Available online at www.sciencedirect.com

ScienceDirect

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

Research Paper

Detecting tomatoes in greenhouse scenes by combining AdaBoost classifier and colour analysis

Yuanshen Zhao, Liang Gong, Bin Zhou, Yixiang Huang, Chengliang Liu *

School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

ARTICLE INFO

Article history:
Received 8 January 2016
Received in revised form
28 April 2016
Accepted 2 May 2016
Published online 11 June 2016

Keywords:
Tomato detection
Haar-like feature
AdaBoost classifier
Colour analysis
I component image

Despite the rapid development of agricultural robotics, a lack of access to automatic fruit detection and precision picking is limiting the commercial application of harvesting robots. An algorithm for the automatic detection of ripe tomatoes in greenhouse was developed for a simple machine vision system. The images of tomato planting scenes were captured by a colour digital camera, and most of the ripe tomatoes were correctly recognised using the proposed algorithm. The proposed tomato detection approach worked in two steps: (1) by extracting the Haar-like features of grey scale image and classifying with the AdaBoost classifier, the possible tomato objects were identified; (2) the false negatives in the results of classification were eliminated using average pixel value (APV) based colour analysis approach. Comparative test results showed that the C style of Haar-like features and I component image were optimum in the proposed algorithm. The results of validation experiments show that combination of AdaBoost classification and colour analysis can correctly detect over 96% of ripe tomatoes in the real-world environment. However, the false negative rate was about 10% and 3.5% of the tomatoes were not detected.

1. Introduction

For autonomous harvesting, fruit detection is the first major task for a harvesting robot. It is difficult to develop a vision system as intelligent as a human to easily recognise fruit in tree canopy. Although some researchers have used various approaches to accomplish fruit detection under indoor conditions, the robustness and precision of these methods are not sufficient for commercial application in greenhouses (Bac, Van Henten, Hemming, & Edan, 2014). Pla, Juste, and Ferri (1993) reported a segmentation method and the extraction of spherical features for detecting citrus. Their proposed detection algorithm proved efficient for identifying more than 50% of citrus in natural conditions. Linker, Cohen, and Naor (2012) developed an image processing method for detecting apples using RGB colour image acquired in orchard. Although the algorithm detected correctly more than 85% of the apples visible, the varying illumination and colour saturation caused a large number of false positive detections.

Various vision sensors have been used to meet the challenges caused by uneven illumination conditions and colour similarities (Gongal, Amatya, Karkee, Zhang, & Lewis, 2015). Because of the similar colour of citrus fruit and its leaves, Bulanon, Burks, and Alchanatis (2009) used a combination of infrared and visible images to detect the green citrus in tree canopy. A machine vision system consisting of a time-of-flight

* *Corresponding author*. Tel.: +86 21 34204350.
E-mail address: chlliu@sjtu.edu.cn (C. Liu).
http://dx.doi.org/10.1016/j.biosystemseng.2016.05.001
1537-5110/

(ToF) camera and a digital colour charge coupled device (CCD) camera was used to recognise apples in the canopy by Feng, Zeng, and Liu (2014). The detection algorithms based on fusion of multiple image data achieve higher correct detection rates. However, the high-cost vision sensors and long processing times for multiple modalities are the key shortcomings for commercial application.

With the development of artificial intelligence technology, Parrish and Goksel (1977) proposed that the pattern recognition approach could be used for detecting fruit in the natural scenes. Slaughter and Harrell (1989) used a Bayesian classifier to discriminate oranges from the natural background. The classification model using chrominance and intensity information correctly identified over 75% of the fruit pixels. Bulanon, Kataoka, Okamoto, and Hata (2004) developed an apple detection approach using a linear classifier and the trichromatic coefficients were used as features. About 80% of fruit pixels were correctly classified under all lighting conditions with a less than 3% error rate. Kurtulmus, Lee, and Vardar (2014) also proposed an algorithm based on support vector machine (SVM) to detect cutting locations of corn tassels in natural outdoor maize canopy. The author argued that the proposed algorithm performed with a correct detection rate of 81.6% for the test set. Lu et al., (2015) developed a combinational algorithm termed mTASSEL for detecting maize tassel. The results of field testing showed that the proposed detection algorithm can alleviate the influence of environment variations. Zhu, Cao, Lu, Li, and Xiao (2016) provided a two-step coarse-to-fine wheat ear detection mechanism. In the coarse-detection step, machine learning technology was used to increase the correct detection rate.

Inspired by the human face detection issue, AdaBoost algorithm is an available pattern recognition approach which could be used to detect ripe tomato for harvesting robot. The goal of this study was to develop a computer vision approach combining AdaBoost algorithm and colour-based analysis to detect tomato objects in canopies using conventional colour images. The proposed tomato recognition algorithm is a combination of extracting Haar-like feature, scanning a single image using a sub-window, using an AdaBoost classifier for identifying tomato objects, and average pixel value (APV) based colour analysis using I component.

2. Materials and methods

2.1. Image acquisition

For developing and testing proposed algorithm, images of tomatoes in greenhouse were acquired in the natural daylight illumination conditions (08:00–17:00) using a common digital camera (J2, Nikon) with a resolution of 3872 × 2592 pixels. A total of 180 images were captured manually in May 2015 in Sunqiao modern agriculture garden, Shanghai, China. Eighty images were randomly selected for training classification, whereas the remaining images were used as test set for evaluating the proposed algorithm. The tomato plants in greenhouse 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. Thus, the tomato scenes were randomly selected between the rows of tomato plants on the sunny side of the plant and also shadow side. As shown in Fig. 1, three typical disturbed conditions were investigated: (a) multiple tomatoes clustering, (b) varying illumination, and (c) occlusion by leaves, twigs, or immature tomatoes. The purpose of this study was to develop a tomato detection algorithm using conventional colour images, so images were resized to 388 × 260 pixels using the bicubic interpolation algorithm for computational convenience.

