The combinations of the F1 and F2 values and the number of quality grades is fixed by the destination market supply and demand situations and time of the year.

Score F1 was one dimensional while F2 was composed of 24 features. Although the values of F1 and F2 were supplied in the dataset the features composition of F2 was not disclosed by the manufacturer. Consequently, we adopted the values of F1 and F2 as measurements by a reference sensor and the grades labels of the machine as reference labels. As the estimating sensor succeeded in estimating F1 very accurately, (less than 1% misclassification) it was decided that there would be no further improvement by adaptively estimating the F1 score. Consequently, the goal of the sorting simulation was limited to adaptively executing phase 2 - estimating F2.

In the machine, the third phase is performed by the sorting line operator. Using a touch screen for input, the operator chooses combinations of F1 and F2 to form particular quality grade. The actual mapping of F1 and F2 to a grade label depends on the destination of the finished product and the market forces. This process is beyond the realm of the adaptive sorting algorithm.

Through consultation with the machine manufacturer, and after considerable experimentation we constructed estimating feature subsets to approximate F1 and F2 and a hierarchical grade classifier similar to the one used in the machine (see Figure 24).

Subsequent to dividing the dates into subgroups by F1, linear discriminant classifiers as described in the SAS package (SAS 1985) were used to determine the estimate of F2:

$$\hat{F}2 = g_{F1}(x) = \min(D_{\omega(i)c}(x)) \text{ where} \quad (14)$$

$$D_{\omega(i)c}(x) = (x - \bar{X}_{\omega(i)c})^T S_{\omega(i)c}^{-1} (x - \bar{X}_{\omega(i)c}) + \log |S_{\omega(i)c}| - 2 \log P(X_{\omega(i)c}),$$

$X_{\omega(i)c}$ is the data in the training set for the grade class ω_i with $F1 = c$,

$\bar{X}_{\omega(i)c}$ is the mean vector of $X_{\omega(i)c}$, $S_{\omega(i)c}$ is the covariance matrix, and

$P(X_{\omega(i)c})$ is the prior probability of $X_{\omega(i)c}$ in the part of the population with $F1 = c$.

The classifier is based on multivariate normal (Gaussian) distributions. However the data is decidedly non-normal. Each feature was examined for skew (the degree of asymmetry around the mean). As some of the features were highly skewed a sequence of Tukey ladder functions ($\dots x^{-2}$, x^{-1} , $x^{-1/2}$, $x^{-1/4}$, $\log(x)$, $x^{1/4}$, $x^{1/2}$, x , x^2 , ...) (Tukey 1977) was applied to each feature until the skew was less than 1. While the grade classifier construction has been described at some length here, the adaptive sorting algorithm is external to the particular grade classifier and assumes the machine manufacturer or the packing house staff can provide a suitable grade classifier. The specific case in-hand demonstrates the adaptive classifier's ability to incorporate even a complicated grade classifier.

A blind stepwise process (patterned after the STEPWISE procedure in SAS (SAS 1985) of feature selection was used to chose the features to be used for estimating F2. An upper limit of 9 features

yielded acceptable performance. Using a global training set, this classifier provided an overall misclassification rate of 35%, versus the reference labels given by the machine.

Partitioning the data into Prototype Populations

The first step in partitioning the data into prototype populations involves choosing an optimal feature for characterizing prototype populations by CDFs and the K-S statistic. Some consideration was given to using a two- or higher-dimensional form of the K-S statistic. These have been developed by Smallwood (1996), Feltz and Goldin (1991), and others. However, one of the prime advantages of the K-S statistic is its non-parametric character: the test does not depend on the underlying distribution of the data. In the tests developed for higher dimensions, this characteristic is not preserved. In addition, the "curse of dimensionality" (as described in Scott 1992) requires more data to adequately describe the two dimensional distribution than the one-dimensional distribution.

The fact that the population classifier utilizes only one feature may not be a severe handicap. In most cases, the actual classification task relies on a main feature, with nuances supplied by the others. Many of the acquired features are highly dependent. Any change in the makeup of the produce stream that has repercussions for sorting, will manifest itself in all the dependent features, including the one chosen for population classification.

Even with one dimensional CDFs, optimal partitioning of the data into prototype populations is a difficult task. A necessary precondition for a feasible partition is that the prototype populations must be "identifiable", i.e. the differences between the resulting CDFs must be sufficiently large. One measure of the identifiability of the prototype populations is the K-S statistic α , another is the misclassification rate in the resultant classifier. Obviously, there may be many identifiable partitions which will invariably provide a "payoff" in terms of increasing the weighted grade purity index P_{ω} versus non-adaptive sorting, wherein the entire dataset comprises just one prototype population.

