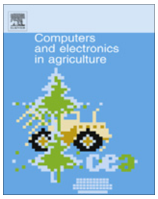
Computers and Electronics in Agriculture 81 (2012) 45–57

Contents lists available at SciVerse ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag



Determination of the number of green apples in RGB images recorded in orchards

Raphael Linker [a,*], Oded Cohen [a], Amos Naor [b]

[a] *Faculty of Civil and Environmental Engineering, Technion – Israel Institute of Technology, Haifa 32000, Israel*
[b] *Golan Research Institute, P.O. Box 97, Katzrin 12900, Israel*

ARTICLE INFO

Article history:
Received 28 November 2010
Received in revised form 9 November 2011
Accepted 19 November 2011

Keywords:
Image processing
Computer vision
Fruit recognition

ABSTRACT

This work details the development and validation of an algorithm for estimating the number of apples in color images acquired in orchards under natural illumination. Ultimately, this algorithm is intended to enable estimation of the orchard yield and be part of a management decision support system. The algorithm includes four main steps: detection of pixels that have a high probability of belonging to apples, using color and smoothness; formation and extension of "seed areas", which are connected sets of pixels that have a high probability of belonging to apples; segmentation of the contours of these seed areas into arcs and amorphous segments; and combination of these arcs and comparison of the resulting circle with a simple model of an apple. The performance of the algorithm is investigated using two datasets. The first dataset consists of images recorded in full automatic mode of the camera and under various lighting conditions. Although the algorithm detects correctly more than 85% of the apples visible in the images, direct illumination and color saturation cause a large number of false positive detections. The second dataset consists of images that were manually underexposed and recorded under mostly diffusive light (close to sunset). For such images the correct detection rate is close to 95% while the false positive detection rate is less than 5%.

1. Introduction

Early estimation of the expected yield has always been a major challenge in agriculture. In orchards, such predictions are of interest to growers, packing and storage houses, and compensation funds. For growers, yield forecast would be valuable at several stages during the season in order to optimize the intensity of chemical and manual thinning. Currently, in the absence of accurate yield forecast, growers tend to perform insufficient chemical thinning at an early stage of fruit development, preferring to be on the safe side rather than to cause irreversibly low yield. This has significant repercussions later in the season when labor-consuming hand thinning is required to compensate for the insufficient chemical thinning. In addition, hand thinning is performed only after the process of natural thinning has ended and the fruit has reached a size where experienced growers can roughly estimated yield by visual inspection. This delay has two serious horticultural consequences: (1) it results in smaller fruit at harvest, which, in addition to having a lower market value, require more time for harvesting, and (2) high crop load before the late thinning reduces flower bud initiation which lowers the potential bloom in the subsequent season. For packing and storing houses, accurate yield forecast is needed toward the middle of the season in order plan adequate allocation of storage spaces. Finally, the compensation funds need a yield estimate early in the season when unexpected fruit drop or hail damage may occur.

To date, yield forecasts are based on manual counting of the number of apples on selected trees. This method is extremely time-consuming and the small number of trees that can be inspected is insufficient due to the high variability of crop yield that occurs in apple orchards. More accurate forecasts of yield will enable growers to perform more adequate early chemical thinning, which will minimize the need for hand thinning, will increase the fruit's size and will increase the flowering potential in the subsequent season.

As a first step toward the development of such a system for estimating orchard yield, the present study focuses on the development of a color imaging system for automatic estimation of the number of apples present on a tree. The more general task of fruit localization on trees using artificial vision has been investigated in numerous studies, and most of these are summarized in the excellent review of Jimenez et al. (2000). These studies used either standard color cameras as in the present study (e.g. Annamalai and Lee, 2003; Bulanon et al., 2001; Pla et al., 1993; Tabb et al., 2006; Zhao et al., 2005), multispectral or hyperspectral imaging (e.g. Kim and Reid, 2004; Safren et al., 2007), thermal imaging (e.g. Stajnko et al., 2004) or a combination of these methods (Wachs et al., 2010). As noted in some of these studies, the task

[*] Corresponding author. Tel.: +972 4 8295902; fax: +972 4 8228898.
E-mail addresses: linkerr@tx.technion.ac.il (R. Linker), oded@odedcohen.com (O. Cohen), amosnaor@research.haifa (A. Naor).

0168-1699/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.compag.2011.11.007

is especially challenging under natural illumination, and under such conditions shading and high contrast transitions are a major problem. For instance, sunlit fruit is 10 times brighter than shadowed leaves, and sunlit leaves are four times brighter than shadowed fruit (Kim and Reid, 2004). When the fruit of interest is green as in the present study, the task is made more complex by the low color contrast between the background and the fruit.

2. Problem description and challenges

Fruit detection for yield estimation presents a challenge that is different from fruit detection for robotic harvesting, and has some unique characteristics. First, in order to minimize problems related to image registration and to enhance performance, each image should contain a large portion of the tree, so that fruit appear as rather small objects in the image. Also, unlike for robotic harvesting, it is not possible to bring the camera closer to a specific suspected fruit, to illuminate it or to improve its image in any way. The envisioned system will consist of a geo-referenced tractor equipped with one or several cameras mounted sideways that will capture images on-the-fly while traveling at low speed. Clearly, with such an ‘‘on-the-fly’’ system there is no ‘‘second chance’’ of detecting partially hidden or shaded apples in other images photographed under more favorable conditions.

In ideal conditions, apples would appear as non-touching circular objects with smooth surface and distinctive color. However, such ideal situations are rarely found in images recorded in natural outdoor conditions (see Electronic Appendix). In such images, most apples are partially hidden by other apples and leaves, and their apparent color is influenced by their environment, the light conditions and the photography parameters. As a result, it is not possible to identify a single characteristic (for instance color or hue) that would enable reliable differentiation of apple pixels from leaf pixels, or apple contour parts from leaf contour parts. In particular, regarding color it must be noted that the color of young leaves is similar to that of apples and that apple surface appears darker, almost colorless, near the shaded margins. Furthermore, as apples constitute about 4% of the image, and leaves over 60%, naïve color-based classification leads to poor detection performance. With respect to contours, since most of the apples are overlapping, partially hidden or shaded, only part of their typical circular contour is visible. Furthermore, circular contour sections can be found occasionally on leaves as well.

Light conditions have a major influence on detection feasibility: the most problematic factor is direct sunlight which results in saturated objects without color information and in shadows that cause standard segmentation procedures to split the apples surfaces into several fragments. Such undesirable effects can be minimized by acquiring the images under diffusive light conditions (for instance, by taking pictures close to sunset) and by manually underexposing the images compared to the camera’s automatic setting. Lowering the amount of light that reaches the sensor in such a manner brings the apples to the middle of the dynamic range of the sensor and avoids saturation. Fig. 1 shows a typical image obtained with such optimal photography parameters. Although to the human eye such an image may seem too dark and ‘‘flat’’, it is much more suitable for computer analysis than a bright and shiny image. Another factor that must be taken into account is that the color of an object is influenced by the light reflected from surrounding objects. In the present case, apples that are deeper within the tree do not only appear to be darker, but are also more similar in color to the leaves surrounding them (see Fig. AE4 in Electronic Appendix).

3. The data

Images were recorded in a Golden Delicious orchard in the Matityahu Research Station located in Northern Israel. The images were taken during two consecutive seasons during the months of June–August. All the pictures were taken under natural daylight conditions.

A first set of images was taken using the fully automatic mode of operation of the camera (Fujifilm FinePix S8000fd). Eight of these images were selected as ‘‘calibration images’’, which were used to develop the algorithm, and another nine were selected as ‘‘validation images’’. About 70% of the apples present in the calibration images were marked manually to provide calibration pixels of apple objects.

