Computers and Electronics in Agriculture 94 (2013) 29–37

Contents lists available at SciVerse ScienceDirect

Computers and Electronics in Agriculture

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

An image segmentation method for apple sorting and grading using support vector machine and Otsu's method

Akira Mizushima [a,*], Renfu Lu [b]

[a] United States Department of Agriculture, Agricultural Research Service, 524 S. Shaw Lane, Room 207, Michigan State University, East Lansing, MI 48824, USA
[b] United States Department of Agriculture, Agricultural Research Service, 524 S. Shaw Lane, Room 224, Michigan State University, East Lansing, MI 48824, USA

ARTICLE INFO

Article history:
Received 18 September 2012
Received in revised form 6 February 2013
Accepted 17 February 2013

Keywords:
Segmentation
Support vector machine
Otsu's method
Automatic thresholding
Apple sorting and grading

ABSTRACT

Segmentation is the first step in image analysis to subdivide an image into meaningful regions. It directly affects the subsequent image analysis outcomes. This paper reports on the development of an automatic adjustable algorithm for segmentation of color images, using linear support vector machine (SVM) and Otsu's thresholding method, for apple sorting and grading. The method automatically adjusts the classification hyperplane calculated by using linear SVM and requires minimum training and time. It also avoids the problems caused by variations in the lighting condition and/or the color of the fruit. To evaluate the robustness and accuracy of the proposed segmentation method, tests were conducted for 300 'Delicious' apples using three training samples with different color characteristics (i.e., orange, stripe, and dark red) and their combination. The segmentation error varied from 3% to 25% for the fixed SVM, while the adjustable SVM achieved consistent and accurate results for each training set, with the segmentation error of less than 2%. The proposed method provides an effective and robust segmentation means for sorting and grading apples in a multi-channel color space, and it can be easily adapted for other imaging-based agricultural applications.

Published by Elsevier B.V.

1. Introduction

Today computer vision technology has been widely used for quality inspection of agricultural and food products, replacing traditional manual operations that are labor intensive, slow and prone to human error (Sun, 2008). Despite the significant advances over the past decade, new image processing algorithms are continuously being developed to improve the accuracy and efficiency of automatic inspection for agricultural and food products (Baranowski et al., 2012; ElMasry et al., 2012; Garrido-Novell et al., 2012).

Image segmentation, which separates the product region from background in the image, is one of the most important tasks in image processing since it is the first step in image analysis after the image capture to subdivide an image into meaningful regions. The segmentation result affects the subsequent image analysis. For instance, the estimation of product size (Moreda et al., 2009) and shape (Moreda et al., 2012) is directly affected by the segmentation result because these morphological features are usually obtained based on the product contour information.

A number of image segmentation methods have been developed in the past (Zheng and Sun, 2008). Among them, thresholding-based segmentation is the most popular for online applications due to its simplicity and fast processing speed. The simplest thresholding selection method is manual selection. Li et al. (2002) applied manual thresholding after subtracting the background image. Manual thresholding, combined with an edge detection technique, was used to remove the background from the grayscale image for grading pre-sliced hams (Valous et al., 2009). Multi-thresholding methods using red and saturation were performed to filter the background for apples (Zou et al., 2010). Currently, manual thresholding method is still being used for image segmentation because of its simplicity. However, it is not practical or convenient in real world applications, where the optimal threshold may vary if the lighting condition and the object color change. Consequently, a number of automatic thresholding selection methods have been developed to adapt the changes in real time.

Otsu's thresholding selection method (1979) is one of the most accurate and widely used methods for image segmentation (Sahoo et al., 1988). Otsu's method automatically finds the threshold using the histogram of a grayscale image, based on the idea of finding the threshold that maximizes the between-class variance $\sigma_B^2(t)$ (or minimizes the weighted within-class variance), which is expressed as follows:

$$\sigma_B^2(t) = \frac{[m_G P(t) - m(t)]^2}{P(t)[1 - P(t)]} \tag{1}$$

where m_G is the average intensity of the entire image, $m(t)$ is the cumulative mean up to level t, $P(t)$ is the cumulative sum of

* Corresponding author. Tel.: +1 517 353 7793.
E-mail address: Akira.Mizushima@ars.usda.gov (A. Mizushima).

0168-1699/$ - see front matter Published by Elsevier B.V.
http://dx.doi.org/10.1016/j.compag.2013.02.009

probability assigned to object (background). The optimum threshold is the value t^*, that maximizes $\sigma_B^2(t)$ as follows:

$$t^* = \arg\max_{0 \leqslant t \leqslant L-1} \sigma_B^2(t) \tag{2}$$

This method works well if the histogram is bimodal and the image has uniform illumination. Many researchers used this method for a variety of color spaces, including the hue color space (Nashat et al., 2011), grayscale images (Yang et al., 2009), near-infrared (NIR) images (Lunadei et al., 2012), hyperspectral images (Garrido-Novell et al., 2012), and linear combinations of RGB colors such as excess green, excess red (Meyer and Neto, 2008) and color index of vegetation extraction (Zheng et al., 2009). Since all these color or multi-channel spaces are selected on a trial-and-error basis with visual inspection, which is time-consuming, there is no specific and effective procedure to find the best color spaces to obtain good segmentation results, Mery and Pedreschi (2005) sought the best linear combination of RGB components that maximizes the variance of a grayscale image using a numerical gradient method. But they did not report the segmentation time, which is very important in online machine vision applications.

Classification-based segmentation is another popular method, especially for RGB color images. It is a pixel-oriented technique, in which each pixel is classified as either object or background. Linear discriminant analysis (or Bayesian classification) is the most popular classification-based segmentation method for color imaging (Blasco et al., 2003; Chinchuluun et al., 2009). This method generates a linear separation (or classification) hyperplane in a multi-dimensional space (e.g. 3-D for RGB color) based on the statistics of each region (object or background) by selecting a few representative pixels of the object or background. This hyperplane generation task, called training, is usually performed off-line before the online image processing is executed. Recently, support vector machine (SVM) is becoming a popular method to calculate the separation hyperplane for agricultural applications (Mitra et al., 2004; Nashat et al., 2011). Unlike linear discriminant analysis, support vector machine only uses ''difficult points'' close to the decision boundary, called support vector, and maximize the margin between support vectors. Because of this criterion, SVM is robust for untrained data. Although the hyperplane calculated by SVM is considered the best hyperplane, the method has limitation in adjusting to variations in the lighting condition and the color of products because the hyperplane is fixed.

