Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/issn/15375110

Research Paper

Automatic fruit recognition and counting from multiple images

Y. Song [a], C.A. Glasbey [a,*], G.W. Horgan [b], G. Polder [c], J.A. Dieleman [d], G.W.A.M. van der Heijden [c]

[a] Biomathematics and Statistics Scotland, Edinburgh EH9 3JZ, UK
[b] Biomathematics and Statistics Scotland, Aberdeen AB21 9SB, UK
[c] Biometris, Wageningen UR, P.O. Box 100, 6700 AC Wageningen, Netherlands
[d] Wageningen UR Greenhouse Horticulture, P.O. Box 644, 6700 AP Wageningen, Netherlands

ARTICLE INFO

Article history:
Received 20 March 2013
Received in revised form
28 November 2013
Accepted 13 December 2013
Published online 20 January 2014

In our post-genomic world, where we are deluged with genetic information, the bottleneck to scientific progress is often phenotyping, i.e. measuring the observable characteristics of living organisms, such as counting the number of fruits on a plant. Image analysis is one route to automation. In this paper we present a method for recognising and counting fruits from images in cluttered greenhouses. The plants are 3-m high peppers with fruits of complex shapes and varying colours similar to the plant canopy. Our calibration and validation datasets each consist of over 28,000 colour images of over 1000 experimental plants. We describe a new two-step method to locate and count pepper fruits: the first step is to find fruits in a single image using a bag-of-words model, and the second is to aggregate estimates from multiple images using a novel statistical approach to cluster repeated, incomplete observations. We demonstrate that image analysis can potentially yield a good correlation with manual measurement (94.6%) and our proposed method achieves a correlation of 74.2% without any linear adjustment for a large dataset.

1. Introduction

There are an increasing number of robotics applications aimed at detecting fruits from images or videos (De-An, Jidong, Wei, Ying, & Yu, 2011; Ji et al., 2012; Linker, Cohen, & Naor, 2012; Tanigaki, Fujiura, Akase, & Imagawa, 2008). Although various research efforts have been made in this field, challenges still remain for complex scenes with varying lighting conditions, low contrast between fruits and leaves, foreground occlusions and cluttered backgrounds. Most of these applications have been to find the fruits for automatic harvesting. A recently new direction is to find the fruits for plant breeding purposes (Alimi et al., 2013): to automatically recognise, count and measure the fruits in order to assess the differences in quality of the genetic material. When the measurements are made by a computer, this is often referred to as digital phenotyping and the field is growing in importance, e.g. Furbank and Tester (2011).

The aim in our application is to locate and count green and red pepper fruits on large, dense pepper plants growing in a greenhouse. Alimi et al. (2013) described the use of manual fruit measurements (manual phenotyping) for predicting yield in pepper plants. Our work is to automatically detect and

* *Corresponding author*. Tel.: +44 1316504899.
E-mail address: chris@bioss.ac.uk (C.A. Glasbey).
1537-5110/$ – see front matter
http://dx.doi.org/10.1016/j.biosystemseng.2013.12.008

Fig. 1 – Examples in the training data. The top three rows are fruit examples, and the bottom three are background. The background templates are much larger than the fruit templates, and their sizes have been adjusted for display purposes.

count any fruit in images of dense pepper plants, to reduce manual measurement and labour requirements, and to increase objectivity. In a recent paper, van der Heijden et al. (2012) showed that several manual measurements could be replaced by image analysis leading to the same QTL (positions on a genetic map, which shows a relation with the trait under study). Besides they showed that image analysis could aid in the identification of additional physiological traits that are hard or impossible to measure by human operators.

Machine vision applications developed for fruit have been reviewed by Brosnan and Sun (2004) and Lee et al. (2010). Compared with previous fruit applications, e.g. finding red apples in green canopies (Bulanon, Kataoka, Ota, & Hiroma, 2002), we are looking for predominantly green fruits. Stajnko, Lakota, and Hocevar (2004) described the use of thermal imaging for measuring apple fruits. In their work, they used morphological operations and constant shape constraint to separate the round apple fruits from leaves. This is not possible for our images, since the difference in colour and shape between fruits and other plant parts are small.

Jimenez, Jain, Ceres, and Pons (1999) provided a review of different vision systems to recognise fruits for automated harvesting using a laser range-finder. Zhao, Tow, and Katupitiya (2005) presented methods to recognise apples grown on trees, which used the texture and redness colour. It was shown that redness works equally well for green apples as for red ones. Yang, Dickinson, Wu, and Lang (2007) proposed methods to recognise mature fruit and locate cluster positions for tomato harvest applications. Kitamura and Oka (2005) described a picking robot to recognise and cut sweet peppers in greenhouses, but their image analysis methods were developed only for this specific application under fixed lighting conditions.

In this paper we describe a new method to locate and count green peppers in a cluttered complex image, using a two-step approach. In a first step, the fruits are located in a single image and in a second step multiple views are combined to increase the detection rate of the fruits. The approach to find the pepper fruits in a single image is based on a combination of (1) finding points of interest, (2) applying a complex high-dimensional feature descriptor of a patch around the point of interest and (3) using a so-called bag-of-words (Nilsback & Zisserman, 2006; Sivic & Zisserman, 2008) for classifying the patch. For complex images, every object detector will yield both false positives and missing detections. If the application is video-based, one could apply a number of tracking-by-detection approaches (Breitenstein, Reichlin, Leibe, Koller-Meier, & Van Gool, 2009). These methods continuously perform a detection algorithm in individual frames and then associate detections across frames. In our case, we are not using a video-based approach, but since images are recorded every 5 cm, we can use multiple views of the same fruit. We show a new statistical approach to combine information from multiple views to improve the detection rate of the fruits.

2. The datasets

The plant material used in this paper consists of pepper plants of 148 recombinant inbred lines resulting from a cross between a large-fruited bell pepper ('Yolo Wonder') and a small-fruited chilli pepper ('Criollo de Morelos 334'). Including parents and F1, there were 151 genotypes, and they were randomised over four compartments in an incomplete block design. There were 264 experimental plots grown in a standard double-row arrangement, and each plot consisted of eight plants. The four plants in the centre of a plot were experimental plants, and the other four were border plants, making 1056 experimental plants in total. Two trials were conducted in 2009. The first trial was in spring (from December 2008 to May 2009) and the second in autumn (from June to September 2009). Further information regarding the trials can be found in Alimi et al. (2013). In this paper, the first trial was used for training, and the second for validation.

Our aim is to develop a high-throughput image analysis tool and record images of plants in their growing conditions without transporting the plants to a controlled environment. We used an imaging robot known as SPYSEE to capture images of pepper plants. A trolley was equipped with cameras, computer, illumination and wheel encoder to record images of pepper plants in the greenhouse, while moving between the rows of pepper plants. The trolley was pushed along heating pipes, 60 cm apart, by a human operator. As pepper plants can grow to more than 3 m in height, and the distance between plants and camera is relatively small due to the narrow space between two adjacent rows, we used a vertical stack of four cameras. Every 5 cm, the wheel encoder on the trolley automatically triggered the flashes and cameras. The distance from the cameras to the plants was short, requiring a large field of view lens. Colour cameras with a high resolution, type CSFS20CC2 Teli FireWire, were used with a resolution of 1024×1280 pixels and Lensagon lens, type CMFA0420ND, with a 75° field of view. The region of interest was set to 480×1280 (width $\times$ height) pixels. For illumination, a Xenon flashlight (VIGI-LuxTM MVS 5002 Strobe, PerkinElmer) with a light pulse duration of 30 μs was used, which allowed a short shutter time of 70 μs for the cameras, resulting in sharp images, hardly influenced by ambient light. For further details, see Polder, van der Heijden, Glasbey, Song, and Dieleman (2009).