2.2. The dataset

In order to implement AdaBoost classifier training, 156 ripe tomato samples and 877 background samples were manually cropped from 80 tomato images for constructing a dataset. These samples were labelled separately, 1 for ripe tomato and −1 for background. Examples of the dataset images are shown in Fig. 2. The 156 tomato samples were cluttered with some background pixels in addition to the fruit. The background samples were randomly selected in the greenhouse scenes, and they contained leaves, twigs, immature tomatoes and other objects. Since the tomato diameters in the dataset varied from 40 to 120 pixels, all samples in dataset were resized to 40 × 40 pixels to unify the size of samples.

2.3. Overview of the proposed algorithm

Flow diagram of the proposed tomato recognition algorithm is shown in Fig. 3, which can be encapsulated in 5 general steps:

1 Sliding a sub-window on the entire image
2 Extracting Haar-like features within each sub-window
3 Detecting tomatoes using AdaBoost classifier
4 Colour analysis using average pixel value (APV) of I component
5 Merging the detection results.

In pre-processing stage, colour image was converted to grey image and YIQ colour space for splitting I component. The proposed approach was to allocate a sub-window (a square patch) for every pixel in an image using a sliding window approach, and then verify whether the pixels within the sub-window were classified as ripe tomato. In this study, 40 × 40, 80 × 80, and 120 × 120 pixels of sub-window sizes were selected to scanning the entire image due to the tomato diameters in image. Twenty, forty, and sixty pixel increments were carried out as shifting step in the scanning process when sub-window sizes were the three scales respectively. The value of the shifting step was half of the sub-window, so that fruit pixels could not be skipped in the image.

The classifiers of the proposed algorithm relied on features called Haar-like features. The Haar-like features were extracted within each sub-window, as described in detail in the following section. In detection phase, the sub-window was tested by the AdaBoost classifier to determine whether it was target tomato or background. The AdaBoost is an algorithm for constructing a “strong” classifier as linear combination of “weak” classifiers. The “weak” classifier could be thought as a simple threshold operation on a specific type of feature, and the training process of weak classifiers was called

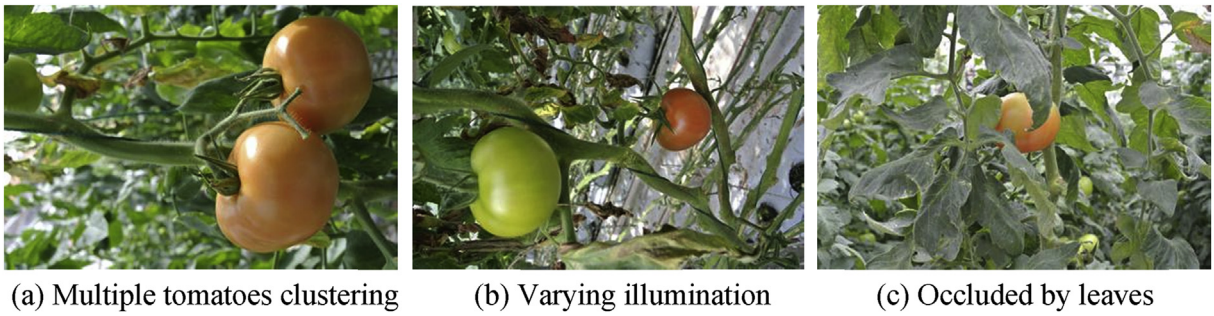

Fig. 1 – Images having different disturbed conditions: (a) multiple tomatoes overlapped, (b) varying illumination, and (c) occlusion by leaves.

Fig. 2 – Examples of the dataset. The top row tomato templates, lower rows are background templates including leaves, branches, immature tomatoes, and other obstacles in greenhouse.

"WeakLearn", which was described in more detail in Section 2.5. If the result of AdaBoost classification was negative, the sub-window was considered to be background. When the result was positive, APV-based colour analysis was performed to I component within the sub-window pixels. The positive result represented that target tomato was correctly detected, and the negative result indicated that it was a false positive by AdaBoost classifier. The identified results in different sub-window sizes testing could be the same tomato, so the merging of multiple detections was required for the same fruit. The merging principle of detection results is also presented in the following sections.

2.4. *Haar-like features extraction*

Papageorgiou et al. (1998) firstly proposed to represent objects in terms of a subset of over-complete dictionary of Haar wavelet basis functions. Since then Haar-like features have been popularly employed for object recognition due to its efficiency in the domain of face detection. It is illumination invariant because the mutual ordinal relationship of different pixel within the region is not affected by the varying illumination conditions (Park & Hwang, 2014). The three types of basis Haar-like features are presented in Fig. 4. A, B, and C type of Haar-like features reflect edge, line, and radian features respectively.

Figure 5 shows the process of Haar-like features extraction. Three different scales of each type Haar-like features were employed in our research, which were 2×2, 4×4, 8×8 pixels. A Haar-like feature consisted of two different adjacent square regions (white region and black region as shown in Fig. 5), and the value of a Haar-like feature x is equal to the weighted sum of white region pixels and black region pixels in a feature as in Equation (1).

$$x = \sum_{i=1}^{R} w_i \cdot \mu_i \tag{1}$$

where R is the number of pixels constituting a Haar-like feature and w_i is the weight of the ith pixel. The weight of white region pixels and black region pixels were 1 and -1 respectively. μ_i is the grey value of ith pixel.

