

Chapter 7

Machine Vision System for Orchard Management



Duke M. Bulanon, Tyler Hestand, Connor Nogales, Brice Allen,
and Jason Colwell

Abbreviations

2D	Two-dimensional
3D	Three-dimensional
Cov	Covariance
GPS	Global Positioning System
IR	Infrared
K_I	Integral gain
K_P	Proportional gain
LIDAR	Light detection and ranging
NIR	Near infrared
PI	Proportional-plus-integral
RGB	Red, Green, and Blue
RMS	Root mean square
UAV	Unmanned arial vehicle
UGV	Unmanned ground vehicle
Var	Variance

D. M. Bulanon (✉) · T. Hestand · C. Nogales · B. Allen
Department of Physics and Engineering, Northwest Nazarene University, Nampa, ID, USA
e-mail: dbulanon@nnu.edu; theastand@nnu.edu; cnogales@mnu.edu; beallen@mnu.edu

J. Colwell
Department of Mathematics and Computer Science, Northwest Nazarene University, Nampa, ID,
USA
e-mail: jcolwell@mnu.edu

7.1 Introduction

Machine vision is a technology that provides a visual sensor to machine systems. There are several industries where machine vision can be applied, such as agriculture [1], automotive [2], and industrial [3], each with its own set of applications. The first industrial applications focused on quality inspection and manipulator control while agriculture used machine vision for tractor navigation, product inspection, and fruit harvesting. This chapter focuses on the development of machine vision for orchard management applications and is ordered as follows:

1. Definition of machine vision

The four main elements of a machine vision system are:

- **Scene Constraints**—the physical constraints of the environment in which the machine vision system operates. There are several factors to consider when evaluating the scene constraints which include lighting and the color of the work plane as well as other factors [4].
- **Image Acquisition**—the properties and characteristics of the camera being used. There include color cameras, stereo cameras, NIR cameras, IR cameras, and others. Each of these camera types has different characteristics, so the decision of which camera employed depends on the application.
- **Image Processing and Analysis**—the process of modifying the acquired image to extract the desired information. There are several sub-steps in the image processing and analysis element, which include preprocessing the image, segmenting the region to useful regions, extracting useful features and classifying those features.
- **Actuation**—the physical action the system will take in response to an identified object. In agricultural applications, examples include picking fruit off the tree, sorting already picked fruit by their grade and weed control [5, 6].

2. Machine vision applications in different areas of agriculture

As mentioned in the abstract, there are several different fields for which machine vision can be applied and different applications within each field. Specifically, within the field of agriculture, three primary applications exist. They are:

- **Plant Identification**—analyzes the color, size, and shape of the object within the image to classify the plant type.
- **Process Control**—common application for machine vision and in agriculture tends to focus on evaluating fruit. This evaluation uses the size, shape, and color of the fruit to determine quality for grading and sorting.
- **Machine Guidance and Control**—the most common thought of process for an application of machine vision. Though many forms exist, a common example is a ground vehicle, which could be either manned or unmanned. In an agricultural application, this vehicle would run through a field or an orchard employing

several different inputs and sensors such as GPS, ultrasonic sensors, and a vision system [5]. The vision system could determine what object is in front of the ground vehicle and helps determine what action the ground vehicle should take.

3. Orchard management machine vision system

This example of a machine vision system in an orchard application demonstrates different analysis techniques of images in order to extract useful information. There are several different steps involved when analyzing images for a machine vision system, and these are dependent on the task at hand. The goal, for the example discussed in this chapter, is a machine vision system that can predict a fruit yield of apple and peach trees when the tree is in full blossom. It was hypothesized that the crop yield could be estimated by determining the number of blossoms on the tree, so a machine vision system needed to be created which could count the number of blossoms on a tree. Because every blossom on a fruit tree was essentially the same color, the RGB data from an acquired image could be used to filter out of the scene everything but the blossoms. The remaining information, the blossoms, are then counted and processed to determine the overall yield of the particular tree. Again, depending on the goal of the project, a different application of machine vision may need to be applied, but the method reviewed in this chapter provides various applications which can be extended outside of agriculture.

4. Stereo imaging to identify tree structure and improve individual tree detection

One of the main problems with the machine vision system used to estimate the yield of apple and peach trees is the inclusion of blossoms from multiple trees as each blossom in the acquired image would be counted even if it was on a tree behind the tree of interest. This is because the vision system developed used only the image RGB data to remove the scene. An additional filter, one which focuses on distance, needs to be added. Stereo imaging can be used to address this problem because now the acquired image has both the RGB parameter and the distance parameter. With this distance parameter, trees which are farther away from the tree of interest can be eliminated, so blossoms only from the tree of interest can be counted, thereby increasing the accuracy of the yield estimation.

5. Machine vision system that navigates a robot within orchard rows

Another application for a machine vision system is implementation on a ground vehicle which can navigate through the rows of an orchard. When applying machine vision for this configuration, the scene constraints become extremely important. To successfully navigate the rows of an orchard, the system must account for the symmetry of the rows, the size of the trees, and row separation. The ground vehicle is sized to fit and operate within these constraints. For example, a small ground vehicle can use the sky visible between rows of trees for navigation if the trees are relatively large and the rows are adequately spaced.

7.2 The Machine Vision System

Awcock and Thomas [7] defined a general machine vision system that is shown in Fig. 7.1. The defined system consists of four elements that could be found in a typical machine vision system in any field of application. The four elements are scene constraints, image acquisition, image processing and analysis, and actuation.

7.2.1 Scene Constraints

The scene constraint refers to the environment wherein the machine vision equipment is to be placed, and it is where the information is to be taken. The main aim of this system is to extract from the environment the desired information by the proper controlling of factors that affect the acquisition of data like lighting and the proper installation of the machine vision equipment. Some of the environment may be controlled such as in the sorting lines for product inspection [8] while other environmental parameters such as lighting conditions, fruit location on a tree, and the unstructured nature of the branches are difficult to control in an apple orchard [9].

7.2.2 Image Acquisition

Image acquisition is the element that converts light falling onto the photosensors of a camera into a digitized data, typically a 512×512 pixel image, which is then able to

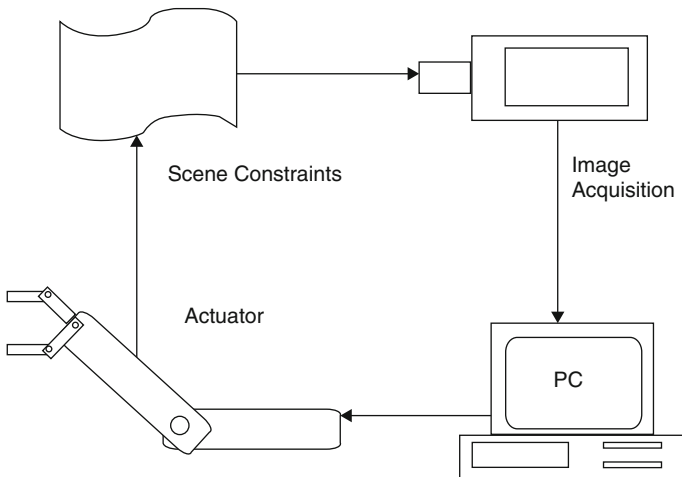


Fig. 7.1 A generic machine vision system

be processed. The camera may be a black and white video camera that is dependent on light intensity, a colored video camera that is dependent on the visible spectrum, or an infrared camera with the selection being based on the relevant information needed. With the advancement of sensor technologies, cameras sensitive beyond the visible spectrum are also available. Hyperspectral and multispectral imaging techniques have emerged as important tools for the safety and quality inspection of food and agricultural products [1].

7.2.3 Image Processing

Image processing deals with the acquired digital image as input and outputs an image that has been enhanced so that the desired information can be extracted. Several steps are involved in the extraction of the data each of which is discussed in the following sections.

7.2.3.1 Preprocessing

Images are preprocessed to modify and prepare the pixel values of the digitized image to produce an output that can be more easily analyzed in subsequent operations. This may consist of contrast enhancement, filtering to remove the noise of the hardware, and correction for camera distortion [10].

7.2.3.2 Segmentation

Segmentation is the process wherein the digitized image is broken down into meaningful regions. It is considered the first step of image analysis because the decision-making process of identifying the foreground and background has already been conducted. The simplest segmentation process is the identification of the foreground and background regions which is often easily achieved by thresholding. A very popular thresholding technique is the Ohtsu method [11].

7.2.3.3 Feature Extraction

After the image is divided into regions, the feature extraction process identifies objects in the region using descriptors. Basic descriptors are typically scalars that include area, centroid, perimeter, major diameter, compactness, and thinness [12]. These descriptors are often used simultaneously providing a good description of the object of interest.

7.2.3.4 Classification

Classification is the task of putting the objects in the image into some predefined categories. This process may be done by template matching or by a statistical method. Template matching is the comparison of the unknown objects to a set of known templates so as to identify the object.

Artificial intelligence, or machine learning, is being used more frequently in image classification in agricultural applications. In many of these applications, supervised machine learning is being used, where the user would enter and label several “training samples,” then the neural network would recognize connections between them. The neural network is then tested with “test samples” which it has not seen before, and the network is evaluated. Agricultural applications so far have primarily been concerned with plant identification, where plants were segmented under different lighting conditions [13], and it has been applied in weed management [14].

7.2.4 Actuation

Once the machine has identified the object, the decision on what the machine will do is known as the actuation process. This is the interaction of the machine with the environment or the original scene either directly or indirectly. This closes the machine vision system shown in Fig. 7.1. Usually, the machine vision is linked to a robotic system which is the basic component of automated operations [15].

7.3 Agricultural Machine Vision Applications

Machine vision systems typically use complex electronic sensors. The rapid development of computer technology and the photosensor has widened the field for machine vision applications. Currently, industries occupy most of the field of usage focusing mainly on product inspection, but other areas such as military science, astronomy, medicine, and the field of agriculture are now investigating the uses for machine vision [16]. For agriculture, researchers have been studying the potentials of machine vision in enhancing production which can be classified into three categories:

1. Plant identification
2. Process control
3. Machine guidance and control

The recent applications developed in these categories are described below.

7.3.1 Plant Identification

Plant identification refers to the process of classifying a certain plant by accurately identifying its component geometrical shape, size, and color. Figure 7.2 shows a schematic diagram of a machine vision system that is used for plant identification. Important parameters analyzed by the system are size, color, and shape, and surface temperatures. Making measurements of these parameters by noncontact visual means is an advantage of machine vision as identification and classification can be done without the risk of damage to the plant.

Several research projects on plant identification using machine vision have been conducted. Guyer et al. [17] developed a machine vision system that identifies plant species such as corn, soybeans, tomatoes, and some weed species at early growth stages using spatial parameters. The image processing stage evaluated the differences in the reflection of radiation from leaves and soil surface and the differences in the number of leaves and the shapes of the different weed species. This plant identification visual system could thus be used for selective herbicide application. A robotic vision-based system was developed to detect crop and weed locations, kill weeds, and thin crop plants [18]. This vision system identified different plant leaves using shape features that included area, major axis, minor axis, area to length ratio, compactness, elongation, length to perimeter ratio, and perimeter to broadness ratio. The system could then differentiate between tomato cotyledons and weeds when attached to a ground vehicle such as a tractor, and the prototype robotic weed control system could identify and treat weeds simultaneously.