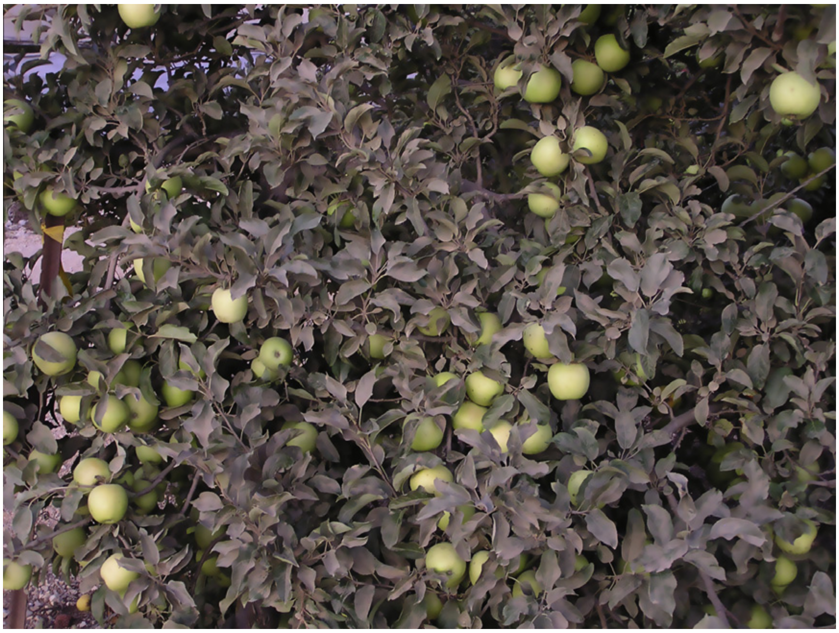


Fig. 1. Typical image acquired close to sunset with manual underexposure.

A second set of images was taken in the following season with a different camera (Olympus C740UZ) and new photographic parameters in an attempt to overcome some of the problems identified when analyzing the first set of images:

1. The camera shutter was set manually to 0.7 units below the automatic camera setting.
2. The pictures were taken close to sunset, when there were nearly no remnants of direct sunlight and lighting was diffusive.
3. The images were taken at a constant distance from the trees, resulting in a more uniform size of the apples.

Nine of these images were used to calibrate the detection algorithm, which was then tested on another nine images.

In order to estimate the algorithm performance, a manual (“ground truth”) count of the apples whose sizes were in the range of the objects sought by the algorithm was performed for all images. While performing this count it became clear that when analyzing complex images containing tens of apples, the longer one inspects an image, the more one finds vague objects that may be apples (although it is difficult to reach a definitive conclusion), so that the process tends to be time-consuming and not entirely reliable. In order to estimate the inconsistency of a manual count performed within a reasonable amount of time, all images were also shown for 30 s to six graduate students who were instructed to count only “normal size” apples on the trees, i.e. not very small undeveloped apples or apples located on trees in the background or on the ground.

4. Algorithm description

As in any typical image-analysis application, only the initial data (in this case the raw image) and the desired result (in the present case the number of apples in the image) are specified and the choice of the specific steps that should be performed in order to produce the desired final result are left to the researcher. For non-trivial problems, several steps are required for progressing from the original image to the final result. At each step, intermediate objects representing increasingly significant parts of the final objects are detected. These intermediate objects, called below “building blocks”, represent identifiable portions of the target objects. Clearly, the decomposition of a target object into building blocks is not unique and depends on the specific properties selected and the order of the analysis. For the present study we selected three building blocks:

- **Pixels** that have a high probability of belonging to apples.
- **Blobs** (sets of connected pixels) that are expected to correspond to parts of apple surfaces.
- **Arcs** that are likely to belong to the apple perimeter.

The sizes of the building blocks should be such that they capture important details and minimize the influence of noise. Arcs, for instance, should be long enough to capture the objects’ curvature and to average out irregularities. On the other hand, these arcs should be small enough to process the smallest objects of interest. Hence at each stage the size of the building blocks is dictated by the expected size of the features to be detected. In the present case, we denote as $D_{\max}$ and $D_{\min}$ as the maximal and minimal diameters of the apples to be detected, respectively. We further assume that the smallest object that must be detected is an apple of diameter $D_{\min}$ that is 50% hidden. These values are used to parameterize the algorithm and building blocks as described in Sections 4.1–4.4.

The algorithm presented here deals with the challenges described in the Introduction by combining *color information, surface smoothness, contour shape analysis and heuristic object building* (Fig. 2).

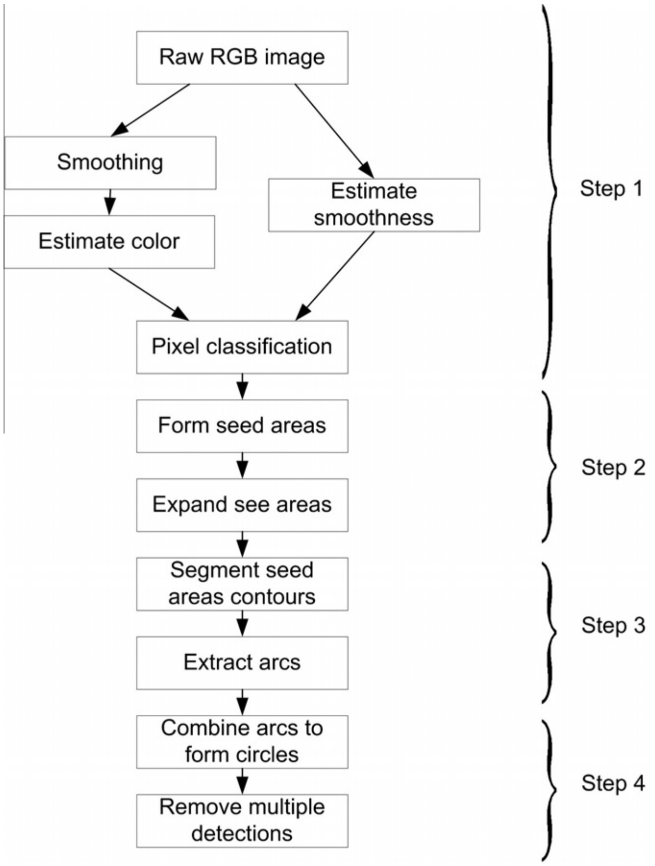


Fig. 2. Algorithm flowchart.

- Firstly, pixels that have a characteristic color and texture are detected. These consist mainly of apple shiny parts and young leaves.
- Secondly, the detected pixels are organized in blobs (sets of connected pixels), and each blob is extended to include neighboring pixels that are deemed similar enough. The main purpose of this stage is to append shaded and saturated areas, which do not have the required typical apple color, to the previously detected regions.
- Next, the contour of each extended area is segmented into arc sections and amorphous sections. The arc sections correspond mainly to apple perimeter sections and artifacts such as arc-like contour sections of leaves.
- Finally, the arc sections are grouped into circular objects, each one representing a single apple. A heuristic model is used to quantify the similarity between each object and the ideal “apple object”.

The algorithm was developed in the ImagingChef environment (a full description of this imaging development environment can be found at: www.odedcohen.com). This platform supported the entire development process, from marking of the reference objects to development of the algorithm and result visualization.

4.1. Step 1 – Detection of apple pixels by color and texture

The purpose of this first step is to detect pixels that have a high probability of belonging to an “apple object”. Such pixels form

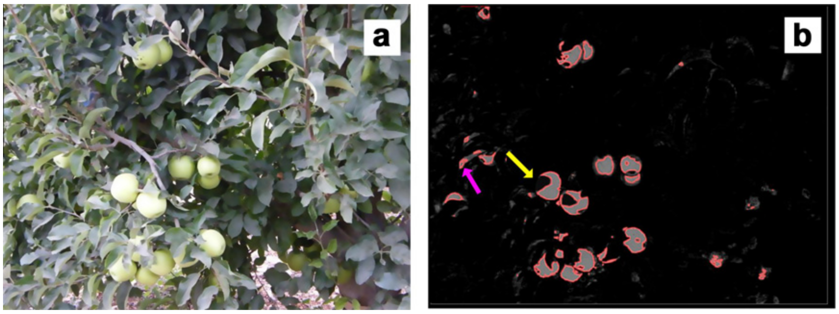


Fig. 3. (a) Detail of an image. (b) Result of the color and texture classifier in which the lighter pixels have a higher probability of belonging to an apple. The red contours correspond to probability of 50%. The yellow and purple arrows point to misclassified areas that correspond to sensor saturation and leaf underside. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

"seed areas" from which detection of whole apples will ensue. The implementation of this step requires (1) choosing the descriptors (features) used to describe each pixel and (2) choosing the type of classifier.