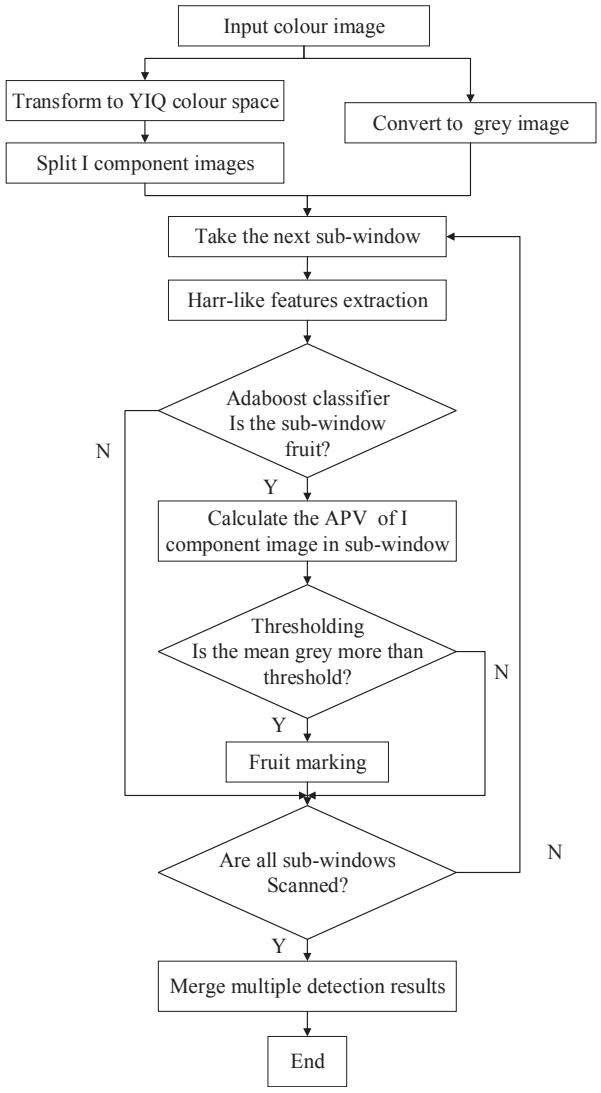

Fig. 3 – Flow diagram of the proposed tomato detection algorithm.

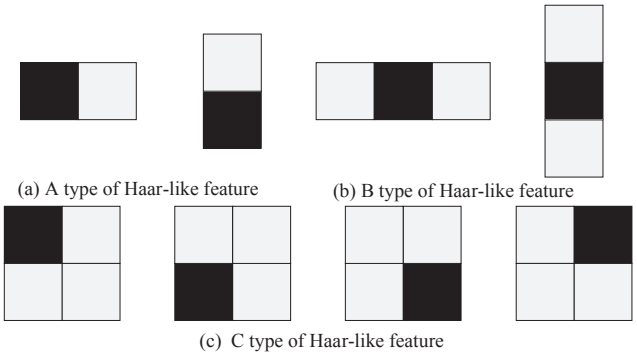

Fig. 4 – Three types of Haar-like features: (a) A type of Haar-like feature reflecting edge feature, (b) B type of Haar-like feature reflecting line feature, and (c) C type of Haar-like feature reflecting radian feature.

Viola and Jones (2001) developed an integral image method to rapidly calculate the value of Haar-like feature. With this method, the sum of grey scale pixel values within any rectangular region could be computed by referencing only four points of rectangular from the integral image. As shown in Fig. 6, the value of integral image at location S1 is the sum of the pixels in rectangle Ra. Similarly, the value at location S2 is Ra + Rb, the value at location S3 is Ra + Rc. At location S4, value of integral image is Ra + Rb + Rc + Rd. Thus, the sum with Rd can be computed by Eq. (2)

$$Rd = S1 + S4 - (S2 + S3) \tag{2}$$

2.5. *AdaBoost algorithm*

The scale of Haar-like features extracted from a sample is usually huge. For a given 40 × 40 pixel sample from dataset, the number of Haar-like feature is over 5000. Although each feature can be used as a weak classifier to identify tomato object, most of these features play a limited role in tomato detection and some features even play the opposite roles. Thus, the main challenge for detecting tomato using Haar-like features is to learn a classification function using effective features. Given a feature set and a training set of positive and negative samples, many number of machine learning approaches could be used to meet this challenge (Sengupta & Lee, 2014). In the proposed approach, AdaBoost algorithm was used to train a strong classifier through training procedure (sometimes called weak learning). The weak learning was designed to select the weak classifier which could best separate the positive and negative sample. For each feature, the weak learning determined the optimal threshold value, such that the minimum number of examples are misclassified. A weak classifier is described by Eq. (3).

$$h_t(x_j) = \begin{cases} 1, & x_j < \theta_t \\ -1, & \text{otherwise} \end{cases} \tag{3}$$

In which $h_t(x_j)$ is the weak classifier, x_j is the absolute value of jth feature, t is a running index which means the iterative times, and θ_t is the threshold which can be calculated from Eq. (4).

$$\theta_t = \arg\min \varepsilon_t \tag{4}$$

In which ε_t is the error rate of the tth iterative process. The error rate ε_t is the sum of weights of features which are misclassified.

$$\varepsilon_t = \sum_{j=1}^{M} D_t(j) \quad \left[y_j \neq h_t(x_j)\right] \tag{5}$$

In which $D_t(j)$ is the weight of jth feature which is misclassified, y_j is the reference value of jth Haar-like feature which is set as 1 or −1, M is the number of samples. The weight of feature need to be updated with each iterative process (Kong & Hong, 2015).

$$D_{t+1}(j+1) = D_t(j)\exp\left(-a_t y_j \theta_t\right) \Big/ Z_t \tag{6}$$

In which Z_t is the normalising factor and is obtained by Eq. (7).

