ELSEVIER Computers and Electronics in Agriculture 14 (1996) 77–89

Computers and electronics in agriculture

Accurate blemish detection with active contour models

Qingsheng Yang *, John A. Marchant

Image Analysis and Control Group, Silsoe Research Institute, Wrest Park, Silsoe, Bedford MK45 4HS, UK

Accepted 31 July 1995

Abstract

This paper presents a novel image analysis scheme for accurate detection of fruit blemishes. The detection procedure consists of two steps: initial segmentation and refinement. In the first step, blemishes are coarsely segmented out with a flooding algorithm and in the second step an active contour model, i.e. a snake algorithm, is applied to refine the segmentation so that the localization and size accuracy of detected blemishes is improved. The concept and the formulation of the snake algorithm are briefly introduced and then the refinement procedure is described. The initial tests for sample apple images have shown very promising results.

Keywords: Blemish detection; Fruits; Image processing; Segmentation; Active contour models

1. Introduction

Machine vision is the ultimate tool for the automation of blemish detection in the fruit grading process. In image analysis blemish detection is essentially a segmentation problem. As the size requirement specified in grading standards is one of the most important quality criteria, accurate segmentation of blemishes from fruit surfaces is highly desired.

There exist several image analysis methods for produce blemish detection, such as global grey-level or gradient thresholding, simple background subtraction, statistical classification and colour classification (Yang, 1992). Depending on the application, certain methods are more accurate than others. However, there is little, if any, work specifically addressing accurate blemish detection.

Blemish segmentation is a difficult problem in image analysis, because various types of blemishes with different size and extent of damage may occur on fruit

* Corresponding author.

SSDI 0168-1699(95)00038-0

surfaces. If a blemish appears as very a dark mark on a fruit surface, a simple thresholding of grey-level intensity of reflected light may allow a direct segmentation of the blemish. However, in most cases, the light reflectance from both blemished and non-blemished surfaces varies considerably, and it is impossible to set a single threshold value for the segmentation. For example, a patch of good surface with a relatively dark colour can have similar reflectance as a slightly discoloured blemish on a light coloured surface. In this case, the thresholding method will fail.

In previous work, Yang (1994) developed a region-based segmentation algorithm, called the flooding algorithm, for fruit blemish detection. The main advantage of the technique reported is that it can overcome the difficulties caused by the variation in light reflectance. It is suitable for detection of blemishes with reasonable large variation in size, and it allows the shape and other features of individual blemishes to be collected. However, the algorithm is not particularly accurate. The detected blemish areas may be larger than what the eyes see, because of the smoothing used in the algorithm. In most cases the difference is slight, but in some extreme cases, the areas are significantly larger than the real ones.

In this work, a further study is carried out to improve the accuracy of the segmentation obtained from the flooding algorithm. We investigate applying the "snake", one of the active contour models proposed by Kass et al. (1988), to finding accurate boundaries of blemishes. Active contour models are physically-based models and obey the mechanical laws of elastic media that move and deform in response to applied forces. A snake simulates the fitting of an elastic band to the image feature such as contours to provide a continuous boundary. When guided close to the desired boundary, it will lock onto nearby edges and localizing them accurately. As an example of a more general technique of matching a deformable model to an image by means of energy minimization, snakes are particularly suitable for finding boundaries of biological objects. They have found various applications in agricultural engineering such as animal tracking and monitoring (Marchant and Schofield, 1993).

In this paper we describe how to combine the flooding algorithm and the snake algorithm into a coherent scheme for accurate fruit blemish detection. The flooding algorithm is used for the initial segmentation of blemish areas and the snake algorithm is applied to improve the boundary location of the areas. The experimental results for sample apple images will be presented.

2. Initial segmentation of blemishes

Blemishes can be modelled as water catchment basins in image grey-level landscapes, because they generally appear darker, under diffuse illumination, than their surrounding non-blemished surfaces. Based on this model, a flooding algorithm has been developed to detect the basins. The algorithm works according to the behaviour of water over a topographic relief. A drop of water will always flow downhill until a local minimum is reached. Imagine we gradually flood a topographic relief with water, it will flow along the swiftest descent paths into catchment basins and accumulate until it spills over the lowest points of the basins' rims. When all catchment basins have been filled up, they become lakes of various sizes and depths. The edges

of the lake surfaces delimit the catchment basins, and any extra water will spill out. The lakes represent the blemishes being detected. The algorithm is very effective for detecting patch-like defects without prior knowledge of size, shape and contrast. It is superior to the existing background subtraction method (Rehkugler and Throop, 1989) and thresholding method (Davenel et al., 1988) for fruit blemish detection.

