Low and High-Level Visual Feature Based Apple Detection from Multi-modal Images

J. P. Wachs , H. I. Stern, T. Burks and V. Alchanatis

Abstract. Automated harvesting requires accurate detection and recognition of the fruit within a tree canopy in real-time in un-controlled environments. However, occlusion, variable illumination, variable appearance and texture make this task a complex challenge. Our research discusses the development of a machine vision system, capable of recognizing occluded green apples within a tree canopy. This involves the detection of "green" apples within scenes of "green leaves", shadow patterns, branches and other objects found in natural tree canopies. The system uses both thermal infra-red and color image modalities in order to achieve improved performance. Maximization of mutual information is used to find the optimal registration parameters between images from the two modalities. We use two approaches for apple detection based on low and high-level visual features. High-level features are global attributes captured by image processing operations, while low-level features are strong responses to primitive parts-based filters (such as Haar wavelets). These features are then applied separately to color and thermal infra-red images to detect apples from the background. These two approaches are compared and it is shown that the low-level feature based approach is superior (74% recognition accuracy) over the high-level visual feature approach (53.16% recognition accuracy). Finally, a voting scheme is used to improve the detection results, which drops the false alarms with little effect on the recognition rate. The resulting classifiers acting independently can partially recognize the on-tree apples, however, when combined the recognition accuracy is increased

Keywords: Mutual information, multi-modal registration, sensor fusion, Haar detector, apple detection.

Introduction

Automatic harvesting based on object recognition methods has been evolving in the last few years, where the main emphasis is the efficient detection of fruits and vegetables in their natural environment. Nevertheless, real-time systems capable

J. P. Wachs
School of Industrial Engineering, Purdue University, West Lafayette, IN
e-mail: jpwachs@purdue.edu

H. I. Stern
Department of Industrial Engineering, Ben Gurion University of the Negev, Beersheva, Israel
e-mail: helman@bgu.ac.il

V. Alchanatis
Institute of Agricultural Engineering, Agricultural Research Organization, the Volcani Center, Bet-Dagan, Israel.
e-mail: victor@volcani.agri.gov.il

T. Burks
Agricultural and Biological Engineering, University of Florida, Gainesville, FL
e-mail: TFBurks@ifas.ufl.edu

of recognizing partially occluded apples; regardless of position, scale, shadow pattern and illumination within a tree canopy are still in the prototyping phase due to the complexity of the task. Labor for orchard tasks constitutes the largest expense (Jiménez *et al., 2000a*), and hence there is a need to address technological challenges leading to autonomous robotic fruit picking systems. In this research, we address the problem of on-tree green apple detection using a real-time machine-vision approach. This involves the detection of "green" apples within scenes of "green leaves", shadow patterns, branches and other objects found in natural tree canopies. Unfortunately, traditional approaches capitalizing on color and edge features alone are not successful in this environment. They are highly dependent on illumination, while texture is highly sensitive to the proximity (scale) of the object. An excellent review regarding apple recognition systems was presented in (Jiménez *et al., 2000b*). The concept of background modeling using Gaussian mixture color distributions in RGB images was used in Tabb *et al.*, (2006). This algorithm detected 85 to 96 percent of both red and yellow apples assuming a uniform background in an artificial environment. Color distribution models for fruit, leaf and background classes were used in Annamalai and Lee, (2003) in a citrus fruit counting algorithm. In Stanjnko *et al.*, (2004) pixel thermal values were mapped to red green blue (RGB) color model values and detected using the normalized difference index. However the efficiency of the algorithm was affected by the apple's position on the tree and degree of sunlight. In Sapina (2001), textural features extracted from the gray level co-occurrence matrix were used to discriminate between warm objects and their background in infra-red (IR) images. In the same vein, a threshold selection approach was proposed by Fernandez-Maloigne *et al.* (1993) based on the texture features of an apple in grayscale images. The authors assume that all the apples have a bright spot (due to their exposure to sunlight) and the apple region is practically homogenous and spherical. These assumptions have limited validity in natural un-controlled scenarios. Texture-based edge detection combined with a measure of redness were used in Zhao *et al.*, (2005) for the detection of green and red apples in trees. The authors claim that their method can deal with occluded apples, clustered apples and cluttered environments. However, no recognition rates are reported. A robust system using an infra-red laser is presented in Jiménez *et al.*, (2000a) which considers illumination, shadows and background objects. The authors report a rate of 80-90% detection when used with an artificial orange tree.
This paper discusses two approaches: high-level and low-level feature-based apple recognition. In each of the two approaches, thermal infra-red and color information are considered. Thermal infra-red provides clues regarding the physical structure and location, while color images provide pattern geometry.

Materials & Methods

Initially, we register between the two modalities (color and thermal) using the method of maximization of mutual information. To detect the apples, we first demonstrate a high-level approach relying heavily on image processing operations. Later, we compare this approach to a low-level feature based approach, using a Haar detector (Viola and Jones, 2004). In the low-level approach, color detections are converted to hypotheses that are each tested by a voting scheme. The resulting detections are combined with the thermal results and transformed using the registration parameters.

Multi-modal image registration using mutual information

In order to detect the apples, 180 color images (RGB) and thermal images were acquired using thermal infrared camera (SC2000, FLIR Inc., Sweden). The resolution of the color images was 3264 columns by 2448 rows, while the thermal images resolution was 320 columns by 240 rows. The images were captured at the Matitiahu farm in the Western Galilee region in North Israel, during August-Nov, 2007. The apples tested were Golden Delicious and Granny Smith type. The camera was mounted in one row at a time in the orchard and orthogonal to the tree row. The fruit was as close as 2.2-2.3 m from the camera under direct sun light and shade, and a total of 7 rows were traversed, see Fig. 1. Seven images from the original dataset were discarded since there was no matching image in the other modality.

To detect the apples automatically, the first step involved 'registration' of the images.

Multi-modal image registration is a fundamental step preceding detection and recognition in image processing tasks used by the pattern recognition community (Brown, 1992). This pre-processing stage concerns the comparison of two images – the base and sensed images - acquired from the same scenario at different times or with different sensors. In order to align the images, a correspondence problem must be solved in which every point in one image is aligned with a corresponding point in the other images.