4.1.1. Choice of the pixel descriptors

Most apples have a visible area in which their color and smooth surface can be observed clearly. Hence for classification purposes, each pixel is characterized by both its color and smoothness, where smoothness is calculated as the local color variance in a square region of R_S by R_S pixels centered at the pixel of interest. R_S should be small enough to minimize the inclusion of neighboring objects, but large enough to contain the texture details of the object. The choice of R_S is further discussed in Section 5.1. For classification purposes smoothness is treated as an additional color component "S" where $S = 0$ corresponds to perfect smoothness. Close examination of apple surfaces shows local color fluctuations at the pixel level. These fluctuations are caused by camera noise as well as by natural color variations (light and dark spots on the apple surface). Therefore, when measuring the color for classification purpose, better results are achieved by measuring the average color on a small region of R_C by R_C pixels rather when using the raw pixel data. The considerations for the choice of R_C are similar to those of R_S, namely minimizing the inclusion of neighboring objects while averaging out the noise (see Section 5.1). An example of this noise and the result of the filtering operation are included in the Electronic Appendix (Fig. AE5).

4.1.2. Choice of the classifier type

The first decision in classifier selection is whether to select a parametric or a non-parametric classifier. A parametric classifier is the preferred choice when distributions can be described mathematically using statistical parameters, and apple color seems a natural candidate for such representation. However, apple pixels are only ~4% of the total image and there are many other types of objects (ground, sky, rocks, branches, leaves, etc.), each with its own color distribution. It is impractical to match a color distribution to each type of object, and impossible to match a single parametric distribution to the entire non-apple objects. Therefore, a parametric classifier is not suitable for the present task and a non-parametric classifier is more appropriate. We use a k-nearest neighbors (KNN) classifier that uses data stored in a KD-tree (Moore, 1991). The KD-tree is a data structure used to store and manipulate sets of points in multidimensional space. It is especially useful for non-parametric classifiers that have to handle large numbers of samples. In order to create the KD-tree, the K-dimensional space is split recursively into smaller cells, hence creating a hierarchy of sub-cells and each cell stores only the samples that fall within the respective range. The ending sub-cells are termed "leaves" while the others are termed "branches" of the tree. Several KD-tree variants exist with different split and data-storage policies which can be defined in an ad hoc fashion in order to minimize memory usage and access time for the specific problem at hand. In the present application, the sample set was very large (~3 million points per image) and fast classification was crucial since this operation had to be performed for each pixel in the image. Therefore our KD-tree did not store the samples themselves, but only their count in each leaf (i.e. the number of "apple" and "non-apple" pixels with color and smoothness values that fall within the range of the leaf). In this manner the classification process of each pixel was reduced to accessing the corresponding leaf and checking how many samples of each population it contained, which determined the apple probability P_A of the pixel. Attempt to access a non-existing leaf (i.e. a color and smoothness not encountered in the calibration set) returned a "non-apple" result.

"Apple" objects were marked manually in the images belonging to the calibration sets, and pixels from these "apple" and "non-apple" objects (unmarked pixels) were used to create the KD-tree. Smoothness was rescaled to values similar to the RGB attributes (0–255) so that the initial cell of the KD-tree had a length of 256 in each dimension. During the calibration process, this cell and all subsequent sub-cells were split into 16 "children" by splitting each dimension in the middle. Splitting was performed recursively five times, so that the leaf sub-cells had length 8 in each dimension (further splitting was investigated but increased the computation burden considerably without noticeable improvement of the classification results). Only leaves that contained data were actually created, and at the end of the calibration stage the KD-tree contained only approximately 15% of the potential number of leaves (16^5). Furthermore, about 40% of the existing leaves contained only non-apple objects.

A typical result of the classification procedure is shown on Fig. 3, in which the lighter pixels have a higher probability of belonging to an apple object. Two typical problems can be observed in Fig. 3:

1. Saturated (bright) areas are associated with low apple probability, as their color is practically white (e.g. the area indicated by a yellow arrow).
2. Pixels that belong to the underside of leaves are incorrectly recognized as "most probably apple" since this side of the leaves is lighter and its color is similar to that of apples (e.g. the area indicated by a pink arrow).

4.2. Step 2 – Apple surface detection by seed area detection and growth

In this step connected sets (blobs) of apple pixels are detected and extended to cover the apple area to which they belong.

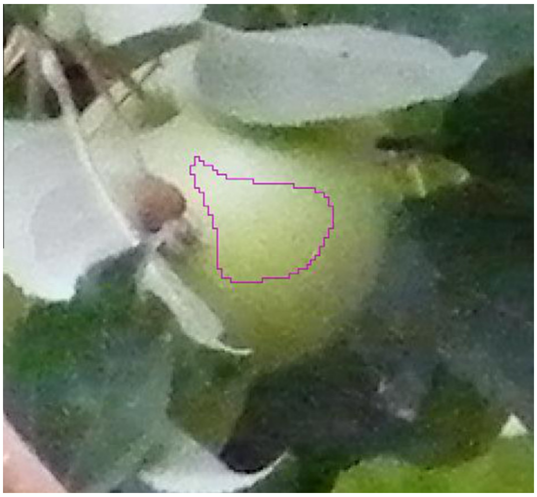


Fig. 4. Detail of an image in which the contour of the detected seed area is shown.

Initially, a seed area consists of connected pixels for which $P_A(x) = P(x \in A \mid C_x) \geq \tau$, where C_x denotes the "color and smoothness" feature vector RGBS as described in Section 4.1 and τ is a user-defined threshold. Ideally, each apple would result in a seed area that covers most, if not all, of its surface. In practice, light reflectance varies considerably across the apple surface and its edges might be considerably darker. Such areas have less color information, and are initially misclassified as "non-apple", resulting in a smaller seed area. Fig. 4 shows a seed area that includes only a small part of an apple due to low light reflection. To compensate for this initial misclassification, each seed area is subsequently expanded by annexing similar neighboring pixels. This is done by re-classifying pixels close to a seed area as belonging or not belonging to this seed area. Rather than building a dedicated classifier for this task, we use the color classifier described in the previous section with a small adaptation: we assume that the feature distributions of the two populations (apple and non-apple pixels) remain the same, but their relative part in the overall population changes. Namely, we assume that the relative part of apple pixels increases in regions closer to seed areas. Indeed, inspection of images such as Fig. 4 shows that the *a priori* probability for a pixel to belong to an apple is much higher for pixels that are close to detected seed areas. Assuming that the feature distributions remain the same and only the ratio between the populations is changed, the probability of a pixel x to belong to the seed area ψ_i can be calculated as:

$$P_{\psi_i}(x) = \frac{P_\psi(x) \cdot F}{P_\psi(x) \cdot F + (1 - P_\psi(x))}$$

where $P_\psi(x)$ is the initial probability of the pixel x to belong to an apple object (calculated by the color and texture classifier in previous section), and F is the factor by which the ratio between the apple population and the non-apple population is increased. F can be calculated as:

$$F = \frac{\frac{\hat{P}_{a_s}(x)}{1-\hat{P}_{a_s}(x)}}{\frac{\hat{P}_a(x)}{1-\hat{P}_a(x)}}$$

where $\hat{P}_a$ is the *a priori* probability for apple pixels in the whole population (set to 0.03 based on inspection of the images), and $\hat{P}_{a_s}$ is the *a priori* probability for apple pixels close to a seed area. By inspecting the images, we determined that $\hat{P}_{a_s}$ can be expressed as a function of the distance D to the seed area and the equivalent diameter of the seed area d_{ψ_s} which is calculated as the area of the seed area divided by π:

$$\hat{P}_{a_s}(D) = \frac{\hat{P}_0(x)}{1 + \frac{D}{d_{\psi_i}} \cdot \frac{1}{r_{0.5}}}$$

where $\hat{P}_0$ is the *a priori* probability of a pixel which is on the boundary of a seed area to belong to an apple object and $r_{0.5}$ is the ratio $\frac{D}{d_{\psi_i}}$ at which the *a priori* probability for apple decreases by half (set to 0.25 based on inspection of images such as Fig. 4).