The objective of this research was therefore to develop an automatic adjustable algorithm for segmentation of color images for apple sorting and grading, using Otsu's method and SVM. The method takes advantage of the adjustability of Otsu's method and the effectiveness of selecting the optimal classification hyperplane by SVM. The proposed algorithm consists of three main steps: (1) the optimal linear separation hyperplane in the 3-D RGB space is calculated by using linear SVM; (2) a contrast enhanced grayscale image is calculated from the linear hyperplane, and (3) the optimal threshold around the fruit boundary is automatically estimated using Otsu's method.

2. Materials and methods

2.1. Color segmentation by linear support vector machine

Support vector machine is a supervised machine learning algorithm originally designed to solve the two-group classification problems by generating the optimal separation hyperplane in a multi-dimensional space (Vapnik, 1995). The basic idea of SVM is to find the optimal hyperplane to separate a dataset, since, in theory, there exist many hyperplanes. A hyperplane is such chosen that the margin between it and the nearest data points, termed support vectors, of both classes is maximized. Because of this concept, linear SVM, which finds the ''linear'' optimal separation surface, is considered the best ''linear'' classification or segmentation algorithm in terms of accuracy and robustness for the untrained data. Although nonlinear SVM can solve a more complicated decision boundary, we selected linear SVM because the former requires more computational time when there are many support vectors. In contrast, the computational time for linear SVM classification is irrelevant to the number of support vectors.

The soft margin hyperplane algorithm (Cortes and Vapnik, 1995) is applied for non-separable data. Given a training set of instance-label pairs $(\mathbf{x}_i, t_i)$, $i = 1, \ldots, l$ where $\mathbf{x}_i \in R^n$ is the training sample and $t_i \in \{-1, +1\}$ is the class label, the linear SVM requires to find the solution of the dual representation of the maximum margin problem:

$$L_D(\alpha) = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j t_i t_j \mathbf{x}_i \cdot \mathbf{x}_j \tag{3}$$

Subject to:

$$0 \leqslant \alpha_i \leqslant C, \tag{4}$$

$$\sum_i \alpha_i t_i = 0 \tag{5}$$

where α_i is the positive Lagrange multipliers and parameter C (>0) controls the trade-off between the training error and the margin; a larger C means assigning a higher penalty to errors. α_i can be obtained by solving the quadratic programming problem of Eq. (3).

The separation hyperplane is given by:

$$f(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + b \tag{6}$$

$$\mathbf{w} = \sum_{i=1}^{N_S} \alpha_i t_i \mathbf{x}_i \tag{7}$$

$$b = t_i - \mathbf{w}^{\mathrm{T}}\mathbf{x}_i \tag{8}$$

where $\mathbf{w}$ is the surface normal to the hyperplane, $|b|/||\mathbf{w}||$ is the perpendicular distance from the hyperplane to the origin, $||\mathbf{w}||$ is the Euclidean norm of $\mathbf{w}$, and N_S is the number of support vectors. The new data point $\mathbf{x}$ is classified by evaluating the sign of Eq. (6).

When linear SVM is applied to the color RGB image segmentation, it calculates a classification hyperplane in the 3-D RGB space shown in Fig. 1. Eq. (6) is rewritten as a linear combination of red, green and blue as follows:

$$Z(x,y) = w_R R(x,y) + w_G G(x,y) + w_B B(x,y) + b \tag{9}$$

$$w_R = \sum_{i=1}^{N_S} \alpha_i t_i R_i, \; w_G = \sum_{i=1}^{N_S} \alpha_i t_i G_i, \; w_B = \sum_{i=1}^{N_S} \alpha_i t_i B_i \tag{10}$$

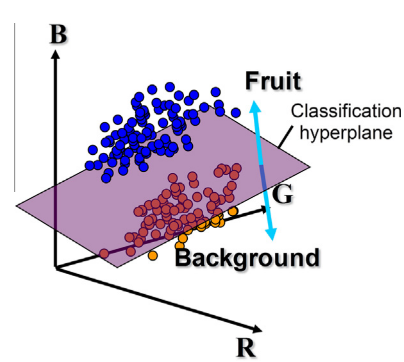

Fig. 1. Linear classification hyperplane in the 3-D RGB space.

$$b = t_k - (w_R R_k + w_G G_k + w_B B_k), \; 1 \leqslant k \leqslant N_S \tag{11}$$

where $R(x,y)$, $G(x,y)$ and $B(x,y)$ are the red, green and blue values at pixel position (x,y), respectively; R_i, G_i and B_i are support vectors of red, green and blue obtained by training, respectively; and R_k, G_k and B_k are the red, green and blue of any given support vector, respectively. Note that coefficients w_R, w_G, w_B and b are constant so that the computational time of Eq. (9) is not affected by the number of support vectors. The output of the Eq. (9) is a plus or minus value. If Z is more than 0, the pixel is the object; if Z is less than 0, the pixel is the background.