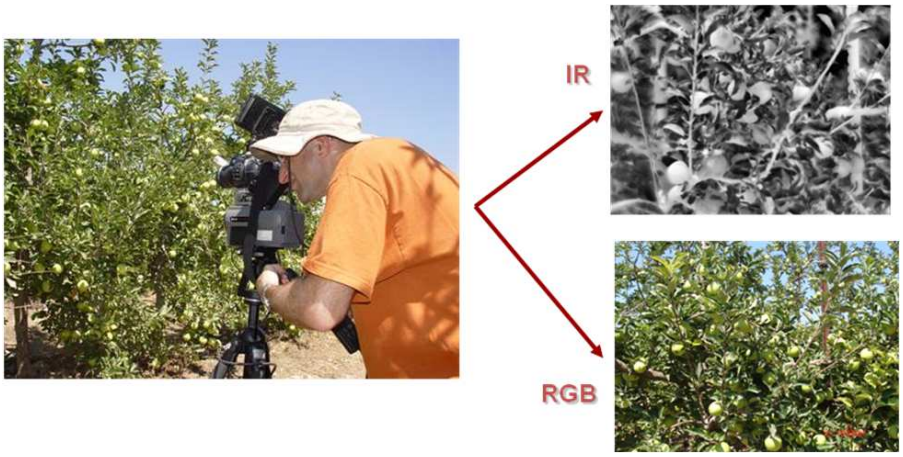


Figure 1. IR and RGB images obtained from an apple tree

In our problem, the transformation between two images of different modalities is affine which means; rotations, translations and scaling are allowed (Sonka *et al.*, 1999). Transformation of the co-ordinate sets P_A and P_B from the sensed image A to the base image B is given by Equation 1.

$$(P_B - C_B) = sR(\theta).(P_B - C_B) + t$$
$$R(\theta) = \begin{pmatrix} cos(\theta) & sin(\theta) \\ -sin(\theta) & cos(\theta) \end{pmatrix} \quad t = \begin{pmatrix} t_X \\ t_Y \end{pmatrix} \qquad (1)$$

Where C_A and C_B are the co-ordinates of the centers of the images (the origin of the co-ordinates is the pixel (0,0) in the bottom left corner of the image), s is a

scaling factor (same for both axes), $R(\theta)$ is the rotation matrix, and t is the translation vector.

We shall compare five different registration methods using the similarity indices: cross correlation normalized (CC_1), correlation coefficient (CC_2), correlation coefficient normalized (CC_3), the Bhattacharyya coefficient (BC) and the Mutual Information index (MI). We will discuss in detail MI since this was the method adopted for registration in this work. However the other measures are also included as these are standard methods to compare similarity between images (Cha and Srihari, 2002; Wachs *et al.*, 2009).

We first describe the method of mutual information (MI) (Viola and Wells, 1995). Let a and b be two discrete random variables (intensities) drawn from images A and B, respectively, with probabilities $p_A(a)$ and $p_B(b)$ and joint probability $p_{AB}(a,b)$. The degree of dependence between A and B can be obtained by the MI, according to Equation 2.

$$I(A,B) = \sum_{a,b} p_{AB}(a,b) \log \frac{p_{AB}(a,b)}{p_A(a).p_B(b)} \qquad (2)$$

A data set including 173 color and thermal images of apple trees were acquired from a digital RGB camera and an IR FLIR camera. These images were registered by the five indices mentioned earlier. Table 1 shows the root mean squared errors (RMS) of the five indices for each registration parameter. Given the true parameters obtained from manual registration, $\alpha_{ij\mu}$*, we denote the error of registration as $\Delta_{ij\mu} = (\alpha_{ij\mu} - \alpha_{ij\mu}*)$ where entries $\alpha_{ij\mu}$ are registration parameter i, for the pair of images j, using the measure m_μ. For a data set of size N, the root mean square error (RMS) is:

$$RMS_{i\mu} = \frac{1}{N} \sum_{j=1,..,N} \sqrt{(\alpha_{ij\mu} - \alpha_{ij\mu}*)^2} \qquad (3)$$

The measures m_μ with $1 \leq \mu < 4$ correspond to bc, m_i, cc_1, cc_2 and cc_3.

The registration parameters $\alpha_{ij\mu}$, with $1 \leq i \leq 4$ correspond to Δs (scale), $\Delta\theta$ (rotation), Δt_x (translation in the x axis), Δt_y (translation in the y axis).

Table 1. Registration parameters RMS error using five 5 similarity indices.

Measure	RMS			
	Δs	Δθ	Δtx (%)	Δty (%)
bc	0.226	2.205	3.958	4.328
mi	**0.175**	**1.701**	**3.547**	3.912
cc_1	0.196	1.929	3.985	**3.868**
cc_2	0.196	1.715	6.030	6.875
cc_3	0.196	1.713	6.067	6.848

Figure 2. Color and thermal registered image

By observing the results in Table 1, the mutual information technique performed better than the other four methods for three parameters (Δs, $\Delta\theta$, Δt_x), and comparable to cc_1 for the last parameter (Δt_y). Therefore MI was selected as the preferred method for registering the whole set of images. Fig. 2 shows an example

of a pair of images registered from the dataset. Note, the terms “measures” and “indices” are used interchangeably through the text.

Method 1: High-Level Visual Features

The first method relies on standard image processing operations and classification techniques to detect apples in both RGB and IR images. The steps are very similar for both image modalities, after minor modifications are made. Figure 3 describes the different steps.