Among the various identifiable partitions, the optimal partition should provide the maximal "payoff": $\max(P_{\omega})$. However, the partition with the minimal population misclassification rate is not necessarily the one which provides the maximal payoff $\max(P_{\omega})$. A full exploration of these fine details, for finding optimal prototype populations will be included in the doctorate sponsored by this project, to be published around the end of 1997.

In the context of this BARD project a semi-optimal partition of the date dataset was determined by choosing an optimal feature for generating CDFs and reducing the number a-priori prototype populations, as follows:

The maximum distance in the training-set sample CDFs between all possible binary combinations of *a-priori* known date populations was determined. Then an agglomerative process (based on the CLUSTER process of SAS (1985)) of combining populations based on this distance measure was carried out. This resulted in a dendrogram depicted in Figure 25, which shows the estimated confidence that the populations are different. Each feature's dendrogram was examined, and the feature with the maximum separation of the *a-priori* populations was chosen.

The dendrogram also provided a rationale for combining populations. The original 20 populations were reduced to 16 by combining those with a low (<0.6) probability of coming from different populations.

Population Classifier Performance

The performance of the population classifier was tested by dividing the data into a training and test set. The test set data was formed into a indexed sequence of data, and the data fed into a FIFO stack for classification. Both the proportion of data used to train the classifier and the length of the stack were varied. The results of these tests are depicted in Figures 26.27. It may be seen that the population classifier exhibits the expected behavior. For a given training set size an optimum stack length is obvious. The classifier performance improves with larger training sets, till an asymptotic minimum is reached.

The misclassification rate has a minimum as a function of the stack size, since on the one hand increasing the stack length improves the accuracy of the classification, but it also increases the number of classifications made while the stack is in transition between two populations (for a given data set size). The dependence of misclassification rate on training set size is another expression of the same phenomenon: as more data is taken into the training set the classifier accuracy improves. However, since less data is left for the test set, the proportion of data classified while the

stack is in transition grows, increasing the classification errors. This is similar to the behavior depicted in Figures 22,23.

Adaptive vs. Non-Adaptive Classifier Performance

The particular grade classifier chosen depends on an adequate estimation of the parameters of each F1 value's mean $\bar{\mathcal{X}}_{\omega(i)c}$ and covariance matrix $\mathbf{S}_{\omega(i)c}$. The classifier was trained by selecting a random sample from the entire ~20,000 vector data set, executing phase 1 of the classifier (see

Figure 24), and for each value of F1 calculating the mean vector $\bar{X}_{\omega(i)c}$ and the sample covariance matrix $S_{\omega(i)c}$ for each reference F2 value.

The 20 (five F1 values and four F2 values) different vector/matrix combinations were stored. These were the actual parameters used in the grade classifier. Subsequently, the test set data were classified and the estimated F2 value compared to the known, reference value. Various sizes of training sets were run through the process. The classifier was also trained and tested on each *a-priori* population separately. Again, the incorrect decisions were counted as a percent of the data. Table 9 consolidates the misclassification rates of the grade classifier, by training and test set.

Table 9 - Non-Adaptive Grade Classifier Performance - Misclassification rates for each of the hierarchical grade classifiers trained.																	
Training set from Population	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	all
Test set from only the trained population	22.4	29.0	20.4	23.6	15.8	13.5	23.3	16.2	27.9	19.5	29.3	21.0	11.6	31.6	24.3	21.8	33.3
Test set from the entire data	39.2	45.8	43.2	43.6	39.5	49.9	64.9	39.8	36.1	51.9	47.3	43.8	48.2	51.6	46.7	58.4	

The misclassification rates clearly demonstrate the advantage of adaptive sorting. Using the same training set a clear reduction in misclassification can be obtained by "tailoring" the classifier to the particular feature space currently being sorted. The table also emphasizes the price of a mistake in the population classifier. Using the wrong grade classifier for any extended period will undo all the reduction in overall misclassification.

Table 10 presents the overall comparison of the different classification strategies. The two optimal classifiers are compared to the adaptive classifier. In real-life situations, the non-adaptive classifier performance will be significantly worse, as the training set cannot anticipate all possible variations in the produce. The adaptive classifier avoids this problem through the identification of new populations. It can notify the packing house staff that something has changed and adjustments are necessary much before post-sorting quality control (if it exists).