2.2. Support vector machine grayscale image

To apply automatic adjustable thresholding by Otsu's method, the SVM gray scale image is generated using the classification hyperplane calculated by linear SVM [Eq. (9)]. Instead of calculating a binary image by the classification hyperplane, a gray scale image is calculated by normalizing the Z value to between 0 and 255 using the maximum and minimum Z of the entire image as follows:

$$Gray_{SVM}(x,y) = \frac{Z(x,y) - Z_{min}}{Z_{max} - Z_{min}} * 255 \tag{12}$$

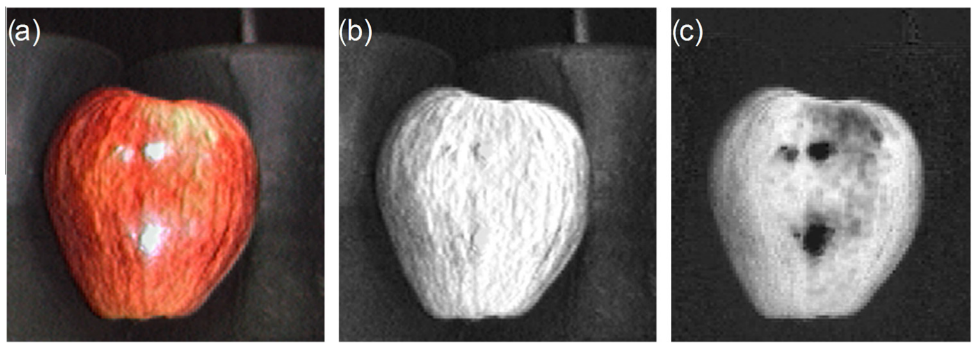

Fig. 2. Example of a support vector machine (SVM) grayscale image for an apple fruit: (a) original color image, (b) red color, and (c) SVM grayscale image. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

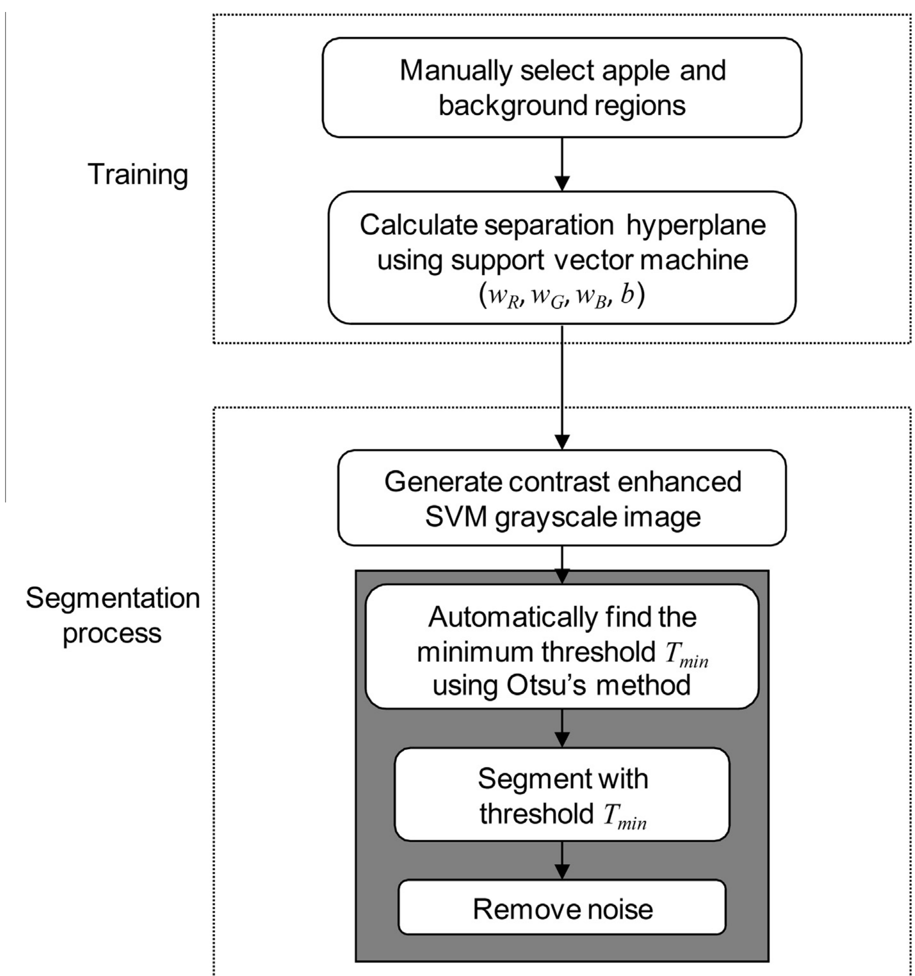

Fig. 3. Overall procedure for automatic adjustable SVM segmentation. The three segmentation procedures shown in the shaded box are further explained in Fig. 4.

