On-tree Fruit Recognition Using Texture Properties and Color Data*

Jun Zhao, Joel Tow and Jayantha Katupitiya
ARC Centre of Excellence for Autonomous Systems
School of Mechanical and Manufacturing Engineering
The University of New South Wales, Sydney, NSW 2052, Australia
J.Katupitiya@unsw.edu.au

***Abstract*— As a prelude to using stereo vision to accurately locate apples in an orchard, this paper presents a vision based algorithm to locate apples in a single image. On-tree situations of contrasting red and green apples as well as green apples in the orchard with poor contrast have been considered. The study found out that the redness in both cases of red and green apples can be used to differentiate apples from the rest of the orchard. Texture based edge detection has been combined with redness measures, and area thresholding followed by circle fitting, to determine the location of apples in the image plane. In the case of severely cluttered environments, Laplacian filters have been used to further clutter the foliage arrays by edge enhancement so that texture differences between the foliage and the apples increased thereby facilitating the separation of apples from the foliage. Results are presented that show the recognition of red and green apples in a number of situations as well as apples that are clustered together and/or occluded.**

***Index Terms*— Texture properties, redness, image processing, fruit recognition**

I. Introduction

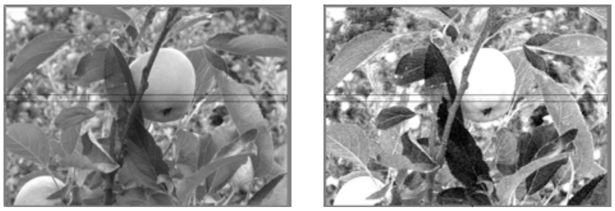
The methodologies and results presented in this paper are directed towards robotic fruit picking. Specifically, the problem of recognizing apples in an orchard as a fruit picking machine gradually traverse the orchard is addressed. The ultimate aim is to use a pair of stereo cameras as part of the set up shown in Fig. 1 so that the manipulator can pick the apples. The manipulator is mounted on a pair of telescopic linear stages so that the robot can reach areas that are outside the width of the tractor. By lifting and lowering the loader, the height of the robot can be adjusted. As a result the cameras looking perpendicular to the direction of travel of the tractor will be able to image the trees with apples.

In the recent past, imaging has been used in recognizing objects in a natural environment. A review of fruit recognition methods can be found in [1]. Most reported are the sorting or grading systems for already harvested fruit where the background is more structured. Among these, [2] reports sorting of two types of apples using machine vision. Electronic means of odor sensing to detect the ripeness using neural networks for on-shelf fruit is presented in [3]. The odor sensor has been developed using a tin oxide chemical sensor array. They claim to have categorized fruit with a success rate of 90%. Other work on fruit grading and shape recognition is reported in [4], [5], [6]. They use neural networks, principal component analysis of Fourier coefficients and thermal imaging, respectively. In other work, near infrared has been used in combination with Haar transforms to classify fruit based on their treatment during growth in either pre or post harvest situations. The work in [7] reports the use of vision, based on a laser range-finder data, to recognize on-tree apples. Use of red color difference to detect on-tree red Fuji apples is presented in [8].

Fig. 1. Fruit picking system

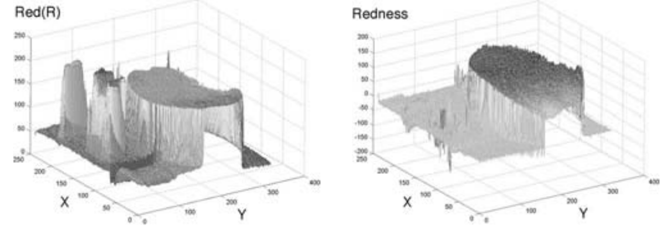
This study presents ways of locating both red and green apples in a single image frame. Having experimented with a number of ways, a procedure that combines texture properties [9] and redness has been chosen as the tool to locate apples in an image. The method is equally applicable to red apples as well as green apples without requiring any change. The apples may be occluded in some cases or may exist in a bunch. Dilation and erosion has been used to eliminate the effects of partial occlusion as much as possible. When apples exist in a bunch, necking regions of the outer contour of the bunch has been used to separate the apples. To locate the centre of the apples that is required to calculate the distance to the apple in a stereo imaging set up, circles of appropriate

*This work is supported in part by the ARC Centre of Excellence programme, funded by the Australian Research Council (ARC) and the New South Wales State Government.