With the advancement of aerial systems, a machine vision system can also be attached to an unmanned aerial vehicle (UAV) for the purpose of plant identification. Crop monitoring and assessment platform were developed to identify apple trees

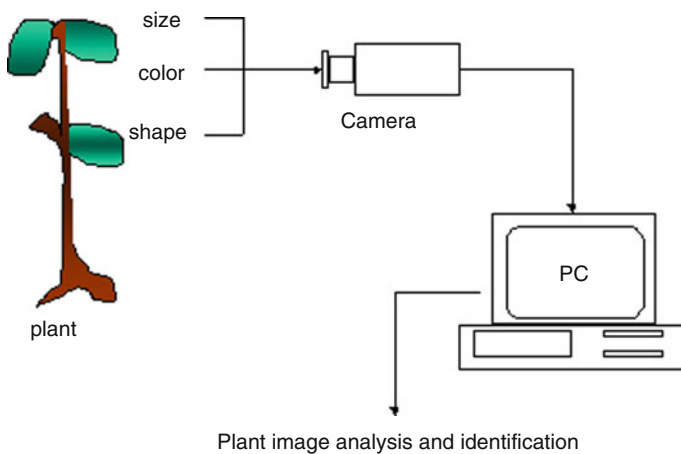


Fig. 7.2 Machine vision system for plant identification

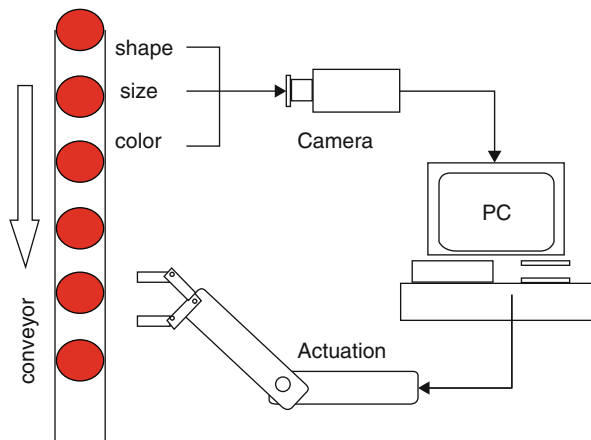
and monitor irrigation types [19]. This UAV machine vision system was composed of a multispectral camera (near-infrared, green, blue) and an image processing and analysis unit. The image processing calculated the enhanced normalized difference vegetation index to identify the tree crops and estimate the irrigation level and was able to differentiate between the full drip and 50% sprinkler irrigated trees. When identifying plants on a UAV system, the images will likely be acquired with a color camera, but using a color camera for the navigation of the UAV has potential problems. Concerning the navigation, using a laser triangulation system has several advantages compared to the color camera navigation. The main advantage is distance measurement, which can be measured to a high degree of accuracy, where a color camera system would estimate the distance [20]. Of course, there are errors when using a laser triangulation system for UAV navigation, such as the static and dynamic friction within the DC motors used in the system; but these errors can be estimated and accounted for, thus increasing the overall accuracy of the system [21].

7.3.2 Process Control

Industrial applications rely on visual systems for process control when the control is dependent on a visual parameter, for example, the inspection of circuit boards in a production line [22]. The system is able to make an intelligent action spotting and removing abnormal products. Generally, in visual sensing, the parameters being assessed are color, shape, and size.

In agriculture, evaluation of the color information indicates qualities such as maturity, sweetness, and wholesomeness. As shown in Fig. 7.3, a machine vision system may be used for the inspection of fruits by allowing the fruits to pass in front of a camera so that its quality may be evaluated.

Fig. 7.3 Machine vision for process control



Miller and Delwiche [23] studied a color vision system that inspects and grades fresh market peaches. Digital color images of the peaches taken as the fruit moved on a conveyor belt analyzed the peach for color, size, and surface characteristics. Compared to visual inspection by human senses, this system gave a high output rate, high reliability, and high uniformity and was additionally capable to make critical measurements.

There are machine vision systems that can detect wavelengths outside the visible electromagnetic spectrum. Bulanon et al. [24] developed a machine vision system to detect citrus black spot using hyperspectral imaging. Hyperspectral imaging allows the acquisition of spatial information across a sequence of individual bands covering a broad wavelength range, resulting in a three-dimensional image data with a very high spectral resolution. In the study, five different surface conditions including citrus black spot were evaluated. Linear discriminant analysis and an artificial neural network were then developed using wavelengths of 493 nm, 629 nm, 713 nm, and 781 nm. Both pattern recognition approaches had an overall detection accuracy of 92%. Rehkugler and Throop [25] developed a machine vision system that detects the defects in an apple.

In addition to the spectral properties of agricultural products, size, shape, form, freshness condition, and absence of visual defects are normally evaluated. Costa et al. [26] developed an automated shape processing system which could be used for both scientific and industrial purposes. This tool would be very useful for grading and sorting agricultural products especially if they were coupled with pattern recognition techniques [27]. It offers many advantages over conventional and mechanical sorting devices. Furthermore, evaluating the shape of agricultural products is a key parameter for allocating packaging and shipping resources [28].

7.3.3 Machine Guidance and Control

One of the important features of a robotic harvesting system is recognizing and locating a fruit. The commonly used camera gives a two-dimensional picture. Since three coordinates are required to fully locate the object, the distance dimension is lacking. This third dimension is typically acquired through the use of another sensor such as a range finding sensor, acoustics, radio frequency, or a stereographic vision system.

Researchers are trying to eliminate the need for an additional sensor by developing the range of information utilizing the object's geometric shape property, reflectance intensity, chrominance, and emissivity. The goal is to take a digital image of the object and then use image processing to identify and locate the position of the objects. Parrish and Goksel [29] conducted the first experimental system for an apple harvesting system. A black and white video camera was used to detect the apples. The image coordinates of the apple and its centroid were determined by image processing, and then the trajectory planning and the actuation routines directed the robotic arm to the apple. Figure 7.4 shows the generic machine vision system applied to fruit

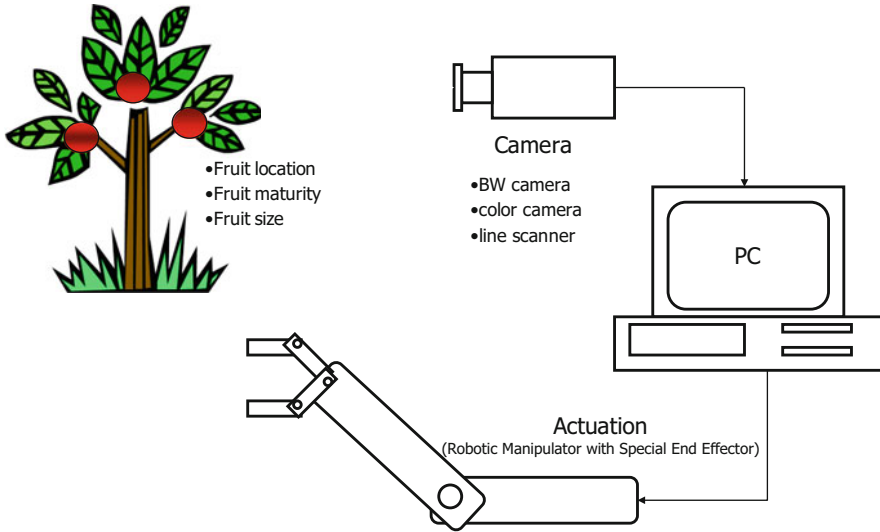


Fig. 7.4 Machine vision for fruit harvesting

harvesting. Similar to the development made by Parrish and Goksel, the features that are extracted from the image included color, shape, centroid location, and depth information. These features were then used to guide a robotic arm toward the fruit and pick it. Slaughter and Harrel [30] later improved the black and white video camera by using a colored video camera. This time, the detection of apples was dependent not only on gray-level intensity but also on color. The color factor is an important parameter in differentiating the object from its background. Another example of machine vision-based fruit harvesting is the apple robotic harvesting system developed by Bulanon and Kataoka [9]. The segmentation method was based on the chromaticity coefficients red and green combined with a decision-theoretic approach method to threshold the apple fruits from the background under variable lighting conditions. The vision system was used to guide a customized end effector that picked the fruit in a manner similar to the way a human would pick the apple.

One of the problems encountered in a robotic vision system is the similarity of the spectral reflectance between the object and its background specifically the leaves of the tree. Recent studies have focused on using the thermal characteristics of the fruit to separate it from the foliage. Bulanon et al. [31] studied the thermal characteristics of the citrus tree. A 24-hour temperature profile between the fruits and the leaves was obtained, and it was found that the fruits had a higher surface temperature than the leaves during the nighttime. Thus a unique image processing approach which combined color and thermal images using fuzzy logic was developed.

Another robotic system that could be guided by machine vision is an agricultural ground vehicle. The vehicle could be manned or unmanned. If the vehicle is manned, the machine vision system is used to assist the driver in steering the system while

an unmanned vehicle would be fully autonomous. The last section of this chapter discusses the development of a machine vision system for steering an unmanned ground vehicle in a commercial peach orchard.

7.4 Machine Vision Development for Fruit Yield Estimation: An Example of Plant Identification Application

Section 7.3 discussed the different applications of machine vision, those being plant identification, process control, and machine guidance and control. This section will discuss a plant identification application of machine vision specific to orchard management. This development was created under a research project of the Robotics Vision Lab at Northwest Nazarene University, where the goal of the project was early fruit yield estimation. Yield estimates are important for growers to help in the production planning and marketing of the fruits. There are several ways of estimating fruit yield [32, 33], and machine vision is one of the popular tools available [34–38]. Most of these vision-based yield estimators [39] count the fruits when they are almost ready to harvest; however, an early yield estimate [40] is more important to the growers. The hypothesis of the project was that by counting the number of blossoms of a fruit tree, an early yield estimate could be derived. The fruits of interest in this project were apples and peaches: specifically, Pink Lady apples grown in a high-density orchard and Snow Giant peaches grown in a standard orchard. Both orchards were located in Caldwell, Idaho, and were planted in a north-south direction. Thirty trees were selected randomly from a block in each orchard and photographed throughout the blossom period during the 2018 growing season. A 12-megapixel 24-bit digital color camera was used to photograph each tree on the east and west sides. Later in the season when the fruits were mature, a ground truth yield was obtained by manually counting the fruits on the selected trees.

The images were processed using MATLAB and its Digital Image Processing Toolbox [41]. Figure 7.5 displays a sample image of a blossoming apple tree in a high-density orchard. The height of each apple tree is approximately 8 ft, and there are approximately 4 ft between each tree. In this orchard, images were acquired approximately 10 ft from the tree. Figure 7.6 displays a blossoming peach tree in a standard orchard. The height of each peach tree is approximately 15 ft, and there is approximately 10 ft between each tree. In this orchard, images were acquired approximately 13 ft from the tree.

7.4.1 Image Processing for Blossom Isolation

With image acquisition completed, the next step is to isolate and count the blossoms for each apple and peach tree.



Fig. 7.5 Sample image of a blossoming apple tree in a high-density orchard



Fig. 7.6 Sample image of a blossoming peach tree in a standard orchard

7.4.1.1 Methods of Data Transformations

Before the blossoms could be isolated, a set of sample data needed to be collected to determine the color properties of each category in the image, so that a color filter can be created from this data, which will isolate the blossoms. This sample data was collected manually, where 600 different pixels for each category of the image were

manually selected, and the RGB values of those pixels were recorded. Within the 600 pixels selected for each category, 300 pixels were selected from images taken from the east side of the tree, and 300 pixels were selected from the images taken from the west side of the tree. The five main categories of classification in each image for both the apple and peach images are the sky, blossoms, leaves/grass, branches, and dirt. A 3D scatterplot displaying the recorded RGB values for the apple images is displayed in Fig. 7.7. As seen in Fig. 7.7, the RGB values of the sky is not included. This is because by manually analyzing the images, it has been noticed that the sky is a relatively large category in size and that the pixels are all connected. Because the pixels of the sky are all connected, an area feature extraction method, which is explained later in this section, can easily be implemented, which will remove the sky from the image.

The goal is to isolate the red circle data points, which represent the blossom's RGB values, so that when analyzing the entire image, the blossoms can be isolated. There are several image analysis functions in MATLAB that could be used to isolate the blossoms, but because MATLAB is not an open source software, it is preferred to use a method of isolation not using these functions.

One method of blossom isolation investigated was to apply a transformation matrix to each sample data point, written mathematically as

$$Ax = b \quad (7.1)$$

where b is the new value of the sample data point, A is the transformation matrix, and x is the red, green, and blue values of the sample data point. In this form, these matrices have the form:

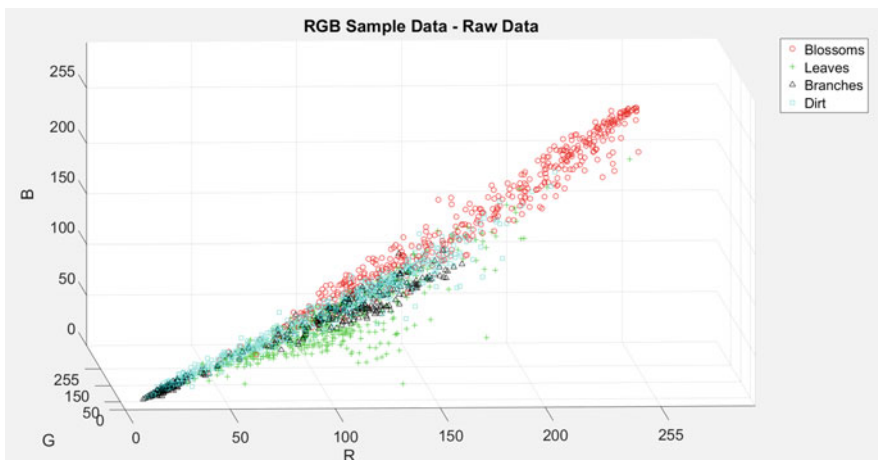


Fig. 7.7 Sample RGB values of objects in the apple orchard

$$A = \begin{bmatrix} a_{1,1} & a_{1,1} & a_{1,3} \\ \vdots & \vdots & \vdots \\ a_{n,1} & a_{n,1} & a_{n,3} \end{bmatrix} \quad (7.2)$$

$$\mathbf{x} = \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (7.3)$$

$$\mathbf{b} = \begin{bmatrix} Ra_{1,1} + Ga_{1,2} + Ba_{1,3} \\ Ra_{2,1} + Ga_{2,2} + Ba_{2,3} \\ \vdots \\ Ra_{n,1} + Ga_{n,2} + Ba_{n,3} \end{bmatrix} \quad (7.4)$$

where the element $a_{n,3}$ is an element in A occupying the n th row and 3rd column. When transformation matrix A is applied to the sample data matrix \mathbf{x} , the image is \mathbb{R}^n .