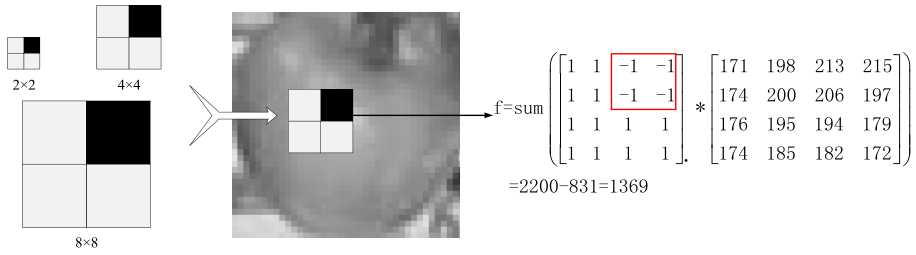

Fig. 5 – Illustration of Haar-like features and the feature value computation.

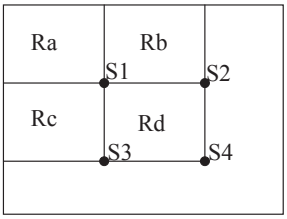

Fig. 6 – The principle of integral image computation.

$$Z_t = \sum_{j=1}^{M} D_t(j) \tag{7}$$

Haar-like features which were selected in early rounds of the training process had error rates between 0.1 and 0.4. With the weak learning processing, the error rates decreased obviously. In practice no single weak classifier could perform the classification task with low error. These weak classifiers multiply their weights and combine linearly for constructing a strong classifier. The final strong classifier is described by Eq. (8).

$$H(x) = \begin{cases} 1, & \sum_{t=1}^{T} a_t \cdot h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} a_t \\ -1, & \text{otherwise} \end{cases} \tag{8}$$

In which, $H(x)$ and h_t (x) denote the function of strong classifier and weak classifier, T is the number of weak classifiers or features, a is the weight of weak classifier and it updated in each round of iteration (Park & Hwang 2014). The update rule of a is

$$\frac{a_t = \frac{1}{2}\ln(1-\varepsilon_t)}{\varepsilon_t} \tag{9}$$

2.6. Colour analysis using colour space components

In the proposed algorithm, two types of classification were used to identify the ripe tomatoes from background. The first one was AdaBoost classifier as mentioned in previous section. The other type of classifier was a colour-based analysis approach which can enhance the success rate of tomato detection. Colour feature played an important role in fruit detection especially the fruit owned the different colour from the background. Since the colour of Fuji apple was red, Bulanon, Kataoka, Ota, and Hiroma (2002) used a R–Y (red–yellow) colour model to discriminate the fruits from canopy. Payne, Walsh, Subedi, and Jarvis (2013) also presented a pixel segmentation algorithm based on color analysis for detecting mango. Referring to the previous literatures (Arefi, Motlagh, Mollazade, & Teimourlou, 2011), colour subspaces of RGB, L*a*b*, and YIQ were used to develop a colour classifier for detecting ripe tomato in this study.

In the preliminary works of this study, tomato images captured in greenhouse conditions were observed by decomposing them to various colour model such as RGB, L*a*b*, and YIQ. Images were transformed from RGB to L*a*b* colour model and then to extract a* component. The transformation relationship was nonlinear and L*a*b* colour model was calculated in two steps. Firstly, XYZ colour model which was an intermediate matrix was obtained from RGB colour model by following linear transformation. Images were converted from RGB to YIQ colour space by the linear transformation. The detail transformation processes have been described in the previous work (Zhao, Gong, Huang, & Liu, 2016).

Total 1033 samples from the dataset were converted to these three colour spaces respectively, then the R component, a* component, and I component were exacted. For finding the difference between ripe tomatoes and background, the average pixel value of those component images were calculated as the follow Eq. (10).

$$v = \frac{1}{N} \sum_{i=1}^{N} p_i \tag{10}$$

where N is the number of pixels within a sample image, and p_i is the absolutely value of ith pixel of each, v is the average pixel value of each component.