(a) (b)

Fig. 2. (a) Grey scale image, (b) Redness image.



(a) Red plot (b) Redness plot

Fig. 3. Effectiveness of redness[10]

sizes are fitted to each isolated apple. The centres of circles may also be useful as the grasping points to pick apples. The results presented in this study show the cases of red apple recognition, green apple recognition, dealing with occlusion and bunching as well as detecting apples, especially green apples, in extremely cluttered environments.

II. Methodology

This section describes in detail the procedure followed to locate a region in an image that is suspected to be that of an apple. As primary tools texture and color data have been chosen. When a human recognizes a red apple, he or she sees a smooth surface (a texture property) and red (a color property). Thus it is justifiable to attempt to recognize both these simultaneously in an image.

Unlike the color, which is a single pixel based property, texture properties are based on area. The definition of three texture properties, namely; energy, entropy and contrast are presented in the Appendix. Having tried these three experimentally, texture contrast was chosen as the property to be used in fruit recognition. The texture property plays two roles in the recognition procedure. First it isolates areas that have the same texture. Note that as different color regions can have the same texture it alone cannot identify apples. Secondly, an averaging technique has been used to define a quantity for each pixel which is a *texture measure* in the surrounding area. The texture measure is then used in an edge detection algorithm to isolate areas of same texture. In the end, it helps identify the shape of the apple and its centre.

Before calculating the texture measure, the type of image must be decided. All images used were obtained in 24-bit RGB color format. These can be transformed into various forms such as grey level, red and redness. Shown in Fig. 2(a) is the grey level version of green apples in green foliage background. Fig. 2(b) shows the redness version of the same image, generated using (1) which for pure white objects will give a redness value of 255.

$$r = 3\ R - (G + B) \tag{1}$$

It can be clearly seen that apples have much higher contrast in Fig. 2(b). Shown in Fig. 3 are two plots generated from an image of a red object in a cluttered background. The large approximately circular plateau represents the red object. As can be seen, all spurious prominency in red (e.g. due to white objects) in Fig. 3(a) have disappeared in Fig. 3(b). Based on this comparative study, for all further processing, redness buffer will be used.

A. Texture Measure Calculation

Let the redness buffer be $\mathbf{R}$ with elements $r(i,j)$ and the corresponding texture buffer be $\mathbf{T}$ with elements $t(i,j)$. $\mathbf{R}$ is converted to $\mathbf{T}$ as follows. First, all possible redness values are quantized to 16 coarse redness levels. Thus $G = 0, 2, \ldots, 15$ (see Appendix). Ignoring the irregularities at the image borders, each time a 5×5 pixel region is chosen. The region chosen first is at the top left hand corner of the image. For this region $p(a, b|d, q)$ is calculated for $d = 3$ and $q = 0$. The resulting $\mathbf{P}$ matrix is 16×16. Substituting the elements of $\mathbf{P}$ in (10)the texture contrast measure for the centre pixel of the 5×5 region is calculated. Next the 5×5 region is moved to right by 1 pixel and the procedure is repeated. When one entire horizontal line is finished, the very first region is moved downwards by one pixel and the procedure is repeated. This way, excluding a 2 pixel wide region right around the image, an entire texture buffer was generated.

B. Edge Detection

The buffer $\mathbf{T}$ is then processed to detect the texture measure transition boundaries by processing one row of data at a time. To keep the boundaries from shifting a Canny detector is used [11]. Since Canny detector is based on differential operators, the data in $\mathbf{T}$ must be filtered first. This was carried out with an ISEF [12]. This process gives the edges in the image with sub-pixel values. As redness data is not used in edge detection, they need not be filtered.