An example of a transformation $T : \mathbb{R}^3 \rightarrow \mathbb{R}^1$ defined by $T(\mathbf{x}) = \mathbf{Ax}$ would be a summative transformation that adds the red, green, and blue values of each pixel. The matrix A would take the form seen in Eq. (7.5).

$$A = [1, 1, 1] \quad (7.5)$$

Applying this matrix to the scatterplot displayed in Fig. 7.7 results in the data points being transformed to a single axis. This is difficult to display because the data points are clustered, so the results transformation are displayed with a histogram in Fig. 7.8.

An example of a transformation $T : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ defined by $T(\mathbf{x}) = \mathbf{Ax}$ is to rotate the 3D scatterplot displayed in Fig. 7.7 such that only two of the axes can be seen. If it was desired to display the red and blue axes, matrix A would take the form

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (7.6)$$

Applying this matrix to the scatterplot displayed in Fig. 7.7, the result is a 2D scatterplot displayed in Fig. 7.9.

A transformation $T : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ defined by $T(\mathbf{x}) = \mathbf{Ax}$ is to move the data points in the 3D scatterplot to a different location on the same 3D scatterplot. An example of this is to take a ratio transformation, that is, to take the red, green, and blue components of each pixel and divide it by the sum of its respective red, green, and blue components. The matrix A would take the form

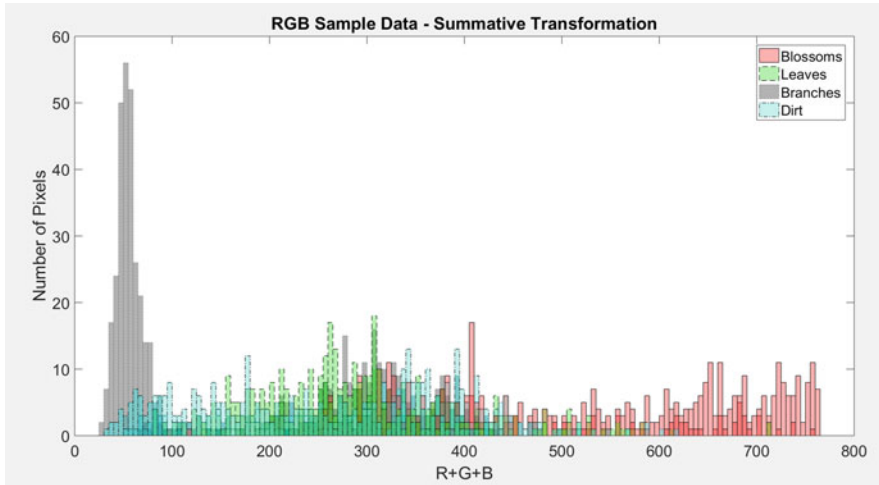


Fig. 7.8 Histogram of the summative transformation of the sample RGB values

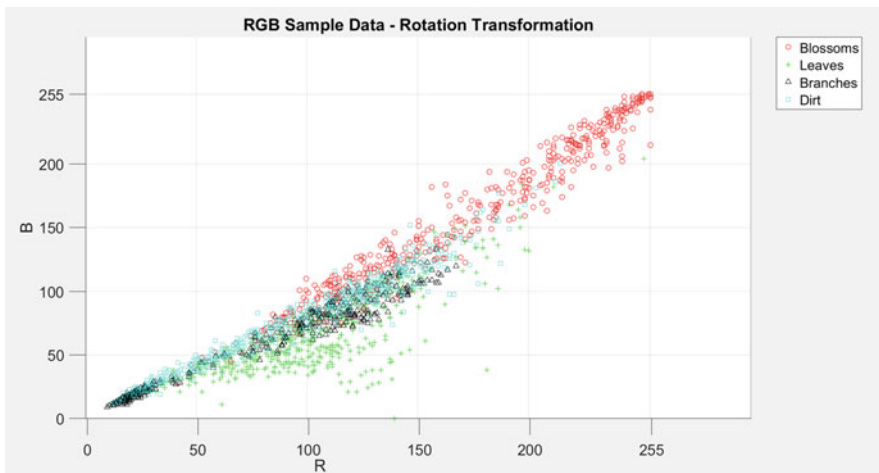


Fig. 7.9 Rotation transformation of the sample RGB values

$$A = \begin{bmatrix} (R + G + B)^{-1} & 0 & 0 \\ 0 & (R + G + B)^{-1} & 0 \\ 0 & 0 & (R + G + B)^{-1} \end{bmatrix} \tag{7.7}$$

This transforms each sample data point onto the plane

$$x + y + z = 1 \tag{7.8}$$

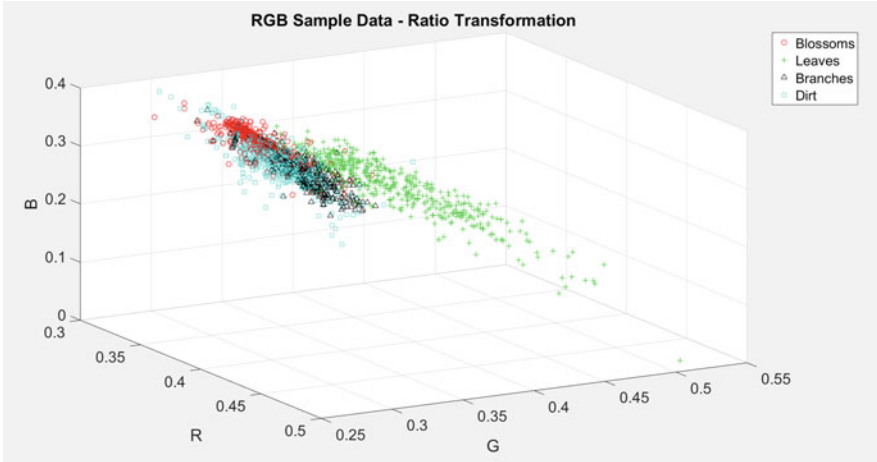


Fig. 7.10 Ratio transformation of the sample RGB values

Applying this matrix to the scatterplot displayed in Fig. 7.7, the result is the 3D scatterplot displayed in Fig. 7.10.

There are infinitely many transformations that can be applied to the set of sample data, such as transformation matrix A yielding a new data point \mathbf{b}

$$A = \begin{bmatrix} 2 & 3 & 7 \\ 5 & 8 & 1 \\ 4 & 6 & 9 \end{bmatrix} \quad (7.9)$$

$$\mathbf{b} = \begin{bmatrix} 2R + 3G + 7B \\ 5R + 8G + 1B \\ 4R + 6G + 9B \end{bmatrix} \quad (7.10)$$

As mentioned previously, the dimension of \mathbf{b} can extend past three. If A is a 4×3 matrix, then \mathbf{b} is in \mathbb{R}^4 . Though these can often be difficult to describe graphically, so examples of this and higher dimensions of \mathbf{b} will not be presented in this chapter.

7.4.1.2 Testing Blossom Isolation

Recall that the goal of applying a transformation matrix is to isolate the red circle data points in the sample data. Looking back at Fig. 7.9, two lines can be drawn which separates the blossom sample data points from the other categories, as seen in Fig. 7.11.

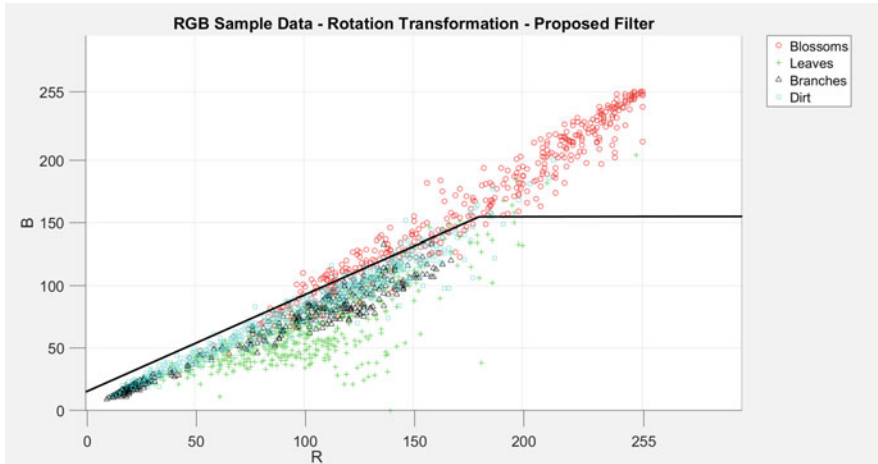


Fig. 7.11 Blossom isolation in the rotation transformation

The equation of these lines is the color filter that will be used to isolate the blossoms from the other objects when filtering an entire image. By using the equations, the points above or below the line can be set to zero, thus isolating a section of the data.

For example, the equations of the lines in Fig. 7.11 are

$$7 \times \text{Red} - 9 \times \text{Blue} - 135 = 0 \tag{7.11}$$

and

$$\text{Blue} = 155 \tag{7.12}$$

Thus, the red circle data points can be isolated by applying the pseudocode:

```

if ((7 × Red - 9 × Blue - 135 > 0) && (Blue < 155)) {
    Red = 0
    Blue = 0
}
    
```

Applying this code to the data set on Fig. 7.11 yields the plot displayed in Fig. 7.12.

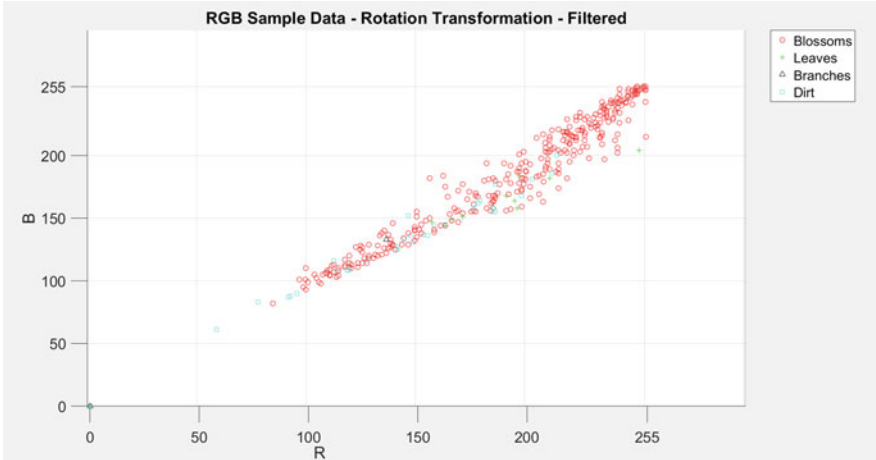


Fig. 7.12 Results of the blossom isolation color filter

7.4.1.3 Tree Isolation

As noticed by the sample image of the apple tree displayed in Fig. 7.5, there are three trees present in the foreground of the image with multiple others in the background. This is common for this style of high-density orchard as the trees are only separated by approximately 4 ft. Because the goal is to count the blossoms on the center tree, a method of isolating the center tree must be derived. Figure 7.8 shows a large clustering of data in the lower region of the histogram, RGB values less than 100 after applying the summative transformation, which is classified as either branches or dirt. This implies the branches can be used as a method of tree isolation, specifically the trunks can be used because they are the most isolated from each other.

Using a copy of the image, a tree isolation algorithm can be created. The results of this algorithm will be applied to the original image before the blossom isolation algorithm is applied. Because the trunks are the means of tree isolation, the first step of the trunk isolation algorithm is to crop out the top two thirds of the copied image. As explained in Sect. 7.2, this first step is the preprocessing in the tree trunk image processing algorithm. The next step of the algorithm is to isolate the trunks, which can be done by applying the following pseudocode to each pixel of the image:

```

if (Red + Green + Blue < 100) {
  Red = 0
  Green = 0
  Blue = 0
}

```

This results in the image displaying the trunks of each tree, but there is some noise, as some dirt samples remain. To remove these samples, a size filter can be applied because the number of dirt pixels passing through the transformation filter is much less than the number of branch pixels passing through the transformation filter. With this process what mostly remains are the three foreground tree trunks.