2.7. Experiments

In this study, all algorithm development steps and experiments were performed on MATLAB (R2013a) with 32-bit Intel® Pentium® G640 2.8 GHz CPU. In order to develop and train the proposed algorithm, 156 ripe tomato samples and 877 background samples from dataset were used as a training set. Experiments were conducted on the training set to obtain thresholds for APV-based segmentation and the aforementioned parameters of the AdaBoost classifier. To test the algorithm, 100 images were used as a test set on which the proposed algorithm was applied. The total number of false negatives (background was misclassified as ripe tomato) and missed tomatoes were manually counted. The proposed algorithm was able to provide these data automatically. The true positive rate, false negative rate, and false positive rate were used to evaluate the results of experiments, and they were defined as Eqs. (11)–(13).

$$\text{True positive rate} = \frac{\text{Correctly identified}}{\text{Tomato count}} \times 100\% \quad (11)$$

$$\text{False negative rate} = \frac{\text{False negative}}{\text{False negative} + \text{Correctly identified}} \times 100\% \quad (12)$$

$$\text{False positive rate} = \frac{\text{Missed detection}}{\text{Tomato count}} \times 100\% \quad (13)$$

3. Results and discussions

3.1. Results of different types of Haar-like features

For studying the influence on the types of Haar-like features, three types of Haar-like features were designed and tested in the AdaBoost classifier training process. During the process of iteration, the training errors were monitored for reflecting the WeakLearn rate and efficiency. Maximum iterations were restricted to 20,000, at which the training errors did not decrease obviously. The WeakLearn curves of three types of Haar-like feature were plotted. As shown in Fig. 7, C type of Haar-like feature achieved the fastest WeakLearn rate and the lowest training error. Thus, C type of Haar-like feature was selected as the optimum and employed in the final tomato detection algorithm.

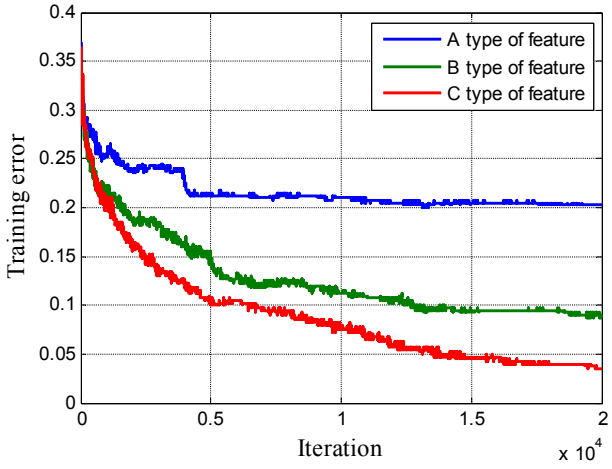

Fig. 7 – WeakLearn curves of three types of Haar-like features.

3.2. Results of AdaBoost classifier

Using the Haar-like features, an AdaBoost classifier for sub-windows was developed according to relatively homogenous features of the tomatoes. The Haar-like feature provided the distinctiveness for tomato from background. An example image before and after applying AdaBoost classifier is shown in Fig. 8. In the scanning process, all Haar-like features were used to determine if the sub-window was tomato or background. A circular region of interest was defined as a maximum circle inside the square sub-window, and was used for excluding background pixels at the corners of the sub-window due to circular shapes of the tomato.

3.3. Results of APV-based colour analysis

The APV-based colour analysis provided some differences between tomato background samples in R, a*, and I colour components. The average pixel value distribution diagrams of those colour components are shown in Fig. 9. The average pixel values of tomato and background sample under R and a* colour components were highly overlapped, thus it was hard to discriminate background and tomato using a certain threshold value. Although the average pixel value of the tomato and background samples were also overlapped, some distinctive pixels between background samples and tomato samples in I component could be observed. Figure 9(c) shows that the threshold of average pixel value was set as 20, about 95% background templates were removed and 98% ripe tomato templates were successfully extracted. Thus, the colour analysis was conducted by classifying the interest of region using APV of I component. The threshold value for segmenting background using I component were found, and a binary mask image (1 for tomato, 0 for background pixels) presenting tomato was created.

3.4. Merging results of classification

During the scanning process, the pixels in the sub-windows which was positively classified are marked in the grey scale

Fig. 8 – An example of the AdaBoost classification: (a) before applying the AdaBoost classifier and (b) after applying AdaBoost classifier.