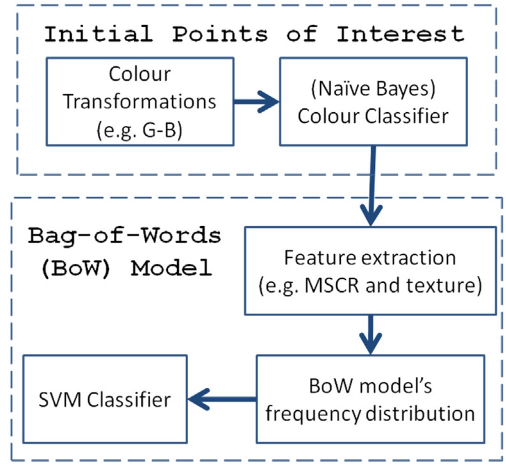


Fig. 2 – Overview of our fruit recognition method in Section 3.

The total number of colour images in each trial exceeded 28,000. Since these plants belonged to different genotypes, their fruits varied greatly in size and shape (Alimi et al., 2013). Some examples can be seen in Fig. 1.

For every plant in the experiment, fruits were physically counted and harvested shortly after image collection. We randomly selected a row with 10 experimental plots from the validation trial. There were 408 images in total, and we manually labelled all fruits visible in each image. This set was created as the ground truth in order to evaluate the performance of our methods.

For algorithm training, we randomly extracted 110 fruit templates that have one fruit (see Fig. 1) and 80 background templates that do not have any fruit. The size of the fruit templates varies from 18×60 to 72×119. The 110 fruit templates were manually classified into red and green fruits. There were 104 green fruits and only 6 red fruits. As seen in Fig. 1, the fruit templates are cluttered with some background pixels in addition to the fruit. Background templates were also collected in the same greenhouse environment, and contained plant parts (e.g. leaf and branch, but no fruits) as well as background objects (e.g. growth pots).

3. Fruit recognition

Our approach is to allocate a support window (e.g. a rectangle patch) for every pixel in an image using a sliding window approach (Dalal & Triggs, 2005; Ferrari, Fevrier, Jurie, & Schmid, 2008), and then verify whether there are sufficient features within the sliding window to classify it as a fruit object. Using a sliding window for every pixel in a 480×1280 image would require more than 600,000 windows per image, which would not be practical for analysing such a large dataset. We therefore applied a simple, fast method to identify approximately 10,000 points of interests (POI) per image, discarding points that clearly were not fruits, thus significantly reducing the number of required operations. An overview of our methods can be found in Fig. 2. Similar approaches have been taken by Leibe, Leonardis, and Schiele (2004), Leibe, Seemann, and Schiele (2005) and others.

3.1. *Initial points of interest*

The steps we have taken to identify POIs are somewhat arbitrary. However, we argue that this is not important. There are typically 10 fruits in an image, so a cautious 60-fold pruning from the original 600,000 points down to about 10,000 by discarding non-interesting points is small compared to the second stage 1000-fold selection process. Alternative approaches that could have been considered include the thresholding used by Reis et al. (2012) to distinguish red and white grapes from leaves, and the linear colour model approach of Teixido et al. (2012).

Colour transformation A classifier is trained on colour information to identify the initial points of interest. Many pepper fruits are green, and we transform the RGB colour intensity in order to distinguish between the fruit and other green plant parts. For each colour pixel (R,G,B), the first transformation $G-B$ quantifies the intensity difference between green and blue. The second transformation $G-R$ quantifies the intensity difference between green and red. The final transformation $G/(R+G+B)$ quantifies the proportion of green. This simple, straightforward colour transformation was chosen because it is less sensitive to changing illumination conditions than the original R, G and B values, but no attempt was made to optimise the transformation (Gevers & Smeulders, 1999).

Colour pixels in a training template are first transformed into a $N \times 3$ vector with columns $G-B$, $G-R$ and $G/(R+G+B)$. For example, for a 1280×480 template, $N = 1280 \times 480 = 614{,}400$. To reduce N, for each template, two clusters were found in the transformed space using K-means clustering with a Euclidean metric. The means and numbers of pixels in the two clusters are then used to represent the template. We found that two clusters were sufficient to capture the variability in pixel values in templates, whereas one cluster, or equivalently sample means, lost this variability. Figure 3 shows the mean values extracted from the fruit and background templates. Note that (R,G,B) has a value range of [0,1] and we applied a linear rescaling so that $G-B$, $G-R$ and $G/(R+G+B)$ also lie in the range [0,1].

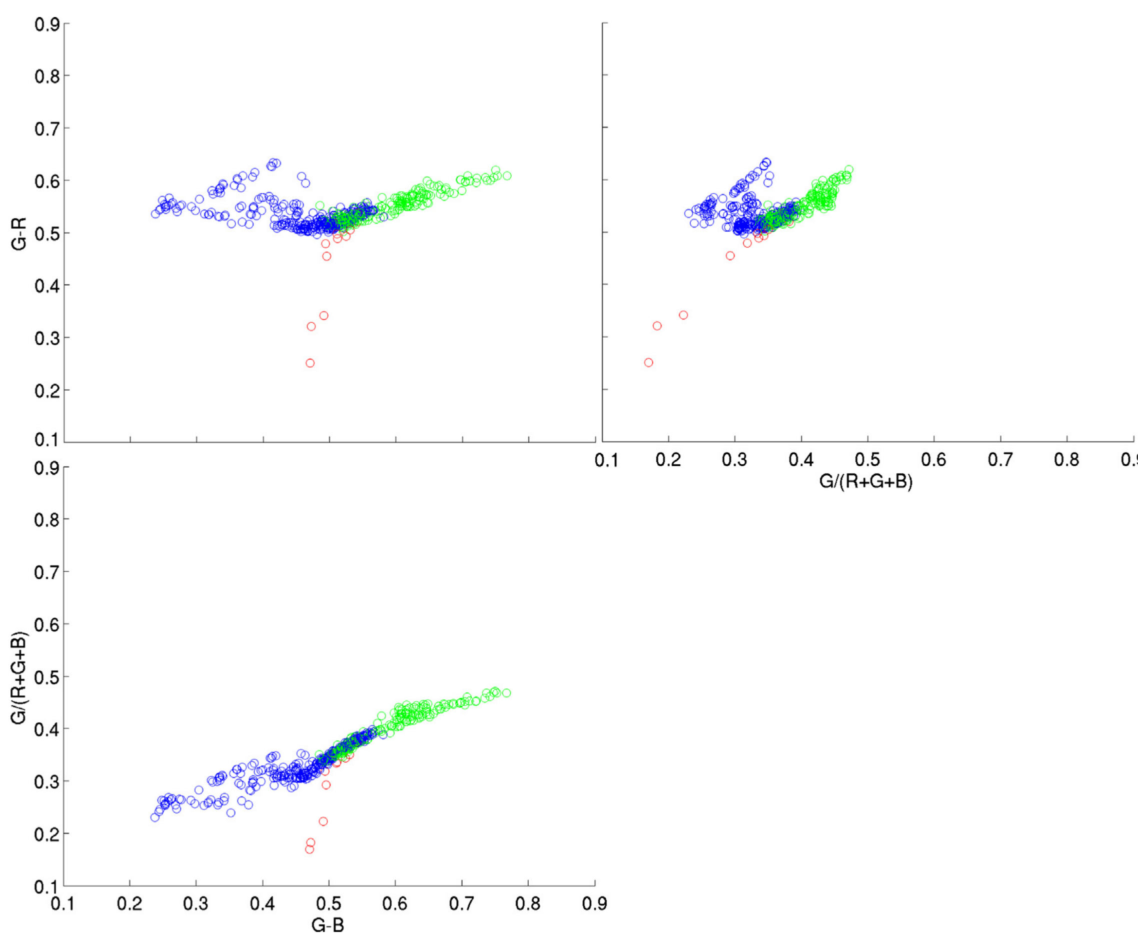


Fig. 3 – Relationship between the fruit group and the background group in $G-B$, $G-R$ and $G/(R+G+B)$ from all training templates. The *x*-axis, *y*-axis and *z*-axis are normalised values for $G-B$, $G-R$ and $G/(R+G+B)$ respectively. Red and green dots are for red and green fruits respectively. The background group is represented by blue dots.