C. Fruit Recognition

All open edge contours resulting from edge detection were discarded. Let the remaining number of closed contours that can be extracted from $\mathbf{T}$ be $C_k^{''}$. Each of these closed regions have its own texture value t_k. As can be seen from (10) uniform color areas give low texture contrast values. Therefore, the set of contours were partitioned into two sets, $C_k^{'}$ in which $t_k \leqslant t_o$ and $\overline{C_k^{'}}$ in which $t_k > t_o$ where t_0 is a sufficiently low texture threshold. Note that thresholding is done after contour extraction and as such the contour shapes are not affected by the thresholding.

Despite the border irregularities mentioned earlier, for all algorithmic purposes both buffers $\mathbf{R}$ and $\mathbf{T}$ are of the same

Fig. 4. Smaller green apples in a cluttered background

Fig. 5. Laplacian filtered image of Fig. 4

size $n_r \times n_c$. Let $R = \{(i,j)\}$ be the set of paired indices with $i = 0, 1, \ldots, n_r$ and $j = 0, 1, \ldots, n_c$ and let $\{C\}$ to denote the set of (i,j) values within a contour C. Using $C_k^{'}$ a new buffer $\mathbf{T}'$ is generated as follows.

$$\forall \quad (i,j) \in R, \quad t'(i,j) = \left\{ \begin{array}{ll} 1; & (i,j) \in \{C_k^{'}\} \\ 0; & (i,j) \notin \{C_k^{'}\} \end{array} \right. \tag{2}$$

Note that some of the contours in $C_k^{'}$ may be too small to be considered as apples. However, they may form part of an apple, for example, separated by a tiny branch. To merge areas that possibly exist adjacent to each other, dilation and erosion [13] is carried out on $\mathbf{T}'$. Subsequent to dilation and erosion, contour extraction must be repeated on the buffer $\mathbf{T}'$ to determine the new set of closed contours. To maintain the simplicity and continuity of notation, the set of newly extracted contours is denoted by $C_k^{'}$. Next, the contours $C_k^{'}$ are further partitioned into two sets; C_k in which $A(C_k^{'}) \geqslant A_{min}$ and $\overline{C_k}$ in which $A(C_k^{'}) < A_{min}$, where A_{min} is the area thresholding value and $A(C)$ gives the area within the

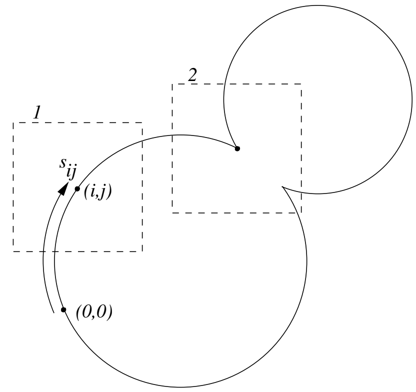


Fig. 6. Separating joined contours

closed contour C. At the end of this process there is a set of closed contours $C_k, k = 1, 2, \ldots, n_k$ which has low enough texture property and large enough surface areas to suspect them as apples.

Next redness values will be used to qualify these contoured areas as those of apples. Using a redness value r_0 as the redness threshold, a new buffer $\mathbf{R}'$ is generated as follows;

$$\forall \quad (i,j) \in R, \quad r'(i,j) = \left\{ \begin{array}{ll} 1; & r(i,j) \geqslant r_0 \\ 0; & r(i,j) < r_0 \end{array} \right. \tag{3}$$