The expansion of the seed area is performed by a dilation operation using a circular structuring element with variable size. The radius of the structuring element is set at each contour point to the maximal value that will include only pixels for which $P_{\psi_i}(x) \geq \tau$. Since the structuring element should not infringe into neighboring apples, the structuring element is further limited so that it does not include pixels with variance (calculated in a R_S by R_S window) greater than a threshold value, defined as the variance for which the apple and non-apple probabilities are equal in the calibration data for the populations of pixels close to the apple edges.

Fig. 5a shows a variance map of Fig. 4, in which the darker pixels indicate higher variance. The seed area contour is drawn in purple, and the structuring element is marked at several points along the contour. Fig. 5b shows the results of the seed area expansion and it can be seen that dark areas that were excluded from the initial seed area have been annexed to it. A similar result for larger portion of an image is included in the Electronic Appendix (Fig. AE6).

4.3. Step 3 – Contour segmentation

A seed area may contain any combination of apple parts and leaves and as Fig. 6 demonstrates there is no one-to-one relation between seed areas and apples. Overall, the ideal situation depicted in Fig. 6a was observed for approximately 15% of the seed areas. The other typical situations correspond to multiple seed areas for a single apple due to partial occultation by a leaf or branch (Fig. 6b), leaf (or leaves) mistaken for apple (Fig. 6c), creation of a large seed area that encompasses two overlapping apples (Fig. 6d), and creation of a seed area that contains both apple and non-apple pixels (Fig. 6e). Therefore, it is necessary to segment the contours of the seed areas into smaller elements, from which apple objects can be recognized.

Except for trivial cases, segmentation does not have a unique correct result and the correctness of this operation should be measured according to the expected usage of its outcome. In particular, the final segments should be large enough to contain significant information, but small enough to be used as flexible building blocks. Since the contour of an ideal apple would be a smooth circle while the contour of a leaf is typically straighter but irregular, in this step local curvature and fuzziness are used to split a seed area contour into arcs (part of circle) and amorphous segments. In Step 4 below, the contours obtained by combining these arcs are compared to the model of an apple contour.

Dividing a contour into segments with low noise and uniform curvature requires calculating these properties locally at each contour point. The calculation is performed by approximating an interval of the contour with length L_{int}, in which the point of interest is centered, by a quadratic polynomial and calculating the following properties[1]:

[1] This simple procedure yields accurate results as long as the interval is completely within a segment. At points closer than $L_{\text{int}}/2$ to the segment endpoints this procedure must be slightly modified as detailed in Appendix A.

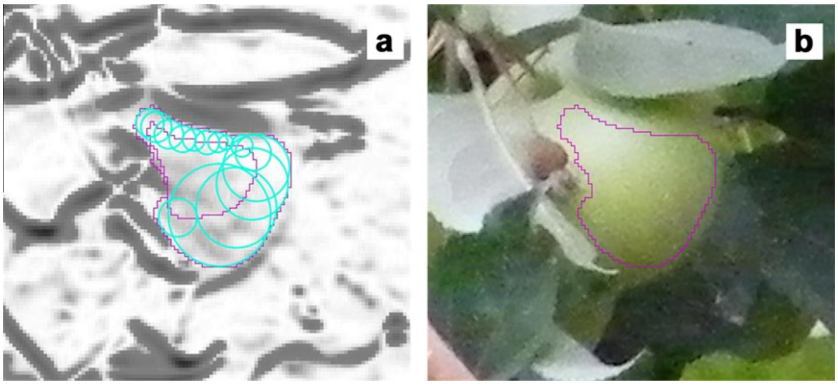


Fig. 5. (a) Variance map of Fig. 4 with the initial seed area, the structuring element at selected locations along the contour and the final contour. (b) Contour of the expanded seed area shown on the original image.

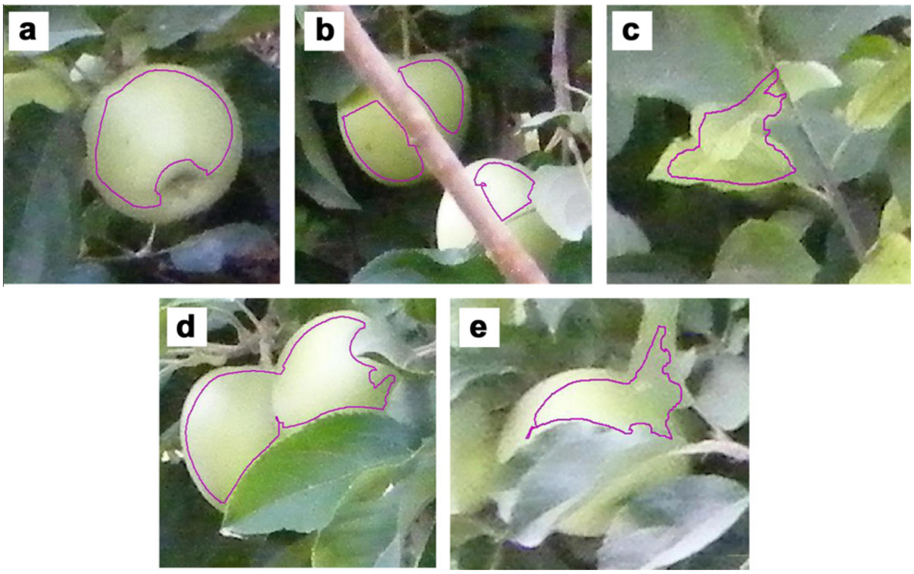


Fig. 6. Typical seed areas: (a) ideal case, (b) two seed areas in a single apple due to partial occultation, (c) leaf erroneously selected, (d) seed area covering two apples and (e) seed area that includes both apple and leaf regions.

1. Curvature, i.e. the polynomial second derivative.
2. Roughness, defined as the mismatch between the actual segment and the polynomial approximation (average of squared differences between the segment points and polynomial approximation).
3. Approximation error of the specific point, defined as the distance between the point and the polynomial approximation.

A point belongs to an arc if its curvature is within the range $[\varphi_{\min}, \varphi_{\max}]$, its approximation error is less than $E_{\max}$ and its roughness is below the threshold $Z_{\max}$.

The procedure is illustrated in Fig. 7 which shows the initial contour (Fig. 7a) and its approximation by a series of quadratic segments (Fig. 7b). It can be seen that the result is a smooth contour which preserves the segment endpoints, such as C_1 and C_2, which is important since these points define the shape of the individual segments.

4.4. Step 4 – Apple detection

In this step, the arcs obtained at the previous stage are grouped into apple objects. In order to do so, the following elements are required:

- A model that describes the features of an ideal apple object.
- A "combining algorithm" that creates potential apple objects from combinations of neighboring arcs.
- A function that quantifies the discrepancy between the combined segments and the model.

Following these steps, a final analysis must be performed to identify apples that were detected more than once.

4.4.1. Apple object model

The contour of an ideal apple object can be described reasonably well as a circle with a radius within a pre-defined range. Although such a heuristic model is clearly very simplistic and apples could be described more accurately using a more complex probabilistic model, such a probabilistic approach is not practical when, as in the present case, the inputs to the probabilistic model are influenced by a larger number of parameters at the previous stages of the analysis. For instance, lowering the probability threshold of the color classifier results in bigger initial blobs, which in turns affects the detected arcs and their size distribution. If a probabilistic model was used, such a change would require a full recalibration of the probabilistic model, i.e. manual tagging of the

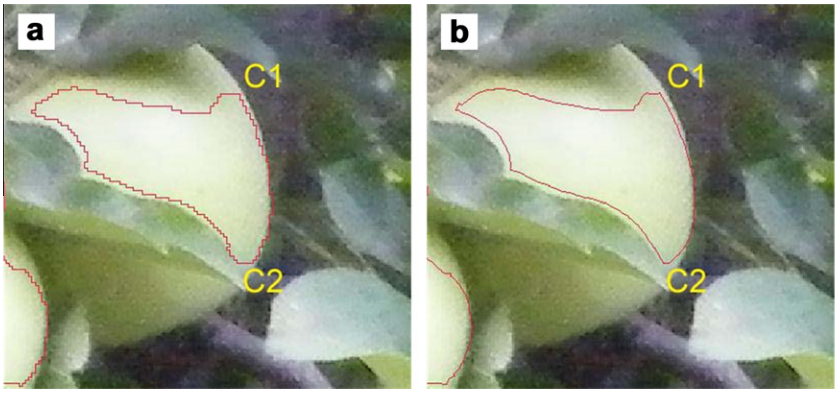


Fig. 7. (a) Detail of an image with original contour of seed. (b) Same image with contour approximated using arcs and amorphous segments.