However, the segmentation is coarse in the sense that the boundary localization of the detected blemish areas may be poor. The reason is that to ensure the success of the flooding algorithm, a two-stage smoothing was applied in the image pre-processing, median filtering and Gaussian filtering. The median filtering removes impulsive noise, such as the image sampling noise and spotty lenticels, and preserves edge information, while Gaussian filtering smooths off small features in images. This smoothing has a negative effect on the results in that the detected blemish areas can be larger than we visually perceive. Therefore, a refinement is required after the segmentation to improve the localization accuracy. An extreme example of the enlargement is shown in Fig. 1. In the middle of Fig. 1a there is a dark blemish and a tiny insect hole adjacent to it. This image has been chosen as an example because the greater part of the edge of the blemish is very sharp and the size of the blemish area can be easily perceived by eye. Thus, the adverse effect of smoothing can be clearly seen. By smoothing the image the edge of the blemish is heavily blurred and the insect hole is merged into the blemish region (in this case the merger is acceptable because in grading requirements the sum of blemish areas is counted). The result from the initial segmentation is shown in Fig. 1b as a bright closed line which indicates the border of the detected blemish. It can be seen that the area enclosed by the line is larger than that shown in Fig. 1a. Note that in this particular case simple thresholding may produce a straightforward segmentation with good boundary localization and area size, because the blemish and the insect hole have very good contrast to the rest of surfaces. However, in most cases this good contrast rarely occurs.

3. Formulation of snakes

A snake can be thought as an elastic band which is under the influence of three types of forces; image forces, internal and external constraint forces, and seeks for its minimum energy state. The internal forces of the band come from the material resistance to stretching or compression and bending so to impose a smoothness constraint. The image forces push the band toward salient image features such as lines and edges. The external constraint forces come from high-level sources, possibly specified by human users, or automatic initialisation procedures which put the band near the desired local minimum. The dynamic behaviour of a snake is governed by an energy minimization procedure. When a snake is placed near the desired position in an image, the interaction of the forces will move and deform the snake to fit local image features until an equilibrium state is reached, at which the energy of the snake is a minimum.

The snake's energy is the sum of internal energy E_{intern} and external energy E_{extern}. The latter consists of the energy E_{image} due to image forces and the energy E_{constr} due to external constraint forces, i.e. $E_{\text{extern}} = E_{\text{image}} + E_{\text{constr}}$. The gradients

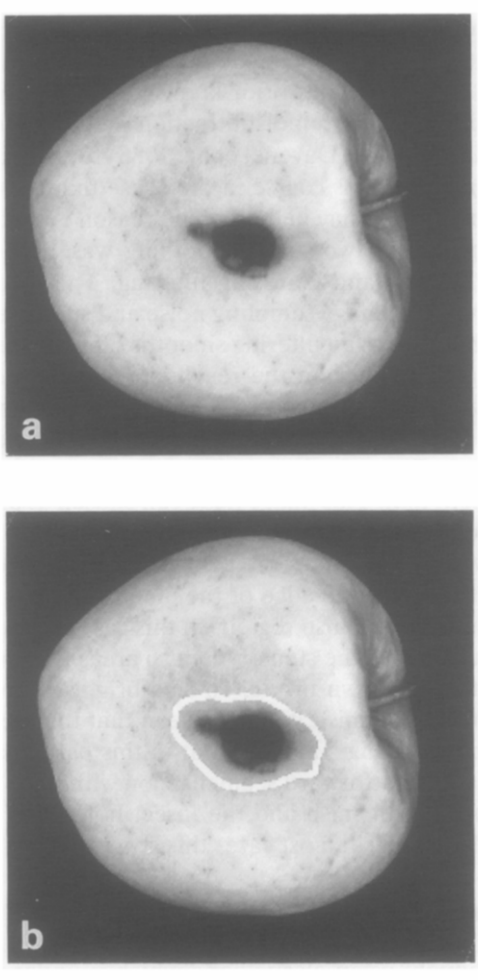

Fig. 1. Result of initial segmentation for an example apple. (a) The original apple image with a blemish in the middle; (b) the detected blemish area which is shown as the area enclosed by the bright line.