Table 10 - Overall Classifier Performance Comparison of C_{ω} , stack length at 100 and training set size at 30%	
Classifier	Misclassification Rate
Optimal Non-Adaptive	33.3%
Optimal Adaptive	22.6%
Adaptive w/Erring Pop. Class	26.2%

The combination of the grade classifier error and the population classifier error gives the overall adaptive sorting error. This was tabulated for the same conditions as the test of the population classifier above. In Figure 28 the adaptive sorting error is plotted as a function of stack length. Eleven runs were made at each stack length. For reference, the optimal non-adaptive and optimal adaptive misclassification rates, which are not dependent on the stack, are shown. The adaptive sorting technique produced results as expected: between the two bounds.

In Figure 29 the adaptive sorting misclassification rate is plotted as a function of training set size. Again, the optimal non-adaptive and optimal adaptive classifiers' performances are also shown. The non-adaptive classifier performance is relatively insensitive to training set size, since a training set size of even 1% on such a large data set yields an accurate estimate of the mean and standard deviation needed for classification. The adaptive classifier performs better than the non-adaptive one, and can utilize the larger training set size to improve its estimate of each single population's parameters, improving performance.

The populations were ordered as presented in the original data. An additional test was run feeding the data into the classifier in reverse order, to test the performance of the population classifier with the same data under different conditions. The differences in misclassification rate C_{ω} were not significant at the 90% confidence level.

2. Evaluation of research achievements

The overall objective of this project as stated in the original proposal to BARD was to lay a scientific and engineering basis for development of a "Programmable Imaging Sensor and Adaptive Classifier System," for optimal sorting of fresh produce.

The specific objectives were:

1. Incorporate adaptive, wavelength selectable imaging into a system capable of measuring unique quality indicators, and test that system with two selected fruit types, e.g., carrots and dates.
2. For the two selected fruits, develop adaptive classification algorithms, based on a finite number of "prototype populations" with distinctly different feature distributions.
3. Develop algorithms for on-line updating of classification feature distributions, by automatic collection of samples from the currently sorted products.

The US team was primarily responsible for objective 1, and the Israeli team for objectives 2 and 3.

We are pleased to report that by and large these objectives were fulfilled. Nevertheless, there are minor difference between the above initially stated objectives and the actual achievements. Particularly, objective 1 was fulfilled using only on carrots, while objectives 2 and 3 were fulfilled using only dates. In other words, each team fulfilled its objectives as planned but we were unable to combine the "adaptive imager" and "adaptive classifier" into an integrated system. This does not detract from the main achievements of the project, inasmuch as the "adaptive imager" and "adaptive classifier" are essentially independent systems, each provides a contribution in its own right, to improved sorting of agricultural products.

Below we summarize in some detail the particular achievements in the adaptive imaging and sorting systems.

2.1 Adaptive imaging

According to the original research plan the objectives of the US team were to design and construct a multispectral adaptive sensor, upgrade the existing carrot inspection system (INSPECT), and build carrot data sets obtained from different packing houses. Accordingly, the research objectives were addressed as follows:

1. The desired specifications of the multispectral sensor were defined in detail. Three different technologies were considered and the most appropriate for the implementation of the sensor was selected.
2. Based on an Acousto-Optic Tunable Filter, a multispectral imaging sensor was constructed. The sensor is capable of acquiring images in multiple wavelengths in the range of 500 to 1000nm.
3. Hardware and software was developed in order to enable multispectral image acquisition. An appropriate illumination chamber was constructed. All hardware components were integrated into one system: camera, filter, and frame grabber. Software for multispectral image acquisition was developed.
4. The existing carrot inspection system (INSPECT v1.0) was upgraded to INSPECT v2.0, by developing and implementing prototype algorithms for high speed image processing and classification. The algorithms were implemented in a pipelined architecture using a line scan camera and a digital signal processor.
5. Carrot samples were obtained from a packing house in South Texas. The upgraded INSPECT system was used to build a data base of carrot features and to archive carrot images. A total number of 2800 carrot images were acquired, processed and stored. The constructed data base included carrot samples that originate from different fields, harvested at different times and have different sample population characteristics.