The next step then is to isolate the central tree trunk. For this task, a MATLAB data structure called “regionprops” was used. This function was used even though it was earlier stated that the use of MATLAB specific features was undesirable. This is because this function is also available through open source methods, such as the OpenCV library [42] or the ImageJ package, Fiji [43].

MATLAB’s regionprops measured the properties of an image’s regions—area, centroid, major and minor axis lengths—and then applied a bounding box to the region. The centroid feature of regionprops can be used to determine the location of each trunk, thereby giving the center position of each tree. Due to the nature of high-density orchards, where the trees tend to be vertical with little overlap of branches, the center tree can now be isolated. Using the position of the trees, the left and right trees can be cropped out by drawing a vertical line at the midpoint of the center tree and trees to the left and right of center.

This method works with the apple trees because they are in a high-density orchard, but peach trees are planted farther apart. Three trunks are not always seen in each image as previously shown in Fig. 7.6. Vertical line trunk isolation is not a viable option as there is no “center” tree. Instead, the natural geometry of the peach tree is used as a method of isolation.

Peach trees have four main branches that stem from the trunk, which makes a shape similar to an upside-down pyramid in the empty space within the four main branches. So, when an image of the tree is taken, the four main branches have a “V” shape. Thus, the tree of interest can be isolated by drawing a line from the top corners of the image to the bottom center, cropping out the bottom two corners. Figure 7.13 displays the result of the tree isolation algorithm applied to the original apple and peach tree images.

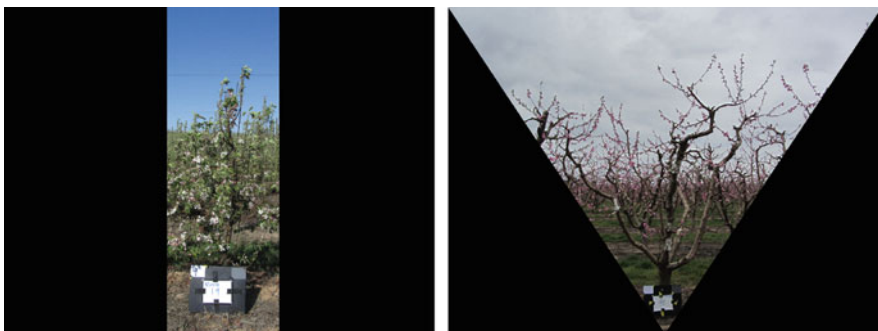


Fig. 7.13 Tree isolation process results for an apple and peach trees

7.4.1.4 Blossom Isolation and Counting for Apple Trees

Now that the center tree is isolated, the blossom filter described in Sect. 7.4.1.2 can be applied to each pixel in the apple image. Referring to Fig. 7.11, the line that isolates the blossoms from the other categories in the image is very close to including data points of the other categories. Because there is no significant gap in the separation of the blossoms and the other categories, a significant amount of noise in the resulting image after applying the color filter can be anticipated, which is exactly what is seen in Fig. 7.14, after the blossom isolation color filter is applied to the image.

As seen in Fig. 7.14, several of the pixels from the leaves passed through the color filter as has the entire sky. Both issues can be resolved by applying a size filter focusing on removing small and large groups of pixels. This size filter uses the `regionprops` data structure mentioned earlier in Sect. 7.4.1.3, where if the area is outside of the range of a specified pixel count, then the pixels are set to zero. It should be noted that the specific range which will allow an area to pass through the size filter varies depending on the size of the image. There are more pixels in a 12-megapixel camera (which was used in acquiring these images) than an 8-megapixel camera, so the allowable area of an image from a 12-megapixel camera should be higher than the allowable area of an image from an 8-megapixel camera. Because of this, care needs to be taken to match the filter parameters to the number of pixels in an image. After applying the size filter, the resulting binary image is displayed in Fig. 7.15.

The remaining areas are the identified blossoms on the tree. The `regionprops` data structure will now be used to label each area and to obtain a count for the total blossoms. In addition, a bounding box around each area can be applied, and the boxes can be overlaid over the original image to visually check the accuracy of the

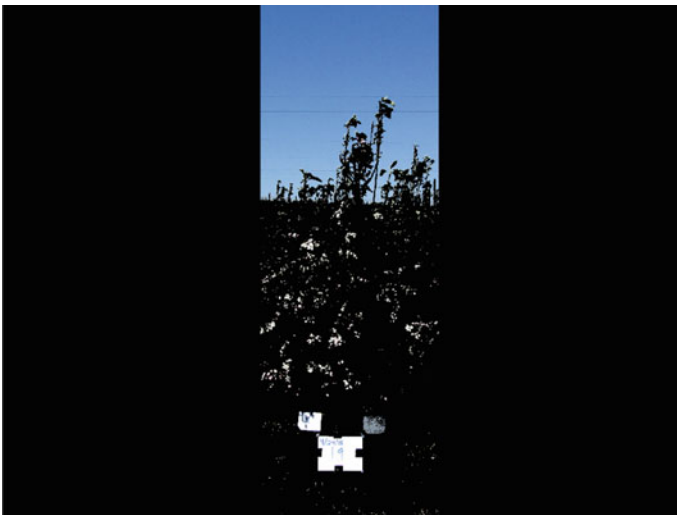


Fig. 7.14 Color filter applied to an image of an apple tree

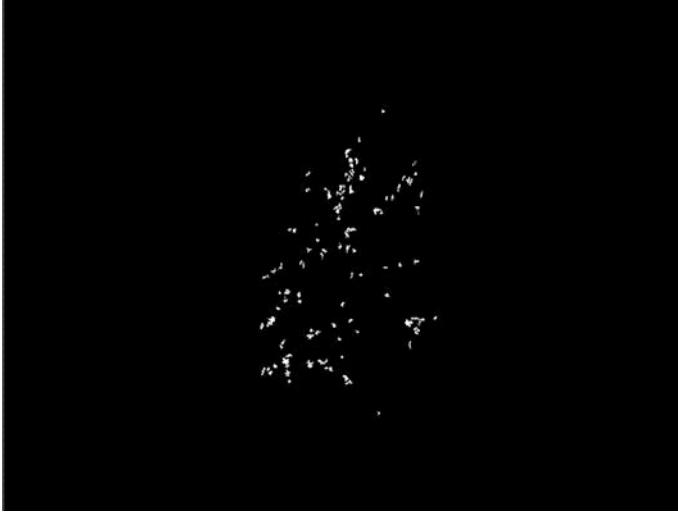


Fig. 7.15 Size filter applied to the apple tree image



Fig. 7.16 Bounding boxes of the identified blossoms overlaid on the original image

program. This image is displayed in Fig. 7.16, where it can be seen that there are very few false positives and false negatives in the image.

7.4.1.5 Blossom Isolation and Counting for Peach Trees

The process for identifying the blossoms on a peach tree is virtually the same as the process for the apple tree. The largest difference is that different transformation matrices may be applied and there will be a different equation applied to each pixel to apply the color filter. In the case of blossom identification, the transformation matrix which was applied is

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (7.13)$$

which rotated the 3D scatterplot to display the red and green color values. Figure 7.17 through Fig. 7.20 displays the overall process of blossom isolation for a peach tree. The rotation transformation of the sample RGB data and the line displaying the color filter are displayed in Fig. 7.17, the tree isolation algorithm and the color filter applied to the sample image seen in Fig. 7.6 are displayed in Fig. 7.18, the result of the size filter to remove the noise is displayed in Fig. 7.19, and the bounding boxes overlaid on the original image is displayed in Fig. 7.20.

After analyzing Fig. 7.20, there appears to be a significant number of false negatives. This observation may lead to the conclusion that the algorithm is not very successful in identifying peach blossoms; however, the false negatives seen in Fig. 7.20 were intentionally produced. This particular type of peach undergoes an intensive thinning process. Thus, to produce a better final yield estimate, fewer regions were desired. If an orchard were to not be significantly thinned, then a different size filter should be applied.



Fig. 7.17 Blossom isolation process for a peach tree

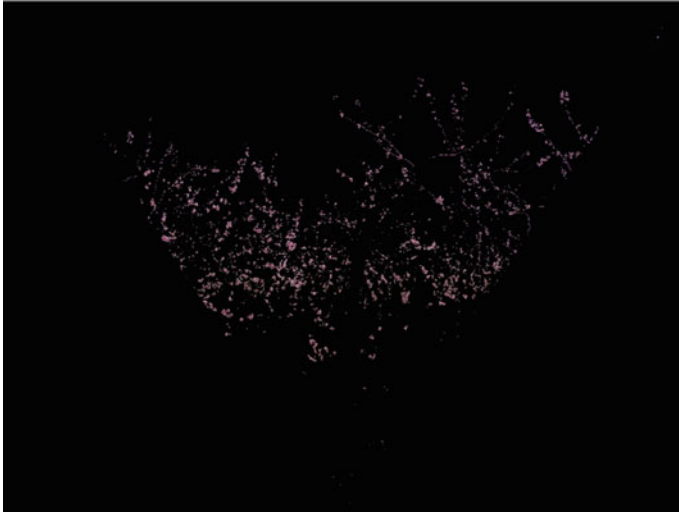


Fig. 7.18 Color filter applied to the peach tree

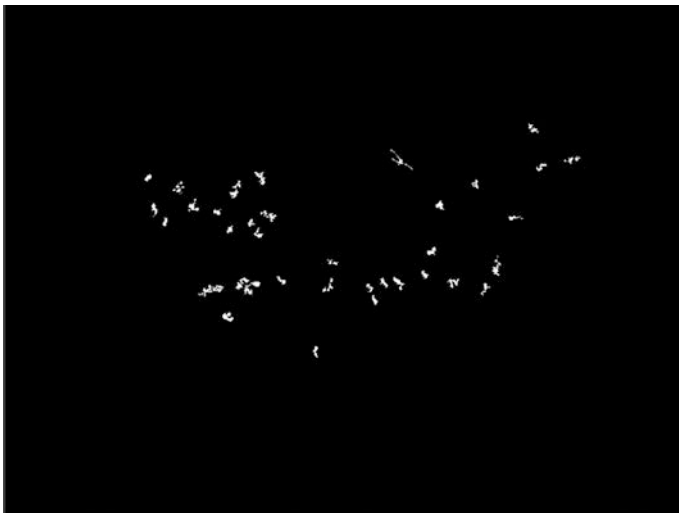


Fig. 7.19 Size filter applied to the peach tree

7.4.2 Results of Yield Estimation

So far, an algorithm has been developed which isolates and counts the number of blossoms on an apple and peach trees. There still remains steps to produce the final result of fruit yield estimate.



Fig. 7.20 Bounding boxes of the identified blossoms overlaid on the original image of the peach tree

7.4.2.1 Transition from Blossom Count to Yield Estimation

Once the blossom isolation process for each image has been applied, a blossom count for each side of the tree has been obtained. Recall from the beginning of Sect. 7.4, it was explained that two images of the tree of interest were acquired, one image from the east side and one from the west side. This is significant because the blossom count from each side of the tree cannot simply be added together to obtain a total fruit yield estimation, because there is a significant risk of double-counting blossoms.

Consider this question, “What if on the east side of the tree, the blossom count is consistent between each tree, but there is a large variance in the blossom count between each tree on the west side?” Intuition would say the blossom count from the east side should play a larger role in the yield estimation, because of the more consistent blossom count. Consistency in the blossom count is important because hypothetically there should only be a small difference in the total blossom count from tree to tree as each tree in a section of an orchard is of the same age and same size.

This is a case where intuition is correct, because the correct method of determining a yield from a set of two blossom counts is to calculate a weight that will be applied to blossom counts from the east side, and a different weight from blossom counts taken on the west side. The derivation for the two weights is described in Sect. 7.4.2.2, and it will be seen that the two weights depend on the variances and covariances [44] of the blossom counts from the east side, blossom count from the west side, and ground truth number of fruits.

7.4.2.2 Derivation of Weight Values

In the following derivation, random variables are denoted by capital letters, actual values of those variables by lowercase letters, and vectors by boldface type. The correlation between the blossom count from the images and the actual fruit count was determined. There are 30 selected trees, which are numbered #1 through #30. For tree # i , there are

$$X_E = x_{E,i} \quad (7.14)$$

blossoms visible from the East and

$$X_W = x_{W,i} \quad (7.15)$$

blossoms visible from the West. The eventual fruit yield of tree # i is

$$Y = y_i. \quad (7.16)$$

The data is represented by vectors in \mathbb{R}^n

$$\mathbf{x}_E = (x_{E,1}, x_{E,2}, \dots, x_{E,n}), \quad (7.17)$$

$$\mathbf{x}_W = (x_{W,1}, x_{W,2}, \dots, x_{W,n}), \quad (7.18)$$

$$\mathbf{y} = (y_1, y_2, \dots, y_n). \quad (7.19)$$

Choose weights α_E and α_W , with $\alpha_E, \alpha_W \geq 0$, and with $\alpha_E + \alpha_W = 1$. Then construct an equation of the form

$$Y' = m(\alpha_E X_E + \alpha_W X_W) + c \quad (7.20)$$

which will be the least-squares regression line of

$$X = \alpha_E X_E + \alpha_W X_W. \quad (7.21)$$

The RMS error of $Y \approx Y'$ will be

$$s_Y \sqrt{1 - r_{\mathbf{x}, \mathbf{y}'}^2} \quad (7.22)$$

where $r_{\mathbf{x}, \mathbf{y}}$ is the correlation coefficient of the data $\mathbf{x} = \alpha_E \mathbf{x}_E + \alpha_W \mathbf{x}_W$ with \mathbf{y} . Accordingly, the RMS error of the linear model will be minimized if α_E and α_W are chosen to maximize the correlation coefficient $r_{\mathbf{x}, \mathbf{y}}$.