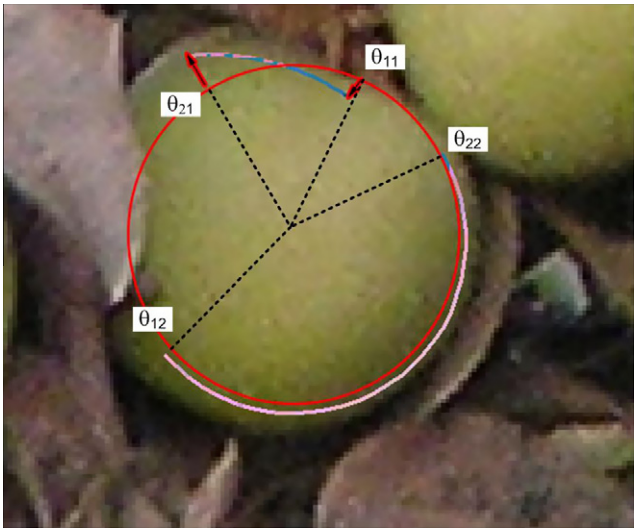


Fig. 8. Detail of an image with two detected arcs and fitted model (red circle). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

apple and non-apple arcs in all calibration images. By comparison, when using the heuristic model, this change only requires an adjustment of the expected apple contour radius.

4.4.2. Combining algorithm

There are typically only several hundreds of arcs in each image, and only a few local combinations make sense. Therefore the algorithm of selecting neighboring arc and grouping them into objects is a simple search that creates a list of all feasible combinations, sorted by ascending model mismatch. The list is processed in a "top-down" fashion, retaining only objects with small model mismatch and discarding objects that use arcs belonging to previous objects.

4.4.3. Calculation of the discrepancy between a detected object and the heuristic model

A set C of n arcs can be approximated by a circle A_c with radius ρ_c and its center located at (x_c, y_c). First, a simple test is applied to reject obvious (very large or very small) non-apple objects. When $R_{min} < \rho_c < R_{max}$, each arc C_i is associated with a section of the perimeter of A_c, denoted as P_i and which ranges from θ_{1i} to θ_{2i} (Fig. 8). In order to determine whether A_c indeed describes an apple, the mismatch function Δ_T between A_c and the ideal apple object A_m is calculated by combining four measures:

- The difference between the radii of A_c and A_m: $\Delta R = \rho_c - R_m$.
- The difference between the length of the detected contour and the contour of a fully visible apple: $\Delta V = 2\pi - P_V = 2\pi - \sum_{i=1}^{n}(\theta_{1i} - \theta_{2i})$, where P_V denotes the detected length of the segment (expressed in radians) (Fig. 8).
- The radial deviation Δr_i of each arc C_i from A_c: $\Delta r_i = \sqrt{\frac{\int_{P_i}(C_i(\theta)-\rho_c)^2 d\theta}{\theta_{2i}-\theta_{1i}}}$, where C_i is expressed in polar coordinates relative to the center of A_c. For instance in Fig. 8, where A_c is indicated in red, there is a small radial difference between the lower arc and A_c.
- The angular deviation Δa_i of each arc C_i from A_c: $\Delta a_i = \sqrt{\frac{\int_{P_i}\left(\frac{\partial C_i(\theta)}{\partial \theta}\right)^2 d\theta}{\theta_{2i}-\theta_{1i}}}$, where again C_i is expressed in polar coordinates relative to the center of A_c. In Fig. 8 the lower arc, which follows closely A_c, has no angular deviation but the upper arc has a significant angular deviation.

The total mismatch Δ_T between A_c and the ideal apple object A_m is obtained by normalizing and summing these deviations:

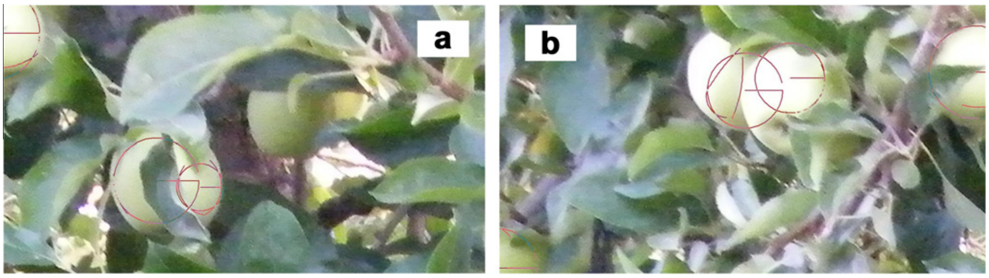


Fig. 9. Details of an image with overlapping detected objects.

$$\Delta T = \left(\sqrt{ \left(\frac{\Delta R}{\delta_{50}^{R}}\right)^2 + \left(\frac{\Delta V}{\delta_{50}^{V}}\right)^2 + \left(\sum_{i=1}^{n} \frac{(\theta_{2i}-\theta_{1i})\Delta r_i}{\delta_{50}^{r} P_{\mathrm{V}}}\right)^2 + \left(\sum_{i=1}^{n} \frac{(\theta_{2i}-\theta_{1i})\Delta a_i}{\delta_{50}^{a} P_{\mathrm{V}}}\right)^2 } \right)$$

The normalization parameters $\delta_{50}^{R}, \delta_{50}^{V}, \delta_{50}^{r}$ and δ_{50}^{a} are chosen in such a way that $\Delta T > 1$ corresponds roughly to an object that has a probability of less than 50% of being an apple. Although this is somewhat arbitrary, this has the advantage of giving a clear meaning to the normalization factors and also provides a way of estimating them. For instance, let A_{c} be identical to A_{m} except where the contour was not identified. δ_{50}^{V} is the missing perimeter fraction beyond which the probability of A_{c} for being an apple is less than 50%. Inspection of the calibration images led to selecting the values listed in Appendix B. For instance, $\delta_{50}^{V} = 0.85 \times 2\pi$ meaning that at least 15% of the contour must be visible for detecting the apple with high reliability.

4.4.4. Removal of multiple detections

Multiple detections of the same object occur in situations such as Fig. 9a, in which the presence of a leaf leads to a non-apple arc that is incorrectly combined with an apple arc to create a circle that meets all the conditions listed in Section 4.4.3. Although the two objects overlap, this fact alone does not guarantee that this is indeed a case of multiple detections. For instance, Fig. 9b shows a situation in which two overlapping objects correspond to two partially overlapping apples. To differentiate between these two typical cases, the degree of overlap is determined as follows. Let us consider two circles with radii A_1 and A_2 ($A_1 > A_2$) and denote the distance between their respective centers as d. The degree of overlap is defined as:

$$\Omega = 1 - \frac{(d + A_2) - A_1}{\sqrt{A_1 A_2}}$$

It is easy to see that when the small circle is completely inside the larger one $\Omega > 1$ and Ω decreases as the small circle moves away from the larger one. The value of Ω depends on the relative size of the two circles and on the distance between their centers, which makes it possible to differentiate, up to a certain extent, between situations such as those shown in Fig. 9a ($\Omega = 0.21$) and Fig. 9b ($\Omega = -0.21$). When counting the objects, pairs of detected circles for which $\Omega > \Omega_{\max}$ are considered to correspond to a single object.

5. Results and discussion

The results of Steps 1 and 4 (pixel classification and apple detection) can be clearly quantified and hence are presented in detail in this section. By comparison, Steps 2 and 3 are only intermediate steps and their performance can not be quantified independently of the overall procedure and is of little interest in and of itself. Therefore the influence of the parameters included in these steps is discussed only with respect to the performance of the overall procedure.