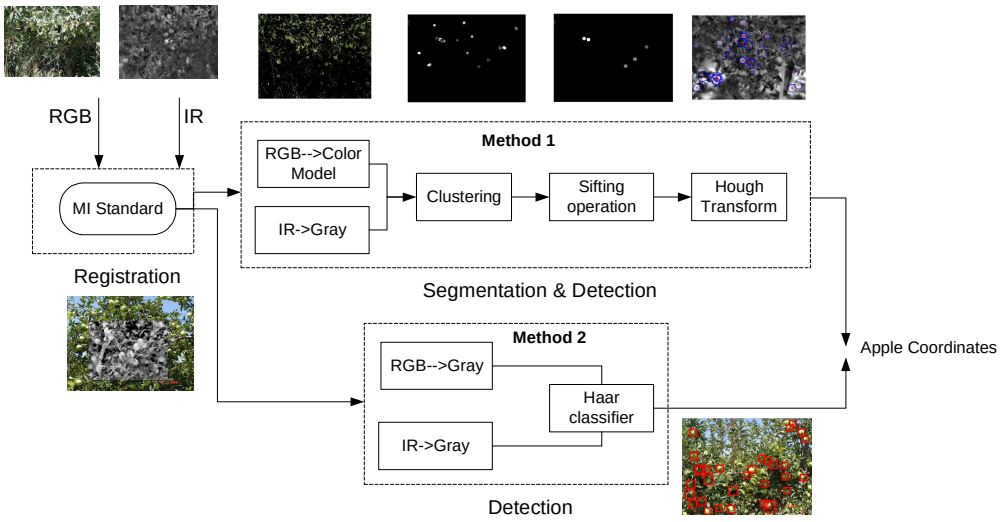


Figure 3. Two different methods for apple detection (Method 1 – high level, Method 2-low level)

Color data was chosen as a first attempt to recognize a single pixel-based property for RGB images (see top left of Figure 3). All the color images acquired were 3264 by 2448 pixels with 24-bit RGB color format (Figure 4a). These can be transformed into several color models such as grayscale, red channel only, and Hue Saturation and Intensity (HSV). The L*a *b color space (Hunter, 1948) was used since it captures the red-green colors (prominent in an apple) in one axis, see Figure 4b.

(a) (b)

Figure 4. Green apples on tree image. (a) RGB. (b) L*a*b channels

In order to detect apples using their temperature, the corresponding grayscale values were used (Figure 3, top left). Thermal infra red images of 320 by 240

pixels with temperature information for each pixel were converted to 8 bit grayscale values.

*Classify the Colors in 'a*b*' Space Using K-Means Clustering*

To initially segment the pixels belonging to apples, the k-means clustering algorithm (Sonka *et al.*, 1999) was used. The *a and *b channels (from the L*a *b color space) were used as 2D data describing the color attributes for each pixel (this corresponds to "RGB->Color Model" and "Clustering" boxes in Figure 3). To reduce the dataset size, the images were reduced to a quarter of their original size.

After clustering the *a and *b components from the training set of 173 images, three separate clusters for each class, namely apple, leaf and background, were obtained (see Fig. 5). For example, the 2D vector [*a *b]=[116 148] represents the centroid of the background class.

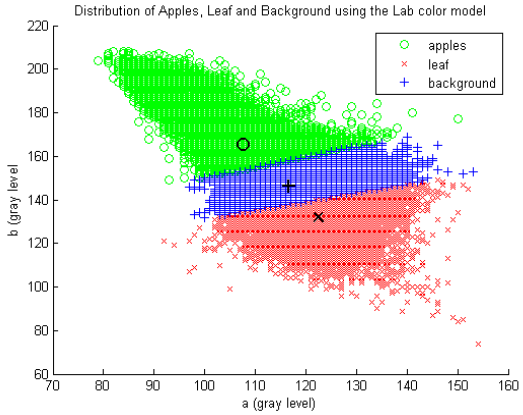


Figure 5. Cluster distribution for the leaves, apples and background classes

Once the k-means was applied to each image in the training set, the classes were labeled. A new image, *T*, was obtained by setting the pixels from the leaf and background classes to '0' (Off) (Fig. 6(b)). In the image *T*, only pixels belonging to the apple cluster were set to '1'. Thus, the image *T* included mostly apples and some leaves with colors similar to the apples. In Figure 6, three views of the same original image are displayed. The views 6a, 6b and 6c show only the pixels belonging to the classes 'leaves', 'apples' and 'background', respectively, while the pixels belonging to the other two classes in each view are 'off' (black).

(a) (b) (c)

Figure 6. Images according to class. (a) Leaves, (b) Apples (Color version of image T), (c) Background

Regarding IR images, the same three classes (apple, leaf and background) were expressed by ranges of grayscale values (0 to 255). Manual selection of apples in every image in the IR training set, indicated that the centroid of the apple class was t_{app}=192. Using this information, the pixels closer to t_{app} than to the other two classes, were considered to belong to the apple class in the thermal infra-red modality. The pixels which belong to that class were set to '1' while the others were set to '0'.

Pre-processing and morphological operations

Although the image *T* includes mostly apples, there are still some branches and leaves included. Much noise is due to a small overlap between the classes in the L*a*b planes, for the RGB modality while, in the IR modality, the overlap occurs between pixels with similar temperatures. To overcome the mis-classification of leaves and branches as apple pixels in *T*, a round shape-structuring element was used to detect rounded objects, supposedly apples, while discarding other objects in the image in both modalities.
First, we maximize the intensity contrast in *T* using histogram equalization (Sonka *et al.*, 1999); later, gaps inside the blobs are filled. The rounded blobs in *T* are separated by applying the 'opening' operation (Sonka *et al.*, 1999) using a kernel consisting of a size equal to the diameter of an apple, and with a 'disk' structuring element (see Figure 7). Different apple sizes were obtained by using the granulometry approach, or sifting (this corresponds to "Sifting operation" box in Figure 3).

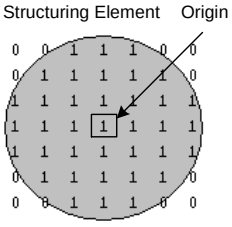


Figure 7. The 'disk' structuring element

Granulometry assesses the intensity surface area distribution of apples as a function of its radii. Granulometry approximates blobs to disks whose radii can be determined by pushing (sifting) them through screens of increasing size and collecting what remains after each pass. This means that, in order to obtain all the apples of radius r^*, we subtract images I_1 and I_2, where I_1 was created after applying the 'opening' operation on *T* with a structuring element of size r^* and I_2 when applying the element of size r^*+1.
The minimum and maximum size of the structuring element is selected by analyzing the manually selected apples from the training set. The maximum and minimum size is the mean radius of the manually selected data set plus 2 standard deviations, and minus 2 standard deviations, respectively. The radii distribution is presented in Figure 8.