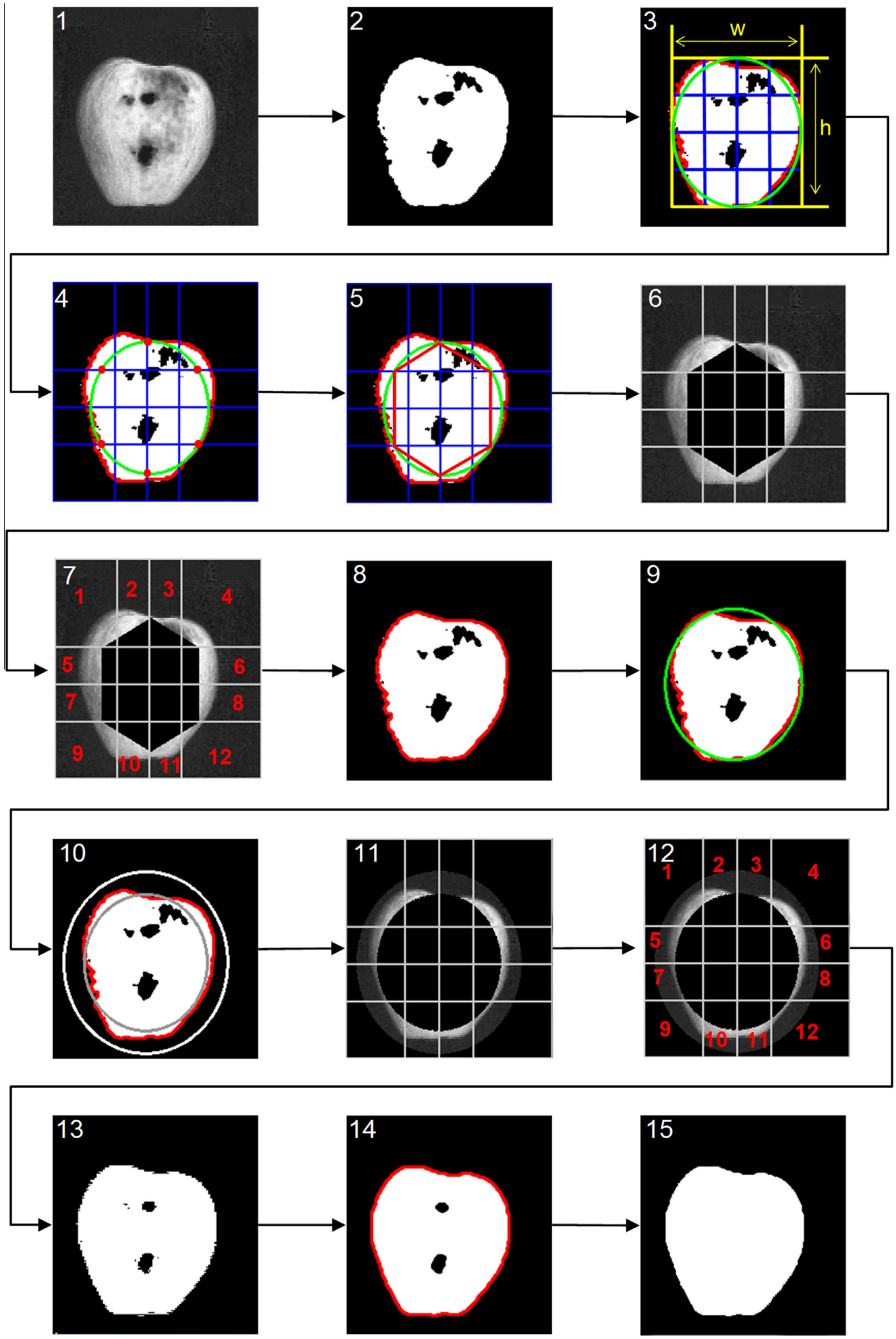

Fig. 4. Procedures to estimate the minimum threshold (Section 2.3).

where Z_{max} is the maximum value of $Z(x,y)$ and Z_{min} is the minimum value of $Z(x,y)$. This SVM grayscale image enhances the contrast between objects and background since SVM estimates the best separation hyperplane. The example result of a SVM grayscale image for an apple is shown in Fig. 2. The original color image (left) and red channel image (center) are also shown for comparison.

2.3. Automatic adjustable SVM segmentation

The overall procedure for automatic adjustable SVM segmentation is shown in Fig. 3. First, an SVM grayscale image is calculated from the captured color image. Even though the SVM grayscale image enhances the contrast between the fruit and background, the contrast around the fruit boundary is not uniform. Simply applying Otsu's method globally does not provide a good result if there are variations in brightness (lighting) and color, especially for a darker color fruit which has low contrast with the black background. To obtain the best segmentation result, we estimated the minimum threshold $T_{\min}$ around the fruit boundary. It was then applied to the SVM grayscale image globally to achieve segmentations. The final result was obtained by removing noises inside the fruit and background. The last three procedures of the segmentation process, as shown in the gray shaded area of Fig. 3, are graphically explained in Fig. 4, which consist of the following 15 steps:

(1) Calculate the SVM grayscale image using Eq. (12).
(2) Otsu's method is applied to the SVM grayscale image globally for obtaining a binary image.
(3) Contour (red) is extracted from the binary image using the border following algorithm of Suzuki and Abe (1985) and the minimum up-right bounding rectangle (yellow) of the contour is calculated. Then, the rectangle is divided into 4×4 windows (blue) and the inscribed ellipse (green) of the rectangle is estimated.
(4) The ellipse is downscaled to 90%.
(5) A hexagon is generated from the intersections between the 90% ellipse and 4×4 windows.
(6) The hexagon is used as mask and applied to the SVM grayscale image.
(7) Find the minimum threshold T_1 from the 12 windows around the boundary using Otsu's method. The inside of the hexagon is excluded for the Otsu's method calculation.
(8) Binary image is calculated by applying the minimum threshold T_1 to the SVM grayscale image and contour is extracted again.
(9) A fitted ellipse (Fitzgibbon and Fisher, 1995) from the contour is calculated instead of the inscribed ellipse of the bounding rectangle.
(10) 90% (gray ellipse) and 120% (white ellipse) of the fitted ellipse are generated.
(11) 90% of the inside of the fitted ellipse and 120% of the outside of the fitted ellipse are masked for the SVM grayscale image.
(12) Again, find the minimum threshold T_2 from the masked 12 windows around the boundary using Otsu's method.
(13) The SVM grayscale image is segmented using the minimum threshold T_2 globally.
(14) Extract contour from the binary image. The contour extraction algorithm is the same as step 3.
(15) Pixels inside the contour are assigned as fruit and the pixel outside of the contour are assigned as background.

2.4. Imaging for apple samples

To evaluate the performance of the proposed segmentation algorithm, RGB color images were obtained using a CCD color camera (Fire-I Digital Camera, Unibrain Inc., California, USA) with the maximum image size of 640×480 pixels. The camera was installed 38 cm above the fruit and the spatial resolution was 0.49 mm/pixel. Images were captured through an IEEE 1394 interface. Eight LED lights (BXRA-C0400, Bridgelux Inc., California, USA), arranged in two lines at an equal angle of about 40–50° away from the vertical direction, were used to illuminate the fruits. Two computers (Pentium M 1.7 GHz, 2 GB Memory, Windows XP 32 bit; Corei7–2600 K 3.4 GHz, 16 GB Memory, Windows 7 64 bit) were used, separately, for image processing to compare the computational speeds. The image processing algorithms were implemented in C++ using the OpenCV library (version 2.3).