5.1. Pixel classification and seed area formation

The initial pixel classification has a direct impact on the performance of the overall procedure since apples for which a sufficient number of pixels are not correctly recognized as "apple" at this first stage are irremediably lost. The classifier was based on the assumption that apple color and smoothness would be sufficiently distinctive to enable correct classification of enough apple pixels for creating a seed area. The results show that this assumption is correct as long as the light is mostly uniform and diffusive. However, taking pictures of the shaded side of the tree results in strong backlight (Fig. 10a), while taking pictures under low light conditions yields images with relatively high noise in the internal and lower parts of the tree (Fig. 10b), and such situations should be avoided since they deteriorate the classification performance appreciably.

As mentioned in Section 4.1, the classification performance depends on (1) the pixel properties used (color only or color and texture), (2) the size of the filtering window, R_{C}, applied before estimating the pixel color and (3) the size of the window used to estimate smoothness (R_{S}). Tables 1 and 2 summarize the results of the sensitivity analysis that was conducted to determine the optimal parameters. In both tables, the second column is the probability of misclassifying an apple pixel as non-apple and the third column is the probability that a pixel detected as apple is actually non-apple. The fourth column is the average of second and third columns and it represents the average probability of misclassification of apple pixels. Table 1 shows the classification results for various values of R_{C}. Table 2 shows the classification results obtained without smoothness information and with smoothness calculated using various window sizes. The first observation that can be made is that the probability of misclassification is almost three times lower in the second dataset than in the first one due to the improved photography conditions that emphasized the difference between the apples and the background. For both datasets the best results are obtained with R_{C} and R_{S} higher than $0.12 \cdot D_{\min}$, with only very small and inconsistent variations of the average misclassification rate with the actual value of R_{C} and R_{S}. Since using values greater than $0.12 \cdot D_{\min}$ blurs the image details and eventually degrades the final detection results, $0.12 \cdot D_{\min}$ was selected as the value to be used in the subsequent stages of the analysis for both R_{C} and R_{S}. Table 2 shows that not using texture information and relying on color only degrades the overall results by 12% and 22% for the first and second datasets, respectively, compared to

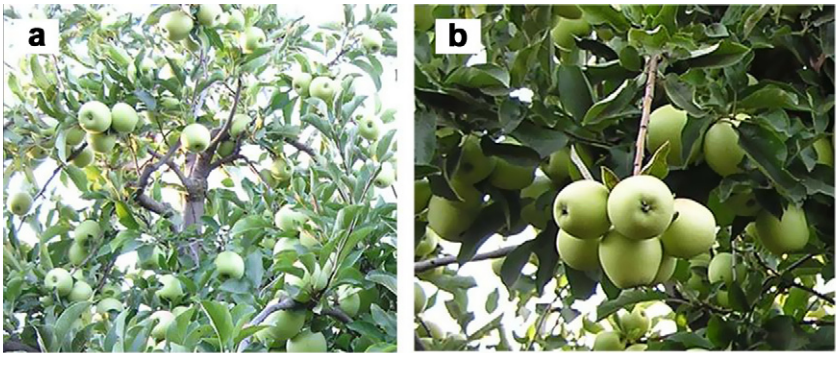


Fig. 10. Examples of poor quality images due to (a) to strong backlight and (b) low light.

Table 1
Influence of the size of the filter used for color-smoothing (R_C) on the rate of misclassification. The bold rows correspond to the classifiers that were used in the analysis presented in the following sections.

R_C	False negative rate	False positive rate	Average misclassification rate
Dataset #1, images recorded in automatic exposure mode: D_{min} = 75, R_S = 9			
2 (0.027 D_{min})	0.4782406	0.2127250	0.3454828
5 (0.067 D_{min})	0.4757716	0.1859929	0.3308822
7 (0.093 D_{min})	0.4799017	0.1847689	0.3323353
9 (0.12 D_{min})	**0.4743216**	**0.1833192**	**0.3288204**
11 (0.147 D_{min})	0.4750884	0.1859475	0.3305180
13 (0.173 D_{min})	0.4790149	0.1854742	0.3322446
Dataset #2, images recorded with manual underexposure: D_{min} = 50, R_S = 6			
2 (0.04 D_{min})	0.1776754	0.0887648	0.1332201
4 (0.08 D_{min})	0.1655643	0.0847855	0.1251749
6 (0.12 D_{min})	**0.1570037**	**0.0798168**	**0.1184103**
8 (0.16 D_{min})	0.1517603	0.0760471	0.1139037
11 (0.22 D_{min})	0.1463461	0.0755470	0.1109466
15 (0.30 D_{min})	0.1481784	0.0771337	0.1126561

Table 2
Influence of the size of the window used to estimate smoothness (R_S) on the rate of misclassification. The bold rows correspond to the classifiers that were used in the analysis presented in the following sections.

R_S	False negative rate	False positive rate	Average misclassification rate
Dataset #1, images recorded in automatic exposure mode: D_{min} = 75, R_C = 9			
Not using smoothness	0.4984354	0.2368744	0.3676549
5 (0.067 D_{min})	0.4917527	0.2073179	0.3495353
7 (0.093 D_{min})	0.4782596	0.1946854	0.3364725
9 (0.12 D_{min})	**0.4743216**	**0.1833192**	**0.3288204**
11 (0.147 D_{min})	0.4693209	0.1812798	0.3253004
13 (0.173 D_{min})	0.4735874	0.1753816	0.3244845
15 (0.2 D_{min})	0.4728587	0.1733178	0.3230882
Dataset #2, images recorded with manual underexposure: D_{min} = 50, R_C = 6			
Not using smoothness	0.1817551	0.1082139	0.1449845
4 (0.08 D_{min})	0.1609944	0.0836199	0.1223071
6 (0.12 D_{min})	**0.1570037**	**0.0798168**	**0.1184103**
8 (0.16 D_{min})	0.1559397	0.0762835	0.1161116
11 (0.22 D_{min})	0.1556744	0.0757016	0.1156880
19 (0.38 D_{min})	0.1600955	0.0806056	0.1203505

using smoothness calculated in a $0.12 \cdot D_{min}$ by $0.12 \cdot D_{min}$ window. This is illustrated in Fig. 11 that shows the detail of an image (Fig. 11a), the classification results when using only color information (Fig. 11b), and the classification results when using both color and smoothness (Fig. 11c). It can be clearly seen that pixels that do not have the typical light-green color are not recognized as apple pixels when using only color information.

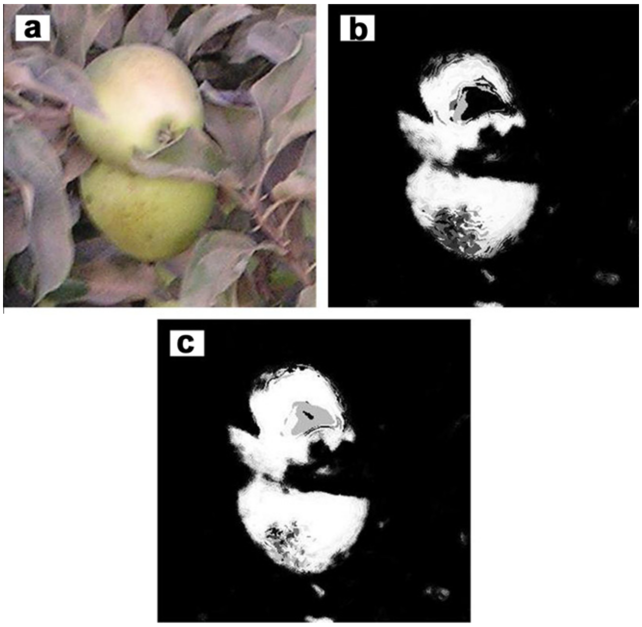


Fig. 11. (a) Detail of an image and classification results using (b) color information only and (c) color and smoothness information.

Fig. 12 shows a typical image and the detected seed areas (after expansion) and it can be verified that a "seed area" was correctly detected for almost all the apples, although in some cases the seed areas encompasses two overlapping apples .

