Precision Agric (2014) 15:662–683
DOI 10.1007/s11119-014-9361-x

Texture-based fruit detection

Supawadee Chaivivatrakul · Matthew N. Dailey

Published online: 18 July 2014

Abstract In this paper, a technique based on texture analysis is proposed for detecting green fruits on plants. The method involves interest point feature extraction and descriptor computation, interest point classification using support vector machines, candidate fruit point mapping, morphological closing and fruit region extraction. In an empirical study using low-cost web camera sensors suitable for use in mechanized systems, 24 combinations of interest point features and interest point descriptors were evaluated on two fruit types (pineapple and bitter melon). The method is highly accurate, with single-image detection rates of 85 % for pineapples and 100 % for bitter melons. The method is thus sufficiently accurate for precise location and monitoring of textured fruit in the field. Future work will explore combination of detection and tracking for further improved results.

Keywords Texture analysis · Object detection · Fruit detection · Keypoint extraction · Keypoint descriptors · Support vector machine classification

Introduction

Fruit crops, due to their high value, stand to gain the most from the techniques of precision agriculture. In the limit, if it is possible to map and track individual fruits in a given field, the farmer would be presented with a wealth of information for decision support at high resolution, down to the individual plant. However, new advances in sensors and sensor information processing are needed to take precision agriculture to this level.

S. Chaivivatrakul (✉) · M. N. Dailey
Computer Science and Information Management, Asian Institute of Technology, P.O.Box 4, Klong Luang, Pathumthani 12120, Thailand
e-mail: supawadee.chaivivatrakul@ait.ac.th

M. N. Dailey
e-mail: mdailey@ait.ac.th

One aspect of sensor information processing requiring improvement is fruit detection and mapping in the field. It is difficult and costly to develop mechanized sensory systems for uncontrolled outdoor environments. With some notable exceptions, the research that does exist has rarely been commercialized, due primarily to slow speeds and high costs. This work thus focuses on the development of low cost techniques for real time or nearly real time fruit detection in the field.

Image processing-based post-harvest fruit and vegetable analysis is a robust area for research in computer vision and machine learning. The main applications involve characterization of fruit by features such as type, size, quality and maturity (Cubero et al. 2011; Slaughter et al. 2008; Rocha et al. 2010; Du and Sun 2006; Zhang and Wu 2012). For example, Cubero et al. (2011) found it effective to use ultraviolet or near-infrared spectral images to evaluate the quality of fruits and vegetables. Slaughter et al. (2008) used ultraviolet fluorescence images to detect freeze-damaged oranges. Rocha et al. (2010) used color, texture and structural appearance information to classify fruit types. In the area of post-harvest analysis of bitter melon, Torii et al. (2009) proposed a technique to acquire 3D shape data from web camera images of complex objects including bitter melons. For pineapple, Kaewapichai et al. (2006) employed a technique based on pineapple skin color for maturity grading. Kaewapichai et al. (2007) further enhanced their earlier work using active contour methods.

Post-harvest methods, though effective in their scope, are not designed to deal with the issues of occlusion, variable pose, close proximity and variable lighting that arise in the field, making in-field analysis much more challenging than post-harvest analysis. A method to apply in-field shape analysis to detection of tomato plants in early growth stages was proposed by Lee et al. (1999). Texture analysis has also been applied to in-field data; for example, Delenne et al. (2008) used texture processing of remote sensing data to identify vine crop areas. Pla and Marchant (1997) applied a close-view region and feature matching method for image series captured from video enabling a crop protection vehicle to detect and track plant rows. These image analysis techniques could be adapted to in-field textured fruit detection with some adjustment.

Automatic in-field fruit analysis systems are faced with the difficult challenge of detecting fruit on the plant in an unpredictable outdoor environment. The outdoor setting requires detection and analysis methods that are robust to pose and lighting variability and partial occlusion. Several groups of researchers have applied their work to detect fruits on plants. Non-green mature fruit (e.g. apples, oranges, peaches and yellow mature pineapple) and flowers can be segmented from leaves and other plant material (though not necessarily from adjacent fruit) using color (Payne et al. 2013; Li et al. 2010; Zhou et al. 2012; Aggelopoulou et al. 2011). For example, Payne et al. (2013) proposed a method to detect mature mangoes on trees. Their method calculates pixel properties from RGB and YCbCr color spaces then applies a cascade of filters to progressively classify non-fruit pixels as background and fruit pixels as foreground, producing a binary mask. Then connected components, subject to size constraints, are counted. However, green fruit like sweet peppers, cucumbers, green pineapples and bitter melons are more challenging. Some researchers have attempted more sophisticated color analysis to distinguish green fruit from leaves and other background (Zhang et al. 2007; Kitamura and Oka 2005). Van Henten et al. (2002) used a reflectance filter sensor to detect cucumbers. Jiménez et al. (2000a, b) concluded that while color- and intensity-based classification is useful for non-green fruit, and while shape analysis, though time consuming, is useful for green fruit, range image processing works well for both green and non-green fruit. Range sensor images are useful because they provide reliable shape information even when fruit are partly or mostly occluded, and they are also very useful for separating fruits from the rest of a plant. Nevertheless, range sensors are generally larger and much more expensive than video sensors. Range sensors vary in price from 100 USD to thousands or more, but prices for video sensors such as web cameras begin at a mere 10 USD.

Bansal et al. (2012) proposed a technique based on symmetry for detecting immature green citrus fruits on trees. They assumed that the fruits have smooth spherical surfaces, making them symmetrical. Thus their method searches for objects that have a symmetric intensity profile about a vertical axis. The method works well for smooth symmetrical objects embedded among asymmetrical background objects and tolerates some occlusion of fruit boundaries. However, detecting non-smooth (textured) or asymmetric green fruits would require different techniques.

Fruit quality determines the price a farmer will receive for a crop, and crop monitoring at the level of the individual plant, combined with precision agriculture, can help the farmer maximize quality. However, detailed monitoring of large fields at the individual plant level by humans would require an exorbitant amount of labor. We are thus investigating the applicability of machine vision techniques to support automated monitoring of large crops, with a focus on fruit that are green at maturity. Pineapple and bitter melon are two such fruits with great economic significance in Thailand.

Methods and materials

In view of cost concerns, the real time requirement for eventual industrial application, and the ineffectiveness of shape cues for green fruit detection in the field, this work focuses on texture-based detection of green fruits on plants in the field using low cost video sensors such as web cameras. The work uses web cameras because they are cheap and easily controlled by desktop or embedded system software implemented in C/C++. Videos containing two types of green fruit were collected for the experiment. The first type is the popular and healthy Smooth Cayenne variety of pineapple (*Ananas comosus*). The second is the bitter melon (*Momordica charantia*), widely eaten and used for medicinal purposes in Asia. Both fruits are major crops that are yellow when very ripe but still green up to harvest time in Thailand.

The three main approaches to detection of fruit on plants in images are color analysis, shape analysis and texture analysis. Pineapple and bitter melon are both green fruit covered with distinctive texture. In the field, fruit shape boundaries are frequently occluded by other plant material. These features make color-based and shape-based detection difficult, but since the fruit surface texture is quite different from the rest of the plant, texture-based detection is feasible. Texture analysis techniques generally extract statistics on local gradients around image points. The experiments in this paper thus evaluate the utility of several point selection and local-gradient-based texture descriptors for discriminating fruit surfaces from the background. The point selector is typically called a feature detector, keypoint detector or interest point detector, and the vector of local gradient statistics for a particular feature point is typically called a feature descriptor, keypoint descriptor or interest point descriptor.

The features and descriptors were selected from among the methods well-known to be effective from the literature. Harris feature extraction (Harris and Stephens 1988) is a simple method returning corner-like pixels as features. SIFT (Lowe 2004), SURF (Bay et al. 2008) and ORB (Rublee et al. 2011) are more modern techniques proposed in the last decade to solve a variety of machine vision problems such as matching and object recognition. They are competitive in terms of accuracy, runtime and resource usage. Harris is only a feature detector. SIFT, SURF and ORB combine feature detection and feature descriptor calculation. These feature detection and feature descriptor calculation methods are explained in more detail in "Interest point features and interest point descriptors" section.

In this work, the system is trained to discriminate the fruit and background feature points (a point is a position in the image plane, which will generally lie between four pixel locations) in a training set and is then evaluated by its ability to generalize to a test set. The support vector machine (SVM; Vapnik 1995) is the classifier used for the discrimination task. The SVM induces a non-linear function predicting a feature point's category (fruit or non-fruit) based on the feature point descriptor as a vector of input variables.