The results of this project were presented at the 1995 International ASAE Meeting:

- Alchanatis, V. and Searcy, S.,W. (1995). High Speed Inspection of Carrots with a Pipelined Image Processing System. ASAE International meeting, June 18-23, Chicago, IL. Paper #953170
- Alchanatis, V. and Searcy, S.,W. (1995). A Selectable Wavelength Imaging Sensor for Multispectral Inspection of Agricultural Products. ASAE International meeting, June 18-23, Chicago, IL. Paper #953210

The two first objectives were fulfilled according the original plan. The construction of carrot data sets was not completely met as originally planned due to several reasons: the construction and evaluation of the multispectral imaging system was finished only at the end of the second year due to technical problems encountered from using newly developed AOTF imaging technology. Therefore, the last year of the project was left for extensive multispectral image acquisition. The work was planned to be performed by a graduate assistant during 1996. Unfortunately, it was difficult to find graduate assistants, and even when found, they remained for a limited time of a couple of months, which was not adequate to gain sufficient expertise with the system. Therefore, the extensive multispectral image acquisition of carrots was not accomplished as planned.

2.2 Adaptive sorting

The Israeli research team focused on developing the adaptive classifier strategy as an integral part of a wider D.Sc. research thesis project of Matt Picus, under the direction of the cooperating investigator. This thesis is entitled: "Optimal Adaptive Classification by Prototype Populations", it will be completed approximately by the end of 1997. Most of the theoretical and all the experimental parts of the project have been completed in the context of this BARD project, as reported above. Some of the finer theoretical points, (as required for a D.Sc. thesis) still remain to

be completed. The practical aspects, which will allow implementation of the adaptive sorting approach in a packing house sorting line are complete, as summarized below:

1. The adaptive classifier algorithm has been modeled and reduced to a series of modules that can be added to any existing sorting line. A simulation of the entire process was created in Matlab using graphical user interface techniques to promote the accessibility of the difficult theoretical subjects.
2. Typical Grade classifiers based on k-Nearest Neighbor techniques and linear discriminants were implemented. While a necessary part of the overall simulation, the adaptive sorting algorithm assumes that in practice conventional grade classifiers already exist and can be provided by the sorting machine manufacturer or developed in cooperation with the management of the packing house.
3. The sample histogram, estimating the cumulative distribution function (CDF), was chosen as a characterizing feature of prototype populations, whereby the Kolmogorov-Smirnov statistic was employed as a population classifier.
4. Simulations were run on artificial data with two-dimensions, four populations and three classes. A quantitative analysis of the adaptive classifier's dependence on population separation, training set size, and stack length determined optimal values for the different parameters involved. A clear advantage was shown for the adaptive classifier, even after taking into account the errors in the population classifier stage.
5. Extensive simulations were also run on actual data collected over a 4 month period from an Israeli date packing house. The same type of study as above was again undertaken, and the results again showed a clear reduction in classification error by using the adaptive technique versus non-adaptive sorting.
6. Some of these results were presented at two International meetings:
 - Peleg. K. "Automated packing house for fruits and vegetables" International Conference on Harvest and Postharvest Technologies for Fresh Fruits and Vegetables, Guanajuato Mexico, February 1995.
 - Picus M. and Peleg K. "Adaptive Classification of Agricultural Produce" International Congress for Computer Technology in Agriculture 1996 on Information and Communication Technology applications in Agriculture, Wageningen, The Netherlands June 16-19, 1996.

3. Description of cooperation

The cooperation between the two research teams was firmly established by a post-doctoral one-year visit at Texas A&M of Dr. Victor Alachantis, a former D.Sc. student of the cooperating investigator. After returning to Israel, Dr. Alachantis accepted a research post at the Volcani Institute ARO. He continued to participate in the project and helped to prepare this report.

Additionally, the cooperating investigator visited Texas A&M and was introduced to the AOTF adaptive imaging system. The principal investigator visited the Technion and was introduced to the finer points and difficulties in developing the adaptive sorting system.

The issues of exchanging implementations of the "adaptive imager" in Israel and "adaptive sorter" in the US were extensively discussed. It was decided that due to the technical difficulties this exchange will be impossible, within the budget and time frame of the project. Instead, it was decided to implement the adaptive sorter by a simulated adaptive classification software package, which may be used as a vehicle for introducing adaptive sorting in commercial packinghouses. This software package was transferred to Texas through the electronic communication channels that were established during the first year of the project. The package was successfully run on local SUN workstations and the results obtained in Israel were duplicated.

Some carrot data sets were transferred to Israel for analysis. Unfortunately, the source of the carrots was not from significantly different populations and it was not possible to test the adaptive classification algorithms with these datasets. Consequently, adaptive sorting experiments were

initially conducted on artificially generated datasets and finally on real datasets from a newly developed date sorting machine.