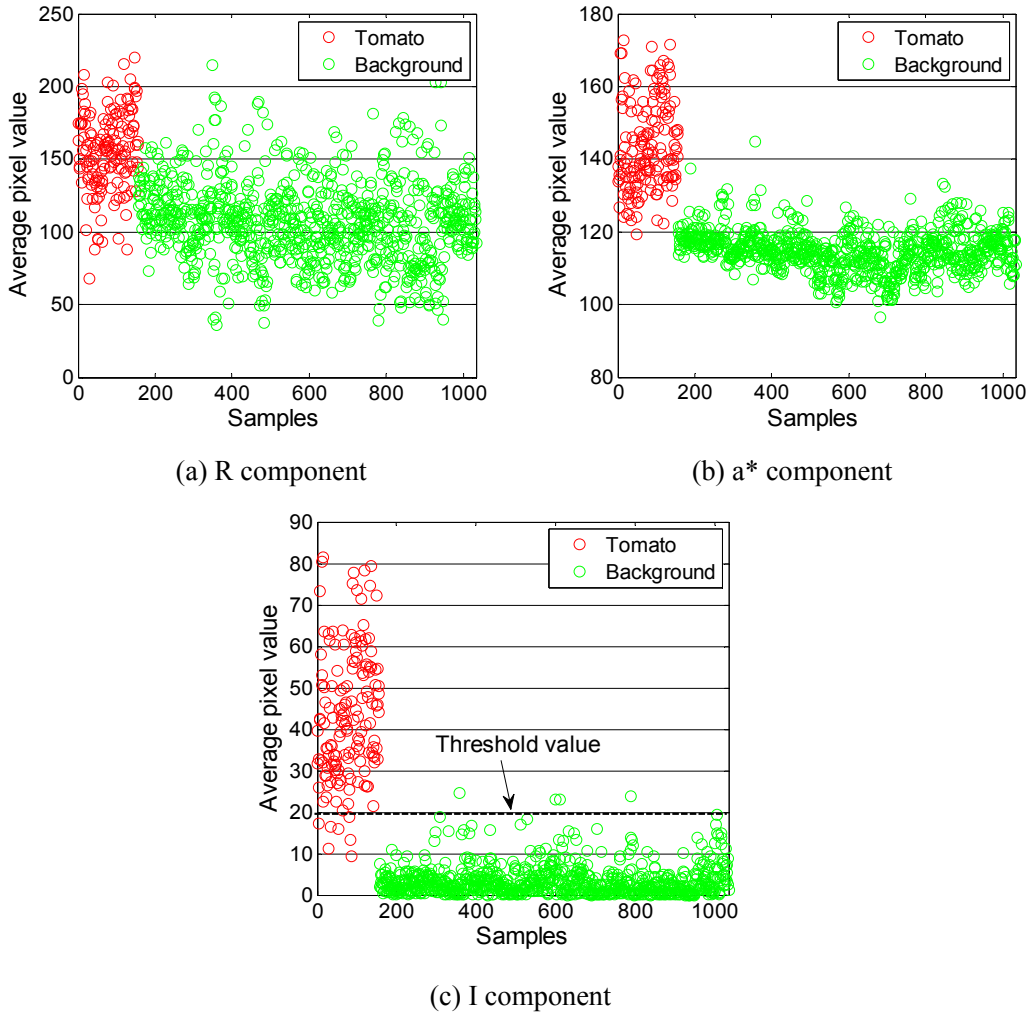

Fig. 9 – APV-based colour analysis of samples from dataset: (a) results of R component, (b) results of a* component, (c) results of I component.

image. The adjacent marked pixels form a connected region which is regarded as a single object and the centre of the connected region is considered as the centre of the tomato object. An example process for merging multiple detections was shown in Fig. 10. It can be seen in this Fig. 10(a), there were two detection results pointing to the same ripe tomato. The detection results can be merged by morphological operations. Firstly, two white square sub-windows were marked as shown in Fig. 10(b), which stands for the two detection results. Because the white square sub-window was connected. The centre of the connective white region is obtained and marked using a red star. The red star is also the centre of the final detection result. The final detection result is shown using a yellow circle.

3.5. *The performance of proposed algorithm*

The experiments were conducted on the training and test image sets. The performance of the proposed algorithm was evaluated according to true positive, false negatives, and false positive. In evaluation of the experiments, some objects were considered as background if they could not be identified clearly as fruit or background due to much-occlusion or visual complexity of the canopy. Table 1 shows the results when applied to the testing images from validation sets. For the proposed detection algorithm, 96.5% of the actual number of the ripe tomatoes were successfully detected. The false negative rate corresponded to about 10.8% (determined manually). About 3.5% of the ripe tomatoes were not detected. The results also showed that the performance of proposed approach was increased compared to the other two approaches using either the AdaBoost classifier or APV-based colour analysis alone. Although correct identifications by the proposed algorithm were the same as the results of AdaBoost classifier, the false detections were decreased.

To make comparison with the proposed tomato detection algorithm, some recent tomato detection approaches such as feature images fusion, BP neural networks and support vector machine (SVM) were employed to test. In this work, a SVM which uses a linear kernel and a BP neural network which were constructed using the training set containing two classes. Table 1 also lists the true positive rates, false negative rates and false positive rates of the three referencing approaches (Kurtulmus et al., 2014; Zhao et al., 2016). It can be seen that the proposed tomato detection algorithm achieved

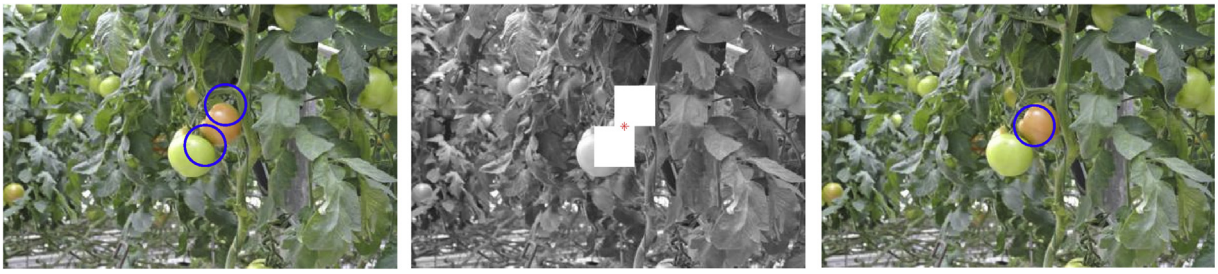

Fig. 10 – Steps of merging multiple detections: (a) multiple detections, (b) grey scale image with detection centre, and (c) resulting image.

Table 1 – Results of compared tests using various tomato detection algorithms.

Approaches	Tomato count	True positives		False negatives		False positives	
		Amount	Rate %	Amount	Rate %	Amount	Rate %
AdaBoost classifier	171	166	97.1	31	15.7	5	2.9
APV-based colour analysis	171	142	83.0	47	24.9	29	17.0
Proposed algorithm	171	165	96.5	20	10.8	6	3.5
Feature images fusion	171	158	92.4	19	10.7	13	7.6
BP neural works	171	147	86.0	39	20.1	24	14.0
SVM	171	156	91.2	22	12.4	15	8.8

the second highest true positive rate while it maintained the second lowest false positive rate which is 0.1%, slightly lower than that of feature images fusion approach.