The test samples were 300 'Delicious' apples were obtained from a commercial packinghouse in Michigan. They represented three basic commercial grades, i.e., 100 of 'U.S. Extra Fancy' grade, 100 of 'U.S. Fancy' grade, and 100 of 'U.S. No. 1' (Cull) grade, based on the United States Standards for Grades of Apples (USDA, 2002).

The three images shown in Fig. 5 (i.e., dark red, stripe and orange color) were used as training sets. Four segmentation hyperplanes were generated from the three images and all of the three by using linear SVM. $C = 0.1$ in Eq. (4) was used. Pixels inside the green rectangle were trained as apple, whereas those inside the white rectangle were trained as background. Means of RGB for the fruit and background of each image are shown in Table 1, and they are quite different. For the orange and stripe images, the RGB value of the fruit was greater than that of the background, while the RGB value of the background was greater than that of the apple for the dark red image.

The best separation hyperplanes were calculated by using linear SVM for each sample image with manual inspection and used as the ground truth. Hence, the ground truth is the best possible segmentation result by the linear separation hyperplane in the RGB space for the specific sample image.

3. Results and discussion

3.1. Accuracy of segmentation

Example segmentation results for the orange training set are shown in Fig. 6. Segmentation results for the linear SVM without automatic adjustment (Fixed SVM) were also calculated and are shown in Fig. 6. The original color image, SVM grayscale image, and Fixed and Adjustable SVM images are shown from left to right in Fig. 6. Green pixels for the fruit were misclassified as background (false negative), while red pixels for the background were misclassified as fruit (false positive). Since the training set was orange, Fixed SVM worked well for cull apples, which usually have the orange to yellow color. But a slight under-segmentation error (green region) was noticed for Fancy apples and there was considerable error for Extra Fancy apples using the Fixed SVM segmentation. On the other hand, the proposed Adjustable SVM segmentation method worked very well for all three grade categories, even though the training set only used an orange region of the apple. The average segmentation errors (%) for Fixed and Adjustable SVM for the orange training set are shown in Table 2. False negative is the percentage of pixels for the fruit being misclassified as background. False positive is the percentage of pixels for the background being misclassified as fruit. Since Fixed SVM tended to under-segment the fruit, most errors were false negative (the fruit was segmented as background) and there were fewer false positive errors (the background was segmented as fruit). For Extra Fancy, there was a large false negative error of 55.23% for Fixed SVM, while Adjustable SVM only had an error of 1.61%.

Example segmentation results for the dark red training set are shown in Fig. 7. Since the hyperplane was calculated from the dark red region, Fixed SVM worked well for Extra Fancy apples that had a dark red skin color, but it became noisy around the boundary for Cull apples. However, Adjustable SVM worked very well for all grade categories; it had zero error for the cull apples. The average segmentation errors (%) for Fixed SVM and Adjustable SVM for the dark red training set are shown in Table 3. Compared with the orange training set, the dark red training set worked better with Fixed SVM. But Adjustable SVM improved the accuracy in all grade categories and nearly removed all noise for cull apples, as shown in Fig. 7.

Average segmentation errors (%) for Fixed SVM and Adjustable SVM with a stripe apple as a training set are shown in Table 4. Even though the stripe training set covered the dark red to yellow color, Fixed SVM could not correctly detect the dark red color as apple,

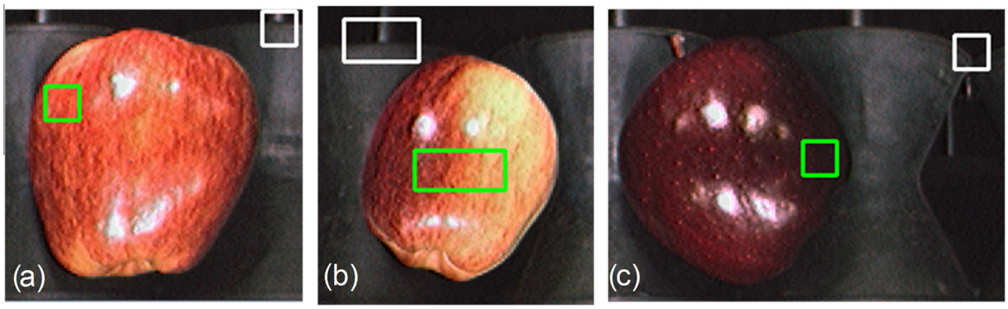

Fig. 5. Images of three training apples with different color characteristics: (a) orange, (b) stripe, and (c) dark red. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 1
Means of RGB values for three training apples of different color characteristics.

	Orange		Stripe		Dark red	
	Apple	Background	Apple	Background	Apple	Background
Red	235	68	205	45	42	34
Green	83	65	110	42	26	31
Blue	58	66	71	43	28	32

resulting in 22.63% error for Extra Fancy, whereas Adjustable SVM achieved outstanding results for all categories with the error of no greater than 1.71%.

Average segmentation errors (%) for Fixed SVM and Adjustable SVM for all three color training sets are shown in Table 5. Since the training set used all three images, the training sets contained a more wide range of color information from yellow to dark red. However, Fixed SVM still could not segment Extra Fancy well with 16.90% error, while it gave good results with 1.07% error for Cull and 2.18% for Fancy. Meanwhile, Adjustable SVM showed excellent results with the error of 0.98%, 1.68% and 1.59% for Cull, Fancy and Extra Fancy, respectively.