Writing

$$x_i = \alpha_E x_{E,i} + \alpha_W x_{W,i}, \quad (7.23)$$

$$\mathbf{x} = (x_1, \dots, x_n), \quad (7.24)$$

and

$$\mathbf{x} = \alpha_E \mathbf{x}_E + \alpha_W \mathbf{x}_W, \quad (7.25)$$

define the means

$$\bar{x}_E = \frac{\sum_{i=1}^n x_{E,i}}{n} \quad (7.26)$$

$$\bar{x}_W = \frac{\sum_{i=1}^n x_{W,i}}{n} \quad (7.27)$$

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n} \quad (7.28)$$

$$\bar{y} = \frac{\sum_{i=1}^n y_i}{n} \quad (7.29)$$

the mean vectors

$$\bar{\mathbf{x}}_E = (\bar{x}_E, \dots, \bar{x}_E) \quad (7.30)$$

$$\bar{\mathbf{x}}_W = (\bar{x}_W, \dots, \bar{x}_W) \quad (7.31)$$

$$\bar{\mathbf{x}} = (\bar{x}, \dots, \bar{x}) \quad (7.32)$$

$$\bar{\mathbf{y}} = (\bar{y}, \dots, \bar{y}), \quad (7.33)$$

and the deviation vectors

$$\tilde{\mathbf{x}}_E = \mathbf{x}_E - \bar{\mathbf{x}}_E \quad (7.34)$$

$$\tilde{\mathbf{x}}_W = \mathbf{x}_W - \bar{\mathbf{x}}_W \quad (7.35)$$

$$\tilde{\mathbf{x}} = \mathbf{x} - \bar{\mathbf{x}} \quad (7.36)$$

$$\tilde{\mathbf{y}} = \mathbf{y} - \bar{\mathbf{y}} \quad (7.37)$$

so that

$$\bar{\mathbf{x}} = \alpha_E \bar{\mathbf{x}}_E + \alpha_W \bar{\mathbf{x}}_W, \quad (7.38)$$

and

$$\tilde{\mathbf{x}} = \alpha_E \tilde{\mathbf{x}}_E + \alpha_W \tilde{\mathbf{x}}_W. \quad (7.39)$$

Then

$$r_{\mathbf{x},\mathbf{y}} = \frac{\tilde{\mathbf{x}} \bullet \tilde{\mathbf{y}}}{\sqrt{\tilde{\mathbf{x}} \bullet \tilde{\mathbf{x}}} \sqrt{\tilde{\mathbf{y}} \bullet \tilde{\mathbf{y}}}}, \quad (7.40)$$

while

$$\tilde{\mathbf{x}} = \alpha_E \tilde{\mathbf{x}}_E + \alpha_W \tilde{\mathbf{x}}_W. \quad (7.41)$$

The first expression is the cosine of the angle

$$\theta_{\tilde{\mathbf{x}},\tilde{\mathbf{y}}} \quad (7.42)$$

between $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{y}}$, all vectors $\tilde{\mathbf{x}}$ in the plane spanned by $\tilde{\mathbf{x}}_E$ and $\tilde{\mathbf{x}}_W$ need to be maximized. To do this, $\tilde{\mathbf{y}}$ must be projected into this plane, and an orthogonal basis for the plane is desired. Using the Gram–Schmidt Method [45] on the $\tilde{\mathbf{x}}_E$ and $\tilde{\mathbf{x}}_W$ an orthogonal basis

$$\left\{ \tilde{\mathbf{x}}_E, \left(\tilde{\mathbf{x}}_W \bullet \tilde{\mathbf{x}}_E \right) \tilde{\mathbf{x}}_W - \left(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_W \right) \tilde{\mathbf{x}}_E \right\} \quad (7.43)$$

is obtained for

$$\text{span} \left\{ \tilde{\mathbf{x}}_E, \tilde{\mathbf{x}}_W \right\}. \quad (7.44)$$

Project $\tilde{\mathbf{y}}$ onto each of these two orthogonal basis vectors, and add the projections to obtain

$$\begin{aligned}
 & \frac{\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{y}}}{\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E} \tilde{\mathbf{x}}_E \\
 & + \frac{(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E)(\tilde{\mathbf{x}}_W \bullet \tilde{\mathbf{y}}) - (\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_W)(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{y}})}{(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E)^2(\tilde{\mathbf{x}}_W \bullet \tilde{\mathbf{x}}_W) - 2(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E)(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_W)^2 + (\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_W)^2(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E)} \\
 & \left((\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E) \tilde{\mathbf{x}}_W - (\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_W) \tilde{\mathbf{x}}_E \right) \\
 & = \frac{\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{y}}}{\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E} \tilde{\mathbf{x}}_E + \frac{(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E)(\tilde{\mathbf{x}}_W \bullet \tilde{\mathbf{y}}) - (\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_W)(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{y}})}{(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E)^2(\tilde{\mathbf{x}}_W \bullet \tilde{\mathbf{x}}_W) - (\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E)(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_W)^2} \\
 & \left((\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E) \tilde{\mathbf{x}}_W - (\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_W) \tilde{\mathbf{x}}_E \right). \tag{7.45}
 \end{aligned}$$

Since $r_{\tilde{\mathbf{x}}, \tilde{\mathbf{y}}}$ will be unaffected by multiplying $\tilde{\mathbf{x}}$ by a scalar, multiply the last vector by the denominator

$$(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E)^2(\tilde{\mathbf{x}}_W \bullet \tilde{\mathbf{x}}_W) - (\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E)(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_W)^2 \tag{7.46}$$

to simplify the expression

$$\begin{aligned}
 & \left((\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E)(\tilde{\mathbf{x}}_W \bullet \tilde{\mathbf{x}}_W)(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{y}}) - (\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_W)^2(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{y}}) \right. \\
 & \left. - (\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_W)(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E)(\tilde{\mathbf{x}}_W \bullet \tilde{\mathbf{y}}) + (\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_W)^2(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{y}}) \right) \tilde{\mathbf{x}}_E \\
 & + \left((\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E)^2(\tilde{\mathbf{x}}_W \bullet \tilde{\mathbf{y}}) - (\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E)(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_W)(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{y}}) \right) \tilde{\mathbf{x}}_W \tag{7.47}
 \end{aligned}$$

Two terms may be canceled to obtain the following:

$$\begin{aligned}
 & \left((\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E)(\tilde{\mathbf{x}}_W \bullet \tilde{\mathbf{x}}_W)(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{y}}) - (\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_W)(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E)(\tilde{\mathbf{x}}_W \bullet \tilde{\mathbf{y}}) \right) \tilde{\mathbf{x}}_E \\
 & + \left((\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E)^2(\tilde{\mathbf{x}}_W \bullet \tilde{\mathbf{y}}) - (\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E)(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_W)(\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{y}}) \right) \tilde{\mathbf{x}}_W \tag{7.48}
 \end{aligned}$$

Finally, divide by the scalar $n^2 \tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E$

$$\begin{aligned} & \left(\left(\frac{\tilde{\mathbf{x}}_W \bullet \tilde{\mathbf{x}}_W}{n} \right) \left(\frac{\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{y}}}{n} \right) - \left(\frac{\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_W}{n} \right) \left(\frac{\tilde{\mathbf{x}}_W \bullet \tilde{\mathbf{y}}}{n} \right) \right) \tilde{\mathbf{x}}_E \\ & + \left(\left(\frac{\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E}{n} \right) \left(\frac{\tilde{\mathbf{x}}_W \bullet \tilde{\mathbf{y}}}{n} \right) - \left(\frac{\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_W}{n} \right) \left(\frac{\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{y}}}{n} \right) \right) \tilde{\mathbf{x}}_W \end{aligned} \quad (7.49)$$

Then let

$$\begin{aligned} \beta_E &= \left(\frac{\tilde{\mathbf{x}}_W \bullet \tilde{\mathbf{x}}_W}{n} \right) \left(\frac{\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{y}}}{n} \right) - \left(\frac{\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_W}{n} \right) \left(\frac{\tilde{\mathbf{x}}_W \bullet \tilde{\mathbf{y}}}{n} \right) \\ &= \text{var} [\mathbf{x}_W] \text{cov} [\mathbf{x}_E, \mathbf{y}] - \text{cov} [\mathbf{x}_E, \mathbf{x}_W] \text{cov} [\mathbf{x}_W, \mathbf{y}] \end{aligned} \quad (7.50)$$

and

$$\begin{aligned} \beta_W &= \left(\frac{\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_E}{n} \right) \left(\frac{\tilde{\mathbf{x}}_W \bullet \tilde{\mathbf{y}}}{n} \right) - \left(\frac{\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{x}}_W}{n} \right) \left(\frac{\tilde{\mathbf{x}}_E \bullet \tilde{\mathbf{y}}}{n} \right) \\ &= \text{var} [\mathbf{x}_E] \text{cov} [\mathbf{x}_W, \mathbf{y}] - \text{cov} [\mathbf{x}_E, \mathbf{x}_W] \text{cov} [\mathbf{x}_E, \mathbf{y}] \end{aligned} \quad (7.51)$$

Setting

$$\alpha_E = \frac{\beta_E}{\beta_E + \beta_W} \quad (7.52)$$

and

$$\alpha_W = \frac{\beta_W}{\beta_E + \beta_W} \quad (7.53)$$

yields the desired vector

$$\mathbf{x} = \alpha_E \mathbf{x}_E + \alpha_W \mathbf{x}_W, \quad (7.54)$$

and of course

$$\bar{\mathbf{x}} = \alpha_E \bar{\mathbf{x}}_E + \alpha_W \bar{\mathbf{x}}_W, \quad (7.55)$$

with $\alpha_E + \alpha_W = 1$. To finish, take the least-squares regression line of \mathbf{y} on \mathbf{x} . Writing

$$m = \frac{\text{cov} [\mathbf{x}, \mathbf{y}]}{\text{var} [\mathbf{x}]} \quad (7.56)$$

for its slope, the line is

$$y' = m(x - \bar{x}) + \bar{y}. \tag{7.57}$$

Given the counts X_E from the East and X_W from the West, the prediction for the fruit yield Y is

$$Y' = m(\alpha_E X_E + \alpha_W X_W - \bar{x}) + \bar{y}. \tag{7.58}$$

7.4.2.3 Statistical and Probabilistic Results

The values of the weights (α_E and α_W) for the apple and peach orchard are displayed in Table 7.1. As seen from this table, the weights applied to the peach blossom count are very similar, where the weights applied to the apple blossom count have a large difference. This is because the images were acquired at 9 a.m., and there was slight overcast when acquiring the peach orchard images, where there were clear skies when acquiring the apple orchard images. So for the apple orchard, images from the west side were looking into the sun, thus increasing the blossom count variance between each tree, and lowering the weight. For the peach orchard, there was overcast, so the images from the east and west sides were consistent, thus resulting in consistent weights.

Table 7.2 elaborates on the results for the apple and peach orchards. In this table, the probability of underestimation is the confidence which the program will not overestimate the number of fruit. This number was calculated by performing a right-tailed binary hypothesis test, by using the two averages of each orchard along with the sample standard deviation to determine a Z-score, which then can be used to determine a significance level [44]. It is important to have a high probability of underestimation, so the farmer is not misled in the number of fruit he/she has, while maintaining a low percent error, so the farmer can have an accurate revenue estimation and be accurate in resource allocation.

Table 7.1 Value of weight applied to the blossom counts

	Apple orchard	Peach orchard
East side	0.639	0.475
West side	0.361	0.525

Table 7.2 Additional results of the early yield estimation program

	Apple orchard	Peach orchard
Coefficient of correlation between ground truth yield and predicted yield	0.699	0.606
Ground truth average fruit per tree	114.17	66.733
Estimated average fruit per tree	103.47	61.440
Percent error	-9.37%	-7.93%
Probability of underestimation	97.19%	95.25%

7.4.3 General Image Processing Techniques for Other Projects

What are the main points which can be taken away from the process which was described in the previous sections? If the images which are being processed have the same set of objects, the RGB data can be used to isolate the objects in the image. By manually collecting a set of sample data and applying different transformations to them, the transformation which best isolates the object of interest can be determined. Then after applying an n -dimensional transformation matrix to the sample data, an equation with n independent variables can be determined that filters out the other objects in the image.