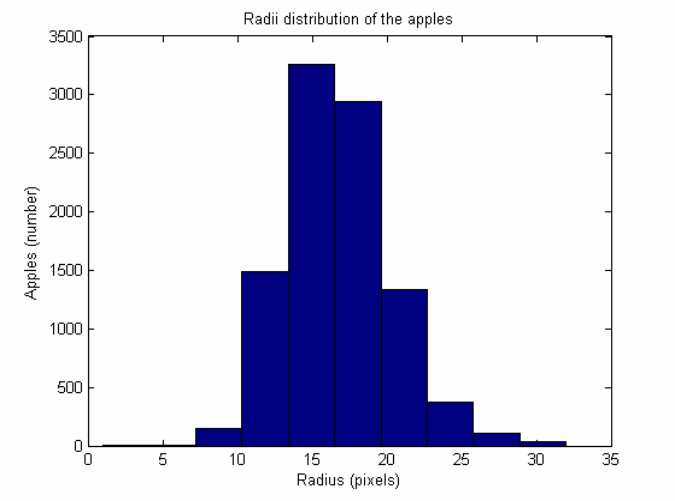
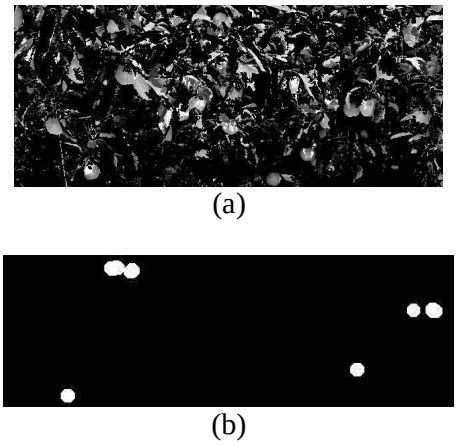


Figure 8. Radii distribution of the apples in the training set

As a result of the sifting process, we obtained an image for each radii considered. To obtain connected components for each image, the pixel grayscale values were converted to binary values through thresholding using Otsu's method (Sonka *et al.*, 1999).

Overlapping areas between apples may also appear after the sifting operation. To discard them, the sifting process is applied again on the binary image (see Figure 9).



(a)

(b)

Figure 9. (a) Pixels from the class 'apples'. (b) Result after 'sifting'

Examining the blobs, one may realize that touching apples will appear like one single connected component consisting of two apples, see the top part of Fig.10(a). To divide the connected component into different objects, the following steps are executed: (1) find the distance transform of the image (Sonka *et al.*, 1999). (2) complement the distance transform, force non-object pixels to be –Inf; and (3) compute the watershed transform (Sonka *et al.*, 1999). Figure 10 shows the different steps to divide touching apples.

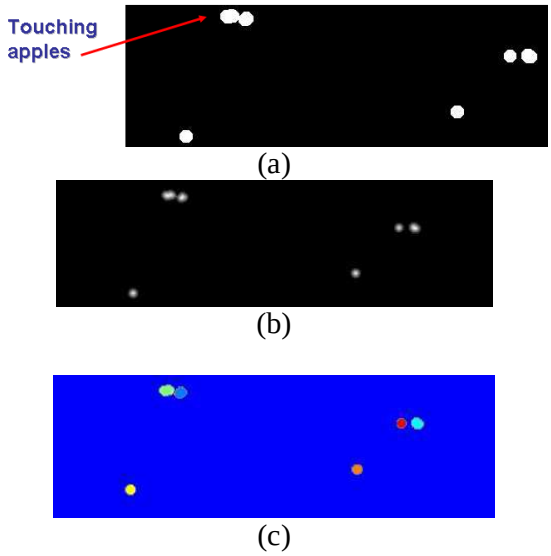


Figure 10. Apple detection process ; (a) BW image with touching apples; (b) distance transform image; (c) apples after Watershed operation

This sequence of operations yielded a set of BW images, where each image corresponded to one discrete radius from the range of radii searched. Since we knew the radii corresponding to each image, the area of the circle was calculated, and each blob was compared to this value. Moreover the following rule was applied: For each blob, check if the area is in the range $[\Pi r^2-\upsilon, \Pi r^2+\upsilon]$ where $\upsilon=(2\Pi r^2)/3$; otherwise discard the blob. The meaning of this rule is to verify that the area of the candidate apple is within a range. If the candidate is out of the range (too small or too big), then it is likely that the blob represents a leaf rather than an apple. All the blobs that passed this rule were stored in a buffer *B* together with the corresponding radii.

The data in the buffer *B* may contain more than one instance of an apple. This occurs when the radius of an apple is between two discrete consecutive radii. For example, an apple with radius 5.5 could be included in both images associated with radius 5 and radius 6. For this case, a grouping procedure was adopted: a hierarchical cluster tree based on the distance matrix between all the blobs was adopted. Blobs separated by a distance shorter than twice the minimum radius were grouped to one blob with the highest radius from the blobs in the same cluster.

Apple detection using Hough Transform

The apples were detected in this stage using the Hough Transform technique (HT) (Sonka *et al.,* 1999). The purpose of the technique was to locate instances of objects following a standard shape (circle or line) by a voting procedure.

Similar to the buffer *B* including the co-ordinates of each circle and its radius, a new buffer *B'* was created containing the co-ordinates of a bounding box surrounding each apple candidate, where the length of the box was the diameter of the candidate. A set of images was obtained by cropping sub windows twice the size of each bounding box. The value of two (twice the bounding box size) was found empirically for the best match between the Hough Transform (HT) and the ground truth. We found that a smaller value than two constrained the circle search procedure to the one found using the sifting approach, and a higher value than two increased significantly the running time of the HT.