Total segmentation errors (average error of 300 apple images) for all four training sets are shown in Table 6. The error varied from 3.31% to 25.50% for Fixed SVM; the worst result was obtained with the orange training set at 25.50% error and the least error was obtained at 3.31% for the dark red training set. For the orange, stripe and dark red training sets and their combination, the largest error came from under segmentation for Extra Fancy. It was noted that the training set using all the three color images did not yielded the best result. Generally, the wider the range of the training data

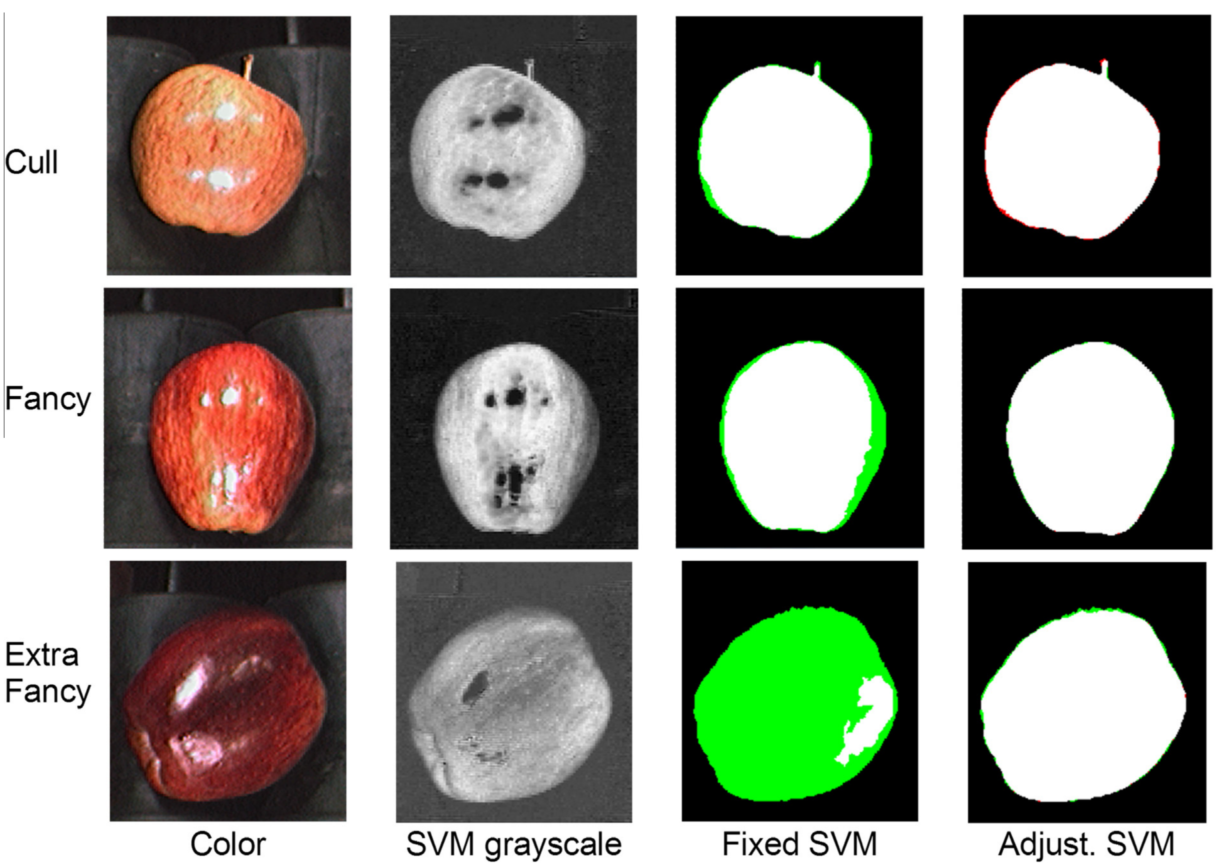

Fig. 6. Example segmentation results for the orange training set using the fixed and adjustable support vector machine (SVM) methods. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 2
Average segmentation errors for the fixed and adjustable support vector machine (SVM) using an orange training set.

	Fixed SVM (%)			Adjustable SVM (%)		
	False negative[a]	False positive[b]	Total	False negative	False positive	Total
Cull	7.37	0.01	7.38	0.55	0.31	0.86
Fancy	13.90	0.01	13.90	1.14	0.37	1.51
Extra Fancy	55.23	0.00	55.23	1.00	0.61	1.61

[a] The percentage of pixels for fruit was misclassified as background.
[b] The percentage of pixels for background was misclassified as fruit.