Table 1 – Means (standard deviations) of normalised, transformed colours.

Template group	Probability %	G − B	G/(R + G + B)	G − R
Background	99.27	0.463 (0.078)	0.332 (0.036)	0.532 (0.024)
Red fruits	0.04	0.508 (0.024)	0.305 (0.073)	0.451 (0.093)
Green fruits	0.69	0.583 (0.065)	0.398 (0.036)	0.547 (0.027)

Colour classifier Given the transformed vectors for the templates of red fruit, green fruit and background, we then constructed a Naive Bayes classifier with these three classes using prior parameters given in Table 1. For simplicity and speed, we ignored correlation between variables. The classifier was applied separately to the transformed colours of every pixel in an image, and the posterior probabilities of red and green fruits were combined. Figure 4 shows an example of the posterior probabilities for the combined fruit group and the background group. We applied a thresholding on the posterior probabilities T_p to obtain the initial points of interest.

3.2. Bag-of-Words model

Our approach is inspired by, and builds on, Nilsback and Zisserman (2006), who used a ‘bag of words’ (BoW) approach to classify flowers. We combine this with the use of Maximally Stable Colour Region (MSCR) features (Forssén, 2007). The methodology is well established in detection and tracking of pedestrians, and here we explore its potential to obviate the need for many different shapes and sizes of templates to detect our highly variable fruit shapes.

For each initial point, we allocate a support window centred at the point in order to provide sufficient image information for recognition. In this work, the size of the support window used was 40×90 pixels, which was based on the average size of the fruit templates.

Feature extraction To determine whether a fruit is present in a support window, we describe the window using two different feature sets: MSCR features (Forssén, 2007) and texture features obtained by local range filters.

An MSCR feature set is a set of descriptors of coloured ellipses in a window. These descriptors are found using an MSCR detector, which is an extension to colour of the maximally stable extremal region (MSER) covariant region detector (Matas, Chum, Martin, & Pajdla, 2002). The original MSER detector finds regions (ellipses) that are stable over a wide range of thresholdings of a grey-scale image. In MSCR, regions are

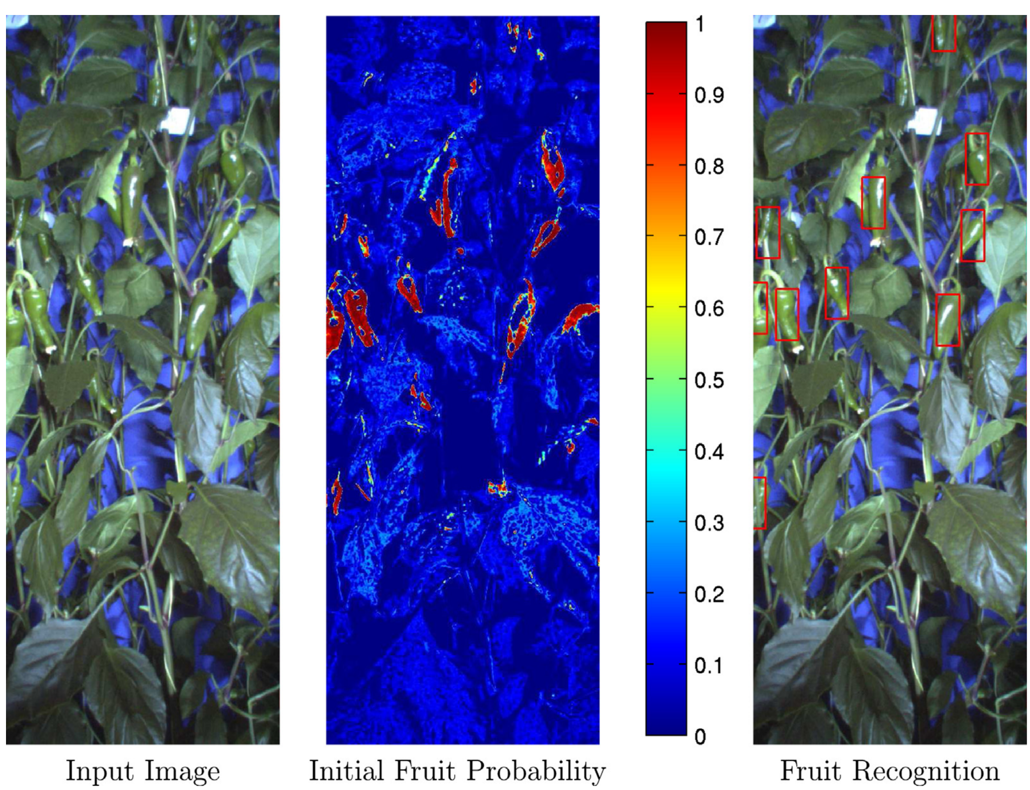


Fig. 4 – An example illustrating the fruit recognition method. We first identify a number of possible fruit positions (initial points of interest), and then verify each fruit position for the removal of false and duplicated estimates by a Bag-of-Words model. Initial fruit probability is calculated based on G − B, G − R and G/(R + G + B). Fruit recognition is obtained by applying the Bag-of-Words model on the initial points of interest.

detected that are stable across a range of time-steps in an agglomerative clustering of image pixels, based on proximity and similarity in colour (Forssén, 2007). Default parameters as described in Forssén (2007) were used. The obtained feature set provides an approximate description of the ‘objects’ in a window (see Fig. 5(b) in the form of ellipses, which constitute an affine-invariant object representation when viewed from different angles. We used the geometric shape (five variables) and mean colour (three variables) of the fitted ellipses as a feature set.

Besides MSCR features, texture features from local range filters are also used. A local range filter simply calculates per colour the difference between the largest and smallest intensity in the filter window. Nilsback and Zisserman (2006) used a set of filters with 4 sizes: 3, 7, 11, 15, to define the texure. To reduce the amount of computational load and memory, we used only two filter sizes and good results were obtained with the filter sizes 5×5 and 9×9 (see Figures 5(c) and (d)).

Bag-of-Words (BoW) frequency distribution Next a so-called ‘bag of words’ approach is used as proposed by Zisserman and collaborators (Nilsback & Zisserman, 2006; Sivic & Zisserman, 2008). Nilsback and Zisserman (2006) describe a set of flower images by creating a flower vocabulary, using three different vocabularies for colour, shape (SIFT features) and texture (using the filter bank). Each vocabulary vector is quantised (discretised) to obtain so-called Visual Words. The frequency histogram of the visual words form a so-called bag of words. These frequency histograms can then be used to calculate similarities, yielding a quick search method for images or videos (Sivic & Zisserman, 2008).

We used K-means clustering to construct a vocabulary with 1000 ‘words’ to represent the MSCR features, and similarly for the local range features, following Nilsback and Zisserman (2006). Then, using the constructed vocabularies, we learned the frequency distribution of the combined vocabularies (2000 words) for the training data.

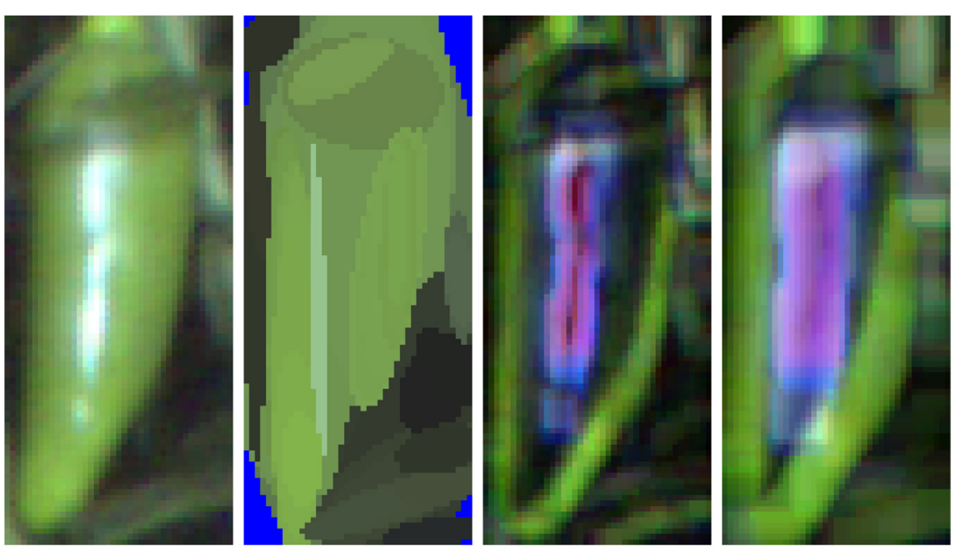
SVM classifier Finally a support-vector-machine (SVM) classifier was used on all the frequency distributions of the training data to represent two groups, *Fruit* and *Others*. In effect, the BoW model represents each image by a frequency distribution of its visual vocabularies.