The HT was applied to an image I_3 obtained from the logical union (OR) of two other images g_1 and g_2. The image g_1 was the result of the green channel (G) of the cropped image after applying the Canny edge detector, while g_2 was the result of the greenness channel (G) using Eq. (4) after applying the Canny edge detector (Sonka *et al.*, 1999). For the case of IR images, the edge detector was applied directly to the grayscale image.

$$g = 3\,G - (R + B) \tag{4}$$

For each cropped image I_3, the HT filter was applied where all the circles in the range $[r^*-\Delta, r^*+\Delta]$ were searched. The value of r^* was obtained directly from the buffer B for the corresponding candidate, and $\Delta=2$ was found empirically.
Once the circle was obtained, the bounding box was aligned so the circle was at the center of the box, and the HT was applied once again. Finally, the bounding box was aligned with the response of the last run of the HT. Convergence between the last circle found and the first indicated a strong candidate of an apple, while a divergence between the circles indicated a weak candidate. Figure 11 displays examples of true (converged HT) and false apples (not converged HT) detected using this approach in RGB images. This process corresponds to "Hough Transform" box in Figure 3. The number of false alarms could be reduced by adopting a more robust edge detector. This will be studied in future work.

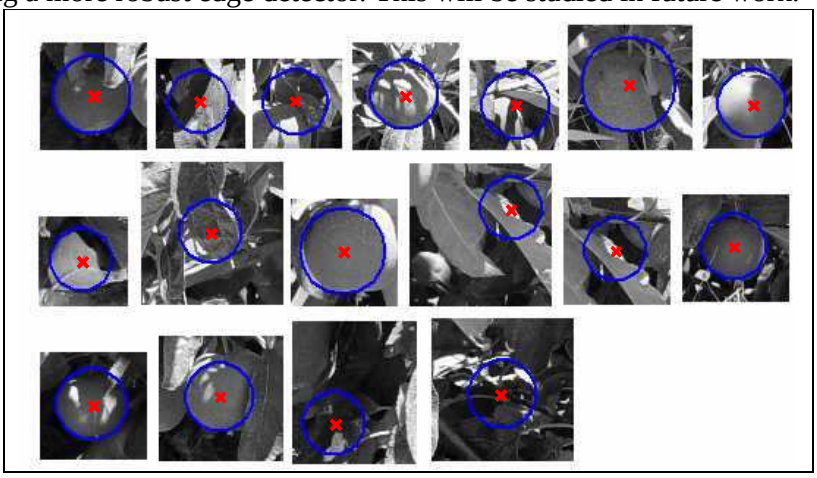


Figure 11. Responses obtained using the Hough Transform filter.

Method 2: Low-Level Visual Features

The approach in method 2 using low-level features consists of a sequence of registration, detection, color space voting and combining stages as shown in Fig. 12. This is a close-up view of box "method 2" in Fig. 3.

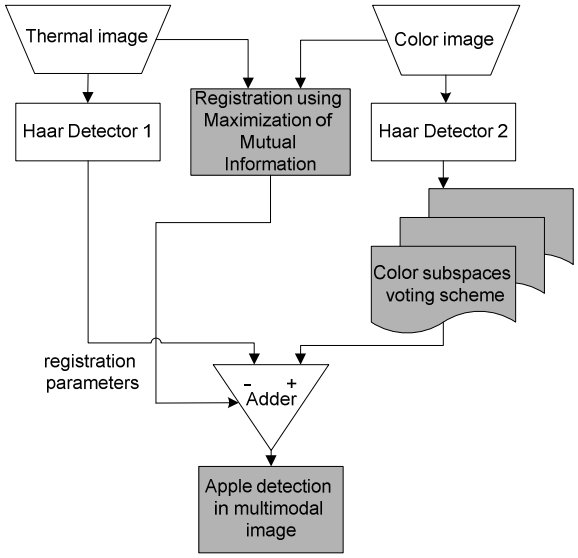


Figure 12. Flowchart of multimodal apple detection procedure

Apple detection is achieved using Haar classifiers (Viola and Jones, 2004), which are applied separately in color and thermal images. This step corresponds to the "Haar classifier" box in Figure 3.This classifier relies on features called Haar-like, since they follow the same arrangement as the Haar basis. The eleven basis features, i.e. edge, line, diagonal and center surround features, are presented in Fig. 13.

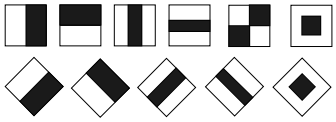


Figure 13. Eleven Haar features: edge, line, diagonal, center surround and rotated features

Since the number of features to be computed is quite large, integral images are adopted for fast computation (Lienhart and Maydt, 2002). A feature is detected when the computation of the weighted differences between the white and black areas of the rectangles (see Fig. 13) are higher than a threshold. This threshold is determined during the training process in such a way that the minimum number of samples is misclassified. The set of selected features is learned through a Classification and Regression Tree (CART) technique (Timofeev, 2004), which is a form of binary recursive tree. To achieve a given detection and error rate, a set of simple CARTS is selected through the Gentle Adaboost algorithm (Freund and Shapire, 1996).

In order to improve the overall performance of the classifiers, they are arranged in a cascade structure, where in every stage of the cascade, a decision is made whether the sub-window includes the object to be detected. At every stage, at least a high hit rate is assured, e.g., 0.995 and at least half of the false alarms are discarded. In spite of the hit rate and the reduction in false alarms, the hit rate decreases slower than the false alarms rate (FA). For example for 20 stages, since every stage keeps the hit rate to 0.995 at least, after 20 stages, the hit rate is 0.995^{20}=0.904. The false alarms rate (FA) is decreased in every stage so half of

the FA detections are rejected at every stage. For every stage, the classification function is learned until the maximum number of stages is reached or the minimum acceptable FA rate is obtained. For more details about this technique see Viola and Jones (2004).

Learning color sub-spaces using a voting scheme

In this section, separate artificial neural network classifiers are trained and tested for each of the three color spaces; L*a*b, HSV and RGB. Since, as we will show, the accuracies obtained for all the color spaces are identical, it was decided to see if a fusion method would provide any advantage. We will show that combining the output of the three classifiers as an ensemble by "majority voting" will decrease the false alarms without affecting the recognition rate. Thermal images are not considered here since their intensity information can lead to ambiguity between classes.

Training the classifiers

For each window obtained from the Haar detector in the RGB images, the hypothesis of whether the window is or is not an apple was subsequently tested. For this purpose, three classifiers of the type MLP (feed forward multi-layer perceptrons) (Werbos, 1990) were used. Each was trained and tested by splitting a sample set of vectors each of dimension three. The dataset was constructed using the following procedure: 1) user selected and manually labeled rectangular regions of interest (sub-windows) from the color image dataset according to 5 classes: apples, leaves, branches, sky and ground, and 2) each selected window was resized to 10x10 pixels and the values of each of the three channels of all pixels was stored as a set of 3D vectors. This process was repeated for three color models: L*a*b, HSV and RGB; and hence three datasets were obtained. Each classifier was trained and tested with a different dataset; therefore each classifier is used for one color space. The details of the datasets are given in Table 2. There were 3 such data sets, one for each of the color models.