The feature detector, feature descriptor and classification techniques work at the individual pixel level, with a need to merge the results for neighboring pixels to get candidate fruit regions. In image processing, mathematical morphology techniques are commonly used to solve this problem.

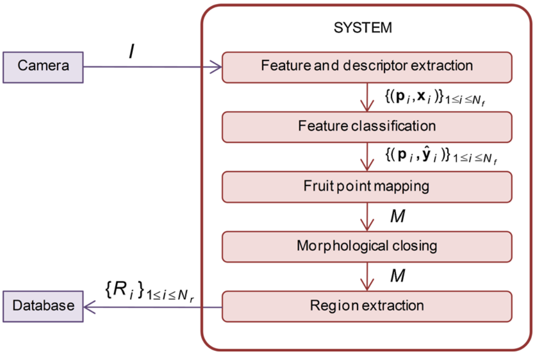
Based on the above analysis, the paper evaluates methods for detection of green fruits on plants that involve interest point feature extraction and descriptor computation, interest point classification using SVMs, candidate fruit point mapping, morphological closing and fruit region extraction. The experiment reported upon in this paper explores the ability of 24 combinations of interest point feature selectors and interest point descriptors to detect two fruit types.

Overview

The system works with video data that can be converted into a series of single images. The video frames were selected manually for purposes of experimentation. However, in a real commercial application, every frame of the video would be processed, so there would be no need for a selection process. For each input image, features are extracted, descriptors are calculated and then the descriptors are classified by a pre-trained classifier. The features predicted as fruit points are used to create a binary image indicating the locations of candidate fruit features and non-fruit points (background pixels). Typically, the candidate fruit features are tightly clustered on the fruit surface (true positive points) and spread sparsely on the background (false positive points). The tightly clustered true positive points can be merged into regions using the morphological closing operation. In some cases, a fruit's visible surface is broken into more than one region due to occlusion by leaves. When morphological closing is unable to merge such regions into a single detected region, the detection will be classified as a fragmentation error (see "Results" section for details). False positive points may also be merged into small regions, but these can be removed using a region size threshold. Figure 1 and the following algorithm summarize the processing performed by the system in brief.

1. Acquire input image I.
2. Apply interest point operator to obtain N_f candidate feature points $\{\mathbf{p}_i\}_{1 \leq i \leq N_f}$, where N_f is the number of features found in image I.
3. For each feature point $\mathbf{p}_i$, obtain a feature descriptor vector $\mathbf{x}_i$.
4. For each feature descriptor vector $\mathbf{x}_i$, obtain predicted label $\hat{\mathbf{y}}_i$ (1 for positive or 0 for negative).
5. Create a binary image M the same size as I and set all pixels to 0.
6. For each $\mathbf{p}_i$ for which $\hat{\mathbf{y}}_i = 1$, set $M(\mathbf{p}_i)$ to 1.

Fig. 1 System overview


SYSTEM
Camera
I
Feature and descriptor extraction
$\{(\mathbf{p}_i, \mathbf{x}_i)\}_{1\le i\le N_f}$
Feature classification
$\{(\mathbf{p}_i, \hat{\mathbf{y}}_i)\}_{1\le i\le N_f}$
Fruit point mapping
M
Morphological closing
M
Region extraction
$\{R_i\}_{1\le i\le N_r}$
Database

7. Perform morphological closing with appropriately shaped structuring element on M.
8. Extract connected components from M and return large positive regions $\{R_i\}_{1\le i\le N_r}$ as output, where N_r is the number of regions surviving size thresholding.
9. Repeat steps 1–8 for each image.

Interest point features and interest point descriptors

In the experiment, the 24 options combining interest point extraction (Mesh, Harris, SIFT, SURF, ORB, and IORB) and interest point descriptor computation (SIFT, 64-element SURF, 128-element SURF, and ORB) are abbreviated as Mesh+SIFT, Mesh+SURF64, Mesh+SURF128, Mesh+ORB, Harris+SIFT, Harris+SURF64, Harris+SURF128, Harris+ORB, SIFT+SIFT, SIFT+SURF64, SIFT+SURF128, SIFT+ORB, SURF+SIFT, SURF+SURF64, SURF+SURF128, SURF+ORB, ORB+SIFT, ORB+SURF64, ORB+SURF128, ORB+ORB, IORB+SIFT, IORB+SURF64, IORB+SURF128, and IORB+ORB.

Szeliski (2011) defines a “feature point,” “keypoint feature” or “interest point” as a point that has strong, complex local gradient compared to the surrounding points. Examples of good feature point include mountain peaks, building corners and doorways. He defines a feature point, keypoint or interest point “descriptor” or “descriptor vector” as a description of a detected feature calculated from a region around the feature that provides a compact and stable description of the region and can be matched to or grouped with other descriptors.

The “mesh” feature approach is a very simple baseline method that picks features in a mesh of specific locations over the image, without considering any pixel properties. For example, for an image with size 640 × 480 pixels, the method takes feature points every 11 pixels horizontally and vertically, returning a grid of 27840 feature points over the image. This method has the benefit of very small computational requirements.

Harris features are simple corner features with complex local gradient structure. There is no computation for multiple scales, smoothing levels or orientations.

SIFT, SURF and ORB feature extraction is more sophisticated. These interest point detectors return points at multiple scales and smoothing levels with orientations. A point is a position in the image plane which does correspond to some pixel. ORB is the fastest, SIFT is the slowest and SURF requires an intermediate amount of runtime. SIFT features are local extrema of difference of Gaussian filters. SURF approximates the SIFT detector with box filters that are faster to evaluate. The ORB detector is an oriented version of the

FAST corner detector (Rosten and Drummond 2006), which searches for high-gradient points.

The descriptor calculation for these methods takes a region around the feature point, divides it into sub-regions, then returns a vector of numbers to represent gradient magnitudes and orientations over the region. SIFT computes the gradient orientations and magnitudes directly. SURF approximates gradients using block differences. Two options for the SURF descriptor were included: SURF64 and SURF128. SURF64 is the default descriptor algorithm, producing a 64-element output. SURF128 extends the SURF descriptor to 128 elements. ORB calculates an improved version of the BRIEF descriptor (Calonder et al. 2010) based on binary tests between pixels in a smoothed patch around the feature point. IORB (“Improved ORB”) is a simple optimization to ORB that adjusts the non-maximum-suppression mask size and the FAST feature detection threshold for data sets with weak (relatively low gradient) feature points.

For all feature types, thresholds were tuned to retrieve approximately the same number of features over the training set. The result of this step is a set $\{(\mathbf{p}_i, \mathbf{x}_i)\}_{1 \le i \le N_f}$ of feature points (Mesh, Harris, SIFT, SURF, ORB or IORB) paired with feature descriptors (SIFT, SURF64, SURF128 or ORB).

Feature descriptor classifier

After computing feature points and descriptors, each descriptor is classified using a SVM classifier that was trained offline. For SVM training, feature points and descriptors were extracted from a set of training and test images, then each feature point was hand-labeled as positive (the feature point is on a fruit’s surface) or negative (the feature point is on a plant or the background). Note here that the image region used to compute the feature descriptor may overlap the boundary between a fruit and the background. There is no special processing to prevent this one. Once the set of training and test descriptors is formed, the training set is randomly subsampled subject to a constraint of balance between positives and negatives. Then a series of SVMs (Vapnik 1995) is trained to perform a grid search using *k*-fold cross validation in the SVM’s hyper-parameter space, to find the best set of hyper-parameters in terms of overall F1 score over the cross validation test sets.

k-fold cross validation is a method for classifier validation. First, the data are divided into *k* partitions (groups), onefold is taken as a validation set, and then the classifier is trained on the remaining *k*-1 folds and tested on the validation fold. The procedure is executed *k* times, each time with a different validation set, and the result is the average validation set classification accuracy over *k* rounds. This process is repeated for each combination of feature point extraction and feature descriptor methods, and the method and hyper-parameters obtaining the best cross validation test set performance are selected. Finally, the selected method and hyper-parameters are used in an evaluation on the final test set.

At runtime, the resulting pre-trained SVM model is simply loaded into memory and used to classify each feature point in the input image as positive or negative. The result of this step is a predicted label $\hat{\mathbf{y}}_i \in \{0, 1\}$ for each feature point $\mathbf{p}_i$.

Fruit region extraction

For each input image, after feature location identification, feature descriptor computation and feature classification using the pre-trained SVM, the points classified as positives (on a fruit surface) are used to construct a binary image M in which each pixel corresponding to a positive is set to 1 and all other pixels are set to 0. Then morphological operations are used to connect regions with dense positive detections and discard regions too small to be considered candidate fruit regions. In the experiments reported on in this paper, morphological closing with a disc- or ellipse-shaped structuring element is used. The result of the detection process for image I is a set of fruit regions $\{R_i\}_{1 \leq i \leq N_r}$.