3.3. *Using bag-of-words model*

For processing a validation image, we first find the initial points of interest in an image, as described in Section 3.1. Next the MSCR and local range features are calculated per window at each initial point. From the quantised vector of these two vocabularies, the frequency distribution in the bag-of-words frequency histogram is calculated, which subsequently is classified to a fruit class or not, using the SVM.

The outputs include fruit locations, and each estimate also has a weight threshold for the two classes. The weight threshold W is the (arbitrary) distance computed from the SVM classification and a higher value means that the window is more likely to belong to that class. In fact, the values quantify how far an object of interest is from the decision line separating the two groups. A smaller value means closer to the borderline, while a larger value means it is more likely to belong to that class.

When points of interest are ‘close’ together, we obtain multiple classifications. In that case, we select the point/window with the highest weight threshold W. ‘Close’ is defined here as the overlap between the two windows, and we consider that two windows which have more than 50%



(a) Image (b) MSCR (c) 5×5 texture (d) 9×9 texture

Fig. 5 – MSCR features and image textures. Each ellipse in (b) is an MSCR feature, and its region is filled by the average colour of that ellipse. The MSCR features provide an approximation to an image. Regions not covered by the MSCR features were shown as blue. The 5×5 and 9×9 textures are obtained by a range filter on the colour image. The colour indicates the magnitude of local colour variation, e.g. green regions indicate large variation in green colour. (a) Image (b) MSCR (c) 5×5 texture (d) 9×9 texture.

overlap with each other are 'close'. Overlap is calculated by the intersection of two detection windows divided by their union.

Overall, the recognition method performs reasonably well given the challenges we faced (see discussion section). The relationship between successive views must be investigated to help filter out isolated false positives, find occluded fruits and produce total fruit count.

4. Fruit counting from multiple views

Since we observe the same plant/fruit in multiple images, we need to combine this information into a single result. The aim is to count the correct number of fruits K in a plot, while preventing double counting of the same fruit which may appear in multiple images as well as correctly counting fruits that are possibly missed in certain views (e.g. because of occlusion). Note that a fruit is approximately shifted by a fixed amount (γ) in the horizontal direction in consecutive images, depending on its distance from the camera. This property will be used to find the same fruit in multiple images.

Consider a contiguous sequence of images from a single experimental plot at one of the four camera heights. Let (x_{ij},y_{ij}) denote the column & row coordinates of the jth fruit located in the ith image, where $i = 0,\ldots,I$ and $j = 1,\ldots,J_i$. For example, Fig. 6 shows illustrative data for a short sequence of 3 images. It is likely that some of these data are repeat observations of the same fruit in different images, because some row coordinates (y) are very similar and column coordinates (x) shift by similar amounts between images. In order to estimate the number of fruits we need to determine which are repeat observations.

Suppose that there are K fruits observed at least once in an experimental plot, indexed by $k = 1,\ldots,K$. To simplify exposition, we will only consider a single camera height, though in the results we sum the K's at the 4 heights to obtain total fruit count. Let (α_k,β_k) denote the true column & row coordinates of fruit k in image 0, and γ_k the true shift in column coordinate between consecutive images. We propose as our observation model:

$$\begin{pmatrix} x_{ij} \\ y_{ij} \end{pmatrix} \sim N\left(\begin{bmatrix} \alpha_{k(ij)} + i\gamma_{k(ij)} \\ \beta_{k(ij)} \end{bmatrix}, \begin{bmatrix} \sigma_x^2, & 0 \\ 0, & \sigma_y^2 \end{bmatrix} \right), \quad (1)$$

where $k(ij)$ denotes the correct fruit label of the observation indexed (i,j), and (σ_x^2, σ_y^2) denote the variances of the normally distributed observation errors. There are 3K parameters (α, β, γ) associated with each experimental plot, together with 2 variance parameters which are common to all plots. The challenge is to estimate the number of fruits, K, in the presence of the remaining nuisance parameters.

In principle, we could estimate all parameters by maximising the likelihood of the model specified by (1), conditional on K, and estimate K using likelihood ratio tests. However, this would involve an enormous combinatorial search to assign fruit labels $k(ij)$, so direct optimisation is computationally infeasible. Reversible jump Markov chain Monte Carlo is another possible approach to tackle this problem, but is problematic because of the large dataset of 40,000 observations, so we have instead developed a simpler, much faster, more *ad hoc* method. We first estimate σ_y^2 by fitting a mixture distribution to differences in y between pairs of observations. Similarly we estimate σ_x^2 by considering triplets of observations. Finally, for each plot we apply a 95% significance threshold rule to identify non-overlapping sets of observations of a single fruit, starting with the largest possible

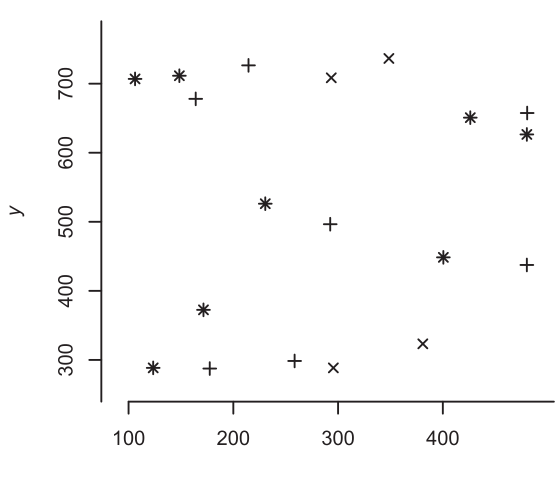


Fig. 6 – Illustrative data for a sequence of 3 images from one experimental plot at a single camera height, with row coordinates (y) plotted against column coordinates (x) for image 0 (*), 1 (+) and 2 (×).

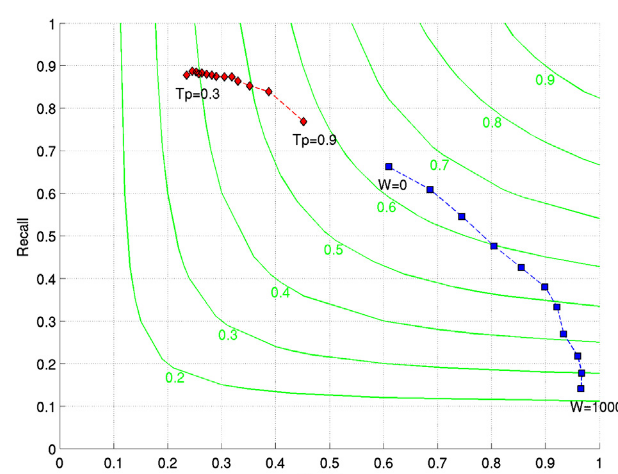


Fig. 7 – Precision-recall curves showing detection performance of our fruit recognition method for 10 experimental plots. Red and blue curves represent initial points and the BoW model applied on initial points respectively. The parameter T_p for initial points (described in Section 3.1) was in the range of [0.3, 0.9], and the range of parameter W for the BoW model is shown in Table 2. The green contours outline F_1 scores from 0.2 to 0.9. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

set size (l) and progressively reducing until we are left with singletons. The number of sets is our estimate of K. See the Appendix for details.