of the three energy terms correspond to the three types of forces described above. Following Kass et al. (1988) we represent a snake by $\boldsymbol{v}(s) = (x(s), y(s))$, having the arc length s as parameter ranging from 0 to 1. The x and y are position coordinates of a snake point. The energy functional (a functional is a function of one or more function variables) of the snake is formulated as

$$E_{\text{snake}} = \int_0^1 \left(E_{\text{intern}}[\boldsymbol{v}(s)] + E_{\text{image}}[\boldsymbol{v}(s)] + E_{\text{constr}}[\boldsymbol{v}(s)]\right)\mathrm{d}s. \tag{1}$$

The exact energy function terms come in various forms depending on the problem in hand.

The internal energy which describes the pseudo-material properties of a snake can be written as

$$E_{\text{intern}} = \alpha(s)|\boldsymbol{v}_s(s)|^2 + \beta(s)|\boldsymbol{v}_{ss}(s)|^2 \tag{2}$$

where the subscripts s and ss indicate first-order and second-order differentiation, respectively. The first term represents stretching and the second term bending. $\alpha(s)$ and $\beta(s)$ are weights controlling the relative importance of the two terms. For example, setting $\beta(s) = 0$ allows the snake to have no resistance to bending and to develop corners, so the snake material acts like an elastic string.

A raw image $I(x, y)$ can be considered as a spatially varying potential field. The gradient of the potential will apply spatially varying conservative forces on a snake, which drive the snake to desired image features. The total potential energy can be expressed as a weighed combination of the three energy functionals corresponding to the lines, edges and terminations:

$$E_{\text{image}} = w_{\text{line}}E_{\text{line}} + w_{\text{edge}}E_{\text{edge}} + w_{\text{term}}E_{\text{term}}. \tag{3}$$

By adjusting the weights, a snake will be more keen to search for certain image features rather than others. The simplest energy functional for finding lines is the image intensity itself, $E_{\text{line}} = I(x, y)$, depending on the sign of w_{line} the snake will be attracted either to the lightest or darkest nearby contour. In order to find terminations of line segments and corners, smoothed curvature of plane contours can be used as the energy functional.

We specifically consider edges as image features, which can be expressed as the image gradient magnitude $|\partial I/\partial \boldsymbol{v}|^2$ because edges normally present large image gradients. To attract the snake to edges, the energy functional can be set as:

$$E_{\text{edge}} = -\int_0^1 \left|\frac{\partial I}{\partial \boldsymbol{v}}\right|^2 \mathrm{d}s. \tag{4}$$

The negative sign occurs because minimising the integral with the sign is equivalent to maximising the integral without the sign.

The external constraint forces can be applied from both automatic and manual supervision. For example, to insert a spring-like force between a snake element and a point p in an image, the energy term E_{constr} can be expressed as a tensile energy $c|p - \boldsymbol{v}|^2$, according to the sign of parameter c, the force can be either attractive or repulsive.

The energy minimization of a snake can be carried out by several methods (Williams and Shah, 1992). Kass et al. (1988) used a variational calculus approach which mathematically yields a system of differential equations governing the evolution of the snake through iterations (time):

$$-\alpha\boldsymbol{v}'' + \beta\boldsymbol{v}'''' + \frac{\partial E_{\text{extern}}}{\partial \boldsymbol{v}} = 0.$$

This can be rewritten in (x, y) coordinates:

$$-\alpha x_{ss} + \beta x_{ssss} + \frac{\partial E_{\text{extern}}}{\partial x} = 0 \tag{5}$$

$$-\alpha y_{ss} + \beta y_{ssss} + \frac{\partial E_{\text{extern}}}{\partial y} = 0 \tag{6}$$

Unfortunately, these implicit equations are difficult to solve analytically because the (x, y) must be known before the partial differentiation of E_{extern} can be found. However, the equations can be easily solved with iterative numerical methods.