Processing time of the algorithm was not considered, since the focus of this study was exploring the feasibility of the detection performance. During experiments, the overall processing time varied approximately from 15 to 24 s for one image depending on the number of Haar-like features and scanning sub-window. Using a low-level programming language could provide higher computation speed.

3.6. *Successful detection for tomato*

Successful results were obtained on clustered tomatoes in greenhouse environment. Figure 11 shows that two ripe tomatoes overlapped were successfully detected, while a ripe tomato clustered with other two immature tomatoes was also detected by the proposed algorithm. Proposed algorithm could detect ripe tomatoes occulted less than 50% by leaves or twigs. Tomatoes partially occluded by the leaves or twigs were correctly identified in Fig. 12.

With respect to sunny side and shadow side of the plant, results show that almost all the ripe tomatoes were correctly identified as shown in Fig. 13. Images from sunny side of the plant yielded higher false positive percentage. Although sunlight was the source of the illumination, it cannot guarantee homogenous illumination over the canopy. Tomatoes in the inner canopy receive varying illumination compared with tomatoes on the outer surfaces because the dense canopy and the branches block some of the light penetrating the canopy. The side exposed to sunlight directly has higher contrasted regions rather than shady side of the plant.

3.7. *False negatives and missed detections*

While the proposed algorithm yielded successful results for most tomatoes occluded by leaves and twigs, some much-occluded tomato regions could not be detected successfully.

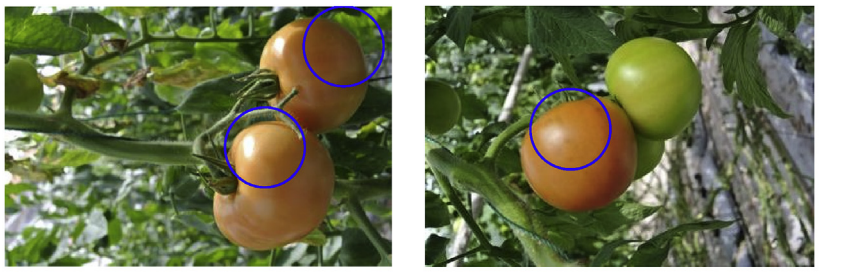

Fig. 11 – Detection result of overlapped tomatoes: (a) detection results of two ripe overlapped tomatoes and (b) a detection result for ripe tomato overlapped by immature tomatoes.

Fig. 12 – Detection results of occluded tomatoes: (a) two detection results of partially-occluded tomato and (b) a detection result for ripe tomatoe occluded by twigs.

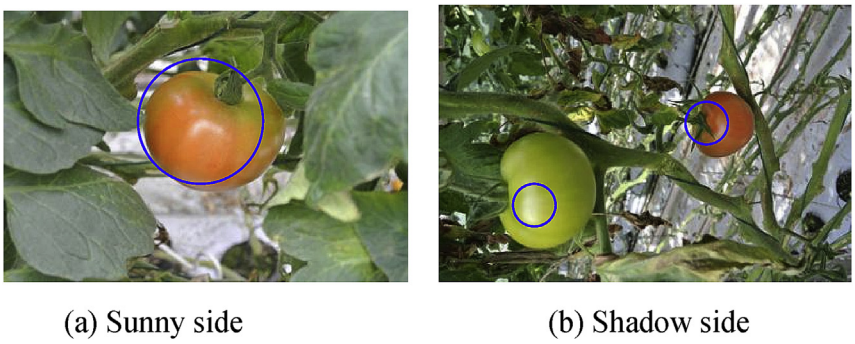

Fig. 13 – Detection results of varying illumination: (a) detection the ripe tomato in sunny side of the plant and (b) a detection result for a ripe tomato in shady side and one false positive.

There were also some false negatives existing in the detection results. Some reasons for misclassifications were occlusion, uneven illumination, and varying fruit sizes. Illumination changes affected both colour and texture features of the images. Sometimes shapes of some leaves and leaf clusters were very similar to tomatoes and this caused inaccurate detections. As shown in Fig. 14(a), these kinds of leaves along with similar shape and texture features cause false negatives. Figure 14(b) shows the missed detection result; the reason for missed detection is serious occlusion.

3.8. *The influence of varying image resolutions*

In the actual application, the resolution of camera is often restricted to about 10 thousand to 100 thousand pixels for computational convenience. Two image resolutions of 388 × 260 pixels and 311 × 208 were selected for a comparative test. The scanning steps depend on image resolution. The scanning step number needed to be changed when varying resolutions of images were employed in the test. Figure 15 shows two detection results with resolutions of 388 × 260

Fig. 14 – Examples of false negatives and missed detection: (a) two detections of occluded tomatoes with one false positive and (b) a missed detection of a mostly-occluded tomato.

Fig. 15 – Comparative experiments of tomato detection using varying image resolutions.

and 311 × 208 pixels. It can see that the ripe tomatoes were detected totally while the centres of the detection results with varying image resolutions had a slight change. Thus, the influence of varying image resolutions on detection results was insignificant.

4. Conclusions and future work

An approach to detecting ripe tomatoes using conventional colour images was developed. To use shape, texture and colour information, Haar-like features, an AdaBoost algorithm and an APV-based colour analysis were implemented. Multiple detections were merged for the final result. Using greenhouse scene images, experiments were conducted and the results were evaluated.