A new buffer $\mathbf{F}$ with elements $f(i,j)$ is created by,

$$\forall \quad (i,j) \in R, \quad f(i,j) = \min\{r'(i,j), t'(i,j)\} \tag{4}$$

This represents element-wise ANDing of $\mathbf{R}'$ and $\mathbf{T}'$. Then for each C_k, a v_k and a u_k were calculated as follows:

$$v_k = \sum_{(i,j)\in\{C_k\}} t'(i,j) \tag{5}$$

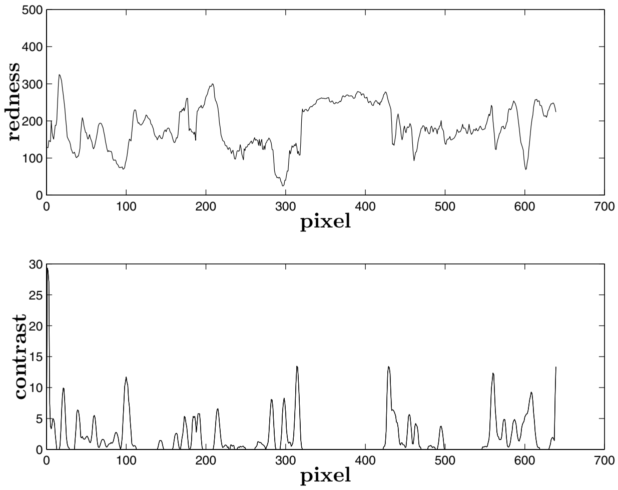
$$u_k = \sum_{(i,j)\in\{C_k\}} f(i,j) \tag{6}$$

C_k is considered an apple contour if $u_k/v_k \geqslant 0.9$. This means that 90% or more pixels in each of the contours of C_k were qualified by redness property. However, entire contours extracted from the buffer $\mathbf{T}'$ are kept as the shapes of apples.

D. Pre-processing

In some situations, the texture in redness image show poor contrast. This is particularly true when the apples are imaged at an increased distance requiring the use of a smaller A_{min}. Further, most of the background areas may have the same texture property as well as the redness values. This will tend to confuse background areas as apples or it will tend to merge areas of apples with those of the background. Such images require pre-processing before applying the algorithm described in the last few sections. A color image that has these complications are shown in Fig. 4. Texture and redness of some of the background areas are very close to those of apples. A solution is to carry out edge enhancement so that the background, which has barely noticeable edges will become

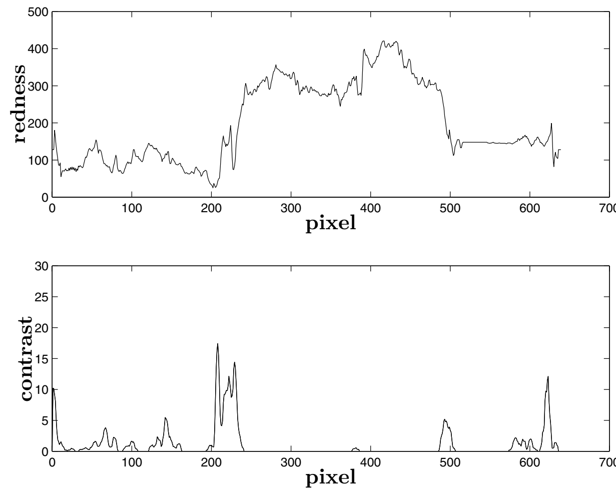
(a) On-tree green apples



(b) Redness plot of (a)

Fig. 7. Redness of green apples

(a) On-tree red apples



(b) Redness plot of (a)

Fig. 8. Redness of red apples

more prominent there by changing the texture properties while due to absence of edges on the apples, their texture will remain relatively unchanged. In this study, Laplacian filters have been used to pre-process the original image. The resulting image is shown in Fig. 5. The results of processing these images is presented in Section III-D.

E. Post-processing

Post processing involves the separation of bunched apple contours and determining the location of each apple in the original image by circle fitting. To separate the contours formed by merging apples, above normal sized contours will be represented as parametric curves of length s. Thus, each point (i, j) of the contour will be assigned a length s_{ij} along the contour (see Fig. 6). At each test point (i, j), a rectangular search region of 20×20 pixels centered around (i, j) is used to locate other data points (k, l) of the array which has excessively large s_{kl} values. By locating (i, j) and (k, l) that corresponds to minimum $(i-k)^2 + (j-l)^2$ is chosen as the necking point of the contour and were used to split the contour for circle fitting.