For implementation on a computer, the snake is discretized into N elements $\{x(s_i), y(s_i)\}$, $i = 1 \ldots N$ by digitizing the interval [0, 1] of parameter s into $N - 1$ equally-spaced subintervals $s_i = (i - 1)h$, where $h = 1/(N - 1)$ is the length of s_i. Each element $(x(s_i), y(s_i))$ can be a polynomial and between the elements are *nodes* $\{(x_i, y_i)\}$, $i = 1 \ldots N$. Thus the discrete formulation of the energy functional in Eq. 1 can be written as:

$$E_{\text{snake}} = \sum_{i=1}^{N} \left(E_{\text{intern}}(i) + E_{\text{extern}}(i)\right).$$

We further approximate $x(s_i)$ and $y(s_i)$ to be constant within individual subintervals, i.e. $(x(s_i), y(s_i)) = (x_i, y_i)$, and replace derivatives with finite differences:

$$\begin{aligned}
x'' &\rightarrow (x_{i+1} - 2x_i + x_{i-1})/h^2 \\
y'' &\rightarrow (y_{i+1} - 2y_i + y_{i-1})/h^2 \\
x'''' &\rightarrow (x_{i+2} - 4x_{i+1} + 6x_i - 4x_{i-1} + x_{i-2})/h^4 \\
y'''' &\rightarrow (y_{i+2} - 4y_{i+1} + 6y_i - 4y_{i-1} + y_{i-2})/h^4.
\end{aligned}$$

Let $\boldsymbol{v}(0) = \boldsymbol{v}(N)$ and $f_x(i) = \partial E_{\text{extern}}/\partial x_i$, $f_y(i) = \partial E_{\text{extern}}/\partial y_i$, Eqs. 5 and 6 are converted to

$$\begin{aligned}
&-a(x_{i+2} - 2x_i + x_{i-2}) + b(x_{i+2} - 4_{i+1} + 6x_i - 4x_{i-1} + x_{i-2}) \\
&\quad + f_x(x_i, y_i) = 0
\end{aligned} \tag{7}$$

$$\begin{aligned}
&-a(y_{i+2} - 2y_i + y_{i-2}) + b(y_{i+2} - 4y_{i+1} + 6y_i - 4y_{i-1} + y_{i-2}) \\
&\quad + f_y(x_i, y_i) = 0 \quad i = 1, \ldots, N
\end{aligned} \tag{8}$$

where $a = \alpha/h^2$ and $b = \beta/h^4$, and for simplicity are assumed constant. For all snake elements we have a set of $2N$ nonlinear differential equations. Solving the equations iteratively results in convergence to an equilibrium configuration $\{(x_i, y_i)\}$ for the snake, defining the shape and the location of a contour in the image.

This discrete model has the physical interpretation of a system of N hinges, connected by springs, and constrained to move in an image potential field. The coordinates (x_i, y_i) represent the positions of the hinges. This physical analogy to a system of springs and hinges guarantees the convergence of the system.

The snake's localization capability depends on the scale of desired low-energy image features. By increasing the scale a snake can be attracted from a considerable distance but this produces poor localization. Reducing the scale makes the snake quickly fall into a local minimum. Kass et al. (1988) employed the scale-space

continuation to enlarge the capture region surrounding a feature. They firstly allowed the snake to reach to equilibrium on a very blurred energy functional (very coarse scale) and then gradually reduced the blurring. This scale-space continuation provides a good medium for obtaining both long-range attraction and good localization.

4. Refinement of the initial segmentation

In our current system, blemishes are detected by two consecutive steps. Firstly the flooding algorithm is used to segment out approximate areas of blemishes. Then a snake is implemented to refine the segmentation. Once the first step is completed the boundary of a segmented area is available. In the second step we place a snake on the boundary of the area and let it go. In the process of seeking an energy minimum, the snake will move to a more accurate location of the boundary. Because the initially segmented area is normally larger than the real one, the snake's movement is basically contraction.