The AdaBoost classification approach for ripe tomato detection using just colour images was novel. The Haar-like features have potential for extracting shape and texture information of the ripe tomato in natural scenes which contains various visual features due to complex structure of the leaves, twigs and other objects. It can be concluded that APV of I component is a useful classifier for discriminating tomato regions from background pixels.

This research was conducted to explore the feasibility of detecting ripe tomato using just regular colour images. According to the proposed algorithm and the results, using colour images for ripe tomato detection in greenhouse conditions is promising because success rates were enough to constitute a vision detection system for a harvesting robot. The developed algorithm would be able to guide the manipulator to move to the correct position. Although the proposed detection algorithm achieved high correct detection rate, one limitation of the work was that only one cultivar of tomato was considered. Future work will include enhanced detection rates, reducing the processing time, various cultivars of tomatoes and

accommodate more varied unstructured environments. Future studies are needed to develop a real application with sensors installed in a robot platform with images taken continuously.

Acknowledgements

This work was mainly supported by a grant from the National High-tech R&D Program of China (863 Program No. 2013AA102307), and partially supported by the National Key Technology R&D Program of China (No. 2014BAD08B01 and 2015BAF12B02) and Shanghai Science Program (No. 14111104603). The authors would also want to give thanks to Sunqiao modern agricultural park of Shanghai for the experiment samples.

REFERENCES

Arefi, A., Motlagh, A. M., Mollazade, K., & Teimourlou, R. F. (2011). Recognition and localization of ripen tomato based on machine vision. *Australian Journal Crop Science, 5*(10), 1144–1149.

Bac, C. W., Van Henten, E. J., Hemming, J., & Edan, Y. (2014). Harvesting robots for high-value crops: state-of-the-art review and challenges ahead. *Journal of Field Robotics, 31*(6), 888–911.

Bulanon, D. M., Burks, T. F., & Alchanatis, V. (2009). Image fusion of visible and thermal images for fruit detection. *Biosystems Engineering, 103*, 12–22.

Bulanon, D. M., Kataoka, T., Okamoto, H., & Hata, S. (2004). Development of a real-time machine vision system for apple harvesting robot. In *Proceedings of SICE annual conference* (pp. 595–598).

Bulanon, D. M., 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.

Feng, J., Zeng, L., & Liu, G. (2014). Fruit recognition algorithm based on multi-source images fusion. *Transactions of the Chinese Society of Agricultural Machinery, 45*(2), 73–80.

Gongal, A., Amatya, S., Karkee, M., Zhang, Q., & Lewis, K. (2015). Sensors and systems for fruit detection and localization: a review. *Computers and Electronics in Agriculture, 116*, 8–19.

Kong, K. K., & Hong, K. S. (2015). Design of coupled strong classifiers in AdaBoost framework and its application to pedestrian detection. *Pattern Recognition Letters, 68*, 63–69.

Kurtulmus, F., Lee, W. S., & Vardar, A. (2014). Immature peach detection in colour images acquired in natural illumination conditions using statistic classifiers and neural network. *Precision Agriculture, 15*, 57–79.

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.

Lu, H., Cao, Z., Xiao, Y., Fang, Z., Zhu, Y., & Xian, K. (2015). Fine-grained maize tassel trait characterization with multi-view representations. *Computers and Electronics in Agriculture, 118*(C), 143–158.

Papageorgiou, C., Oren, M., & Poggio, T. (1988). A general framework for object detection. In *Proceedings of international conference on computer vision* (pp. 555–562).

Park, K. Y., & Hwang, S. Y. (2014). An improved Haar-like feature for efficient object detection. *Pattern Recognition Letters, 42*, 148–153.

Parrish, E. A., & Goksel, J. A. K. (1977). Pictorial pattern recognition applied to fruit harvesting. *Transactions of the American Society of Agricultural Engineers, 20*(5), 822–827.

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

Pla, F., Juste, F., & Ferri, F. (1993). Feature extraction of spherical objects in image analysis: an application to robotic citrus harvesting. *Computers and Electronics in Agriculture, 8*, 57–72.

Sengupta, S., & Lee, W. S. (2014). Identification and determination of the number of immature green citrus fruits in a canopy under different ambient light conditions. *Biosystems Engineering, 117*, 51–61.

Slaughter, D. C., & Harrell, R. C. (1989). Discriminating fruit for robotic harvest using colour in natural outdoor scenes. *Transactions of the American Society of Agricultural Engineers, 32*(2), 757–763.

Viola, P., & Jones, M. (2001). Rapid object detection using a boost cascade of simple features. In , *Vol. 1. Proceedings of conference on computer vision and pattern recognition* (pp. 511–518).

Zhao, Y., Gong, L., Huang, Y., & Liu, C. (2016). Robust tomato recognition for robotic harvesting using feature image fusion. *Sensors, 16*(2), 173–185.

Zhu, Y., Cao, Z., Lu, H., Li, Y., & Xiao, Y. (2016). In-field automatic observation of wheat heading stage using computer vision. *Biosystems Engineering, 143*, 28–41.

Glossary

APV: average pixel value
SVM: support vector machine
ToF: time of flight
CCD: charge coupled device
x: value of a Haar-like feature
R,N: number of pixels
w: weight of pixel
μ: grey value of pixel
Ra, Rb, Rc, Rd: rectangles
S1, S2, S3, S4: locations in image
h: weak classifier
t: running index
θ: threshold
ε: error rate
D: weight of feature
y: reference value
M: number of samples
Z: normalizing factor
H: strong classifier
a: weight of weak classifier
v: average pixel value
p: absolutely value of pixel
BP: back propagation