A simple circle fitting algorithm determines the size and centre of the apple.

III. Results

A. Redness

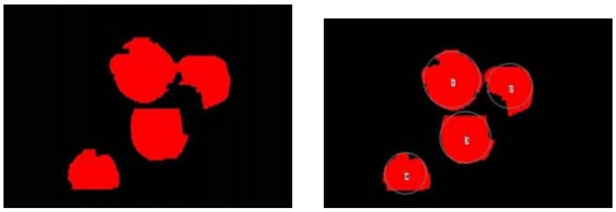
It was mentioned earlier that the redness property works equally well for on-tree green apples as well as red apples. Fig. 7(a) shows on-tree green apples and Fig. 8(a) shows on-tree red apples. For the two cases the redness and texture contrast are plotted in Fig. 7(b) and Fig. 8(b). Fig. 7(a) shows a rectangular region in the middle of the image. The color values plotted are for the centre horizontal line of pixels of the rectangle. The texture values plotted are for the same horizontal line. The plots for Fig. 8(a) were generated in a similar manner. As can be seen in both red and green apples, the redness curve produced the highest color contrast. The texture contrast shown are already filtered using ISEF. It can be seen that the texture stayed steady within an apple region at a very low value. It is readily evident that apples can be identified by combining texture contrast and redness.

B. Green apple recognition

The first case is the close up image of the green apple in the orchard (see Fig. 7). Note that in this figure, there is a tiny branch of the apple tree separating the apple into two regions. The result obtained without area thresholding and dilation

(a) (b)

Fig. 9. Green apple detection using redness



(a) Connected apples (b) Separated apples

Fig. 10. Processed images of Fig. 8

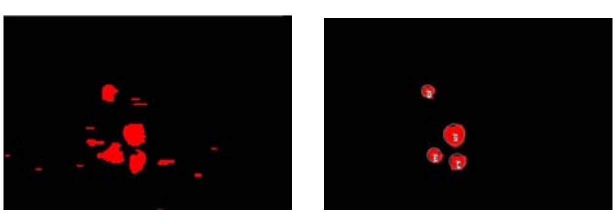
and erosion, is shown in Fig. 9(a). The result with dilation and erosion, area thresholding and circle fitting is shown in Fig. 9(b). It is clear from the result that dilation and erosion significantly reduces the contour complexity and delivers a more realistic result. Further, despite the apple being green, the redness has been used to correctly identify both green apples.

C. Clustered Red Apple Recognition

An image of clustered red apples are shown in Fig. 8. Fig. 10(a) shows the result when post processing is not used. As can be seen two of the apple contours have overlapped. In most situations, the overlapped regions are very small. This is due to a texture and color variation as the apples' surfaces curve away. Fig. 10(b) shows the result with post processing. The circles identify the approximate size of the apple and its centre. In obtaining this result, to maintain computational simplicity no dilation and erosion was used.

D. Laplacian Filtering

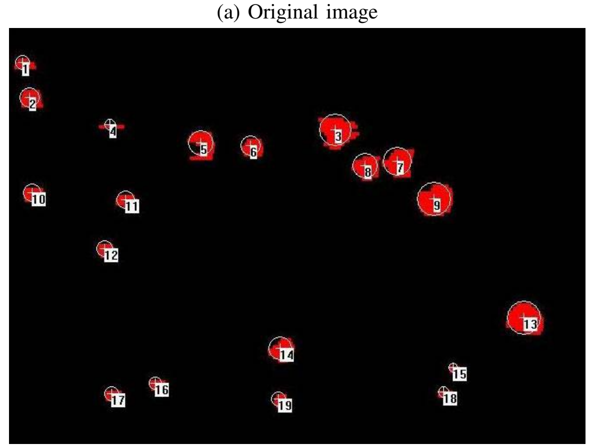
The effect of using Laplacian filtering to deal with distant backgrounds that resemble apple texture and redness is demonstrated by processing the images in Fig. 4 and Fig. 5. The results are shown in Fig. 11(a) and Fig. 11(b), respectively. When the original image was processed as is,