For our snake the general formulation described above can be simplified. As the first-step segmentation provides an excellent automatic initialization mechanism for the snake and no further supervision or interaction is needed, the energy term E_{constr} due to external constraint forces can be eliminated and the external energy E_{extern} is only related to image forces. The selection of image forces depends on the features which are searched for in the images. We expect the snake to move to the positions where the image gradients are locally maximum. These positions usually correspond to the boundary of the real blemished area, so we select a large w_{edge}. As we use closed loop snake and we are not interested in corners, we set the w_{term} in Eq. 3 to zero. This results in

$$f_{\boldsymbol{v}} = \partial E_{\text{extern}}/\partial \boldsymbol{v} = \partial E_{\text{image}}/\partial \boldsymbol{v} = w_{\text{line}}\partial E_{\text{line}}/\partial \boldsymbol{v} + w_{\text{edge}}\partial E_{\text{edge}}/\partial \boldsymbol{v}$$

where

$$w_{\text{line}}\partial E_{\text{line}}/\partial \boldsymbol{v} = w_{\text{line}}\partial I/\partial \boldsymbol{v}$$

$$w_{\text{edge}}\partial E_{\text{edge}}/\partial \boldsymbol{v} = -w_{\text{edge}}\partial(\partial I/\partial \boldsymbol{v})^2/\partial \boldsymbol{v}$$

$$\left(\frac{\partial I}{\partial v}\right)^2 = \left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial}{\partial y}\right)^2 = I_x^2 + I_y^2$$

here subscripts represent partial differentiations. Approximating the derivatives with finite differences:

$$I_x = [I(x_{i+k}, y_i) - I(x_{i-k}, y_i)]/(2k)$$

$$I_y = [I(x_i, y_{i+k}) - I(x_{i-k}, y_{i-k})]/(2k)$$

$$I_{xx} = [I(x_{i+k}, y_i) - 2I(x_i, y_i) + I(x_{i-k}, y_i)]/(k^2)$$

$$I_{yy} = [I(x_i, y_{i+k}) - 2I(x_i, y_i) + I(x_i, y_{i-k})]/(k^2)$$

$$I_{xy} = \{[I(x_{i+k}, y_{i+k}) - I(x_{i-k}, y_{i+k})] - [I(x_{i+k}, y_{i-k}) - I(x_{i-k}, y_{i-k})]\}/(2k^2)$$

where k is the step length, then f_x and f_y in Eqs. 7 and 8 become:

$$f_x = w_{\text{line}} I_x - 2w_{\text{edge}}(I_{xx} I_x + I_{xy} I_y)$$
$$f_y = w_{\text{line}} I_y - 2w_{\text{edge}}(I_{xy} I_x + I_{yy} I_y).$$

To emphasize edges, w_{line} is tuned to be small. To preserve corners and irregularity of a boundary, the weight b for bending terms in Eqs. 7 and 8 is also set to a small value. As a result, the snake in our blemish detection is basically under the influence of two forces; one is the natural tendency of the elastic band to contract, and the other is the image force which works against the contraction and tries to hold the snake at the position where the image gradient is a maximum.

The boundary of an initially segmented area can sometimes be quite a distance from the desired location. This can be seen in Fig. 1. Therefore, a snake has to move a fairly large distance from its initial position. To prevent the snake from sticking on undesired local image features we use a series of smoothings similar to the scale-space continuation. Recall that in the initial segmentation a two-stage smoothing is used, so a Gaussian smoothed image and a median-filter smoothed image are available. Because of this convenience, we first use the Gaussian smoothed image in the calculation of image forces, so the snake can move a large distance. When equilibrium is reached, the median-filter smoothed image is used, and after this, the low-pass filtered image from the original one.

5. Test results

The proposed method has been tested with a laboratory imaging system for apples of several varieties. On those sample apples there were various patch-like defects such as bruises, insect bites and scabs. The fruits were manually placed in the middle of the viewing field in a lighting chamber which provided diffuse illumination. The images were captured with a CCD monochromatic video camera, which was mounted at the top of the lighting chamber and looked downwards. The video signal was sent to and digitized in a transputer-based image processing system. The images were stored in computer memory in the size of 256×256 pixels and in the grey-level range [0, 255].

Some sample images are shown in Fig. 2. There is a russet patch on the apple surface in Fig. 2a, a bruise with diffused edge in Fig. 2b, a scab in Fig. 2c and a bruise in Fig. 2d. These blemishes are of different sizes, shapes and contrasts.