5. Results

To quantify the performance of our fruit recognition method (i.e. first finding the points of interest and then classifying the bag-of-words), we used the precision-recall curve and the ground truth consisted of manually labelled fruit positions for a single row of 10 experimental plots (408 validation images in total, see Section 2). If the overlap between a window classified as fruit (detection) and a similar sized window around the ground truth position is greater than 50%, then the detection is considered a true positive; otherwise the detection is a false positive. We treat each detection as unique for a fruit: if there are multiple detections satisfying the overlap criterion, the one with maximum overlap is the true positive and the others are considered false positives. A false negative represents a ground truth position which has no corresponding detection. The precision is defined as,

$$\text{Precision} = \text{TruePositive}/(\text{TruePositive} + \text{FalsePositive})$$

The recall is:

$$\text{Recall} = \text{TruePositive}/(\text{TruePositive} + \text{FalseNegative})$$

We also combine both precision and recall into a single score F_1,

$$F_1 = 2 \times \text{Precision} \times \text{Recall}/(\text{Precision} + \text{Recall})$$

Figure 7 presents the precision-recall performance for our fruit recognition methods. The result for the initial points of interest was obtained by varying the threshold T_p (0.3,0.4,0.5,…0.9). In case of overlapping detection windows, we used the one with the highest posterior probabilities T_p.

There were many false positives using the initial colour classifier, resulting in low precision ($\leq$0.45). However, precision is less relevant, as this only provides initial estimates. We do want a high recall however for initial points, to make sure that we do not miss fruits. For the BoW model, T_p was set to 0.9, and the weight threshold W was in the range from 0 to 1000 as in Table 2. The BoW model eliminated 2/3rds of the false positives in the initial estimates and the minimum precision is 0.61. False positives can almost be eliminated, but the number of true positives reduces and the miss detection rate can become quite high. For example, the highest precision was 0.97, but the recall was only 0.17 and the F_1 score was lower than 0.3. For the 10 experimental plots, the highest F_1 score was 0.65 at $W \geq 100$, and the F_1 score was also above 0.6 for thresholds {0,200,300}.

It should be noted that the fruit count of a plot could not be estimated using single images only. Plants were visible in a successive sequence of images, and there were multiple plants in an image. We therefore applied the multiple-view fruit counting algorithm to the 10 experimental plots where locations of fruits had been visually identified from images (i.e. the ground truth for evaluating fruit recognition method). Using the methods described in the Appendix, we obtained $\hat{\sigma}_x^2 = \hat{\sigma}_y^2 = 5^2$. We note that these are smaller than those for automatic fruit detection, showing the superiority of the human eye. Figure 8 shows the results, with 94.6% correlation between K and $\hat{K}_{VIS}$. We see that fruit numbers are overestimated for all but one plot.

This is most likely due to fruits from border plants appearing in images although they are excluded from manual counting. Also, as we had four vertical levels of images, some fruits may have appeared at both the top of one image and the bottom of the one above. However, these sources of overestimation will be convolved with those from underestimation, mainly from occlusion due to fruits being hidden behind leaves.

We applied the multiple-view algorithm to all experimental plots in the validation trial with at least 12 images at each of the 4 camera heights, totalling 435, for a range of weight thresholds W. This is more than the 264 plots stated

Table 2 – Choice of weight threshold W to maximise correlation between manually counted number of fruits per experimental plot (K) and estimated value ($\hat{K}$).

$W \geq$	No. data	% Correlation
0	38,600	63.1
100	28,600	67.8
200	21,400	72.4
300	16,300	74.2
400	12,900	74.0
500	10,300	73.0
600	8300	71.6
700	6600	70.0
800	5200	67.4
900	4000	64.6
1000	3000	62.3

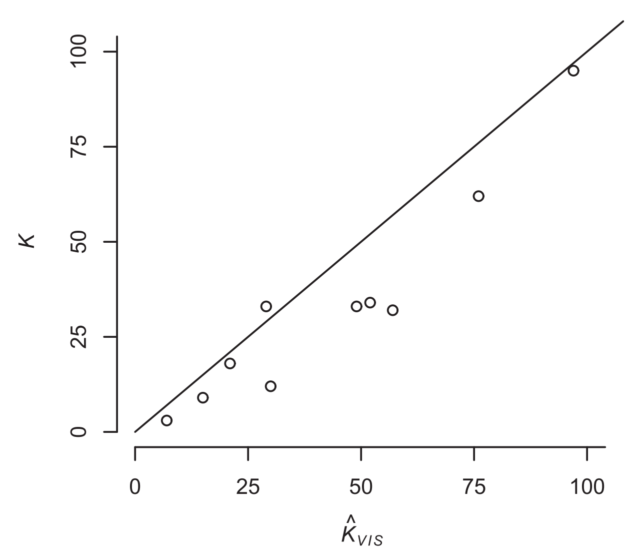


Fig. 8 – Plot of manually counted number of fruits per experimental plot (K) against estimated value using visually identified fruits in images ($\hat{K}_{VIS}$) for 10 experimental plots, together with 1:1 line.

in Section 2 because we view each plot twice, once from the aisle on either side, and have treated these views separately. Recall that, although our exposition has only considered a single camera height, in practice we sum the K's at the 4 heights to obtain total fruit count. Table 2 shows the correlation between observed and estimated numbers of fruits per experimental plot for a range of thresholds, from which we see that a threshold of 300 is best, maximising the correlation at 74.2%.

Figure 9 shows the agreement between K and $\hat{K}$ for each of the 435 experimental plots in the validation trial, from which it is striking that not only is the correlation maximised but also $\hat{K}$ is a good estimate of K without needing linear adjustment for intercept and scale. The standard error of prediction is 11.3 fruits. In Fig. 9 four plots from each of three genotypes with large numbers of fruits have been displayed using different symbols, from which we see that prediction errors are not independent of genotype, as all plots for two genotypes lie above the 1:1 line whereas for the other they all lie below. Some bias is to be expected, as fruits are more likely to be occluded in genotypes with denser foliage, and some shapes and sizes of fruits may be harder to detect by our automatic algorithm. Of the overall variability in K, 55% (i.e. 74.2^2, the squared correlation) is explained by $\hat{K}$, and a further 38% by the 146 genotypes, but this is only 0.26% per genotype. When manual and automatic fruit counts are averaged over replicates of each genotype, correlation rises to 79.4%. Remaining sources of over- and under-estimation are as we discussed for Fig. 8. However, unlike in Fig. 8 where over-estimation predominated, in Fig. 9 on average the two approximately balance out, with as many points above as below the 1:1 line, though this is only the case when a specific threshold value of $W \geq 300$ is used in the BoW model.

We conducted a simulation trial to further validate the multiple-view algorithm. Table 3 shows the bias in $\hat{K}$ averaged

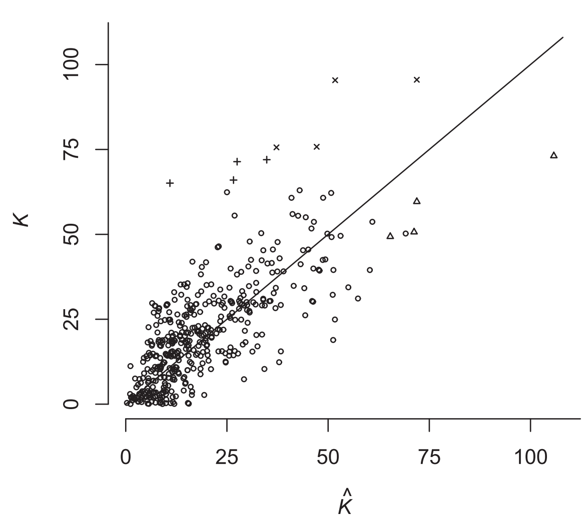


Fig. 9 – Plot of manually counted number of fruits per experimental plot (K) against estimated value ($\hat{K}$) for 435 experimental plots, together with 1:1 line. For three genotypes with large numbers of fruits the four plots have been displayed using △, × and + symbols.