In the example of isolating the blossoms in apple and peach trees, which was presented in Sect. 7.4.1, a transformation $T : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ was applied to the data set. This transformation greatly depends on the scene constraints and the application of the system. Zhang et al. [46] used a transformation $T : \mathbb{R}^3 \rightarrow \mathbb{R}^1$, where they monitored tea leaves to determine the optimal time for harvesting. In this study, the blue plane was subtracted from the green plane. The primary reason why the $\mathbb{R}^3 \rightarrow \mathbb{R}^1$ transformation worked better than the $\mathbb{R}^3 \rightarrow \mathbb{R}^2$ or a $\mathbb{R}^3 \rightarrow \mathbb{R}^3$ transformation is the scene constraints. So when applying machine vision to a system, the scene constraints should have a large influence on which types of transformation is applied.

7.4.3.1 Potential Problems with Over-Constraining the Sample Data

Elaborating more on the equation with n independent variables to isolate the object of interest, this equation can be very involved, but it is often better to use a simple, linear equation. As with the example that isolates blossoms, a linear equation was used because if a parabolic equation with a high degree is used, there is a chance of over constraining the sample data, which may not be useful when applied to the entire image.

Suppose there is a set of images with two objects. A set of RGB sample data is taken, and a rotation transformation to display the red and green values is applied. In this hypothetical situation, this 2D scatterplot may look like the scatterplot displayed in Fig. 7.21. At this point, a line separating the two objects would be drawn, and the equation of this line is the color filter which would be applied to the entire image.

If a straight line is drawn in the sample data set in Fig. 7.21, there is some error, which can be seen in Fig. 7.22. So it would be very tempting to draw a high-degree parabolic function, to get complete separation between the two objects. This high degree polynomial line, which also can also be seen in Fig. 7.22, is an example of over constraining the sample data, and it may have less success when applied to the image because of its complexity.

Suppose the RGB values of each pixel of both objects were known. Of course, this is an impossible task because each individual pixel would have to be analyzed for each image in the set, determining which category the pixel belonged to, and this is potentially millions of pixels. Regardless, suppose these data set exists. If the linear

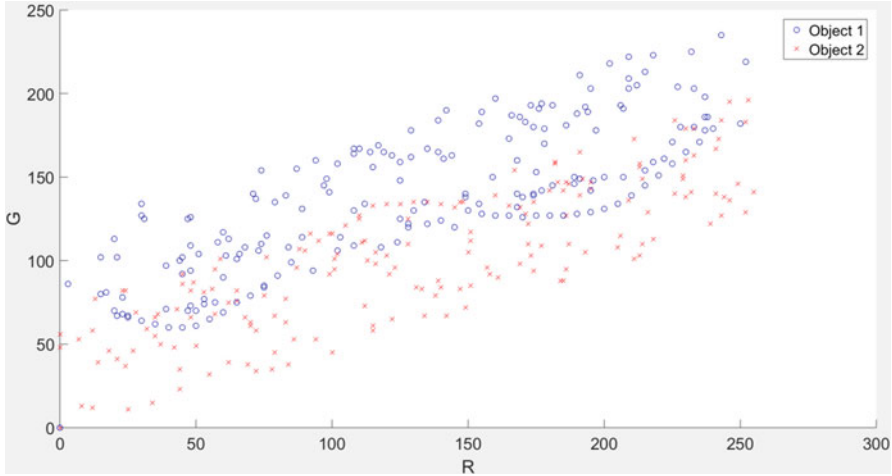


Fig. 7.21 Sample data set of a hypothetical situation

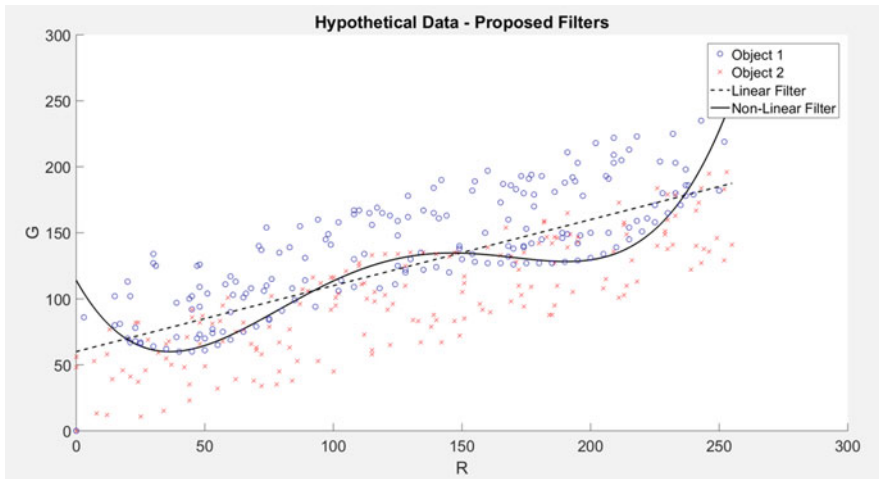


Fig. 7.22 Over constrained and linear equations applied to the sample data set

and parabolic function was applied to this complete data set, seen in Fig. 7.23, it is seen why it is better to use the linear equation compared to the parabolic equation.

But this is just a hypothetical situation used to prove the point of how linear equations are sometimes better than a parabolic equation. How is this proved? Looking back at Fig. 7.21, there are not a lot of sample data points, compared to the complete data set seen in Fig. 7.23. The small sample data size is the reason for the error when applied to the complete data set. The more sample data points which are collected, the higher the confidence which can be had in a complex, nonlinear filter. Going back to the apple and peach trees, there are 12 million pixels in the image and

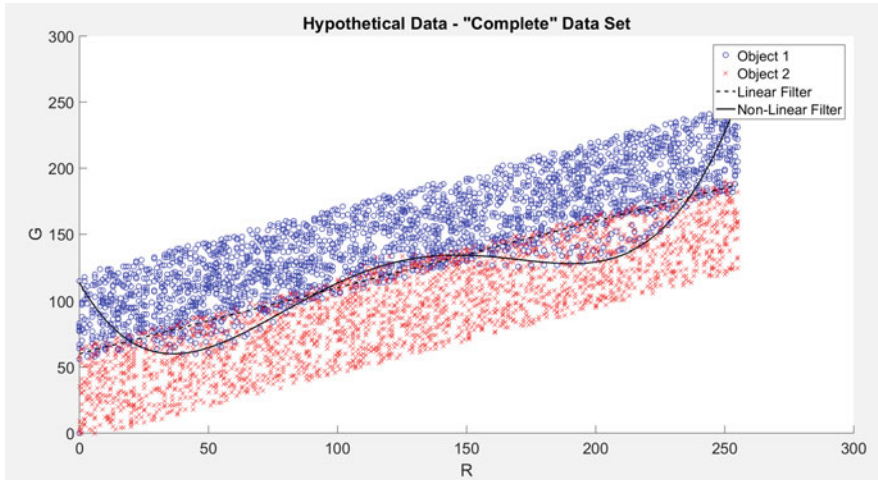


Fig. 7.23 Over-constrained and linear equations applied to the “complete” dataset

60 pictures for each orchard. From this, it is very easy to see it would be extremely time intensive to gather enough pixels to have enough confidence to use an advanced function, which is why it is better to use a simple, linear function.

7.5 An Alternate Method of Object Isolation

7.5.1 Introduction

One of the biggest challenges in using machine vision for agriculture is object isolation. The background for an image or video captured outdoors is seldom uniform. There are always objects and features surrounding the object of interest. For example, when capturing an image of an orchard tree, adjacent trees or trees from different rows will appear in the background. If one was taking a picture of a corn field, adjacent fields with different crops could appear in the background. In any case most of the images will have the sky or the ground in them, and these unwanted elements present unique challenges to image processing.

Humans are easily able to identify and isolate objects in an image; however, machines must be given a little more guidance. For instance, consider the car in Fig. 7.24. A simple method of isolating the car from the building in the background is to apply a color threshold filter. However, this method requires fixed RGB threshold values, and depending on the lighting (or the color of the car and building), the thresholds might need to be adjusted for each image to effectively isolate the object. Furthermore, if the car and the background are similar colors, it is even harder to distinguish the two. Notice that in the top right corner of the image there is a portion



Fig. 7.24 Car in front of a building

of gray sky. The sky color closely matches that of the car, and a simple color filter likely would not differentiate the two. For the car in Fig. 7.24, isolation would be possible by using an area or size filter of some kind. However, if the sky and car had similar pixel areas, image processing might classify the sky as a silver car. Of course, in agriculture, a corn stalk or orchard tree might be pictured instead of a car, but the concepts are the same.

7.5.2 *Spatial Mapping*

A more effective way to isolate an object from its background than using a color filter is by using spatial mapping. Spatial mapping is the process of creating a three-dimensional map of a given environment from sensor data. This sensor data most often come in the form of an image or an array of distance measurements from some arbitrary point to different points in the environment. For example, a stereo camera would produce sensor data in the form of two images and a light detection and ranging device (LIDAR) would produce an array of distance measurements [47].

Spatial mapping can be used to isolate an object from its background by using the physical geometry of the object. For instance, the car in the image is closer to the camera than the building in the background, thus if a 3D map of the image was available, a distance filter could be applied, and the car could be isolated. Furthermore, the dimensions of the car such as length, width, and height could be used to further isolate or to classify the car.

In order to use spatial mapping, a 3D map of the image must be acquired. As previously mentioned, a 3D map can be obtained from a stereo camera or by using LIDAR technology. Both technologies are useful, depending on the application.



Fig. 7.25 ZED stereo camera

However, for this section, the focus will be on the stereo camera. Figure 7.25 shows an example of a stereo camera. The camera pictured is the ZED camera designed by Stereolabs.

7.5.3 Stereo Camera Operation

Stereo cameras are devices that use two fixed RGB cameras to generate a 3D map of an image. The general concept behind stereo cameras is that objects that are close will have a large pixel shift between the two cameras, and objects that are far away will have very little pixel shift between the two cameras. In addition to the 3D map, an RGB image is obtained from a stereo camera. Normally each pixel in the RGB image will be assigned an X-Y-Z Cartesian values, and from those values, a 3D map in the form of a point cloud can be generated. Stereo cameras can perform 3D rendering very quickly, and so they are favorable in real-time applications such as robotics and machine vision [48, 49].

Stereo cameras are particularly effective at isolating trees in a fruit orchard. When photographing a tree in an orchard, the center tree (the tree of interest), as well as the tree to the left and to the right, and parts of trees in one row down are unintentionally included in the image as seen in Fig. 7.26, which shows an unfiltered picture of an apple orchard. By applying a simple distance filter, the sky and background trees can be removed from the image without any manual selecting. The filtered image can then be processed without encountering any negative effects from the background. Stereo cameras are advantageous because they do not rely on pixel color values for their object isolation. If image processing was being used to isolate the trees in Fig. 7.26, then the color thresholds might have to be adjusted to account for the

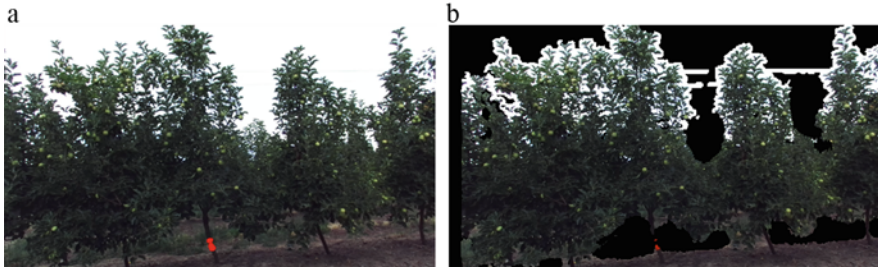


Fig. 7.26 (a) Unfiltered orchard photo. (b) Background filtered orchard photo

overcast sky or for the way the sun reflects off the leaves at different times during the day. Using a distance filter with a stereo camera, however, depends solely on the distance from the camera to the tree and operates independently of the changing light conditions.

Notice in Fig. 7.26b how the trees in the background have been filtered out leaving just the trees in the row of interest. It would have been difficult to isolate the two rows using color and area filters. This is because the two rows are the same color and the trees in the two rows blend together, making it very hard to differentiate them. However, the stereo camera provides spatial information on the location of the trees and makes differentiating between the two rows a relatively simple task.

A further advantage of the stereo camera is that it can obtain basic dimensions of the trees such as height and width. This information can be used to determine the health and canopy volume of a given tree [50]. Tree geometry is also useful when calculating a fruit yield estimate.