Results

Experimental design

In order to compare the various texture classification methods for fruit detection, two fruit types, pineapple and bitter melon, were chosen. These fruit types are green (precluding color-based classification, both fruit types are yellow when very ripe but still green up to harvest time) and have characteristic texture with the potential for automatic texture-based detection. The different feature detector and descriptor methods were evaluated on each fruit type along two dimensions. The first evaluation is how well each classifier discriminates fruit interest points and non-fruit interest points. The second evaluation is the quality of the resulting detected fruit regions in terms of true positives, false negatives and false positives.

Region detection rates require additional attention (Nascimento and Marques 2006). After feature point classification, morphological processing and connected components, any candidate fruit region smaller than some (fruit-type dependent) number of pixels is discarded. The detected fruit regions are compared to a manually created ground truth dataset for the same images. A ground truth region is recorded as a *hit* or true positive whenever there exists a detected region such that the intersection of the detected region and the ground truth region is at least a threshold proportion of the detected region and a threshold proportion of the ground truth. A *false positive* is recorded whenever a detected region cannot be counted as a hit for any ground truth region. A *miss* or false negative is recorded for any ground truth region for which no hit was obtained. Among hits, there are two special cases. *Fragmentations* are ground truth regions that have two or more detected regions qualifying as hits. *Merges* are ground truth regions that have one or more detected regions qualifying as a hit that also qualify as a hit for a different ground truth region. The following results analysis will focus on detection rates and detection error rates for fruit, not emphasizing the background, so true negatives will not be reported.

Data collection

Video data were collected of 240 pineapples on two sub-farms and 120 bitter melons on one sub-farm at 30 and 15 frames/s, respectively, from separate farms with Logitech C200 web cameras and a laptop computer. The video frame resolution was 640 × 480 pixels (providing a resolution of approximately 1 mm per pixel on the fruit surfaces closest to the camera and about 3 mm per pixel on the fruit surfaces furthest from the camera). The velocity was about 1 m/min. In some cases, additional fruits beyond the 240 and 120 fruits of interest were visible in the video data.

The pineapple data were collected at a farm in Chonburi, Thailand; see Fig. 2a. A specialized mobile cart was built to carry the cameras and laptop computer for pineapple data collection as shown in the photographs. The version shown is pushed by hand; mechanizing the cart is envisioned in future work. The tarpaulin covering the cart helps to prevent saturation by sunlight.

The bitter melon video data were collected at a farm in Pathumthani, Thailand; see Fig. 2b. Bitter melon is grown under high-humidity conditions over furrow irrigation channels. This precludes effective use of a mobile cart, so the data were collected by holding the web camera by hand. In this case, protection from the sun was unnecessary due to the dense leaf coverage.

Feature points and feature descriptors

Feature points were extracted using the Mesh, Harris, SIFT, SURF, ORB and IORB algorithms, and feature point descriptors were computed using the SIFT, SURF64, SURF128 and ORB algorithms. The OpenCV library (OpenCV Community 2013) was used for all algorithms without modification except IORB, which required modification of the OpenCV source code to change the mask size for non-maximum suppression and the threshold for FAST feature detection. To avoid bias due to different algorithms returning different numbers of features, parameter sets were chosen for each of the feature point detection algorithms so that each method returned approximately the same number of feature points over the training data. The settings used for each algorithm are shown in Tables 1 and 2. Pineapple has more texture overall than bitter melon, so in some cases, such as Harris feature point detection, a higher threshold for pineapple images was necessary in order to balance the number of feature points returned by the algorithms over the different fruit types. In all cases, the interest points within 36 pixels of the image border were ignored. There is no report of results with ORB feature extraction on bitter melon because the bitter melon texture was too weak to be detected by ORB's default detector. IORB feature extraction, on the other hand, returns a sufficient number of features on the bitter melon video data. Figure 3 shows feature extraction results for sample pineapple and bitter melon images. Figure 4 shows the average and standard deviation of the runtime required for each of the 24 feature extraction and description methods.

SVM feature classifier

First, the images for each fruit type were divided into a training set (200 images for pineapple and 100 images for bitter melon) and a final test set (40 images for pineapple and 20 images for bitter melon). Then feature points and descriptors were extracted according to each of the 24 previously-described methods and hand-labeled as positive or negative. A summary of the distribution of positive and negative features on the training and test sets for each of the four feature methods and two fruit types is shown in Table 3.

Cross validation was performed to find the best SVM hyper-parameter settings for each feature/descriptor combination. For each combination, 2 000 descriptors (1 000 positive and 1 000 negative) were selected randomly from the training set. The experiment used the LIBSVM (Chang and Lin 2001; Muller et al. 2001) RBF classifier, which requires two parameters, γ and C; γ adjusts the RBF kernel width, and C adjusts the penalty for incorrectly-classified training data during the optimization. Fivefold cross validation was run for each (γ, C) pair to find optimal values of these parameters. The data selection is over all three sets: full (unbalanced) training set, balanced training set (a randomly-sampled subset of the full training set) and test set. The pineapple had been planted in a row system, and the data were collected from two subfarms. For each subfarm, 20 fruits were

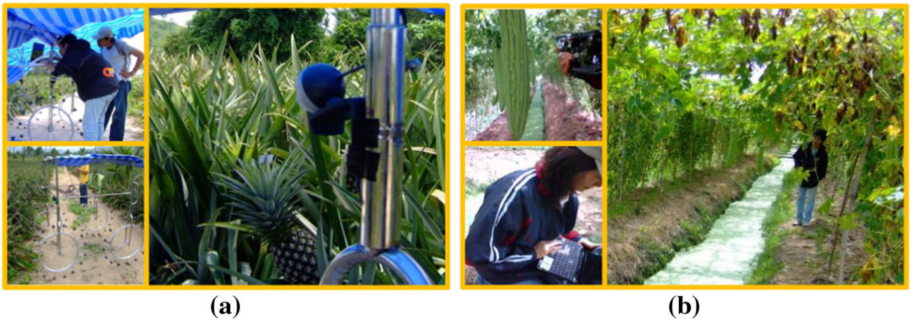


Fig. 2 Video data collection. **a** Pineapple video collection using a specialized pushcart carrying the cameras. **b** Bitter melon video collection using a hand-held web camera

Table 1 Feature extraction parameter settings

Feature	Parameters	Pineapple	Bitter melon
Mesh	Step size	11	9
Harris	Block size	3	5
	Free parameter or constant	0.004	0.0004
	Threshold	0.0018	0.0000015
	Minimum possible Euclidean distance	0	4
SIFT	Threshold	0.035	0.0325
	Number of octaves	1	2
	Number of octave layers	4	2
	First octave	0	0
	Angle mode	First angle	First angle
SURF	Hessian threshold	60	58
	Number of octaves	1	2
	Number of octave layers	4	4
ORB	Number of levels	1	No experiment
	Scale factor	–	–
	First level	0	–
	FAST threshold	20	–
	Non-maximum suppression mask	3 × 3	–
IORB	Number of levels	2	2
	Scale factor	1.2	1.2
	First level	0	0
	FAST threshold	18	12
	Non-maximum suppression mask	9 × 9	11 × 11

selected from each of six rows. The bitter melon data were collected arbitrarily. For both fruit types, the subjective best view of each individual fruit was selected and captured from the video stream as a single image. However, some fruits appeared in multiple images due to proximity to others. Figure 5 shows a diagram providing more detail of the image capture and classification process.

Table 2 Feature descriptor parameter settings

Descriptor	Parameters	Pineapple	Bitter melon
SIFT	Magnification	1	1
	Normalization	True	True
	Recalculate angle	False	False
	Number of octaves	4	1
	Number of octave layers	4	4
	First octave	0	0
SURF64	Number of octave layers	2	2
	Number of elements	64	64
SURF128	Number of octave layers	2	2
	Number of elements	128	128
ORB	Number of levels	1	2
	Scale factor	–	1.2
	First level	0	0

In each case, a loose grid search was performed with $\gamma \in \{2^{-15}, 2^{-13}, \ldots, 2^{3}\}$ and $C \in \{2^{-5}, 2^{-3}, \ldots, 2^{15}\}$, followed by a fine grid search in the neighborhood of the best loose search result. The final classifier's F1 score was used over all cross validation folds as the search criterion. Figure 3 shows an example classification result, with the features predicted positive drawn over the images.