(a) Without Laplacian (b) With Laplacian

Fig. 11. Effect of Laplacian filtering

(a) Original image



(b) Result for (a)

Fig. 12. Distant red apples

some parts of the background merged with some of the apples. The reason was that during texture thresholding, some of the background texture areas were retained as areas belonging to apples. As a result the outcome shown in Fig. 11(a) is undesirable. When observed closely in Fig. 4, it can be seen that the background areas have edges that can be enhanced with an edge enhancing filter. As these background edges, in particular due to grass or apple leaves, lie very close to each other, the background texture changed significantly after the application of Laplacian edge enhancing filter. Since there were no such edges within the apple areas, the texture change within the apples were minimal. This increased contrast in texture helped separate the background regions from apple regions. Once again to maintain computational simplicity, dilation and erosion was not used.

E. Distant Apples

In a robotic fruit picking scenario, it is important to take global images of the orchard and to identify the regions where fruit may be present so that cameras that may be mounted on the robot arm itself can be guided to take close up images to precisely locate the apples. Such a global image is shown in Fig. 12(a). The result obtained is shown in Fig. 12(b). Note that once again no dilation and erosion was used. In the original image 20 apples are clearly visible. The system recognized 18 of these apples correctly plus one partially visible apple. It failed to recognize two of the well exposed apples. Given that this is a global view, it is only necessary to detect the *regions* where fruit may be present. To that effect this is a successful result.

IV. Conclusion

An algorithm that can be used to recognize on-tree apples has been presented. As primary tools, it uses the texture property contrast and the color property redness. It was shown that redness works equally well for red apples as well as green apples. The contours that form boundaries of apples have been extracted using edge detection on texture images. To avoid contour shape distortion, thresholding has been carried out on contours rather than on individual pixels. The only modification that was allowed for the contour shapes were through dilation and erosion. This procedure has been followed to maintain a high accuracy of contour shape and size. To eliminate the merging of the backgrounds, edge enhancing filters have been used thereby changing the background texture while keeping the apple texture unchanged. This increased texture contrast helped identify apples separately from background. The algorithm worked equally well for close ups as well as distant images of apples.

Appendix

Texture Properties

A. Co-occurrence Matrix

Let a rectangular region of an image be of size $N_c \times N_r$ pixels. Let the set of grey levels be $G = 0, 1, \ldots, N_q - 1$. The co-occurrence matrix $\mathbf{P}_{(d,q)}$ is a square matrix of size $N_q \times N_q$, that will list the relative frequency of occurrence of a pair of pixels, in the direction q, separated by a distance d to have the grey levels a and b [9]. An element of $\mathbf{P}_{(d,q)}$ can be expressed as $p(a,b|d,q)$. Note that a and b refers to grey levels, d distance between pixels and q the direction. Let two general pixels within the chosen region be (k,l) and (m,n) where $k, m = 1, 2, \ldots, N_c$ and $l, n = 1, 2, \ldots, N_r$. Then,

$$p(a,b|d,q) = \sum [(k,l),(m,n)] \tag{7}$$

where, $[(k,l),(m,n)] = 1$ if $|k-m| = q,\ |l-n| = d,\ g(k,l) = a,\ g(m,n) = b$, otherwise 0. The function $g(k,l)$ gives the grey level of the pixel (k,l).
Using the co-occurrence matrix, energy is defined as,

$$E(d,q) = \sum p(a,b|d,q)^2 \tag{8}$$

The entropy is defined as,

$$H(d,q) = -\sum \{p(a,b|d,q) \log p(a,b|d,q)\} \tag{9}$$

and the contrast is defined as,

$$I(d,q) = \sum (a-b)^2\ p(a,b|d,q) \tag{10}$$

B. Example

For an R of 4×4 with $G = \{0, 1, 2, 3\}$,

$$R = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 2 & 2 & 2 \\ 2 & 2 & 3 & 3 \end{bmatrix}$$