Additionally, the cooperation between Israel and Texas was expressed by:

1. Exchange of information on the technologies upon which the adaptive imager might be based. State of the art technologies, eligible for the imager's implementation were investigated in Texas and shared with the researchers in Israel. Various aspects of the technologies were discussed and a joint decision was made regarding the technology selected.
2. Exchange of information on possibilities for implementing the adaptive classification system in software and hardware and possible options of using commercially available DSP boards were jointly investigated.

4. Conclusions and further research

Three different technologies were investigated for building a selectable wavelength imaging sensor: diffraction gratings, tunable filters and linear variable filters. Each technology was analyzed and evaluated as the basis for implementing the sensor. Acousto optic tunable filters were found to be most suitable for the selective wavelength imaging sensor. Consequently, a selectable wavelength imaging sensor was constructed and tested using the selected technology. The sensor was tested and algorithms for multispectral image acquisition were developed.

A high speed inspection system for fresh-market carrots was built and tested. It was shown that a combination of efficient parallel processing of a DSP and a PC based host CPU in conjunction with a hierarchical classification system, yielded an inspection system capable of handling 2 carrots per second with a classification accuracy of more than 90%.

The adaptive sorting technique was extensively investigated and conclusively demonstrated to reduce misclassification rates in comparison to conventional non-adaptive sorting. Each subprocess in the adaptive sorting algorithm was explored theoretically and empirically using artificially created datasets. The technique was also applied to a real produce sorting problem, e.g. an automatic machine for sorting dates by machine vision.

The adaptive sorting part of this BARD project was an integral part of an ongoing D.Sc. research thesis project of Matt Picus, under the direction of the cooperating investigator. This thesis will be completed approximately by the end of 1997.

The additional work in the near future will involve a more rigorous theoretical analysis for fine tuning the adaptive sorting process. Particularly, we shall further investigate the problem of dynamically updated optimal stack lengths. The problem of optimally partitioning the feature space into prototype populations will also be studied in more depth. Work on a method of adding unknown populations to the pool of known ones will also be explored further. Publications in professional journal on these subjects are expected by the beginning of 1998.

5. References

adaptive imaging

- Alchanatis V and Searcy SW (1995). "High Speed Inspection of Carrots with a Pipelined Image Processing System" ASAE International meeting, June 18-23, Chicago, IL. Paper #953170
- Alchanatis V and Searcy SW (1995). "A Selectable Wavelength Imaging Sensor for Multispectral Inspection of Agricultural Products". ASAE International meeting, June 18-23, Chicago, IL. Paper #953210
- Alchanatis V and Howarth MS 1995. *INSPECT 2.0 Users Manual*. Texas A&M University, Agricultural Engineering Dept.
- Davenel A, Guizard C, Labarre T and Sevila F, (1988). "Automatic detection of surface defects on fruits by using a vision system". *J. Agric. Engng. Res.* 41:1-9.