The best combinations of feature and descriptor according to F1 score over the entire cross validation set were Mesh+ORB, SIFT+SIFT, IORB+SIFT and IORB+ORB for pineapples and Mesh+ORB, Harris+ORB, SIFT+ORB, IORB+SIFT and IORB+ORB for bitter melons. In both cases, the best combinations' F1 scores were 1.0 (perfect classification).

However, since the final target is to process feature points extracted from entire images, not balanced randomized data sets, the best cross validated SVM classifier for each feature/descriptor combination was then applied to the full (unbalanced) training and test sets. Numerically, in terms of F1 score over the full data sets, the best methods were ORB+SURF128 for pineapples, with $C = 2^{4.25}$ and $\gamma = 2^{-0.75}$, and Harris+SURF128 for bitter melons, with $C = 2^{13.25}$ and $\gamma = 2^{-5.25}$.

To analyze the reliability of the F1 results on the full training set, a statistical analysis was performed. The full training set was split into five partitions sequentially, assigning all of the point descriptors for a particular image to the same partition. A two-way analysis of variance (ANOVA) on the cross validation results for each classifier was performed separately for each fruit type. The independent variables were the feature point detector method and the feature point descriptor method, and the dependent variable was the best cross validated SVM's F1 score, measured separately for each partition of the full training set.

The ANOVA results are shown in Tables 4 and 5. With a Type I error threshold of $\alpha = 0.05$, the results showed a significant effect of feature detector type for both fruit types and a significant interaction only for bitter melon. Post-hoc comparisons were performed between selected means with the Tukey HSD correction for $\alpha = 0.05$; the tests indicated that for pineapple, the ORB feature detector is significantly better than the other feature

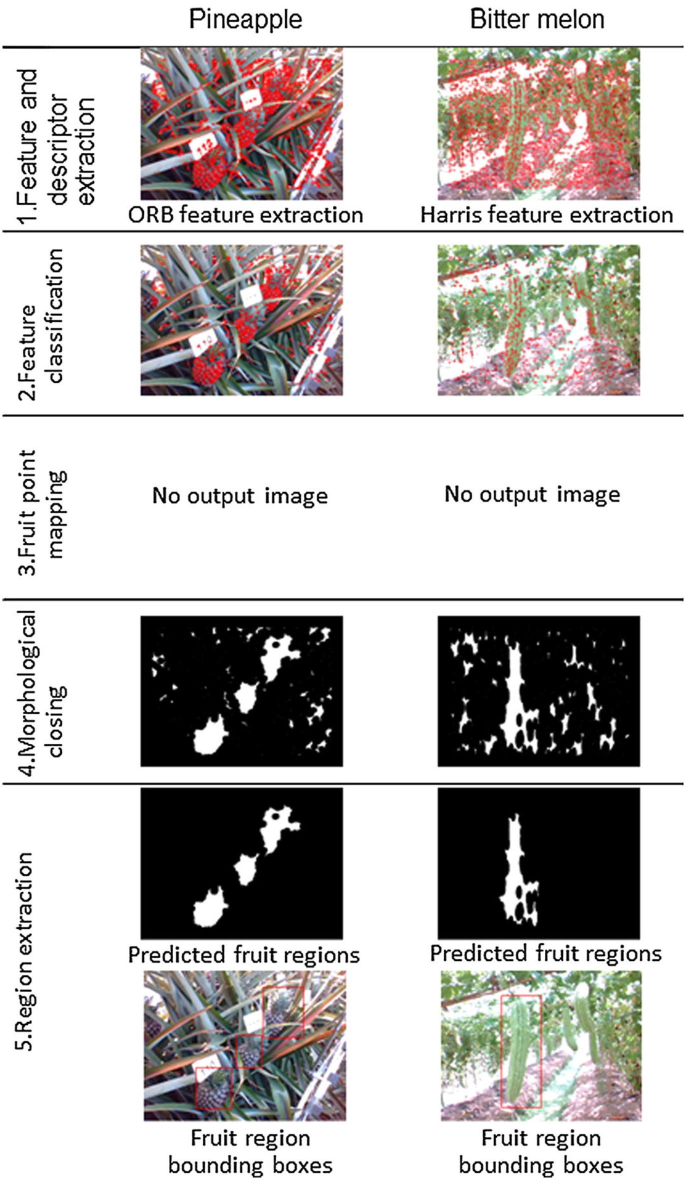


Fig. 3 System overview. Processing steps with sample outputs

detectors, whereas for bitter melon, the Harris feature detector is significantly better than the other feature detectors. Detailed results are shown in Fig. 6.

Note that there was no main effect of descriptor type for either fruit type. For bitter melon, there was a significant interaction; for example, with the Harris feature point detector, the SURF128 descriptor was significantly better than the ORB descriptor. But overall, the feature detector type had a much larger effect on performance than did the descriptor type.

Table 3 Distribution of positive and negative feature points over training and test data for each feature point and descriptor option

Dataset	Number of images	Proportion of positive instances (%)					
		Mesh	Harris	SIFT	SURF	ORB	IORB
Pineapple							
Training Set	200	6.8	24.4	16.7	8.7	27.5	21.0
Test Set	40	6.4	21.6	13.5	8.0	21.5	18.5
Bitter melon							
Training Set	100	14.6	14.3	11.7	14.6	–	9.5
Test Set	20	12.4	12.0	9.7	12.3	–	8.0

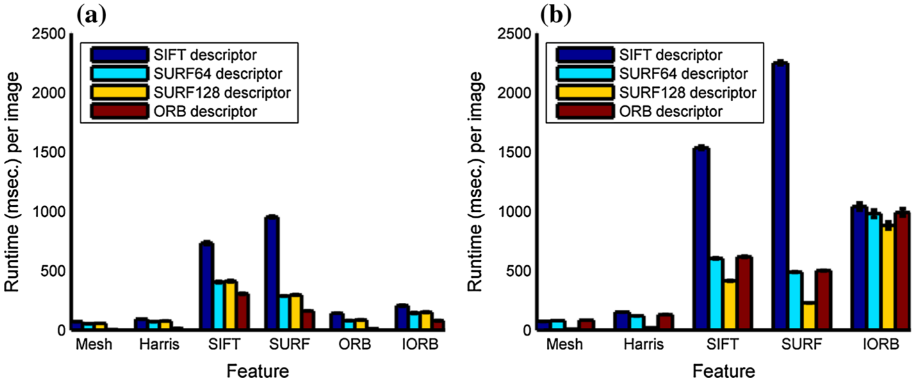


Fig. 4 Average runtime required for each texture classifier. **a** Pineapple. **b** Bitter melon. *Error bars* denote one standard deviation over the image set

To further understand the texture classifiers' performance, Kolmogorov–Smirnov (KS) analyses were performed for the two classifiers on both the training and test sets. Cumulative SVM score distributions for positive and negative feature points are shown in Fig. 7 and Table 6. The analysis shows first that the SVM classifiers trained on the 2000-point cross validation dataset were successful at separating descriptors of positive and negative instances on the full training and test sets. It furthermore shows that the SVM decision thresholds are nearly optimal in terms of KS (the best separation of scores for the unbalanced positive and negative distribution occurs in each case very close to the SVM decision threshold of 0). The analysis finally shows a relatively high degree of overlap between the positive and negative score distributions on the full test sets, with relatively low KS scores of 0.23–0.32.

Expanding the KS analysis to the classifiers for other feature/descriptor combinations, the analysis showed that KS scores for the unbalanced test set generally tracked the cross-validated F1 scores on the full (unbalanced) training set. This leads to the conclusion that ORB+SURF128 and Harris+SURF128 are (numerically) the best methods for pineapple texture and bitter melon texture, respectively.