the co-occurence matrices in horizontal and vertical directions, respectively with $d = 1$ can be calculated using (7) as,

$$\mathbf{P}_{(1,h)} = \begin{pmatrix} 4 & 2 & 1 & 0 \\ 2 & 4 & 0 & 0 \\ 1 & 0 & 6 & 1 \\ 0 & 0 & 1 & 2 \end{pmatrix}; \quad \mathbf{P}_{(1,v)} = \begin{pmatrix} 6 & 0 & 2 & 0 \\ 0 & 4 & 2 & 0 \\ 2 & 2 & 2 & 2 \\ 0 & 0 & 2 & 2 \end{pmatrix}$$

The value of 4 in the $(0,0)$th element of $\mathbf{P}_{(1,h)}$ is the total number of adjacent (i.e. $d = 1$) pixel pairs in horizontal direction which has grey levels $(0,0)$. The $(0,1)$th element gives the total number of adjacent pixel pairs in the horizontal direction which has grey levels $(0,1)$ and so forth. The elements in these matrices can be substituted in (8), (9) and (10) to obtain energy, entropy and contrast.

References

[1] A. R. Jimenez, A. K. Jain, R. Ceres, and J. L. Pons, "Automatic fruit recognition: a survey and new results using range/attenuation images," *Pattern Recognition*, vol. 32, pp. 1719–1736, October 1999.

[2] V. Leemans, H. Magein, and M. F. Destain, "AE–automation and emerging technologies: On-line fruit grading according to their external quality using machine vision," *Biosystems Engineering*, vol. 83, pp. 397–404, December 2002.

[3] J. Brezmes, E. Llobet, X. Vilanova, J. Orts, G. Saiz, and X. Correig, "Correlation between electronic nose signals and fruit quality indicators on shelf-life measurements with pinklady apples," *Sensors and Actuators B: Chemical*, vol. 80, pp. 41–50, November 2001.

[4] T. Morimoto, T. Takeuchi, H. Miyata, and Y. Hashimoto, "Pattern recognition of fruit shape based on the concept of chaos and neural networks," *Computers and Electronics in Agriculture*, vol. 26, pp. 171–186, April 2000.

[5] I. Paulus and E. Schrevens, "Shape characterization of new apple cultivars by fourier expansion of digitized images," *Journal of Agricultural Engineering Research*, vol. 72, pp. 113–118, February 1999.

[6] D. Stajnko, M. Lakota, and M. Hocevar, "Estimation of number and diameter of apple fruits in an orchard during the growing season by thermal imaging," *Computers and Electronics in Agriculture*, vol. 42, pp. 31–42, January 2004.

[7] A. R. Jimenez, R. Ceres, and J. L. Pons, "Vision system based on a laser range-finder applied to robotic fruit harvesting," *Machine Vision and Applications*, vol. 11, pp. 321–329, April 2000.

[8] D. M. Bulanon, T. Kataoka, Y. Ota, and T. Hiroma, "AE–automation and emerging technologies: A segmentation algorithm for the automatic recognition of Fuji apples at harvest," *Biosystems Engineering*, vol. 83, pp. 405–412, December 2002.

[9] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*, 2nd ed. Thompson publishing, 1998.

[10] J. Tow, "Mobile robot localization and navigation through computer vision," BE Thesis, School of Mechanical and Manufacturing Engineering, The University of New South Wales, Sydney NSW 2052, Australia, November 2004.

[11] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679 – 698, 1986.

[12] J. Shen and S. Castan, "An optimal linear operator for step edge detection," *CVGIP: Graphical Models and Image Processing*, vol. 54, no. 2, pp. 112 – 133, 1992.

[13] R. Woods and R. Gonzales, *Digital Image Processing*, 2nd ed. Upper Saddle River, New Jersey: Prentice-Hall, 2002.