Table 3 – Results of 100 simulations of full dataset, using algorithm with range of % limits for $S^2 \leq \chi^2_{2n-3}(\%)$.

Limit %	$\hat{K}$ bias
75	0.83
90	0.24
95	0.00
99	−0.19
99.5	−0.21
99.9	−0.27

over 100 simulations of the full dataset of 435 experimental plots, using both the selected threshold for S^2 of $\chi^2_{2n-3}(95\%)$ and alternative probability levels. We see that 95% is unbiased whereas other values lead to biases in $\hat{K}$.

6. Discussion

The aim of our work was to reliably estimate the number of fruits in a crop. The main challenge is that fruits can have a variety of colours from green to red amidst green leaves of varying colour. A more comprehensive solution might be achieved by using a full 3D approach, as we had earlier attempted (Song, Glasbey, van der Heijden, Polder, & Dieleman, 2011). However, construction of such a 3D map is not trivial and introduces errors in itself. Even if a 3D map can be constructed, it does not fully solve the problem of automatic fruit recognition. Therefore this approach was not adopted here, and focus was on a 2D approach, using multiple views to limit the problem of occlusion.

In earlier stages of this research several methods were adapted, including a template-correlation approach, where the specular reflection was used to locate the fruit, but results proved to be unsuccessful. However, because of the complexities of our images, with green and red fruits in a green canopy, we had to use sophisticated, computationally-intensive segmentation methods. Reis et al. (2012) showed a solution based on relatively simple RGB-thresholds to distinguish between white and red grapes versus leaves. In our case, such a threshold approach was not sufficient, as the green leaves and green fruits were very much alike. Teixido et al. (2012) describe a systematic approach of colour invariance to detect peaches using distances to linear colour models in the RGB colour cube. This enables them to distinguish peaches and leaves in more and less illuminated areas. However, if considerable overlap is present between the different organs in the colour cube (in our case, between the green leaves and green fruits), the linear colour model approach fails and more contextual information is needed to solve the problem. We tried to use correlation based templates to include this contextual information. Standard approaches were however still insufficient and detection rates were low. Therefore we switched to combination of MSCR and BoW which combines invariance, contextual information and advanced high dimensional clustering to obtain a reasonable rate of detection. Still the human vision is in this case superior.

Table 4 – Challenges addressed in this paper and a summary of our methods related to each challenge.

	Challenges	Relevant methods
1	Multiple plants in one image	Multiple views, Section 4
2	Plant spans across several images	Multiple views, Section 4
3	Complex fruit shape	Bag-of-words model, Section 3.2
4	High intraclass variation	Bag-of-words model, Section 3.2
5	Low interclass variation	Colour classifier, Section 3.1
6	Occlusions	Multiple views, Section 4

For finding initial points in our fruit recognition method, approaches other than using colour information could be taken, including corners (Leibe et al., 2004) and robust features (Leibe et al., 2005). In addition, other classifiers could be used instead of Naive Bayes to find the points. Which feature set and classifier works best is generally application specific, but many methods will yield similar results and the method as such is not critical as it is only used to reduce computer time.

We also proposed an algorithm for fruit counting using multiple views, and a correlation of 94.6% was achieved when fruits in images were visually identified, as shown in Fig. 8. This high correlation between estimated number $\widehat{K}_{VIS}$ and true number of fruits K demonstrates that image analysis has the potential to replace manual measurement. The algorithm could be extended to include features of identified fruits, such as shape, size and colour in addition to (x,y) locations to determine which are repeat views.

The pepper fruit considered in this paper is a difficult object to recognise, and there were the six challenges we faced, as shown in Table 4, that have not been tackled by Ji et al. (2012), Linker et al. (2012), Tanigaki et al. (2008) or De-An et al. (2011). For example, pepper fruit (Fig. 1) have a range of curved shapes unlike circle-shaped apples in Ji et al. (2012) and Linker et al. (2012). Moreover, pepper fruit have two distinctive colours and our images were captured under varying lighting conditions (intraclass variation), and the low contrast in colour between the green fruit and other plant parts (interclass variation) led to low precision (see Fig. 3). Our fruit recognition method therefore accounted for blob (i.e. MSCR) and texture features in addition to colour.

Fruit counting for a large number of plants (e.g. we had over 28,000 images recording over 1000 experimental plants) and its challenges (particularly the first two challenges in Table 4) have not been addressed before. We achieved a correlation of 74.2% as shown in Fig. 9, and $\widehat{K}$ was a good estimate of K without any linear adjustment. On average, there were 21.2 fruits in a plot, with a standard deviation of 16.3, while our prediction achieved a standard error of 11.3 fruits.

Although our fruit recognition and counting methods were designed for digital phenotyping, they can in principle be used for other robotics applications such as automatic harvesting. Using a standard PC (3G Hz processor with an 8 GB memory), the running time for our fruit recognition method (MATLAB) was under 10 s for a 480 × 1280 validation image without any reduction in resolution, and the multiple-view algorithm requires few seconds for counting.

In this paper, we have shown how to use the 'bag of words' (BoW) approach for recognising fruits with two distinctive colours and complex shapes in an image. Furthermore, a high-throughput imaging setup such as used by Polder et al. (2009) usually captures a continuous set of images or uses a video camera, which also causes bias in counting when the same fruit is visible in more than one image. To reduce the bias, we have described a multiple-view approach that can aggregate estimates from a number of images or video frames. The multiple-view algorithm also minimises the error caused by the occlusion problem, which improves the detection rate of the fruits. The BoW approach has already been successfully adopted in practical applications in other fields (Nilsback & Zisserman, 2006; Sivic & Zisserman, 2008), and we would like to draw the attention of the biological systems community to the potential use of this method.

7. Conclusions

Our conclusions are:

- Image analysis is one way to automate plant phenotyping, such as to count fruit numbers on plants.
- Our images are of 3-m high peppers with fruits of complex shapes and varying colours collected every 5 cm along greenhouse aisles.
- We have developed a two-stage method to locate and count fruits: 1) find fruits in single images using a bag-of-words model, 2) aggregate estimates from multiple images using a statistical approach to cluster repeated, incomplete observations.
- We achieved a correlation of 74.2% between automatic and manual counts of fruit numbers in 435 plots, whereas stage 2 alone, with stage 1 replaced by manual identification of fruits in images, achieved a correlation of 94.6%.

Acknowledgements

This work is part of the Smart tools for Prediction and Improvement of Crop Yield (SPICY) project supported by the European Community and funded by the KBBE FP7 programme. (Grant agreement number KBBE-2008-211347) We also acknowledge Scottish Government funding.

Appendix. Algorithm to estimate fruit numbers from multiple views

To estimate the parameters in the model specified by (1), we first consider all pairs of observations from the same experimental plot, indexed by (i_1,j_1) and (i_2,j_2), such that $i_1 < i_2$, and compute

$$\Delta = y_{i_2,j_2} - y_{i_1,j_1} \quad \text{and} \quad \widehat{\gamma} = \frac{x_{i_2,j_2} - x_{i_1,j_1}}{i_2 - i_1}.$$