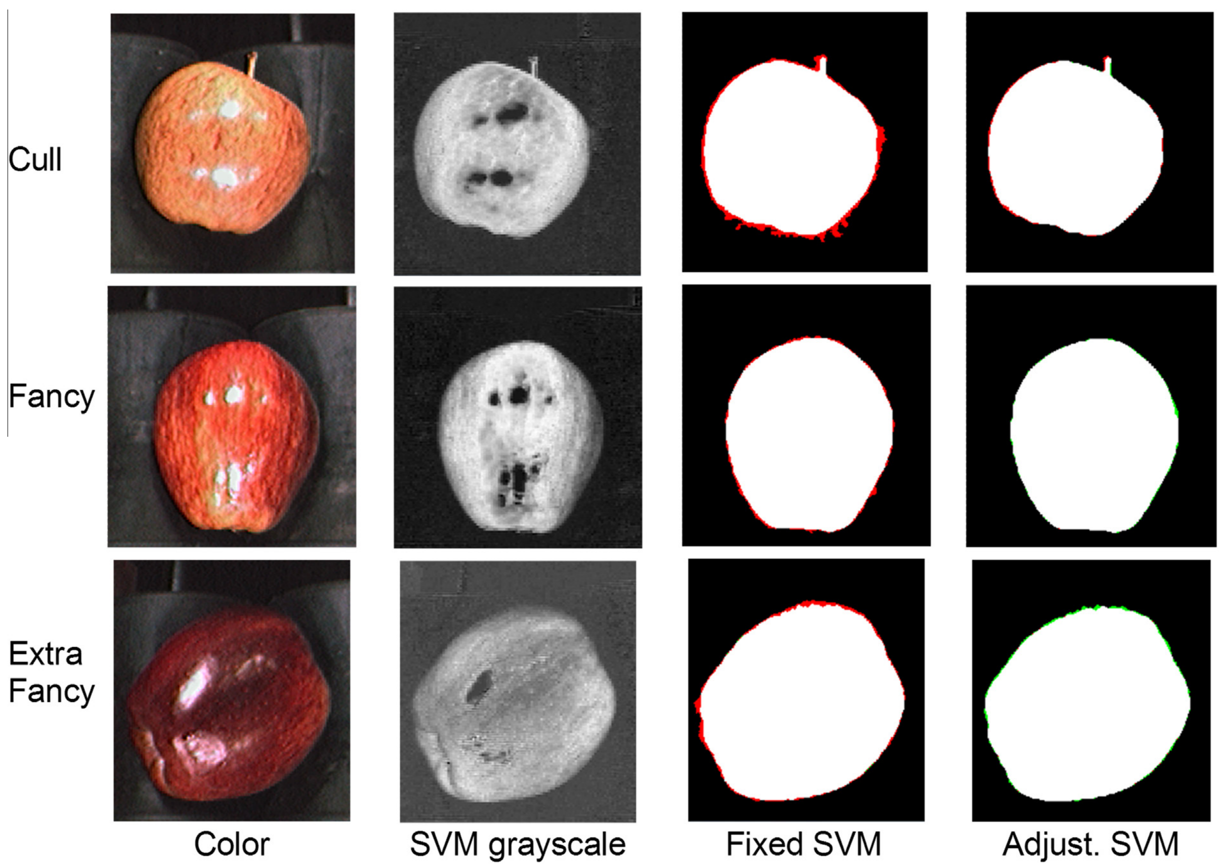

Fig. 7. Example segmentation results for the dark red training set using the fixed and adjustable support vector machine (SVM) methods. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 3
Average segmentation errors for the fixed and adjustable support vector machine (SVM) using a dark red training set.

	Fixed SVM (%)			Adjustable SVM (%)		
	False negative	False positive	Total	False negative	False positive	Total
Cull	0.06	2.58	2.63	0.59	0.38	0.97
Fancy	0.70	3.84	4.54	2.99	0.31	3.30
Extra Fancy	0.17	2.59	2.76	0.46	0.99	1.45

Table 4
Average segmentation errors for the fixed and adjustable support vector machine (SVM) using a stripe training set.

	Fixed SVM (%)			Adjustable SVM (%)		
	False negative	False positive	Total	False negative	False positive	Total
Cull	1.19	0.17	1.36	0.53	0.34	0.87
Fancy	3.24	0.09	3.43	1.36	0.35	1.71
Extra Fancy	22.62	0.01	22.63	0.76	0.75	1.51

Table 5
Average segmentation errors for the fixed and adjustable support vector machine (SVM) using all three training sets.

	Fixed SVM (%)			Adjustable SVM (%)		
	False negative	False positive	Total	False negative	False positive	Total
Cull	0.65	0.42	1.07	0.63	0.35	0.98
Fancy	1.97	0.20	2.18	1.34	0.34	1.68
Extra Fancy	16.88	0.03	16.90	0.87	0.72	1.59

Table 6
Overall segmentation errors (average error of 300 apple images) for the fixed and adjustable support vector machine (SVM) using each of the four training sets.

Training set	Fixed SVM (%)	Adjustable SVM (%)
Orange	25.50	1.33
Stripe	9.14	1.36
Dark red	3.31	1.90
All three	6.72	1.42

is, the better the result would be for a supervised training method. Clearly, it was not the case in this research. One possible reason for this result is that the range of the training data sets was so wide that nonlinearity of the separation hyperplane had increased. There was a trade-off between two conditions for the optimal separation hyperplane; it either optimized for bright color but under-segmented for darker color, or it optimized for darker color but increased the noise around the boundary of the fruit for bright color. Although the dark red training set achieved the best result of 3.31%, this training set worked well only for Extra Fancy apples which had dark red color, and it resulted in a greater error for Cull and Fancy apples, compare with the orange and stripe sets and the combined training set. For Fixed SVM, it was difficult to obtain a fixed single separation hyperplane that would cover a wide range of color for apples, which is a limitation for fixed thresholding methods. Adjustable SVM, on the other hand, achieved stable and accurate results for all grade categories when any one training set was used. The best result was obtained for the orange training set with 1.33% error, while the worst result was acquired for the dark red training set at 1.90% error. Again, the training set with a wider range of color information (all three images) did not lead to the best result for Adjustable SVM. These results confirmed that the proposed segmentation method with the adjustable SVM is very robust and flexible against the color variation of fruit that may be caused by the variation of the lighting condition or the color of fruit. The proposed Adjustable SVM method also dramatically reduced the operational time for training.

3.2. Segmentation speed evaluation

Segmentation times for about 200 × 200 size image for the two computers running the C++ program were calculated for speed evaluation. It took 5 ms using Computer A (Pentium M 1.7 GHz, 2 GB Memory, Windows XP 32 bit), which was an old laptop computer and had only a single core CPU. It took only 1.5 ms using Computer B (Corei7-2600K 3.4 GHz, 16 GB Memory, Windows 7 64 bit), which was a relatively new desktop computer. In addition, the processing time using Computer B was calculated only using one of the 8 cores. The processing time could be further reduced when a multi-core processor is used. These speeds are fast enough to implement further image processing algorithms, such as color grading and defects detection, for real-time application.

3.3. Generalization of the segmentation algorithm

Although the proposed segmentation method was primarily developed for apple grading and sorting, the method can be applied to other agricultural applications. The general procedures for implementing the method are summarized as follows:

(1) Select the number of image channels.
(2) Obtain a linear separation hyperplane of the selected channels space by the linear SVM.
(3) Calculate the SVM grayscale image.
(4) Apply Otsu's method to the SVM grayscale image for segmentation.
(5) Refine the segmentation by finding the minimum threshold with regional operations as needed.

The proposed algorithm is not just limited to RGB images, but can also be applied for other types of multi-channel images such as hyperspectral images, RGB + NIR images.

4. Conclusion

An automatic adjustable color segmentation method based on linear SVM and Otsu's method was developed. A SVM grayscale image was generated using a classification hyperplane in the 3-D RGB space calculated with the linear SVM. An optimal threshold was estimated by finding the minimum threshold around the fruit boundary.