- Delwische MJ, Tang S, and Thompson JF, (1993). "A high speed sorting system for dried prunes". *Trans. ASAE* 36(1):195-200.
- Gunasekaren, Paulsen MR and Shove GC, (1985). "Optical methods for non destructive evaluation of biological materials". *J. Agric. Engng. Res.* 32:209-241
- Horowitz P and Hill W, (1983). *The Art of Electronics* Cambridge University Press: New York.
- Howarth MS and Searcy SW, (1992). "Inspection of fresh market carrots by machine vision". *Proceedings of the 1992 Conference - Food Processing Automation II* pp.106-112.
- Howarth MS, (1991). *Machine vision inspection of fresh market carrots*. Ph.D. thesis, Agricultural Engineering Dept., Texas A&M University, Texas.
- Howarth S and Searcy SW, (1990). "Reflectance characteristics of fresh-market carrots". *Trans. ASAE* 33:961-4.
- Kohonen T, Barna B and Chrisley R, (1988). "Statistical pattern recognition with neural networks: benchmark studies". *Proc. Of the second annual IEEE international conference on neural networks*. Volume 1, pp. 61-68.
- Kurtz I, Dwelle R and Katzka P, (1987). "Rapid scanning fluorescence spectroscopy using an acousto-optic tunable filter". *Rev. Sci. Instrum.* 58(11):1996-2003.
- Miller BK and Delwische MJ, (1991). "Peach defect detection with machine vision". *Trans. of ASAE* 34(6):2588-97.
- NeuralWorks (1993): *Neural computing, a technology handbook for Professional II/plus and NeuralWorks Explorer*. NeuralWare, Inc.
- Ni B, Paulsen MR, Liao K and Reid JF, (1993). "An automated corn kernel inspection system using machine vision". ASAE paper 93-3032.
- OCLI, *Stock filter catalog*, 1993. pp. 30-31.
- ORIEL Corp. Light sources, Monochromators and Spectrographs, Detectors and Detection Systems, Fiber Optics, 1994. Vol. II, Stratford, CI, USA, pp. 2.4-2.13.
- Peleg K and Ben-Hanan U (1993). "Adaptive classification by neural net based prototype populations". *International Journal of Pattern Recognition and Artificial Intelligence*, 7(4):917-933.
- Porteous RL and Muir AY, (1987). "Some measurements of the performance of an experimental quality grading machine for potatoes". *Potato Research* 30:23-24.
- Second Generation TMS320: User's guide, 1989. Texas Instruments Inc.,
- Suhre M Gottlieb, Taylor LH and Melamed NT, (1992). "Spatial resolution of imaging noncollinear acousto-optical tunable filters". *Optical Engineering* 31(10):2118-2121
- Tran CD and Bartelt M, (1992). "Performance characteristics of an acousto-optic tunable filter for optical spectrometry". *Rev. Sci. Instrum.* 63(5):2932-2939.
- Tran CD and Furlan RJ, (1992). "Acousto-optic tunable filter as a polychromator and its application in multidimensional fluorescence spectrometry". *Anal. Chem.* 64:2775-2782.
- Tran CD and Furlan RJ, (1993). "Spectrofluometer based on acousto-optic filters for rapid scanning and multicomponent analyses". *Anal. Chem.* 65:1675-1681.
- Worthington JT, Massie DR and Norris KH, (1976). "Light transmission technique for predicting ripening time for intact green tomatoes". In *Quality detection in Foods* (J.J. Gaffney, Ed.), American Society of Agricultural Engineers, pp. 46-49.

adaptive classification

- Armitage P. (1975), *Sequential Medical Trials*. Wiley NY.
- Conover W.J.(1980). *Practical Nonparametric Statistics 2ed* John Wiley & Sons, NY.

- R. Duda and P. Hart (1973), *Pattern Classification and Scene Analysis*, John Wiley & Sons, NY.
- Eisenberg B.(1991) "Multihypothesis Problems," Chapter 9 in *Handbook of Sequential Analysis*, (Ghosh B. Sen P. editors), Marcel Dekker New York.
- Foley D. Sammon J. (1973), "An Optimal Set of Discriminant Vectors," *IEEE Transactions on Computers*, **C-24:3**, 281-289.
- Ghosh B. Sen P. (editors),(1991) *Handbook of Sequential Analysis*, Marcel Dekker New York.
- Gutman P., Peleg K. and Ben_Hanan U. (1994), "Classification by Varying Features with an Erring Sensor" *Automatica*, **30**(12) pp. 1943-1948
- Mathworks Inc. (1994). *Matlab User Guide Ver.4* The Mathworks Natick Mass.
- Press W, Teukolsky S, Vetterling W, Flannery B., *Numerical Recipes in FORTRAN*, 2nd Edition, Cambridge Univ. Press, 1992.
- Shewhart W.(1931), *Economic Control of Manufactured Product*, Van Nostrand New York.
- Scott D.(1992), *Multivariate Density Estimation*, John Wiley and Sons.
- Siegmund D.(1985), *Sequential Analysis*, Springer-Verlag New York.
- Smallwood R.H (1996)., "A two-dimensional Kolmogorov-Smirnov test for Binned Data", *Phys. Med. Biol.* , **41**:125-135.
- Feltz C., Goldin G.(1991), "Generalization of the Kolmogorov-Smirnov Goodness-of-Fit Test, Using Group Invariance", DIMACS Technical Report 91-25, March 1991.
- Stephens M.A.(1965) *Journal of the Royal Statistical Society, ser.B.* **32**:115-122.
- Tukey J.W. (1977), *Exploratory Data Analysis*. Addison-Wesley, Reading MA.
- Wald A.(1945) Sequential Test of Statistical Hypotheses. *Ann. Math Statist.*, **16**:117-186.
- SAS Institute Inc. (1985) *SAS User's Guide: Statistics, Version 5 Edition*. SAS Institute Inc., Cary NC.
- Peleg K. (1985) *Product Handling Packaging and Distribution*.