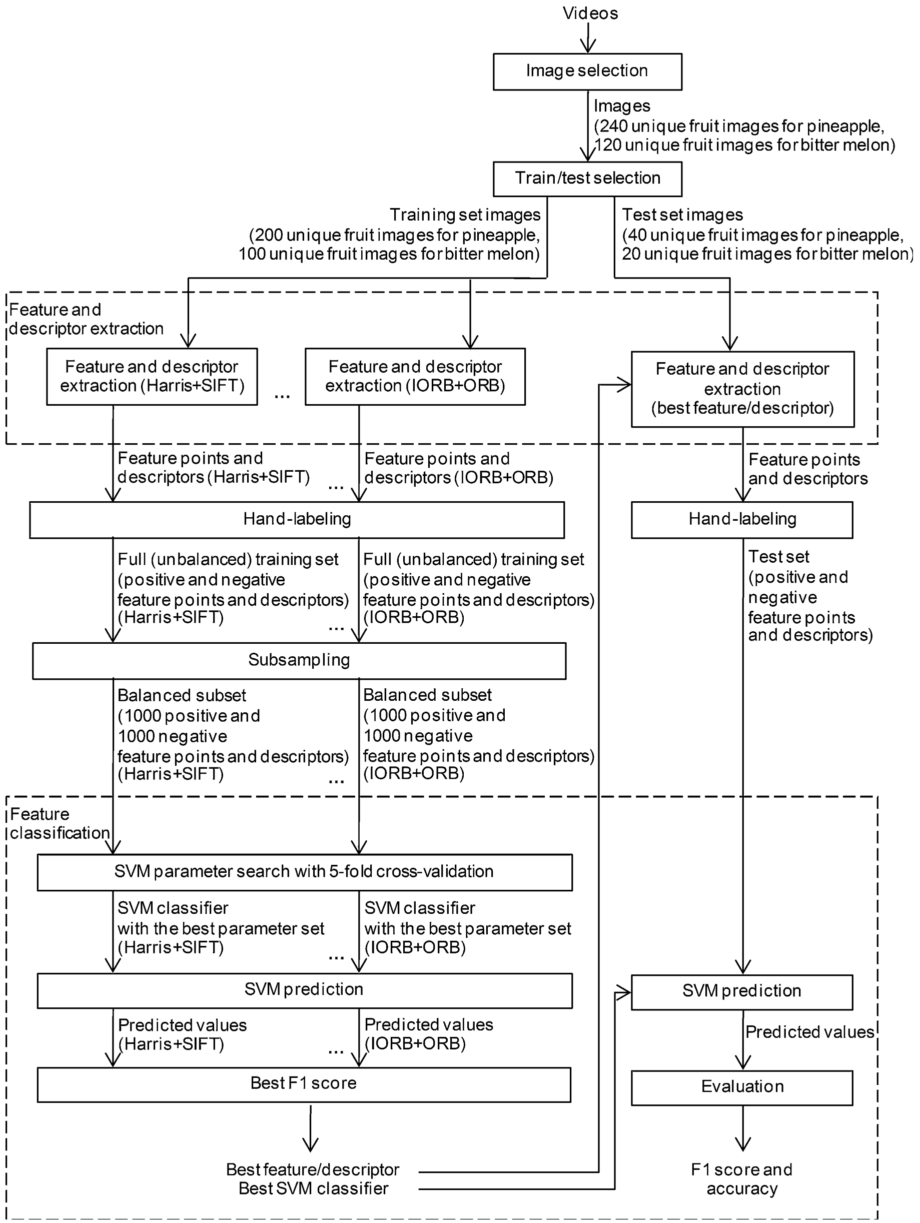

Videos
Image selection
Images
(240 unique fruit images for pineapple,
120 unique fruit images for bitter melon)
Train/test selection
Training set images
(200 unique fruit images for pineapple,
100 unique fruit images for bitter melon)
Test set images
(40 unique fruit images for pineapple,
20 unique fruit images for bitter melon)
Feature and descriptor extraction
Feature and descriptor extraction (Harris+SIFT)
...
Feature and descriptor extraction (IORB+ORB)
Feature and descriptor extraction (best feature/descriptor)
Feature points and descriptors (Harris+SIFT)
...
Feature points and descriptors (IORB+ORB)
Feature points and descriptors
Hand-labeling
Hand-labeling
Full (unbalanced) training set (positive and negative feature points and descriptors) (Harris+SIFT)
...
Full (unbalanced) training set (positive and negative feature points and descriptors) (IORB+ORB)
Test set (positive and negative feature points and descriptors)
Subsampling
Balanced subset (1000 positive and 1000 negative feature points and descriptors) (Harris+SIFT)
...
Balanced subset (1000 positive and 1000 negative feature points and descriptors) (IORB+ORB)
Feature classification
SVM parameter search with 5-fold cross-validation
SVM classifier with the best parameter set (Harris+SIFT)
...
SVM classifier with the best parameter set (IORB+ORB)
SVM prediction
SVM prediction
Predicted values (Harris+SIFT)
...
Predicted values (IORB+ORB)
Predicted values
Best F1 score
Evaluation
Best feature/descriptor
Best SVM classifier
F1 score and accuracy

Fig. 5 Diagram of data selection, feature and descriptor extraction and feature classification details

Fruit region extraction

To evaluate the performance of the fruit region extraction algorithm on the pineapple and bitter melon datasets, the best texture classification models (ORB+SURF128 for pineapple and Harris+SURF128 for bitter melon) were taken from the previous section and used for

Table 4 ANOVA on feature and descriptor performance (pineapple)

Source	*SS*	d.f.	*MS*	*F*	$p < F$
Feature	1.66177	5	0.33235	74.14	<0.001
Descriptor	0.00126	3	0.00042	0.09	0.9632
Feature × Descriptor	0.03947	15	0.00263	0.59	0.5782
Error	0.43015	96	0.00448		
Total	2.13266	119			

Table 5 ANOVA on feature and descriptor performance (bitter melon)

Source	*SS*	d.f.	*MS*	*F*	$p > F$
Feature	0.19410	4	0.04853	193.34	<0.001
Descriptor	0.00104	3	0.00035	1.38	0.2537
Feature × descriptor	0.02435	12	0.00203	8.09	<0.001
Error	0.02008	80	0.00025		
Total	0.23957	99			

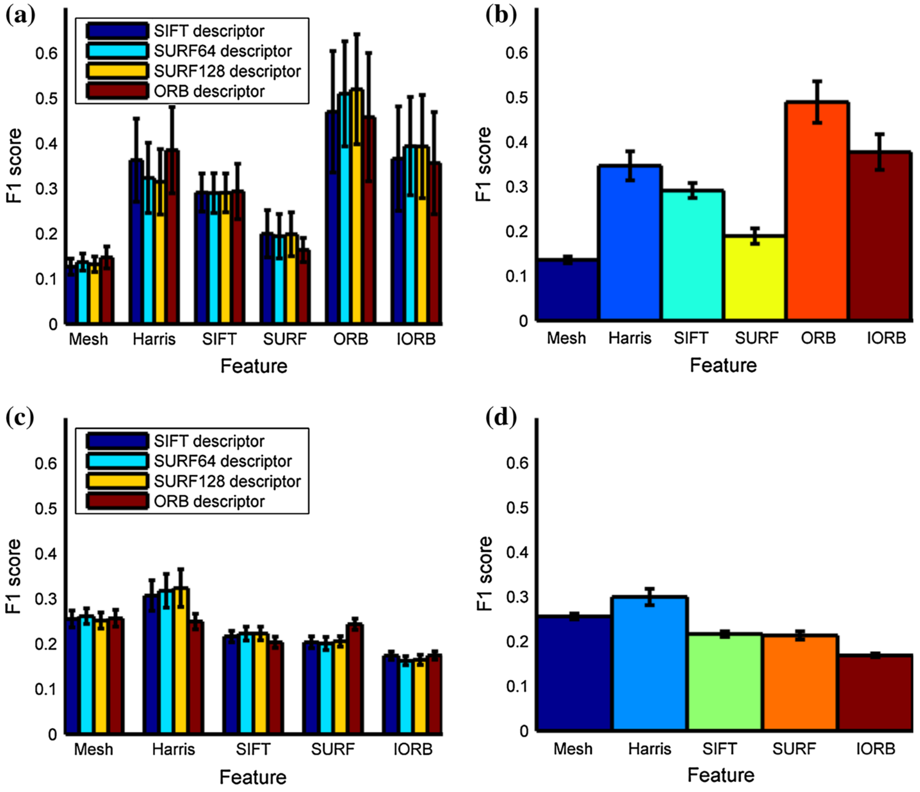


Fig. 6 Average F1 scores on partitioned full (unbalanced) training set. **a** Feature × descriptor F1 for pineapple dataset. **b** Overall feature F1 for pineapple dataset. **c** Feature × descriptor F1 for bitter melon dataset. **d** Overall feature F1 for bitter melon dataset. *Error bars* denote 95 % confidence intervals