Table 2. Dataset used to train the classifiers

Class	Sub-windows	Pixels
1 – apples	1416	141600
2 – leaves	2263	226300
3 - branches	1535	153500
4 – sky	1583	158300
5 – ground	714	71400
All	7511	751100

Each classifier had the same topology: 3-layer perceptron with 3 inputs, 5 outputs and two hidden layers including 100 neurons each. A symmetrical sigmoid activation function was used $f(x)=\beta*(1-e^{-\alpha x})/(1+e^{-\alpha x})$ with α=0.66 and β=1.71. The training consisted of a maximum of 300 iterations resulting in accuracies of 0.784, 0.78 and 0.782 for training and 0.782, 0.78 and 0.78 for testing, for the L*a*b, HSV and RGB classifiers respectively. Since the accuracy values obtained using different classifiers were the same, in the next section, a fusion approach is tested to see if an improved solution can be obtained.

Majority voting in classifier combination

One possible way of combining the output of the three classifiers is in an ensemble that is called "majority voting". For a given triplet of values $z=\{z_1,z_2,z_3\}$, we define a classifier B_i that responds with an output vector y_i such that the entry $y_{ij}=1$ if z is classified as class j, otherwise 0. In our case, $i=1,..,3$ (three classifiers) and $j=1,..,5$ (five classes). We define another type of classifier $D_{i,}$ that produces an output vector $[d_{i,1},\ldots,d_{i,c}]$ where the value $d_{i,j}$ represents the base to the hypothesis that the sub-window w being tested on classifier i belongs to class j, and $c=5$. Each measurement level $d_{i,j}$ can be obtained by Equation 5. Note that the classifier B_i respond to an individual pixel with a vector of values 1's and 0's, while the classifier D_i respond to a set of pixels (a sub-window) with a vector of values between 0 to 1.

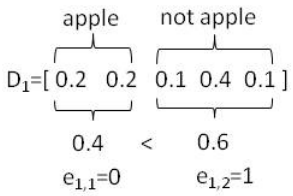
$$d_{ij} = \frac{1}{|w|}\sum_{z\in w} B_{ij}(z) \qquad (5)$$

Where w is a sub-window, and $|w|$ is the number of pixels in the sub-window. For example, for sub-window w_1, the response vector D_1=[0.2 0.2 0.1 0.4 0.1] means that 20%, 20%, 10%, 40% and 10% of the pixels in the sub-window belong to classes "apples", 'leaves", "branches", "ground" and "sky" respectively. However, to discriminate between true hits and false alarms, it is enough to classify the sub-window into two classes "apple" and "not apple". Note that the objective in this step is to re-group the different classes in two main categories only. Therefore vector $[d_{i,1},\ldots,d_{i,c}]$ can be converted to a binary two dimensional vector $[e_{i,1},e_{i,2}]$ such that:

$$e_{i,1} = \begin{cases} 1, & if \ \sum_{j=1}^{k} d_{ij} > \sum_{j=k+1}^{c} d_{ij} \\ 0, & otherwise \end{cases}, \qquad (6)$$

$$e_{i,2} = \bar{e}_{i,1}$$

where k is the partition index between classes, c is the number of classes, and $\bar{e}$ is the boolean complement of e. For example, to consider the first two classes in one group (apple), and all the rest in a different group (not apple), $k=2$, $j=5$ (number of classes) and $i=1$ (one classifier). Using the same example as before, we find that the given vector D_1 is not an apple.



If we consider the ensemble of $N=3$ classifiers ($i=1,\ldots,3$) then the label of the sub-window is given by a majority voting scheme presented in Fig. 14. The label set l includes the final categories that we consider (e.g. l={apples, not apples}) of the sub window detected by the 3 classifiers. Then, the label l_* of the sub-window is found using Eq. 7.

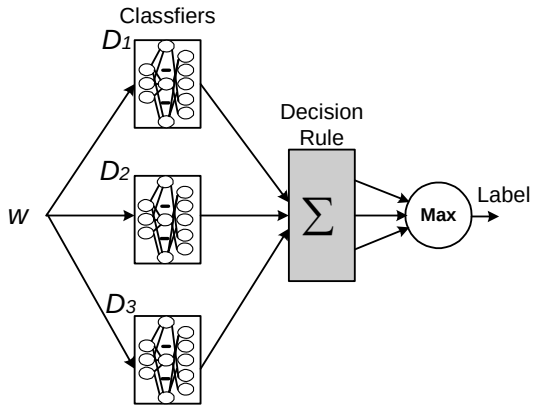


Figure 14. Classification combination scheme

$$l^* = \arg\max_{t=1}^{T}\left(\sum_{i=1}^{N} e_i\right) \qquad (7)$$

For example: if e_1=[0, 1], e_2=[1, 0] and e_3=[1, 0], then the sum in Eq. 7 results in [2, 1], and the maximum value 2 is found in first index (l*=1), corresponding to label "apples".

The majority voting scheme and another two criteria were used to accept or reject the hypothesis about whether the detected sub-windows were or were not apples. The two additional criteria were: a) the detected window does not include sub-windows, b) the detected sub-window size is smaller than k*median(W,H), where we used k=1.5.

Results & Discussion

The following subsections describe the performance of the multimodal apple detection system using two approaches: high and low-level visual features.

Results of Method 1: High-Level Visual Features

Image pre-processing, clustering and morphological operations described earlier were applied to a set of 173 color apple tree images including a total of 7222 green apples under field conditions. The same operations were applied to a set of 173 thermal infra-red images. The high-level visual features extraction methodology was used with the color and IR modality resulting in 38.88% and 50.6% detection accuracies, respectively. For the registered images, only 59 samples were considered since the rest of the images had a very small overlapping area between the two modalities to include enough apples. The intersection of the detected apples in the registered images yielded a 53.16% detection rate which is a small improvement with respect to the results in each modality alone (53.16% > 50.6%>38.8%) (see Table 3).