Evaluation of the algorithm using the linear SVM for 300 'Delicious' apples showed consistent and accurate results with the average errors of 1.33%, 1.36%, 1.90%, and 1.42% for the orange, stripe, and dark red training sets and their combination training set, respectively. Fast processing speeds of 5 ms and 1.5 ms were achieved when using 1.7 GHz and 3.4 GHz CPU, respectively. The algorithm is thus fast enough for implementing subsequent image processing algorithms for such tasks as color grading and defects detection in real-time application.

Although the proposed segmentation method was developed to segment apples from the background, it could be easily adapted for general inspection of fruits, vegetables and other agricultural products, and for segmentation of images acquired in the outdoor environment in such applications as crop row detection, fruit detection in orchard, etc.

Acknowledgements

The research was supported in part by funds from Michigan Apple Committee under the Trust Fund Cooperative Agreement No. 58-3635-9-570 between the Michigan Apple Committee and the USDA Agricultural Research Service.

References

Baranowski, P., Mazurek, W., Wozniak, J., Majewska, U., 2012. Detection of early bruises in apples using hyperspectral data and thermal imaging. Journal of Food Engineering 110 (3), 345–355.
Blasco, J., Aleixos, N., Molto, E., 2003. Machine vision system for automatic quality grading of fruit. Biosystems Engineering 85 (4), 415–423.
Chinchuluun, R., Lee, W.S., Ehzani, R., 2009. Machine vision system for determining citrus count and size on a canopy shake and catch harvester. Applied Engineering in Agriculture 25 (4), 451–458.
Cortes, C., Vapnik, V., 1995. Support-vector network. Machine Learning 20, 273–297.

ElMasry, G., Cubero, S., Moltó, E., Blasco, J., 2012. In-line sorting of irregular potatoes by using automated computer-based machine vision system. Journal of Food Engineering 112 (1–2), 60–68.

Fitzgibbon, A.W., Fisher, R.B., 1995. A buyer's guide to conic fitting. In: Pycock, D. (Eds.). British Machine Vision Conference 1995. Proceedings of the 6th British Machine Vision Conference, British Machine Vision Association Press, Birmingham, UK, pp. 513–522.

Garrido-Novell, C., Pérez-Marin, D., Amigo, J.M., Fernández-Novales, J., Guerrero, J.E., Carrido-Varo, A., 2012. Grading and color evolution of apples using RGB and hyperspectral imaging vision cameras. Journal of Food Engineering 113 (2), 281–288.

Li, Q., Wang, M., Gu, W., 2002. Computer vision based system for apple surface defect detection. Computers and Electronics in Agriculture 36 (2–3), 215–223.

Lunadei, L., Diezma, B., Lleó, L., Ruiz-Garcia, L., Cantalapiedra, S., Ruiz-Altisent, M., 2012. Monitoring of fresh-cut spinach leaves through a multispectral vision system. Postharvest Biology and Technology 63 (1), 74–84.

Mery, D., Pedreschi, R., 2005. Segmentation of colour food images using a robust algorithm. Journal of Food Engineering 66 (3), 353–360.

Meyer, G.E., Neto, J.C., 2008. Verification of color vegetation indices for automated crop imaging applications. Computers and Electronics in Agriculture 63 (2), 215–223.

Mitra, P., Shankar, B.U., Pal, S.K., 2004. Segmentation of multispectral remote sensing images using active support vector machines. Pattern Recognition Letters 25 (9), 1067–1074.

Moreda, G.P., Ortiz-Canavate, J., Garcia-Ramos, F.J., Ruiz-Altisent, M., 2009. Non-destructive technologies for fruit and vegetable size determination – a review. Journal of Food Engineering 92 (2), 119–136.

Moreda, G.P., Muñoz, M.A., Ruiz-Altisent, M., Perdigones, A., 2012. Shape determination of horticultural produce using two-dimensional computer vision – a review. Journal of Food Engineering 108 (2), 245–261.

Nashat, S., Abdullah, A., Aramvith, S., Abdullah, M.Z., 2011. Support vector machine approach to real-time inspection of biscuits on moving conveyor belt. Computers and Electronics in Agriculture 75 (1), 147–158.

Otsu, N., 1979. A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man, and Cybernetics 9 (1), 62–66.

Sahoo, P.K., Soltani, S., Wong, A.K.C., 1988. A survey of thresholding techniques. Computer Vision, Graphics, and Image Processing 41 (2), 233–260.

Sun, D. (Ed.), 2008. Computer Vision Technology for Food Quality Evaluation. Elsevier Inc., New York, 583 pp.

Suzuki, S., Abe, K., 1985. Topological structural analysis of digital binary images by border following. Computer Vision, Graphics, and Image Processing 30 (1), 32–46.

United States Department of Agriculture (USDA), 2002. United States Standards for Grades of Apples. USDA, Washington, DC.

Valous, N.A., Mendoza, F., Sun, D., Allen, P., 2009. Colour calibration of a laboratory computer vision system for quality evaluation of pre-sliced hams. Meat Science 81 (1), 132–141.

Vapnik, V., 1995. The Nature of Statistical Learning Theory. Springer-Verlag, New York, NY.

Yang, W., Lia, D., Zhu, L., Kang, Y., Li, F., 2009. A new approach for image processing in foreign fiber detection. Computers and Electronics in Agriculture 68 (1), 68–77.

Zheng, C., Sun, D., 2008. Image Segmentation Techniques. In: Sun, D. (Ed.), Computer Vision Technology for Food Quality Evaluation. Academic Press, Waltham, Massachusetts, pp. 37–56.

Zheng, L., Zhang, J., Wang, Q., 2009. Mean-shift-based color segmentation of images containing green vegetation. Computers and Electronics in Agriculture 65 (1), 93–98.

Zou, X., Zhao, J., Li, Y., Holmes, M., 2010. In-line detection of apples defects using three color camera system. Computers and Electronics in Agriculture 70 (1), 129–134.