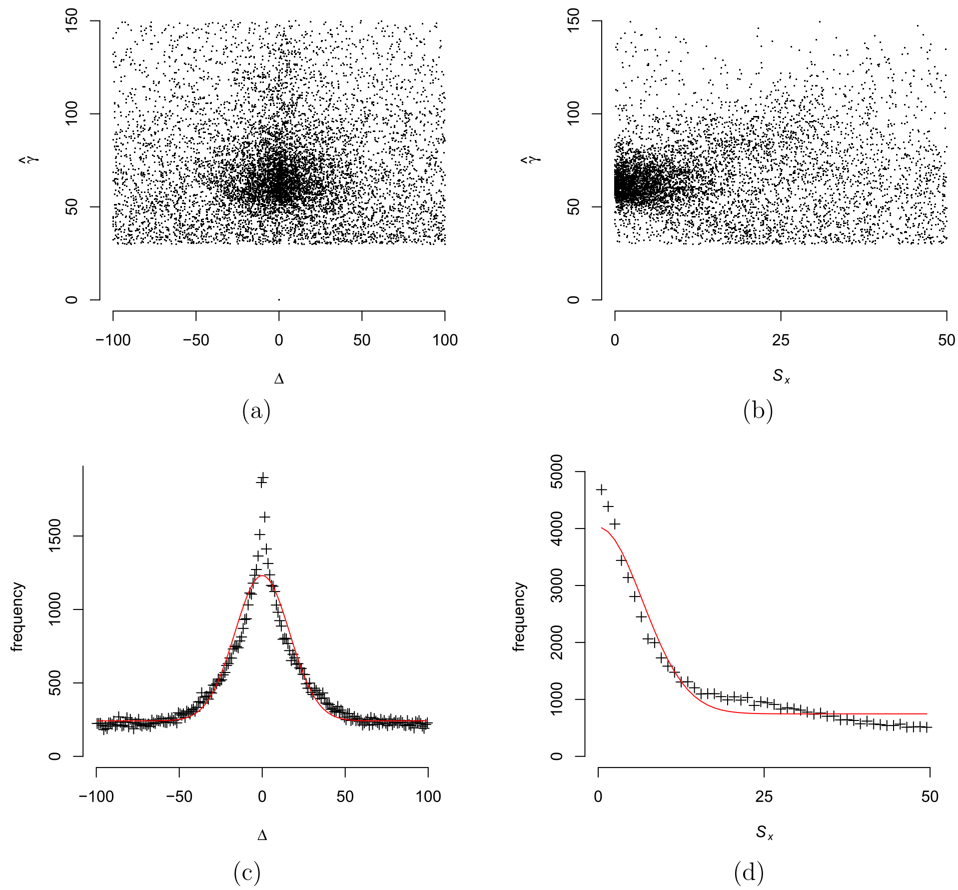


Figure A.1 – Derived data used to estimate $\hat{\sigma}_y^2$ and $\hat{\sigma}_x^2$: (a) differences between pairs of row coordinates (Δ) plotted against estimated shift ($\hat{\gamma}$) for restricted range of values; (b) square-root of residual sums of squares of model fit to triplets of column coordinates (S_x) plotted against estimated shift ($\hat{\gamma}$) for restricted range of value; (c) histogram of values of Δ and maximum likelihood fit (red line) of mixture of normal and uniform distributions; (d) histogram of values of S_x and maximum likelihood fit (red line) of mixture of half-normal and uniform distributions.

Figure A.1(a) shows Δ plotted against $\hat{\gamma}$ for the full dataset, restricted to $|\Delta| \le \Delta_M \equiv 100$ and $30 \le \hat{\gamma} \le 150$, which is a conservative range of values that γ can take for fruits, given the distance from the plants to the cameras. (For clarity, only a random 10% of data are plotted.) We see a cluster of values around $(\Delta,\gamma) \approx (0,60)$, which are likely to be repeat observations of the same fruit. Figure A.1(c) shows the histogram of Δ, which looks well approximated by a mixture of a normal and a uniform distribution, and this agrees with what we expect if distances between neighbouring fruits can be assumed to be approximately uniformly distributed:

$$(\Delta \mid |\Delta| \le \Delta_M) \sim \begin{cases} N\left(0, 2\sigma_y^2\right) & \text{if } k(i_1, j_1) = k(i_2, j_2) \\ U(-\Delta_M, \Delta_M) & \text{otherwise.} \end{cases}$$

We estimate the normal distribution variance ($2\sigma_y^2$) proportion (ρ) by numerically maximising the log-likelihood:

$$\sum_l \ln\left\{\frac{(1-\rho)}{2\Delta_M} + \frac{\rho}{\sqrt{4\pi\sigma_y^2}} \exp\left[-\frac{\Delta_l}{4\sigma_y^2}\right]\right\},$$

where summation over pairs l is restricted to the ranges in Figure A.1(a). Figure A.1(c) shows the fitted distribution, with $2\hat{\sigma}_y^2 = 16.50^2$. We note that there is some evidence for the distribution being more spiked than a normal, but we are not overly concerned with this discrepancy as statistical inference is usually robust to normality assumptions and use of other distributions would greatly complicate estimation to follow.

We next consider all observation triplets from the same plot. However, for subsequent useage, we will express this in the greater generality of n observations $\{(i_1,j_1),(i_2,j_2),\ldots,(i_n,j_n)\}$ such that $i_1<i_2<\ldots<i_n$. Given such a set, we can estimate (α,β,γ) by least squares, analytically using standard formulae, and we can also compute residual sums of squares S_x^2 and S_y^2. For row coordinates (y):

$$\hat{\beta} = \frac{1}{n}\sum_{l=1}^{n} y_{i_l j_l} \qquad S_y^2 = \sum_{l=1}^{n}\left(y_{i_l j_l} - \hat{\beta}\right)^2,$$

and for column coordinates (x):

$$(\hat{\alpha}, \hat{\gamma}) = \underset{(\alpha,\gamma)}{\arg\min} \sum_{l=1}^{n}\left(x_{i_l j_l} - \alpha - i_l\gamma\right)^2 \qquad S_x^2 = \sum_{l=1}^{n}\left(x_{i_l j_l} - \hat{\alpha} - i_l\hat{\gamma}\right)^2.$$

We note that, if the set are all repeat observations of the same fruit, then $S_y^2 \sim \sigma_y^2\chi_{n-1}^2$ and $S_x^2 \sim \sigma_x^2\chi_{n-2}^2$, from which it follows that:

$$S^2 = \left(\frac{S_x^2}{\hat{\sigma}_x^2} + \frac{S_y^2}{\hat{\sigma}_y^2}\right) \sim \chi_{2n-3}^2$$

In particular, for a triplet (n=3), $S_x^2 \sim \sigma_x^2 \chi_1^2$, so $S_x \sim N^+(0, \sigma_x^2)$, the positive half of a normal distribution.

Figure A.1(b) shows a plot of S_x against $\hat{\gamma}$ for triplets from all experimental plots in the dataset, restricted to $S_x \le S_M \equiv 50$ and $30 \le \hat{\gamma} \le 150$. We also restrict to $S_y^2 \le \hat{\sigma}_y^2 \chi_2^2(95\%)$ to ensure that values of y are consistent with repeat observations of a single fruit, at a 95% level of significance. (For clarity, again only a random 10% of data are plotted.) Similar to Δ, the data are consistent with

$$\left(S_x \middle| S_x \le S_M\right) \sim \begin{cases} N^+(0, \sigma_x^2) & \text{if } k(i_1, j_1) = k(i_2, j_2) = k(i_3, j_3) \\ U(0, S_M) & \text{otherwise.} \end{cases}$$

Figure A.1(d) shows the histogram of S_x from the full dataset, and the maximum likelihood fit, with $\hat{\sigma}_x^2 = 6.67^2$. Again, there is some evidence for the distribution being more spiked than a normal, which again does not overly concern us. We also note that $\hat{\sigma}_x^2 < \hat{\sigma}_y^2$, indicating that column locations of fruit are more easily determined than row locations.

Now we have estimates of the 2 variance parameters, we can consider data from each experimental plot separately to estimate K. Although it is possible to estimate the number of pairs and triplets of observations from the same fruit, it is not possible to extend this to direct estimation of K. Instead, by the following algorithm we can identify sets of observations which are inferred to have been of a single fruit, at a 95% level of significance. For each plot:

1. Initialise set size $n \leftarrow (I+1)$ and estimated number of fruits $\hat{K} \leftarrow 0$;
2. Find the set of size n, $\{(i_1,j_1),(i_2,j_2),\ldots,(i_n,j_n)\}$, which minimises S^2, subject to $i_1 < i_2 < \ldots < i_n$ (but no contiguity constraint) and $50 \le \hat{\gamma} \le 130$ (Note, we use this realistic range of values for γ, rather than the conservative range in Figure A.1);
3. If $S^2 \le \chi_{2n-3}^2(95\%)$ then accept this set, remove the n data points from further consideration, $\hat{K} \leftarrow (\hat{K}+1)$, and return to step 2;
4. $n \leftarrow (n-1)$, and return to step 2 provided $n \ge 2$;
5. $\hat{K} \leftarrow (\hat{K} + \text{number of remaining singletons})$.