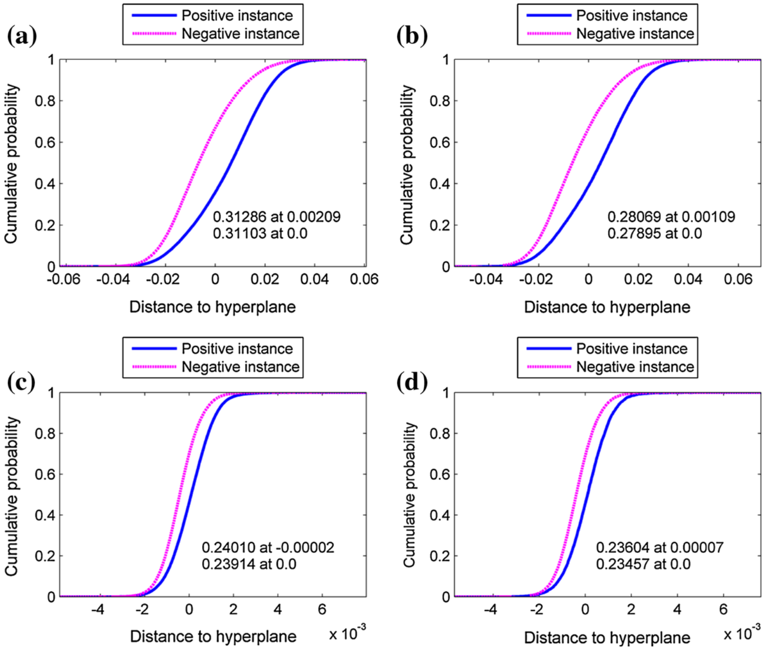


Fig. 7 KS analysis for the best feature point texture classifiers (ORB+SURF128 for pineapple and Harris+SURF128 for bitter melon). **a** Pineapple training set. **b** Pineapple test set. **c** Bitter melon training set. **d** Bitter melon test set

region extraction on the full test set for each type of fruit. The ground truth fruit regions were hand-labeled for each image, then a search for the best region detection parameters for each type of fruit was performed.

There are two free parameters in the region detection algorithm: the structuring element shape (the structuring element is used for morphological closing to fill the gaps between detected fruit feature points) and the minimum region size threshold (used when discarding small regions). The parameters were fit to achieve the best results in terms of per-pixel F1 score over all fruit regions on the full test set.

For pineapple, a disc-shaped structuring element was used with different radii in pixels (5, 6, 7, 8, 9, 10, 11, 12, 13, 14, and 15). Each structuring element radius option was tested with several different minimum region size thresholds (200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000, 2200, 2400, 2600, 2800 and 3000 pixels). The best pixel-based F1 score on the test set was obtained with a radius of 10 pixels and a size threshold of 1600 pixels.

For bitter melon, since the fruit are elliptical and hanging vertically, an ellipse-shaped structuring element was used with a vertical major axis length of 10, 15, 20, 25, 30, 35 and 40 pixels and a horizontal minor axis of 6, 7, 8, 9, 10 and 11 pixels, respectively. Each structuring element shape and size was tested with several different minimum region size thresholds (10000, 10500, 11000, 11500, 12000, 12500, 13000, 13500, 14000, 14500 and 15000 pixels). The best pixel-based F1 score on the test set was obtained with a vertical major axis length of 20 pixels, a horizontal minor axis of 8 pixels, and a minimum region size threshold of 10500 pixels.

Table 6 Distances to the hyperplane from KS analysis for the best feature point texture classifiers (ORB+SURF128 for pineapple and Harris+SURF128 for bitter melon)

Dataset	The largest gap	Distance to the hyperplane at the largest gap	Gap at zero
Pineapple			
Training set	0.31286	0.00209	0.31103
Test set	0.28069	0.00109	0.27895
Bitter melon			
Training set	0.24010	−0.00002	0.23914
Test set	0.23604	−0.00007	0.23457

Table 7 Fruit region detection results

Fruit type	Overlap ratio (%)	Hits (%)	Misses (%)	False positives (per frame)	Merges (%)	Fragmentations (%)
Pineapple	10	85	15.0	0.8	8.8	1.3
	40	67.5	32.5	1.2	8.8	1.3
	70	25	75.0	1.9	0.0	0.0
Bitter melon	10	100	0.0	0.2	0.0	0.0
	40	90.5	9.5	0.3	0.0	0.0
	70	4.8	95.2	1.2	0.0	0.0

Figure 3 shows example results of the morphological closing and fruit region extraction steps.

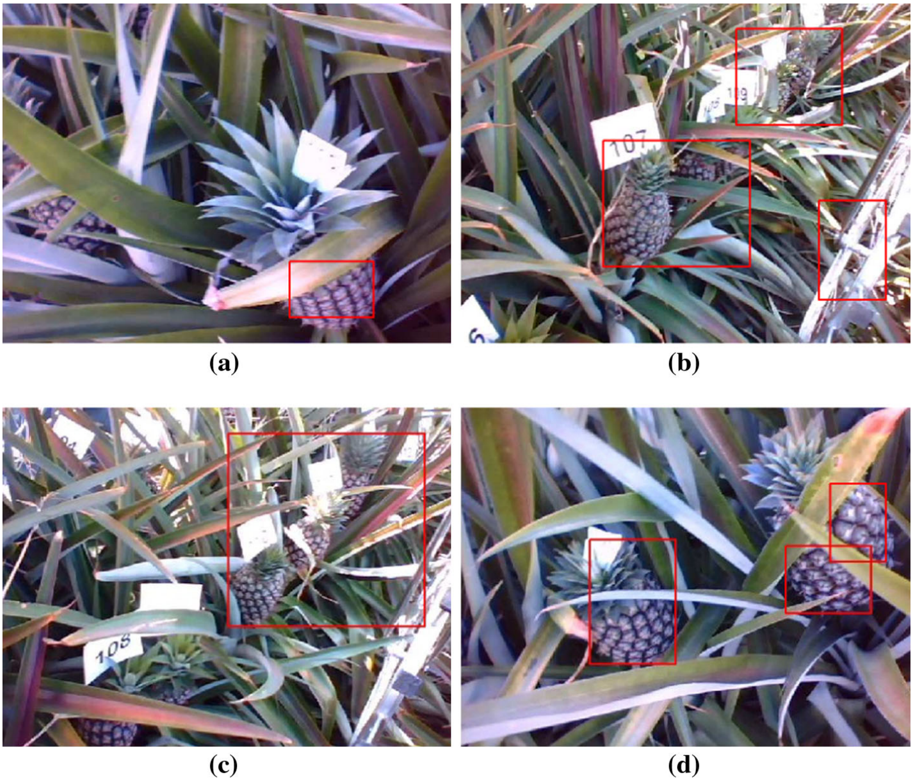
As previously described, the best fruit region detector was evaluated according to true positives (hits), false negatives (misses), false positives, merges and fragmentations. The merge and fragmentation rates are subsets of and with respect to the hits. Table 7 shows the results for the best parameter set (according to hit rate) for each fruit type.

One issue with this evaluation is that it must use the amount of overlap between a candidate detected region and a ground truth region to determine whether a detection is a hit or not. The system's measured detection performance thus increases as the overlap threshold is decreased. This is because the system sometimes detects small isolated regions of the fruit—if these isolated regions are not considered hits, the miss rate increases. Therefore, since different applications could potentially have different requirements for the necessary overlap ratio to be considered a hit, Table 7 reports hit, miss, false positive, fragment and merge rates for three different possible overlap ratio thresholds: 10, 40 and 70 %.

Finally, Figs. 8 and 9 show sample correct detections and detection errors for pineapple and bitter melon, respectively.

Discussion

Fruit detection in the field is a difficult task due to occlusion and variability in appearance and lighting. Some fruit types can be reliably detected by color, but others require alternative approaches. The experiments in this paper have demonstrated the effectiveness of



(a) (b)

(c) (d)

Fig. 8 Sample pineapple detection results. **a** Two fruits, one hit, one miss. **b** One fruit, one hit, two false positives. **c** Two fruits, two hits, two merges. **d** One fruit, one hit, one fragmentation. Images in **b** and **c** contain fruits too small to be considered as ground truth fruits; the "correct" detection of the small fruit in **b** is thus considered a false alarm

texture-based fruit detection in the field using low-cost monocular vision sensors. The sensors are web cameras that are cheap and easily controlled by computer programs. This is good for the construction of a low-cost system. However, web cameras are optimized to work well in indoor environments and work less well in outdoor environments. The cameras tend to produce low quality images when the viewpoint or illumination conditions are inappropriate. Such low quality images may be difficult even for humans to properly interpret. Special care has to be taken to acquire image data clear enough for correct interpretation.

Pineapple and bitter melon are both green fruits covered with distinctive texture. However, the two fruit types pose different challenges to the texture analysis algorithm. On the one hand, pineapple fields are very dense, and images captured under dense circumstances have a great deal of texture throughout, with almost no smooth regions. Nevertheless, pineapple texture is quite distinctive, more so than the background texture, so the difficulty of search for pineapple regions is abetted by the use of groups of strong feature detectors. This explains the effectiveness of ORB on pineapple images and the fact that ORB+SURF128 formed the basis for the best classifier of pineapple texture.