7.5.4 Difficulties of Using Spatial Mapping to Isolate Objects

Using stereo cameras for special mapping and object isolation presents a few unique challenges. For instance, there are intrinsic errors associated with using stereo cameras, and one should always check the device specifications and preform some basic tests to verify that the camera meets the design criteria. Mapping objects that are far away will introduce larger error than objects that are close.

In addition, stereo cameras will occasionally measure a few points that have a dramatically large error. This will create data spikes in the 3D array and if not accounted for can have a huge detrimental effect. To reduce the effect of these data spikes, it is recommended to take many snapshots or even a video of the environment and average together the data that is collected. Some stereo cameras are even able to track the location of the camera with respect to the environment. If that capability is available, it is helpful to move the camera and take measurements at different angles.

Furthermore, stereo cameras work best when imaging objects that have a lot of complexity. Complexity helps the stereo camera determine how much an object

shifts between its two RGB cameras. For example, consider using a stereo camera to measure the distance to a blank white wall. If the two RGB cameras on the stereo imager collect identical images, then the camera will determine that the wall is far away since it was not able to detect any shift between the two RGB cameras. This is a completely unreliable measurement because camera could be an inch away from the wall, and the camera would still determine that the wall is far away. Complexity is important; however, it does not have to come in the form of geometric or texture complexity. Simple color complexity will enable a stereo camera to work properly. For instance, if there were green stripes painted on the white wall, the camera would be able to detect how the stripes shifted between the two RGB cameras and an accurate 3D map of the wall could be made.

7.5.5 Object Isolation Conclusion

In this section, a novel method of object isolation was introduced. Traditional methods of object isolation use color and area filters to uncover objects of interest. The method proposed in this section uses the spatial information of an environment to differentiate between objects. This method isolates objects without relying on the color of the object. It is especially advantageous when the object of interest is a similar color to its background as occurs often in agriculture applications.

7.6 A Machine Vision for Peach Orchard Navigation

7.6.1 Introduction

The process of automating an agricultural operation such as pruning or harvesting in fruit orchards requires a platform that is able to autonomously navigate the orchard. Research on autonomous navigation for an agricultural application was started by using guides that are embedded on the ground [51, 52]. With the development of computer and sensor technologies, autonomous navigation utilized sensors such as limit switches, ultrasonic sensor, lidar, machine vision, and Global Positioning System [53]. In this section, the development of a machine vision system for autonomously navigating a peach orchard is presented. Previous research on navigation using machine vision relied on ground features to use as navigation guides. In this study, a unique approach of an upward looking camera was used to take advantage of the sky features to use as directrix for the unmanned ground vehicle.

7.6.2 Visual Feedback System for Navigation

The block diagram of the visual feedback system of the unmanned ground vehicle (UGV) seen in Fig. 7.27 has three main components: the unmanned ground vehicle platform, the vision sensor, and the controller. The input for navigation is the desired vehicle position, and the visual feedback system is used to correct the position of the vehicle. The error between the desired position and the current position is used by the controller to calculate the position of the vehicle as it moves in between the tree rows.

The visual feedback control system was an image-based position servoing system [12], where features of the image were used as control variables to estimate the vehicle's heading. The image processing of this visual feedback system did not rely on ground features but on sky features. This method is a sky-based approach [54], and the image processing is shown in Fig. 7.28. After acquiring an image, the image was cropped to remove the portion of the sky in the field of view that was closest to the camera. This was done to improve the sensitivity of the control system. It was found that slight changes in the direction of the vehicle were magnified when using the centroid of a point that was further away from the camera. Furthermore, cropping the image reduced the data needing to be processed, resulting in faster processing time and more rapid response of the ground vehicle platform. Since the green color plane provided the higher contrast between the sky and the tree canopy, the green plane was extracted to use for segmentation. A simple thresholding approach was employed to extract the path plane of the vehicle because of the high contrast between the canopy and the sky. The “salt-and-pepper” noise was removed by filtering the thresholded image. Finally, the vehicle's heading was calculated by finding the centroid of the path plane.

After the path plane was extracted, the path plane was inverted and used the position of the difference between the centroid and the set point to find the vehicle's heading and used it to drive the motor actuators, seen in Fig. 7.29. The Proportional-plus-Integral (PI) controller was used to handle the position error and used it to differentially steer the vehicle. The proportional and integral constants, K_P and K_I , of the PI controller were determined by first setting the integral gain to zero and adjusting the proportional gain until the system's response was slightly overdamped

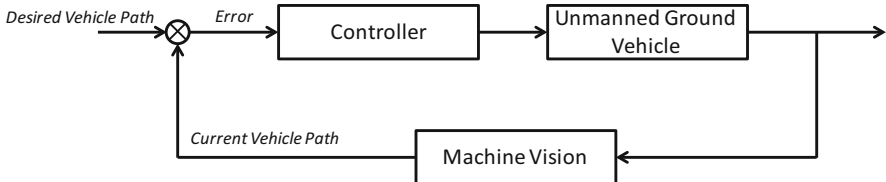


Fig. 7.27 Visual feedback system for unmanned ground vehicle navigation

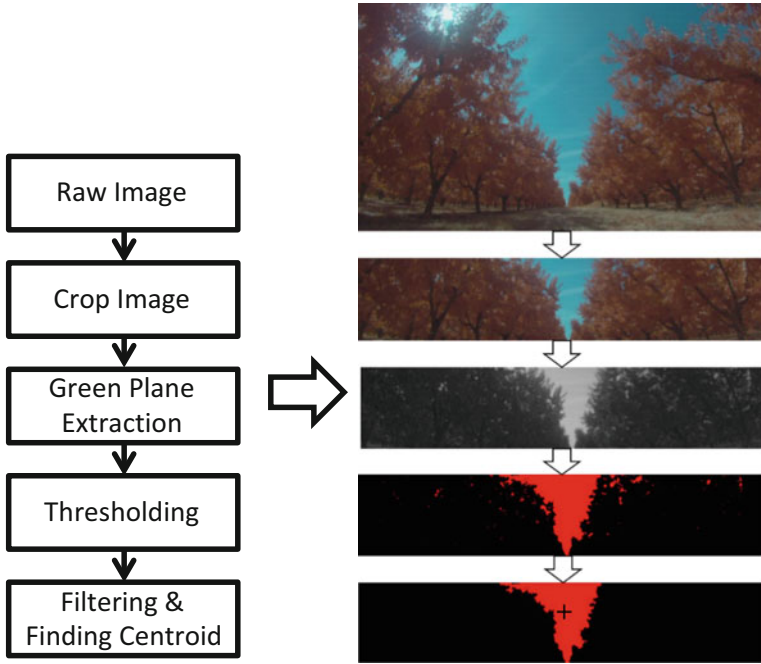


Fig. 7.28 Image processing for finding the centroid of path plane

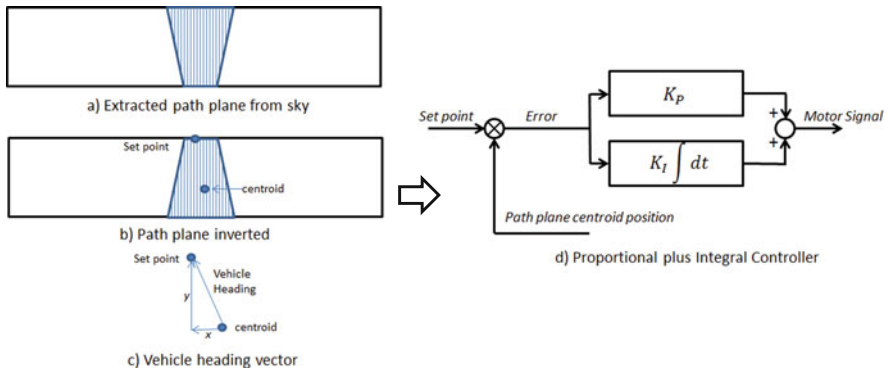


Fig. 7.29 Path plane manipulation and the PI controller for navigation

[55]. The integral gain was adjusted to remove the steady-state error. Once the PI controller had been tuned, a forward speed adjusted to 30% of the maximum value was used as the forward control signal.

7.6.3 *Experimental Ground Vehicle Platform*

The navigation control system was evaluated in a commercial peach orchard located in Caldwell, Idaho, USA. The orchard is well maintained, and one of the rows was randomly selected as a test row. To evaluate the performance of the visual feedback system, the distance from the vehicle to one of the tree rows was measured using an ultrasonic sensor, and cardboard boxes were positioned at a fixed distance from one row of trees. As the vehicle traveled down the row, the distance from the cardboard box was measured via the ultrasonic sensor. Ultrasonic measurements were taken over the first 27 m of travel, and visual observation of the vehicle was done as it finished the whole length of the row.

Figure 7.30 shows that the UGV deviated a maximum of 3.5 cm from its starting point over the 27 m traveled. Based on the test results, it was determined that the image processing algorithm for the vehicle guidance system was sufficient for guiding the vehicle down the orchard row.

The challenges in developing a machine vision system for outdoor application include inconsistent lighting, shadows, and color similarities in features. These difficulties were eliminated by using the sky-based approach where the image contained only the canopy and the sky, thus simplifying the segmentation process. This is a very good example of simplifying the scene constraint, the first component of the machine vision model, to aid segmentation. A simple and effective image segmentation facilitates feature detection. In addition to the test in which ultrasonic data was taken over a set distance, the vehicle was allowed to run the entire length of the row. The vehicle completed the entire row with very little error; however, it was observed that there were larger deviations from the center of the row when the vehicle approached sections where there was a break in the canopy either due to a missing tree or a tree with limited leaf growth. These breaks in the canopy caused the UGV to move away from the center of the row, but when the vehicle would move past that section, it would correct itself and return to the center. The result of a missing tree affected the shape of the path plane. This means that the shape of the path plane

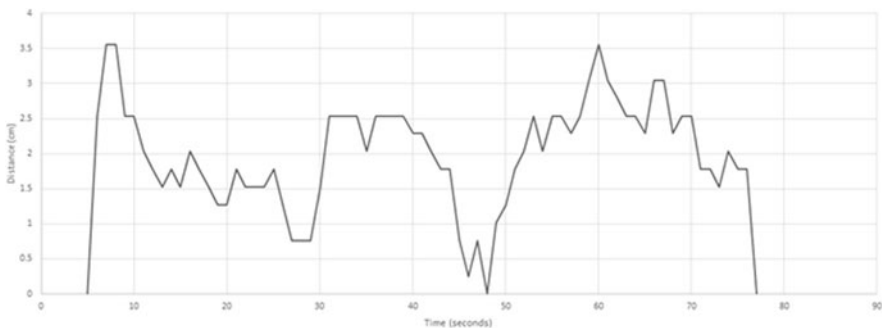


Fig. 7.30 Deviation from starting point for the peach orchard evaluation

could be used to determine a missing tree or end of the row conditions. The result of the test run in the field showed that the sky-based approach in combination with the PI controller was effective in guiding the vehicle down the row.

The sky-based approach machine vision for orchard navigation demonstrated the potential of guiding a ground vehicle in a straightline motion. However, there are some drawbacks to the sky-based approach. It is only effective when the trees have fully developed canopies. Fruit trees that have canopies year-round such as citrus will greatly benefit from this system. On the other hand, fruit trees that lose their leaves in the winter and remain dormant until the spring season will have no canopy during this season. Pruning and other orchard operations are conducted during the dormant period of the trees. To help automate these operations, the ground vehicle should rely on ground features. In this case, a ground-based image processing would be effective. Furthermore, the problems with shadows when the canopy is present can be disregarded. Therefore, for orchards that have deciduous trees, an adaptive image processing approach could be developed to deal with the changing environmental condition. For example, a sky-based image processing will be used when canopies are present, and a ground-based approach will be employed when there are no leaves. The other drawback of the proposed approach is that it only tackles the straightline motion down the row but not the end-of-the-row condition. The end-of-the-row condition could be handled in several ways. An ultrasonic sensor could be used to detect the absence of a series of trees. Another approach would be to observe the path plane of the sky-based approach. The shape of the path plane will be different at the end of the row, and this can be used to trigger the vehicle that it is at the end of the row. Future research would include dealing with a changing environment such as with canopy and without canopy conditions, detecting end-of-the-row conditions, and translating to the next row.

7.7 Conclusion

In this chapter, the different applications of machine vision in agriculture were presented. The vision applications are classified into the following groups: plant identification, process control, and machine control. Concerning plant identification, a machine vision system developed to estimate fruit yield early in the season was discussed. The developed fruit yield estimator identified and counted blossoms and correlated it with the total number of fruits on the tree. A coefficient of correlation of approximately 0.70 was obtained for both apple and peach orchards. An individual tree recognition algorithm combined with stereo-imaging was also discussed. This algorithm removed trees in the background, which could provide false positives of the blossom count. Concerning machine control, a machine vision system to navigate an unmanned ground vehicle prototype was described. The ground vehicle was able to successfully navigate an entire row of commercial peach trees autonomously. These application examples display the potential of machine vision in the field of orchard production. The future of the automation of production agriculture is very bright with machine vision as one of its tools.