Even though the segmentation of the apples detected is very accurate – the full shape is recognized – about half of the apples in shade were missed, since their color distribution overlapped the distribution of the leaves. This indicates that in order to segment the apples under sunlight and shade illumination, an approach

relying on shape descriptors would be more appropriate rather than heavily relying on color information. This was the baseline for the Method 2: "Low-Level Visual Features"

Table 3. Detection rate using the color and thermal infra-red

	Apples	Hits	Hit Rate	FA	Dataset
RGB	9654	3746	38.80%	3552	173
IR	2371	1199	50.60%	1020	173
Together	2709	1440	53.16%	1654	59

The main disadvantage of this approach is that it is based on a color distribution model (or grayscale distribution model) which assumes that the majority of apples in the image are under the same illumination conditions. The assumption does not hold for images including both apples covered by leaves or totally exposed to sunlight.

Results of Method 2: Low-Level Visual Features

To train the RGB detector, a set of 146 color images of apple trees was used which included a total of 9420 green apples under natural conditions. The classifier was tested on 34 images including 1972 apples[2]. There were 30 stages in the detector's cascade, where each stage reached a hit rate of 0.995 with two splits, and its base resolution was 20x20 pixels. Fig. 15 shows the detections found in a sub-region of a testing image.

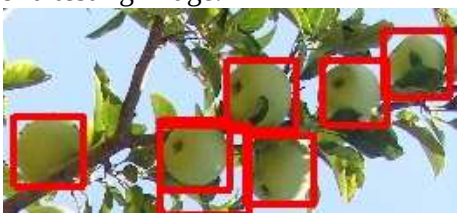


Figure 15. Six apples detected by the RGB Haar detector

The figure shows the classifier's ability to generalize apples (e.g. partially occluded with leaves, non-occluded, pits showing or not). False alarms were reduced using the voting scheme in the classifier combination presented earlier. Table 4 presents the hits over the total number of apples, the missed apples over the total number of apples and the false alarms when using the RGB Haar detector alone (single color space) and after adding the voting scheme (multiple color spaces). The voting scheme affected the correct detections only by less than 0.8% while dropping the FA rate by (536-498)/536=7%.

Table 4. Detection rate using the color Haar detector with and without the voting

	Hits	Missed	FA
RGB Haar	67.24%	32.76%	536
RGB Haar+Voting	66.28%	33.72%	498

[2] The total number of images used is 146 (training) + 34 (testing) = 180 images. This set of images was described in the beginning of the "Materials & Methods" section.

The apple detector classifier with IR images was trained with a training set of 286 images including 2330 apples from the same trees used to train the RGB Haar detector[3]. Due to the lower resolution of the thermal camera, the area captured by the image is much smaller, and hence contained fewer apples. This classifier was trained with a cascade of 20 stages, with a minimum hit rate of 0.995 in each stage, with two splits and a base resolution of 24x24 pixels detection window. Fig. 16 shows apples detected in an IR sub-image.

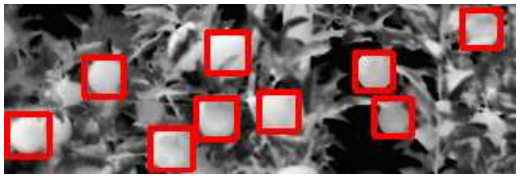


Figure. 16. Nine apples detected by the IR Haar detector

The performance of this detector is given in Table 5 for stages 17-20. For each stage, the total number of apples, the total hits and the false alarms are presented. These results show the dependency between hit rate and false alarms. The cascade with 18 stages was used for the experiments. More stages decrease significantly the hit rate, while increases yield a drastic improvement in the FA.

Table 5. Hit rate and false alarms per stage of the Haar detector

Stage #	Hits	Missed	FA
17	54.37%	45.60%	80
18	**52.18%**	**47.82%**	**61**
19	48.61%	51.39%	51
20	45.83%	54.17%	47

The detected hits resulting from the voting scheme were added to the output of the IR Haar detector after applying the registration parameters (see section "Materials & Methods"). First, the registration parameters for each pair of images (color and IR) were found using mutual information. Then, the RGB and IR Haar detectors were applied to the color and thermal infra-red images respectively. Later, the affine transformation was applied to the set of detections obtained using the IR Haar detector. Finally, the total number of detections was the sum of those found in the RGB and IR cases. The apples considered for the detection in this step are those found in the common area between the color and IR images.

[3] We found that using 146 IR images for training did not include enough apples. The number of apples in the IR images is much lower than the RGB, since the resolution is about 10 times lower. Therefore, a detector trained with approximately 10 times less apples would have had a disadvantage with respect to the RGB detector. Therefore we decided to increase the dataset to another 106 images to have a comparable number of apples between the two detectors.

Table 6. Performance when using single and combined modalities

Modality	Hits	Missed	FA
Color+Voting	66.28%	33.72%	498
IR	52.18%	47.82%	61
Combined	74.37%	25.63%	344

The results are presented in Table 6 for 34 pairs of testing images. The combination approach shows that the recognition accuracy was increased (74%) compared to the conventional approach of detection using either the color (66%) or the IR (52%) modalities alone. We compared our results to similar methodologies and found that Mao *et al.*, (2009) used a similar approach, but using an RGB color modality only. They obtained 90% accuracy, but detected red apples only (which is very easy to discriminate from leaves), and only 10 images were tested. Zhao *et al.*, (2005) was one of the few works found on green apple recognition, however recognition accuracy was not reported nor were the number of images used. Tabb *et al.*, (2006) reported 93.04% recognition on yellow and red apples. Baeten *et al.*, (2008) achieved 80% on detection; however, neither the sample size nor the apple color were given. We are not aware of other research published (Jimenez *et al.*, 2000b) tackling the problem of green apple recognition in non-staged environments with better results reported, using a significant size dataset.

One interesting feature of our methodology is that the three main processes: registration, Haar feature detection in RGB and IR are independent and hence can be easily run in parallel by assigning each process to a different CPU.