On the other hand, bitter melons also have distinctive texture, but since they are grown in sparse fields, the strength of the texture is intermediate, stronger than that in smooth

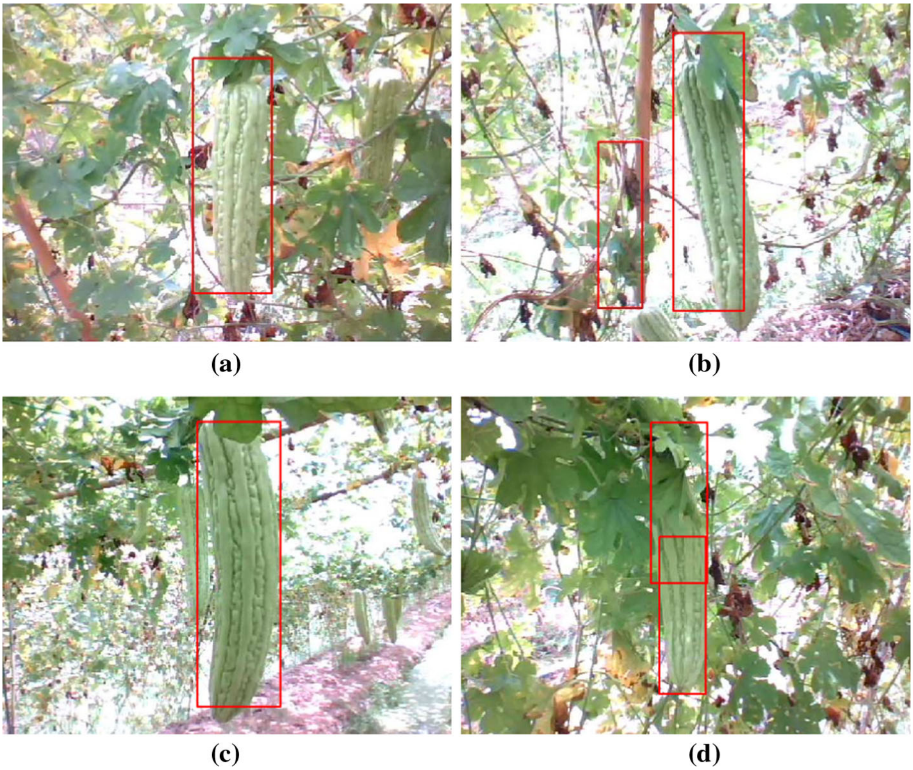


Fig. 9 Sample bitter melon detection results. **a** Two fruits, one hit, one miss. **b** One fruit, one hit, one false positive. **c** Two fruits, two hits, two merges. The fruit behind the large fruit in the center of the image is large enough to be a ground truth fruit, and the overlap between the detection and the second ground truth fruit is more than the overlap ratio threshold (10 % in this case). **d** One fruit, one hit, one fragmentation. (In the bitter melon data set, fragmentation never occurs with the optimal parameters; the fragmentation shown was obtained with a sub-optimal morphological structuring element.) Image in **c** contains several small fruit below the threshold size to be considered ground truth fruit

image regions containing ground or water but less strong than that in plant regions. This makes bitter melon texture segmentation more difficult and also requires detection of relatively weak features in the image, leading to the result that Harris features with a low threshold are best for bitter melon.

Given these characteristics of the problem, the simple mesh feature strategy might have been expected to be effective, because it gives a uniform spread of features across both fruit and non-fruit regions. However, mesh features are not sufficiently precisely located, so they finally perform poorly in comparison to the other detectors.

To further understand the SVM classification of feature points, the results can be considered in more detail. Figure 10 shows close-ups of selected SVM classification results. Each test feature point was run through the SVM model, and then the support vector retained in the model that had the most contribution to the final classification of that feature point was identified. In the case of pineapple, the figure only shows the particular feature points whose most relevant support vector was mapped to by the classifier more than five times in total across the original image. It is easy to see that most of the points are

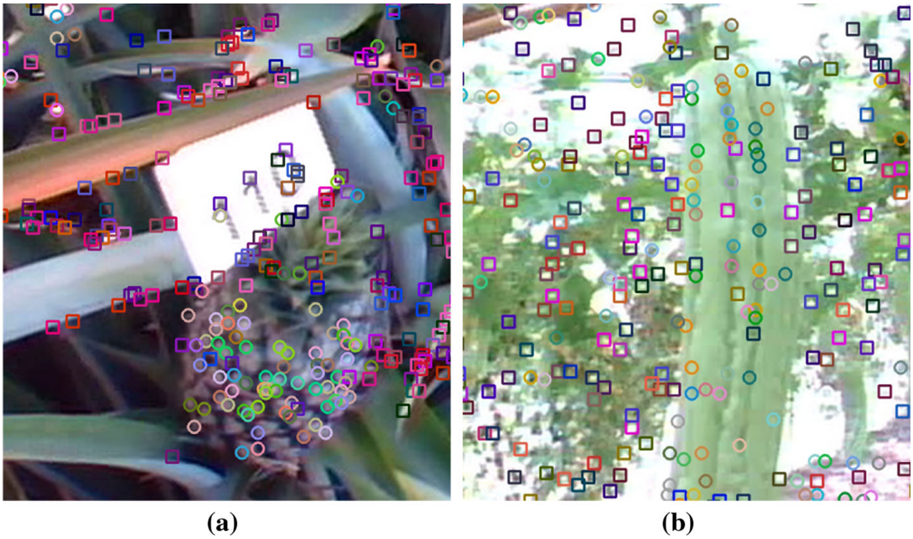


Fig. 10 Most relevant vector classes from SVM classifier for feature points. *Circles* represent positive support vectors, and *squares* represent negative support vectors. *Different colors* indicate different most-relevant support vectors. **a** Close-up of a pineapple and nearby background. **b** Close-up of a bitter melon and nearby background (Color figure online)

correctly classified by the single-most relevant support vector. Only a small number of points are misclassified; the misclassifications are mostly located on the fruit boundary, which is noisy and ambiguous. In the case of bitter melon, the figure shows the positive feature points whose most relevant support vector was mapped to more than two times in total across the original image, and the figure only shows the negative feature points whose most relevant support vector was mapped to more than ten times. Again, most of the bitter melon feature points are also correctly classified by the most relevant support vector, and only a small number of points are misclassified. The visualization shows that the SVM classifier solves the problem using a sophisticated form of weighted k-nearest neighbors classification. With both fruit types, the SURF128 descriptor was numerically the best descriptor, but statistically the performance of SURF128 with ORB features on pineapple was no different from that of any other descriptor. Similarly, with Harris features, the performance of the SURF128 descriptor was statistically equivalent to that of SIFT and SURF64, albeit significantly better than the ORB descriptor. Overall, it can be concluded that the choice of feature type is much more important than the choice of descriptor type.

There is some cause for concern that selecting SVM training parameters and training the classifier on balanced data might lead to sub-optimal results on the highly imbalanced real-world data, but the choice of cross validated F1 on a balanced data set as the criterion for model selection turns out to be quite reasonable, since it is highly correlated with KS test results, which are independent of the data set's balance.

Although best practices (cross validation) were used to avoid overfitting in the evaluation on test data, the method should be further validated by testing on a completely separate data set acquired at a different time and/or on a different farm.

The runtime speed depends on the number of features per image and which combination of extractor/descriptor is used. The runtime performance of the best methods on each

Table 8 Average and standard deviation of runtime for best combinations

Fruit type	Best combination	Average runtime (ms) per image	Standard deviation
Pineapple	ORB+SURF128	84.43	2.12
Bitter melon	Harris+SURF128	18.91	0.04

dataset is reasonable. SURF128 was the fastest method on the bitter melon data set. On the pineapple data set, among methods using the ORB feature detector, the ORB descriptor method was the fastest. Since ORB+SURF128 and ORB+ORB F1 performance was statistically equivalent, ORB+ORB would be a good option for real time implementation in an embedded system. The number of features per image for pineapple is less than that of the bitter melon images. However, the best feature type for pineapple (ORB) requires substantially more runtime than that for bitter melons (Harris). For the descriptor, SURF128 was best in both cases. The overall runtime per image is shown in Fig. 4. The precise average runtimes for the best method for each fruit type are shown in Table 8.