5.2. Apple detection

5.2.1. Dataset #1 – Images recorded in automatic exposure mode

The final results, which were obtained with the parameters listed in Appendix B, are summarized in Table 3. The results are presented in terms of correct and incorrect detections, which were determined manually by inspecting result images such as Fig. 13. Overall, the number of correct detections corresponds to 84% and 88% of the apples counted manually in the calibration and validation images, respectively. However, the number of incorrect detections is unacceptably high, ~25% in the validation images. As can be seen from the fifth column of Table 3, multiple detections of the same apple account for more than 20% of the overall incorrect detections. The other incorrect detections are due to the photography and light conditions that resulted in shiny leaves that are detected as apples. The fourth column of Table 3 gives the number of seed areas that were initially correctly detected but that did not yield an apple detection. Overall, only

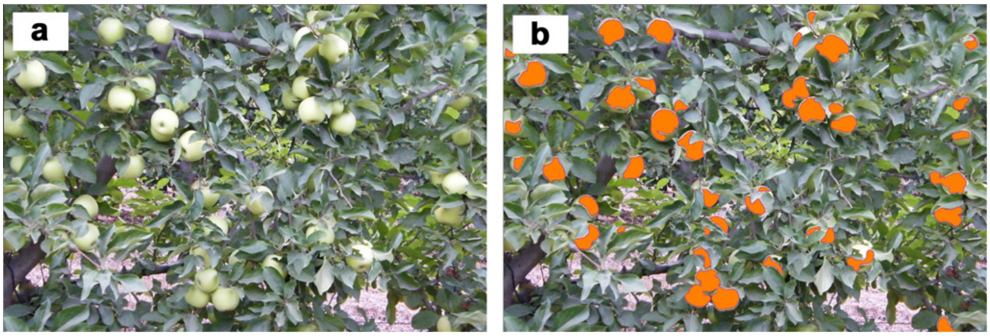


Fig. 12. (a) Original image (validation image #1) and (b) detected seed areas.

Table 3
Final results – Dataset #1.

	Automatic procedure				Ground-truth count	Manual count (six people)	
	Number of correct detections	Number of incorrect detections	Number of apples not detected but for which a seed area was detected	Number of multiple detections		Average	Standard deviation
Calibration images							
Image 1	35	6	2	3	39	35	8
Image 2	29	10	1	4	31	28	4
Image 3	31	0	0	2	50	46	9
Image 4	31	3	1	0	37	39	8
Image 5	22	3	0	1	23	26	3
Image 6	24	6	4	1	30	34	12
Image 7	17	3	0	1	19	22	8
Image 8	35	8	1	2	40	39	14
Total	224	39	9	14	269	269	49
Validation images							
Image 1	37	7	2	4	37	44	7
Image 2	29	5	4	0	33	32	9
Image 3	33	15	7	3	37	39	11
Image 4	16	6	0	1	22	25	11
Image 5	22	9	1	2	22	18	6
Image 6	56	5	3	1	60	49	23
Image 7	34	17	8	1	37	38	13
Image 8	37	3	2	3	49	35	16
Image 9	36	18	2	5	42	44	8
Total	300	85	29	20	339	322	68

approximately 6% of the initial seed areas were incorrectly discarded in the subsequent stages of the analysis. These results show that the most critical stage is the extraction of the seed areas. As explained in Sections 4.1 and 4.2, this stage uses color and texture information, which can be enhanced by diffuse light conditions and manual underexposure.

Table 3 also includes the average and standard deviation of the manual counts performed by the six graduate students (two rightmost columns of the Table). These results show the very large variability that exists in manual counts performed by different people (standard deviation approximately 20% of the average count) and emphasize the difficulty of obtaining a ground-truth estimate to quantify the algorithm performance.

5.2.2. Dataset #2 – Images recorded with diffuse light and manual underexposure

An optimal image, which consists of trees located in the Western edge row photographed close to sunset, is shown in Fig. 1. The results obtained with such images are summarized in Table 4 and a typical result is shown in Fig. 14. The number of correct detections corresponds to 96% and 94% of the ground-truth counts for the calibration and validation images, respectively. The number of incorrect detections is also remarkably low, about 4% of the number of apples, and about 25% of these incorrect detections are due to apples being counted twice.

The significant improvements achieved compared to Dataset #1 show that illumination and exposure are factors that influence strongly the overall performance of the system. While underexposure of the images can be easily achieved, ensuring that the light is mostly diffusive is much more difficult. Working close to sunset as in the present study yields very good images but this approach limits the number of images that can be recorded. Diffuse illumination also occurs naturally on cloudy days, however such cloudy days are rarely observed in Israel during the months of interest. Therefore, although this aspect was not investigated in the present study, placing a screen to block direct sunlight may be the most practical way to enhance the quality of the images and ensure a very high detection performance.

5.2.3. Sensitivity analysis

Tables 5 and 6 summarize the influence of two of the parameters, the threshold applied to the apple probability calculated by

Table 4
Final results – Dataset #2.

	Automatic procedure				Ground-truth count	Manual count (six people)	
	Number of correct detections	Number of incorrect detections	Number of apples no detected but for which a seed area was detected	Number of multiple detections		Average	Standard deviation
Calibration images							
Image 1	54	1	0	0	54	46	9
Image 2	58	1	0	0	58	59	11
Image 3	30	0	3	0	35	33	9
Image 4	31	4	0	0	31	27	3
Image 5	33	2	0	1	33	29	4
Image 6	13	2	0	0	13	12	4
Image 7	34	1	0	1	34	33	6
Image 8	59	3	3	1	62	51	11
Image 9	25	0	4	0	30	30	5
Total	337	14	10	3	350	319	54
Validation images							
Image 1	55	4	0	0	55	52	16
Image 2	39	1	0	1	39	35	10
Image 3	60	0	1	0	61	60	11
Image 4	15	2	1	2	16	14	5
Image 5	20	2	6	2	26	22	4
Image 6	22	1	0	0	22	21	4
Image 7	18	0	3	0	25	25	6
Image 8	47	0	5	0	52	50	13
Image 9	49	2	0	0	49	46	10
Total	325	12	16	5	345	324	70

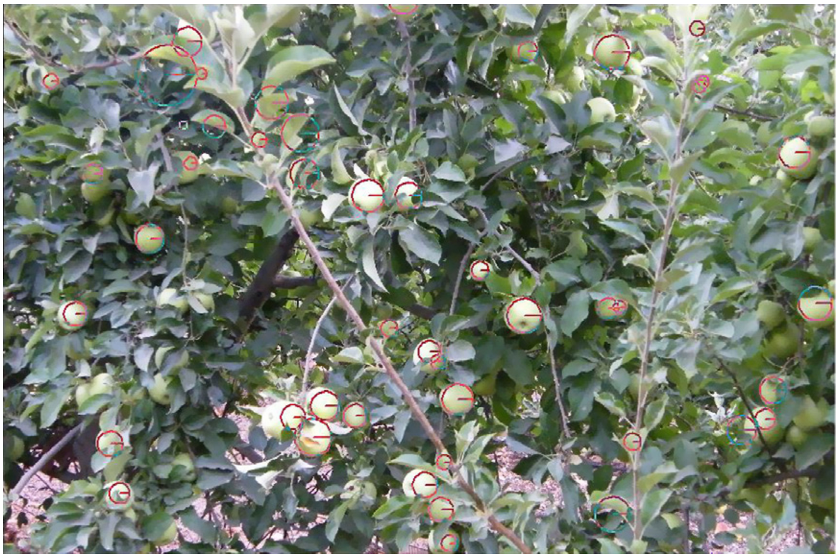


Fig. 13. Validation image #8 (Dataset #1) with detected objects.

the color and texture classifier (Section 4.1) and the length of the interval used when creating the contour arcs, L_{int} (Section 4.3), on the overall performance. This analysis was conducted using only the validation images of Dataset #2. As expected, lowering the classification threshold leads to correct detection of more apples but also causes more incorrect false positives. However, overall the results are quite robust and acceptable results (above 85% of correct detection and less than 10% false positives) are obtained with thresholds ranging from 0.25 to 0.50. The results are very robust with respect to the parameter L_{int} and variations of ±25% around the nominal value ($L_{\mathrm{int}} = 36$) have only a very small influence on the results. Furthermore, variations of ±10% in additional parameters have only very influence on the results (typically up to ±2 detections per image, details not shown).