Although, neither one of the approaches succeeds in detecting all apples and reporting the number of false alarms, the second approach is suitable for implementation in a robotic fruit picking scenario, since it is robust enough for pre-positioning a robot picking arm. Since images will be acquired from cameras mounted on the robotic arm which can be oriented to take close up pictures, gradually all the apples in the tree can be found and false alarms can be identified as the robot arm explores the canopy.

Conclusions

We presented two approaches to address the problem of apple detection in tree canopies under uncontrolled conditions. First, optimal registration parameters were obtained using the maximization of mutual information method. This measure performed better than the other four methods tested. Once the images were registered, two approaches were discussed. The first approach, high-level visual features, used image pre-processing, clustering and morphology operations to recover the shape using mainly color and geometric properties. These operations were applied simultaneously to color and thermal images and the final detection was the intersection of detection sets over the registered image. This approach showed poor performance due to unconstrained illumination in natural scenes (53.16% recognition accuracy). To alleviate this problem, a second approach based on low-level features, which capitalize on Haar wavelet responses, was implemented. Haar features in color and thermal infra-red images were obtained through an Adaboost algorithm. Later, a voting scheme was used to improve further the detection results (66.28% recognition accuracy). Finally, the detection results were fused after applying the best transformation found in the

first step using a voting scheme. The voting scheme decreased the false alarms with little effect on the recognition rate. The resulting classifiers alone could partially recognize the on-tree apples. However when combined together, the recognition accuracy was increased (74.37% recognition accuracy). Even though the approaches presented could not detect the full cadre of apples, the second approach showed satisfactory performance for a robotic fruit-picking scenario.
Future work will include increasing the robustness of the Haar classifiers by increasing the sample set, incorporating morphological information to the voting scheme, and sensitivity analysis studies to assess the effect of the image processing parameters in the recognition accuracy.

Acknowledgments

This research was supported by Research Grant No US-3715-05 from BARD, The United States - Israel Binational Agricultural Research and Development Fund, and by the Paul Ivanier Center for Robotics Research and Production Management, Ben-Gurion University of the Negev.

References

Annamalai, P. and Lee, W.S. (2003). Citrus Yield Mapping System Using Machine Vision. Paper number 031002, ASAE, St Joseph, MI, USA.

Baeten, J., Donné, K., Boedrij, S., Beckers, W., Claesen E. (2008) Autonomous Fruit Picking Machine: A Robotic Apple Harvester. Springer Tracks in Advanced Robotics, 42:531-539.

Brown, L. G. 1992. A survey of image registration techniques, ACM Computing Surveys (CSUR) archive, 24(4): 325 – 376.

Cha, S.-H.; Srihari, S. N. 2002. On measuring the distance between histograms. Pattern Recognition 35, 1355-1370.

Fernandez-Maloigne, C., Laugier, D. and Boscolo, C. (1993). Detection of apples with texture analyses for an apple picker robot. Intelligent Vehicles '93 Symposium: pp. 323-328. ISBN: 0-7803-1370-4

Freund Y. and Shapire, R.E. (1996). Experiments with a new boosting algorithm. In Machine Learning: Proceedings of the 13th International Conference, pp. 148-156. Bari, Italy, July 3-6, 1996. Morgan Kaufmann.

Hunter, R. S. (1948). Photoelectric Color-Difference Meter. JOSA 38 (7): 661. Proceedings of the Winter Meeting of the Optical Society of America.

Jiménez, A.R., Ceres, R. and Pons, J.L. (2000a). A vision system based on a laser range-finder applied to robotic fruit harvesting. Machine Vision and Applications. 11(6) 321-329.

Jiménez, A.R., Ceres, R., and Pons, J.L. (2000b). A survey of computer vision methods for locating fruit on trees. Transactions of ASAE 43(6): 1911-1920.

Lienhart, R. and Maydt, J. (2002). An Extended Set of Haar-like features for Rapid Object Detection. In Proceedings of the IEEE Conference on Image Processing (ICIP ’02), 155-162.

Mao, W., Jia, B.,Zhang, X. and Hub, X. (2009) Detection and Position Method of Apple Tree Image. Computer and Computing Technologies in Agriculture II, 2.

Sapina, R., 2001. Computing textural features based on co-occurrence matrix for infrared images. In: Proceedings of 2nd International Symposium on Image and Signal Processing and Analysis, 2001. ISPA 2001. 373-376. Eds: Loncaric and Babic, University Computing Center, University of Zagreb, Croatia.

Sonka, M., Hlavac, V., and Boyle, R. (1999) Image Processing, Analysis, and Machine Vision, 2nd Ed., Brooks/Cole Publishing, CA, USA, 256-260.

Stanjnko, D., Lakota, M. and Hocevar, M. (2004). Estimation of number and diameter of apple fruits in an orchard during the growing season by thermal imaging. Computers and Electronics in Agriculture 42:31-42.

Tabb, A.L. Peterson, D.L., and Park, J. (2006). Segmentation of Apple Fruit from Video via Background Modeling. Paper number 063060, ASABE, St Joseph, MI, USA.

Timofeev, R. (2004). Classification and regression trees (cart) theory and applications. Master's thesis, Humboldt University Berlin.

Viola, P. and Jones, M.J. (2004). Robust real-time face detection. International Journal of Computer Vision, 57(2):137-154.

Viola, P. and Wells, W.M. III, (1995). Alignment by maximization of mutual information. In Proceedings of 5th International Conference on Computer Vision, pp. 16–2. June 20-23, 1995, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA. IEEE Computer Society.

Wachs, J. Stern, H., Burks, T., and Alchanatis, V. (2009). Multi-modal Registration Using a Combined Similarity Measure. Applications of Soft Computing: Updating the State of the Art Series: Advances in Intelligent and Soft Computing, Avineri, E.; Köppen, M.; Dahal, K.; Sunitiyoso, Y.; Roy, R. (Eds.), Springer, Warsaw, Poland, 52:159-168.

Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it. Proceedings of IEEE 78 1550–1560.

Zhao, J., Tow, J. and Katupitiya, J., (2005). On-tree fruit recognition using texture properties and color data. IEEE/RSJ International Conference on Intelligent Robots and Systems, 263- 268, Aug. 2-6, Alberta, Canada, IEEE Robotics and Automation Society.