Fragmentation occurs when the morphological closing operation fails to connect all detected fruit regions of a particular fruit. In many cases, a fruit fragmented by foliage can be joined if the foliage is thinner than the structuring element. In some cases, it occurs because a bigger/wider leaf than the structuring element crosses over a fruit, making the fruit points split into two groups (see Fig. 8d for an example). In other cases, it occurs because the feature points are too sparse to be joined by the closing operation, making the fruit split into two separate components (see Fig. 9d for an example).

A final limitation of the approach is that different parameter settings are required to get an approximately equivalent number of feature points over each fruit region. Pineapples have strong texture, so high feature point detection thresholds can be used. Bitter melon, on the other hand, has relatively weak texture, so that lower thresholds need to be used. This has the effect of increasing the number of detected feature points in the background, thereby slightly increasing the overall time required to process each image.

Conclusions

This paper describes an empirical study of texture-based fruit detection for green fruits on plants in the field and describes experiments on two green fruit types: pineapple and bitter melon. Image data is captured from web camera video. The method includes five main steps: feature and descriptor extraction, feature classification, fruit point mapping, morphological closing and region extraction. The feature and descriptor methods tested comprised 24 combinations. The classification step used SVMs. The feature type employed was found to be more important than the descriptor type. The method is highly accurate on the data sampled, with the best combinations being ORB+SURF128 for pineapple and Harris+SURF128 for bitter melon. With the best parameter settings (a disc-shaped structuring element with a radius of 10 pixels and a minimum region size threshold of 1600 pixels for pineapple, and an ellipse-shaped structuring element with a vertical major axis length of 20 pixels, a horizontal minor axis of 8 pixels, and a minimum region size threshold of 10500 pixels for bitter melon), the method obtains single-image detection rates of 85 and 100 %, respectively. Robustness of the parameter settings on other data sets must be further validated in future work.

Future work will extend the method to work in a real time system. The method needs to be improved to better handle some disadvantageous conditions such as strong sunlight and occlusion. Temporary occlusions and fragmentations due to leaves can be handled by tracking fruit regions from frame to frame then performing 3D modeling. The run time may also need to be improved in order to increase the speed of processing and/or decrease manufacturing costs. Finally, the detection system will be integrated into a prototype automated fruit crop monitoring system, and an in-field real-time evaluation will be performed.

Acknowledgments SC was supported by the Thailand National Science and Technology Development Agency (NSTDA). The authors thank the members of the AIT Vision and Graphics Lab (VGL) for suggestions and help with data collection.

References

Aggelopoulou, A., Bochtis, D., Fountas, S., Swain, K., Gemtos, T., & Nanos, G. (2011). Yield prediction in apple orchards based on image processing. *Precision Agriculture, 12*(3), 448–456.

Bansal, R., Lee, W., & Satish, S. (2012). Green citrus detection using fast Fourier transform (FFT) leakage. *Precision Agriculture, 14*(1), 59–70.

Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded-up robust features (SURF). *Computer Vision and Image Understanding, 110*(3), 346–359.

Calonder, M., Lepetit, V., Strecha, C., & Fua, P. (2010). BRIEF: Binary robust independent elementary features. In Daniilidis, K., Maragos, P., Paragios, N. (Eds.), *Proceedings of the European Conference on Computer Vision* (pp. 778–792). Heidelberg: Springer.

Chang, C.C., & Lin, C.J. (2001). LIBSVM: A library for support vector machines. Retrieved March 12, 2014, from http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Cubero, S., Aleixos, N., Moltó, E., Gómez-Sanchis, J., & Blasco, J. (2011). Advances in machine vision applications for automatic inspection and quality evaluation of fruits and vegetables. *Food and Bioprocess Technology, 4*(4), 487–504.

Delenne, C., Durrieu, S., Rabatel, G., Deshayes, M., Bailly, J. S., Lelong, C., et al. (2008). Textural approaches for vineyard detection and characterization using very high spatial resolution remote sensing data. *International Journal of Remote Sensing, 29*(4), 1153–1167.

Du, C. J., & Sun, D. W. (2006). Learning techniques used in computer vision for food quality evaluation: A review. *Journal of Food Engineering, 72*(1), 39–55.

Harris, C., & Stephens, M. (1988). A combined corner and edge detector. In *Proceedings of the Fourth Alvey Vision Conference* (pp. 147–151).

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

Jiménez, R. A., Ceres, R., & Pons, L. J. (2000b). A survey of computer vision methods for locating fruit on trees. *Transactions of the American Society of Agricultural and Biological Engineers, 43*(6), 1911–1920.

Kaewapichai, W., Kaewtrakulpong, P., & Prateepasen, A. (2006). A real-time automatic inspection system for Pattavia pineapples. *Key Engineering Materials, 321–322*, 1186–1191.

Kaewapichai, W., Kaewtrakulpong, P., Prateepasen, A., & Khongkraphan, K. (2007). Fitting a pineapple model for automatic maturity grading. In *Proceedings of the IEEE International Conference on Image Processing* (pp. I-257–I-260). New York: IEEE.

Kitamura, S., & Oka, K. (2005). Recognition and cutting system of sweet pepper for picking robot in greenhouse horticulture. In *Proceedings of the IEEE Conference on Mechatronics and Automation* (pp. 1807–1812). New York: IEEE.

Lee, W. S., Slaughter, D. C., & Giles, D. K. (1999). Robotic weed control system for tomatoes. *Precision Agriculture, 1*(1), 95–113.

Li, B., Wang, M., & Wang, N. (2010). Development of a real-time fruit recognition system for pineapple harvesting robots. Paper No. 1009510. ASABE, St Joseph, MI

Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision, 60*(2), 91–110.

Muller, K. R., Mika, S., Ratsch, G., Tsuda, K., & Scholkopf, B. (2001). An introduction to kernel-based learning algorithms. *IEEE Transaction on Neural Networks, 12*(2), 181–201.

Nascimento, J. C., & Marques, J. S. (2006). Performance evaluation of object detection algorithms for video surveillance. *IEEE Transactions on Multimedia, 8*(4), 761–774.

OpenCV Community (2013). Open source computer vision library version 2.3.1, [C source code]. Retrieved December 1, 2013, from http://sourceforge.net/projects/opencvlibrary.

Payne, A. B., Walsh, K. B., Subedi, P. P., & Jarvis, D. (2013). Estimation of mango crop yield using image analysis segmentation method. *Computers and Electronics in Agriculture, 91*, 57–64.

Pla, F., & Marchant, J. A. (1997). Matching feature points in image sequences through a region-based method. *Computer Vision and Image Understanding, 66*(3), 271–285.

Rocha, A., Hauagge, D. C., Wainer, J., & Goldenstein, S. (2010). Automatic fruit and vegetable classification from images. *Computers and Electronics in Agriculture, 70*(1), 96–104.

Rosten, E., & Drummond, T. (2006). Machine learning for high-speed corner detection. In Leonardis, A., Bischof, H., Pinz, A. (Eds.), *Proceedings of the European Conference on Computer Vision* (pp. 430–443). Heidelberg, Germany: Springer.

Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 2564–2571). New York: IEEE.

Slaughter, D. C., Obenland, D. M., Thompson, J. F., Arpaia, M. L., & Margosan, D. A. (2008). Non-destructive freeze damage detection in oranges using machine vision and ultraviolet fluorescence. *Postharvest Biology and Technology, 48*(3), 341–346.

Szeliski, R. (2011). *Computer vision: Algorithms and applications*. London: Springer.

Torii, I., Okada, Y., Mizutani, M., & Ishii, N. (2009). A simple method for 3-dimensional modeling and application to complex objects. In *Proceedings of the 21st International Conference on Tools with Artificial Intelligence* (pp. 41–48). New York: IEEE.

Van Henten, E., Hemming, J., Van Tuijl, B., Kornet, J., Meuleman, J., Bontsema, J., et al. (2002). An autonomous robot for harvesting cucumbers in greenhouses. *Autonomous Robots, 13*(3), 241–258.

Vapnik, V. N. (1995). *The nature of statistical learning theory*. New York: Springer.

Zhang, Y., & Wu, L. (2012). Classification of fruits using computer vision and a multiclass support vector machine. *Sensors, 12*(9), 12489–12505.

Zhang, L., Yang, Q., Xun, Y., Chen, X., Ren, Y., Yuan, T., et al. (2007). Recognition of greenhouse cucumber fruit using computer vision. *New Zealand Journal of Agricultural Research, 50*(5), 1293–1298.

Zhou, R., Damerow, L., Sun, Y., & Blanke, M. (2012). Using colour features of cv. ‘Gala’ apple fruits in an orchard in image processing to predict yield. *Precision Agriculture, 13*(5), 568–580.