6. Conclusions

Detection of green apples in images taken in natural environment and under natural daylight conditions is hindered by two types of difficulties: (1) natural light is not diffusive enough and might cause strong shading and saturation, and (2) apples have variable sizes and color, overlap other objects, and are rarely fully

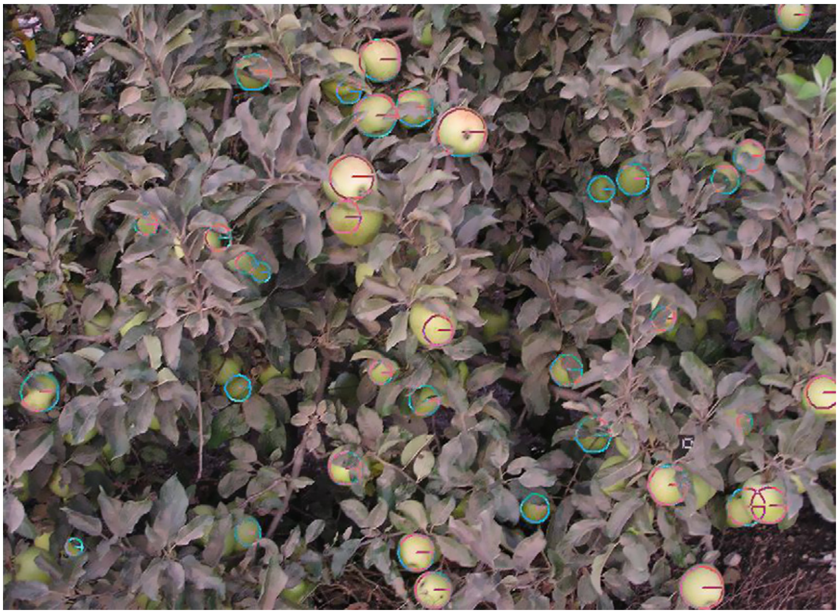


Fig. 14. Validation image #2 (Dataset #2) with detected objects.

Table 5
Detection performance for various values of the threshold applied to the probability calculated by the color and texture classifier. The results correspond to the validation images of Dataset #2 only. The numbers given in parentheses are percentages relative to the ground truth count.

Classifier threshold	Number of correct detections	Number of incorrect detections
0.25	333 (97%)	30 (9%)
0.33	325 (94%)	12 (3%)
0.50	295 (86%)	8 (3%)
0.66	271 (79%)	6 (2%)

Table 6
Detection performance for various values of the length of the interval used when creating the contour arcs, L_{int}. The results correspond to the validation images of Dataset #2 only. The numbers given in parentheses are percentages relative to the ground truth count.

Length of the interval used when creating the contour arcs, L_{int}	Number of correct detections	Number of incorrect detections
28	317 (92%)	18 (5%)
32	322 (93%)	24 (7%)
36	325 (94%)	12 (3%)
40	316 (92%)	20 (6%)
44	317 (92%)	18 (5%)

visible. Nonetheless, the procedure developed in this study detects correctly more that 85% of the apples visible in the images. If care is not taken to prevent direct sunlight and color saturation, the number of incorrect detection may be unacceptably high. However, the quality of the images can be greatly improved by underexposing the images and working under diffusive light. In such a case, the present procedure detects close to 95% of the apples, with a very low rate of incorrect detections (~3%). For the kind of envisioned application (yield estimation), such performance is more than satisfactory and would provide the grower with information that is much more reliable and systematic than the current visual inspection of a small number of trees.

Acknowledgment

This study was supported by the Israel Ministry of Science and Technology, Project 3-3477.

Appendix A. Segmentation near the end points

This simple segmentation procedure described in Section 4.3 yields accurate results as long as the interval of length L_{int} is completely within a segment. At points closer than $L_{int}/2$ to the segment endpoints, such an interval includes points outside the segment, which results in a sharp increase in fuzziness and curvature. Therefore, following the initial segmentation the local properties at such points must be recalculated. The complete procedure consists of four steps:

1. Calculation of the local curvature and fuzziness at every contour point, as explained in Section 4.3.
2. Initial segment detection, by detecting a contour section in which all points have low curvature and fuzziness.
3. Recalculation of the local properties of the points outside the segment detected in Step 2, up to a distance $L_{int}/2$ from its ends, by using an interval that is centered on the initial endpoint of the segment. In other words, the interval centered at the original endpoint is used to calculate the properties of points up to $L_{int}/2$ from it. This extension process is stopped if the distance between the point and the approximating polynomial exceeds a predefined threshold.

Appendix B. Parameter values

Symbol	Value for Dataset #1	Value for Dataset #2	Units
D_{min}	75	50	Pixels
D_{max}	250	120	Pixels
E_{max}	4	2	Pixels

Appendix B (*continued*)

Symbol	Value for Dataset #1	Value for Dataset #2	Units
L_{int}	60	36	Pixels
R_c	9	6	Pixels
R_m	80	40	Pixels
$R_{\max}$	250	15	Pixels
$R_{\min}$	20	70	Pixels
R_s	9	6	Pixels
$r_{0.5}$	0.25	0.25	–
$Z_{\max}$	4	2.9	Pixels^2
Δ_{noise}	4	4.5	Pixels
δ_{50}^{a}	1.0	0.8	Radians
δ_{50}^{R}	$2.1 \times R_{\text{m}}$	$0.75 \times R_{\text{m}}$	Pixels
δ_{50}^{r}	$0.8 \times (R_A + \Delta_{noise})$	$0.33 \times (R_{\text{A}} + \Delta_{\text{noise}})$	Pixels[a]
δ_{50}^{V}	$0.85 \times 2\pi$	$0.85 \times 2\pi$	Radians
$\Omega_{\max}$	0.4	1	–
$\varphi_{\min}$	0.002	0.002	Pixels^{-1}
$\varphi_{\max}$	0.1	0.1	Pixels^{-1}
τ	0.5	0.3	–

[a] R_{A} denotes the estimated radius of the apple.

Besides the lighting conditions, Dataset #1 and Dataset #2 differed by the size of the apples in the images. Dataset #1 included apples with radius ranging from 30 to 200 pixels while the apples in Dataset #2 had a much more consistent radius of approximately 40 pixels. As a result, it is necessary to use different values not only for R_{m}, $R_{\min}$ and $R_{\max}$, but it is also necessary to adjust the weight of the "radius error" in Δ_{T}, which is reflected by the larger values of δ_{50}^{R} and δ_{50}^{r} in Dataset #1.

Appendix C. Supplementary data

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.compag.2011.11.007.

References

Annamalai, P., Lee, W.S., 2003. Citrus yield mapping system using machine vision. ASAE Paper 03-1002.

Bulanon, D.M., Kataoka, T., Zhang, S., Ota, Y., Hiroma, T., 2001. Optimal thresholding for the automatic recognition of apple fruits. ASAE Paper 01-3133.

Jimenez, A.R., Ceres, R., Pons, J.L., 2000. A survey of computer vision methods for locating fruit on trees. Trans of the ASAE 43, 1911–1920.

Kim, Y., Reid, J., 2004, Apple yield mapping using a multispectral imaging sensor, Proceedings of the AgEng'04 Conference.

Moore, A., 1991. An introductory tutorial on kd-trees. In: Efficient memory-based learning for robot control, Ph.D. Thesis, University of Cambridge. Available at <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.10.818&rep1&type=pdf>.

Pla, F., Juste, F., Ferri, F., Vicens, M., 1993. Colour segmentation based on a light reflection model to locate citrus fruits for robotic harvesting. Computers and Electronics in Agriculture 9, 53–70.

Safren, O., Alchanatis, V., Ostrovsky, V., Levi, O., 2007. Detection of green apples in hyperspectral images of apple-tree foliage using machine vision. Trans of the ASABE 50, 2303–2313.

Stajnko, D., Lakota, K., 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., Park, J., 2006. Segmentation of apple fruit from video via background modeling. ASABE Paper 06-3060.

Wachs, J.P., Stern, H.I., Burks, T., Alchanatis, V., 2010. Low and high-level visual feature-based apple detection from multi-modal images. Precision Agriculture 11, 717–735.

Zhao, J., Tow, J., Katupitiya, J., 2005. On-tree fruit recognition using texture properties and color data. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 263–268.