References

1. Chen, Y., Chao, K., & Kim, M. S. (2002). Machine vision technology for agricultural applications. *Computers and Electronics in Agriculture*, *36*, 173–191. [https://doi.org/10.1016/s0168-1699\(02\)00100-x](https://doi.org/10.1016/s0168-1699(02)00100-x).
2. Buluswar, S. D., & Draper, B. A. (1998). Color machine vision for autonomous vehicles. *Engineering Applications of Artificial Intelligence*, *11*(2), 245–256. [https://doi.org/10.1016/s0952-1976\(97\)00079-1](https://doi.org/10.1016/s0952-1976(97)00079-1).
3. Deac, G. C., Deac, C. N., Popa, C. L., Ghinea, M., & Cotet, C. E. (2017). Machine vision in manufacturing processes and the digital twin of manufacturing architectures. *DAAAM Proceedings of the 28th International DAAAM Symposium, 2017*, 0733–0736. <https://doi.org/10.2507/28th.daaam.proceedings.103>.
4. Sharan, R. V., & Onwubolu, G. C. (2014). Automating the process of work-piece recognition and location for a pick-and-place robot in a SFMS. *International Journal of Image, Graphics and Signal Processing*, *6*(4), 9–17. <https://doi.org/10.5815/ijigsp.2014.04.02>.
5. Emmi, L., Gonzalez-De-Soto, M., Pajares, G., & Gonzalez-De-Santos, P. (2014). Integrating sensory/actuation systems in agricultural vehicles. *Sensors*, *14*(3), 4014–4049. <https://doi.org/10.3390/s140304014>.
6. Matache, M., Persu, C., Nitu, M., & Gabriel, G. (2017). Vision system for spraying machines adaptive control. *Engineering for Rural Development—International Scientific Conference*, *24*, 358–363. <https://doi.org/10.22616/erdev2017.16.n071>.
7. Awcock, G. J., & Thomas, R. (1995). *Applied image processing*. New York: McGraw-Hill.
8. Chao, K., Park, B., Chen, Y. R., Hruschka, W. R., & Wheaton, F. W. (2000). Design of a dual-camera system for poultry carcasses inspection. *Applied Engineering in Agriculture*, *16*(5), 581–587.
9. Bulanon, D. M., & Kataoka, T. (2010). Fruit detection system and an end effector for robotic harvesting of Fuji apples', International commission of Agricultural and. *Biosystems Engineering Journal*, *12*(1), 203–210.
10. Ni, Z., & Burks, T. F. (2013). Plant or tree reconstruction based on stereo vision. In *Annual meeting of the American Society of Agricultural and Biological Engineers*, 2013.
11. Bulanon, D. M., Burks, T. F., & Alchanatis, V. (2008). Study on temporal variation in citrus canopy using thermal imaging for citrus fruit detection. *Biosystems Engineering*, *101*(2), 161–171.
12. Corke, P. (2011). Robotics, Vision and Control: Fundamental Algorithms in MATLAB. *Springer Tracts in Advanced Robotics*, *73*(6), 2011.
13. Sabzi, S., Abbaspour-Gilandeh, Y., & Javadikia, H. (2017). Machine vision system for the automatic segmentation of plants under different lighting conditions. *Biosystems Engineering*, *161*, 157–173. <https://doi.org/10.1016/j.biosystemseng.2017.06.021>.
14. Partel, V., Kakarla, S. C., & Ampatzidis, Y. (2019). Development and evaluation of a low-cost and smart technology for precision weed management utilizing artificial intelligence. *Computers and Electronics in Agriculture*, *157*, 339–350. <https://doi.org/10.1016/j.compag.2018.12.048>.
15. Ruiz-Altisent, N., Ruiz-Garcia, L., Moreda, G. P., Lu, R., HernandezSanchez, N., Correa, E. C., et al. (2010). Sensors for product characterization and quality of specialty crops—A review. *Computers and Electronics in Agriculture*, *74*, 176–194.
16. Davies, E. R. (1997). *Machine vision: Theory, algorithms, practicalities*. New York: Academic Press.
17. Guyer, D. E., Miles, G. E., Schreiber, M. M., Mitchell, O. R., & Vanderbilt, V. C. (1986). Machine vision and image processing for plant identification. *Transactions of ASAE*, *29*(6), 1500–1507.
18. Lee, W. S., Slaughter, D. C., & Giles, D. K. (1999). Robotic weed control system for tomatoes. *Precision Agriculture*, *1*, 95–113.
19. Bulanon, D. M., Lonai, J., Skovgard, H., & Fallahi, E. (2016). Evaluation of different irrigation methods for an apple orchard using an aerial imaging system. *ISPRS International Journal of*

Geo-Information, 5, 79.

20. Lindner, L., Sergiyenko, O., Rivas-López, M., Valdez-Salas, B., Rodríguez-Quiñonez, J. C., Hernández-Balbuena, D., et al. (2016, November). Machine vision system for UAV navigation. In *Electrical Systems for Aircraft, Railway, Ship Propulsion and Road Vehicles & International Transportation Electrification Conference (ESARS-ITEC), International Conference on IEEE* (pp. 1–6).
21. Lindner, L., Sergiyenko, O., Rivas-López, M., Ivanov, M., Rodríguez-Quiñonez, J. C., Hernández-Balbuena, D., et al. (2017, June). Machine vision system errors for unmanned aerial vehicle navigation. In *Industrial Electronics (ISIE), 2017 IEEE 26th International Symposium on IEEE* (pp. 1615–1620).
22. Wu, W. Y., Wang, M. J., & Liu, C. M. (1996). Automated inspection of printed circuit boards through machine vision. *Computers in Industry*, 28(2), 103–111.
23. Miller, M. K., & Delwiche, M. J. (1989). A color vision system for peach grading. *Transactions of ASAE*, 32(4), 1484–1490.
24. Bulanon, D. M., Burks, T. F., Kim, D. G., & Ritenour, M. A. (2013). Citrus black spot detection using hyperspectral image analysis. *Agricultural Engineering International: CIGR Journal*, 15(3), 171–180.
25. Rehkugler, G. E., & Throop, J. A. (1986). Apple sorting with machine vision. *Transactions of ASAE*, 29(10), 1388–1397.
26. Costa, C., Antonucci, F., Pallottino, F., Aguzzi, J., Sun, D. W., & Menesatti, P. (2011). Shape analysis of agricultural products: A review of recent research advances and potential to computer vision. *Food and Bioprocess Technology*, 4, 673–692.
27. Morimoto, T., Takeuchi, T., Miyata, H., & Hashimoto, Y. (2000). Pattern recognition of fruit shape based on the concept of chaos and neural networks. *Computers and Electronics in Agriculture*, 26, 171–186.
28. Pallottino, F., Menesatti, C., Costa, C., Paglia, G., De Salvador, F. R., & Lolletti, D. (2010). Image analysis techniques for automated hazelnut peeling determination. *Food and Bioprocess Technology*, 3(1), 155–159.
29. Parrish, E., & Goksel, A. K. (1977). Pictorial pattern recognition applied to fruit harvesting. *Transactions of ASAE*, 20, 822–827.
30. Slaughter, D., & Harrel, R. C. (1987). Color vision in robotic fruit harvesting. *Transactions of ASAE*, 30(4), 1144–1148.
31. Bulanon, D. M., Burks, T., & Alchanatis, V. (2009). Image fusion of visible and thermal images for fruit detection. *Biosystems Engineering*, 103(1), 12–22.
32. Ehsani, R., & Karimi, D. (2010). Yield monitors for specialty crops. *Landbauforschung Völknerode, Special Issue*, 340, 31–43.
33. Maja, J. M., Campbell, T., Neto, J. C., & Astillo, P. (2016). Predicting cotton yield of small field plots in a cotton breeding program using UAV imagery data. In *Proc SPIE 986, Autonomous Air and Ground Sensing Systems for Agricultural Optimization and Phenotyping*, 98660C. Retrieved May 17, 2016.
34. Tumbo, S. D., Whitney, J. D., Miller, W. M., & Wheaton, T. A. (2002). Development and testing of a citrus yield monitor. *Applied Engineering in Agriculture*, 18, 399–403.
35. Annamalai, P., & Lee, W. S. (2003). *Citrus yield mapping system using machine vision, 2003 ASAE Annual Meeting*. American Society of Agricultural and Biological Engineers.
36. Stajanko, D., Rakun, J., & Blanke, M. (2009). Modelling apple fruit yield using image analysis for fruit colour, shape and texture. *European Journal of Horticultural Science*, 74(6), 260–267.
37. Lee, W. S., Alchanatis, V., Yang, C., Hirafuji, M., Moshou, D., & Li, C. (2010). Sensing technologies for precision specialty crop production. *Computers and Electronics in Agriculture*, 74, 2–33.
38. Zaman, Q. U., Swain, K. C., Schumann, A. W., & Percival, D. C. (2010). Automated, low-cost yield mapping of wild blueberry fruit. *Applied Engineering in Agriculture*, 26(2), 225–232.
39. Wang, Q., Nuske, S., Bergerman, M., & Singh, S. (2012). Automated crop yield estimation for apple orchards. In *Proceedings of the International symposium on Experimental Robotics*, June 2012, Quebec City.

40. Cheng, H., Damerow, L., Sun, Y., & Blanke, M. (2017). Early yield prediction using image analysis of apple fruit and tree canopy features with neural networks. *Journal of Imaging*, 3, 6.
41. MATLAB. (2016). *MATLAB and image processing toolbox release*. Natick, MA: The MathWorks.
42. Khodaskar, H. V., & Mane, S. (2017). Human face detection & recognition using raspberry Pi. *International Journal of Advanced Engineering, Management and Science*, 1–2. <https://doi.org/10.24001/icsesd2017.50>.
43. Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., et al. (2012). Fiji: An open-source platform for biological-image analysis. *Nature Methods*, 9(7), 676–682. <https://doi.org/10.1038/nmeth.2019>.
44. Bertsekas, D. P., & Tsitsiklis, J. N. (2008). *Introduction to probability* (2nd ed.). Belmont: Athena Scientific.
45. Lay, D. C. (2003). *Linear algebra and its applications* (3rd ed.). Boston: Addison-Wesley.
46. Zhang, L., Zhang, H., Chen, Y., Dai, S., Li, X., Imou, K., Liu, Z., & Li, M. (2019). Real-time monitoring of optimum timing for harvesting fresh tea leaves based on machine vision. *International Journal of Agricultural & Biological Engineering*, 12(1), 6–9.
47. Llorens, J., Gil, E., Llop, J., & Escolà, A. (2011). Ultrasonic and LIDAR sensors for electronic canopy characterization in vineyards: Advances to improve pesticide application methods. *Sensors*, 11(2), 2177–2194. <https://doi.org/10.3390/s110202177>.
48. Geiger, A., Ziegler, J., & Stiller, C. (2011). StereoScan: Dense 3d reconstruction in real-time. In *2011 IEEE Intelligent Vehicles Symposium (IV)*. <https://doi.org/10.1109/ivs.2011.5940405>.
49. Rovira-Más, F., Zhang, Q., & Reid, J. F. (2008). Stereo vision three-dimensional terrain maps for precision agriculture. *Computers and Electronics in Agriculture*, 60(2), 133–143. <https://doi.org/10.1016/j.compag.2007.07.007>.
50. Lee, K., & Ehsani, R. (2008). A laser-scanning system for quantification of tree-geometric characteristics. In *2008 Providence*, Rhode Island, June 29–July 2, 2008. <https://doi.org/10.13031/2013.25003>
51. Richey, C. B. (1959). “Automatic pilot” for farm tractors. *Agricultural Engineering*, 40(2), 78–79, 93.
52. Tillet, N. D. (1991). Automatic guidance sensors for agricultural field machines: A review. *Journal of Agricultural Engineering Research*, 50, 167–187.
53. Reid, J. F., Zhang, Q., Noguchi, N., & Dickson, M. (2000). Agricultural automatic guidance in North America. *Computers and Electronics in Agriculture*, 25(1/2), 154–168.
54. Radcliffe, J., Cox, J., & Bulanon, D. M. (2018). Machine vision for orchard navigation. *Computers in Industry*, 98, 165–171. <https://doi.org/10.1016/j.compind.2018.03.008>.
55. Bolton, W. (2001). *Mechatronics, electronic control systems in mechanical and electrical engineering* (2nd ed.). Boston: Addison Wesley Longman.