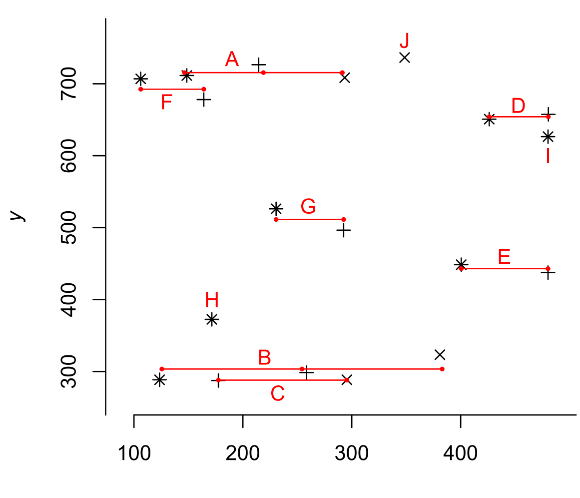


Figure A.2 – Data in Fig. 6, showing sets of points identified by algorithm, with ‘A’…‘J’ denoting identification order (see text), and fitted values (red dots).

The algorithm is fast because in step 2 most sets can be excluded with fewer than n points, because S^2 increases monotonically as points are added to a set. So a tree search can be used. Figure A.2 shows the results of the algorithm applied to the data in Fig. 6. As there are only 3 images in this illustrative example, we start with $n = 3$. The group marked ‘A’ are the first identified, with $S^2 = 2.0$, followed by ‘B’ with $S^2 = 5.3$. No other triple of observations remaining has $S^2 \le \chi_3^2(95\%) = 7.8$, so we then search for sets of size $n = 2$. We find 5 pairs, labelled ‘C’…‘G’ with increasing values of $S^2 \le \chi_1^2(95\%) = 3.8$. Figure A.2 also shows the fitted values. Three unassigned points remain, singletons labelled ‘H’,‘I’,‘J’, and we infer that the total number of observed fruit to be $\hat{K} = 2 + 5 + 3 = 10$.

REFERENCES

Alimi, N., Bink, M., Dieleman, J., Nicola, M., Wubs, M., Heuvelink, E., et al. (2013). Genetic and QTL analyses of yield and a set of physiological traits in pepper. *Euphytica, 190*, 181–201.

Breitenstein, M., Reichlin, F., Leibe, B., Koller-Meier, E., & Van Gool, L. (2009). Robust tracking-by-detection using a detector confidence particle filter. In *IEEE 12th international conference on computer vision* (pp. 1515–1522).

Brosnan, T., & Sun, D.-W. (2004). Improving quality inspection of food products by computer vision – a review. *Journal of Food Engineering, 61*(1), 3–16.

Bulanon, D., Kataoka, T., Ota, Y., & Hiroma, T. (2002). A segmentation algorithm for the automatic recognition of fuji apples at harvest. *Biosystems Engineering, 83*(4), 405–412.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *IEEE computer society conference on computer vision and pattern recognition* (pp. 886–893).

De-An, Z., Jidong, L., Wei, J., Ying, Z., & Yu, C. (2011). Design and control of an apple harvesting robot. *Biosystems Engineering, 110*(2), 112–122.

Ferrari, V., Fevrier, L., Jurie, F., & Schmid, C. (2008). Groups of adjacent contour segments for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 30*(1), 36–51.

Forssén, P.-E. (2007). Maximally stable colour regions for recognition and matching. In *IEEE conference on computer vision and pattern recognition* (pp. 1–8).

Furbank, R. T., & Tester, M. (2011). Phenomics – technologies to relieve the phenotyping bottleneck. *Trends in Plant Science, 16*(12), 635–644.

Gevers, T., & Smeulders, W. M. (1999). Color-based object recognition. *Pattern Recognition, 32*, 453–464.

Jimenez, A., Jain, A., Ceres, R., & Pons, J. (1999). Automatic fruit recognition: a survey and new results using range/attenuation images. *Pattern Recognition, 32*(10), 1719–1736.

Ji, W., Zhao, D., Cheng, F., Xu, B., Zhang, Y., & Wang, J. (2012). Automatic recognition vision system guided for apple harvesting robot. *Computers & Electrical Engineering, 38*(5), 1186–1195.

Kitamura, S., & Oka, K. (2005). *Recognition and cutting system of sweet pepper for picking robot in greenhouse horticulture.* In *IEEE international conference on mechatronics and automation* (Vol. 4); (pp. 1807–1812).

Lee, W., Alchanatis, V., Yang, C., Hirafuji, M., Moshou, D., & Li, C. (2010). Sensing technologies for precision specialty crop production. *Computers and Electronics in Agriculture, 74*(1), 2–33.

Leibe, B., Leonardis, A., & Schiele, B. (2004). Combined object categorization and segmentation with an implicit shape

model. In *ECCV workshop on statistical learning in computer vision* (pp. 17–32).
Leibe, B., Seemann, E., & Schiele, B. (2005). Pedestrian detection in crowded scenes. In *IEEE computer society conference on computer vision and pattern recognition* (pp. 878–885).
Linker, R., Cohen, O., & Naor, A. (2012). Determination of the number of green apples in rgb images recorded in orchards. *Computers and Electronics in Agriculture, 81*, 45–57.
Matas, J., Chum, O., Martin, U., & Pajdla, T. (2002). Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of the British machine vision conference* (pp. 384–393).
Nilsback, M.-E., & Zisserman, A. (2006). *A visual vocabulary for flower classification*. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (Vol. 2); (pp. 1447–1454).
Polder, G., van der Heijden, G. W. A. M., Glasbey, C. A., Song, Y., & Dieleman, J. A. (2009). Spy-See - advanced vision system for phenotyping in greenhouses. In *Proceedings of the MINET Conference: Measurement, sensation and cognition* (pp. 115–117). National Physical Laboratory.
Reis, M. J. C. S., Morais, R., Peres, E., Pereira, C., Contente, O., Soares, S., et al. (2012). Automatic detection of bunches of grapes in natural environment from color images. *Journal of Applied Logic, 10*(4), 285–290.
Sivic, J., & Zisserman, A. (2008). Efficient visual search for objects in videos. *Proceedings of the IEEE, 96*(4), 548–566.
Song, Y., Glasbey, C. A., van der Heijden, G. W. A. M., Polder, G., & Dieleman, J. A. (2011). Combining stereo and Time-of-Flight images with application to automatic plant phenotyping. In *17th Scandinavian conference on image analysis, SCIA 2011* (pp. 467–478).
Stajnko, D., Lakota, M., & 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*(1), 31–42.
Tanigaki, K., Fujiura, T., Akase, A., & Imagawa, J. (2008). Cherry-harvesting robot. *Computers and Electronics in Agriculture, 63*(1), 65–72.
Teixido, M., Font, D., Palleja, T., Tresanchez, M., Nogues, M., & Palacin, J. (2012). Definition of linear color models in the RGB vector color space to detect red peaches in orchard images taken under natural illumination. *Sensors, 12*, 7701–7718.
van der Heijden, G., Song, Y., Horgan, G., Polder, G., Dieleman, A., Bink, M., et al. (2012). SPICY: towards automated phenotyping of large pepper plants in the greenhouse. *Functional Plant Biology, 39*(11), 870–877.
Yang, L., Dickinson, J., Wu, Q., & Lang, S. (2007). A fruit recognition method for automatic harvesting. In *14th International conference on mechatronics and machine vision in practice* (pp. 152–157).
Zhao, J., Tow, J., & Katupitiya, J. (2005). On-tree fruit recognition using texture properties and color data. In *International conference on intelligent robots and systems* (pp. 263–268).