In the test, the boundary of a blemish was divided into 20 snake elements. Since the snake algorithm is essentially a model-based method, the parameters in a built model should be selected according to the desired behaviour of the snake. As discussed in the above section, we expected our snake to grip on the position where the image gradient was locally maximum. Therefore, we chose a relatively large edge weighing constant $w_{\text{edge}} = 0.9$ and a small image grey-level weighing constant $w_{\text{line}} = 0.1$. We also wanted to preserve the boundary shape of blemishes, so we selected a small bending coefficient $b = 0.1$ and a tensile coefficient $a = 0.5$. The algorithm was also tried with small variations in these values, with only a minor

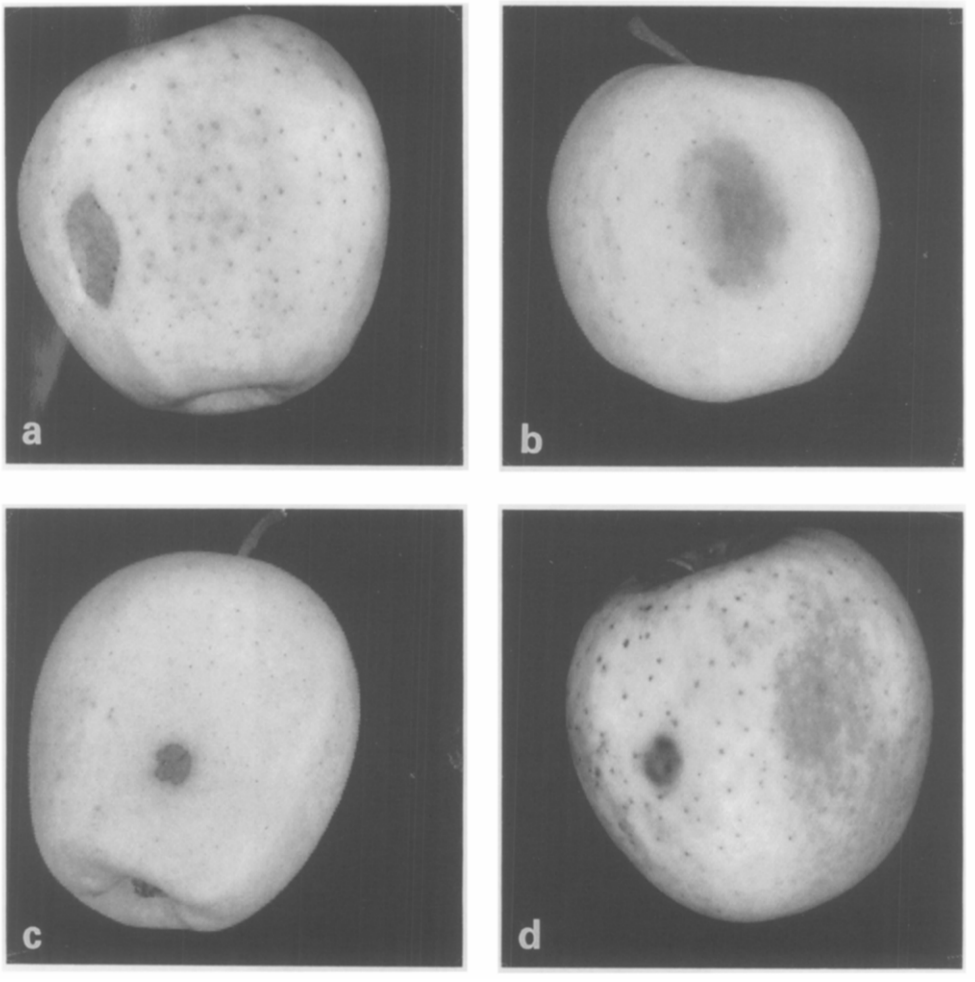

Fig. 2. Example apple images with various blemishes: (a) a russet patch; (b) a bruise with diffused edge; (c) a scab; and (d) a bruise.

effect on the snake's final position. Therefore, it was assumed that within a small range the exact choice was not very critical and the above values were used for the rest of the work. As the parameters were all constant, our snakes were of uniform properties.

Fig. 3 shows the test result of Fig. 1a. The bright dots in Fig. 3a indicate the final location of refined boundary. The radial lines in Fig. 3b show the path of the snake elements' movement from the initial location shown in Fig. 1b to the final boundary. Fig. 4 shows the output of the system for sample images in Fig. 2. The initial boundaries of blemishes are highlighted by the bright closed lines and the final boundaries are represented by the dots. Comparing the bright contours obtained from initial segmentation, i.e. just using the flooding algorithm alone, we

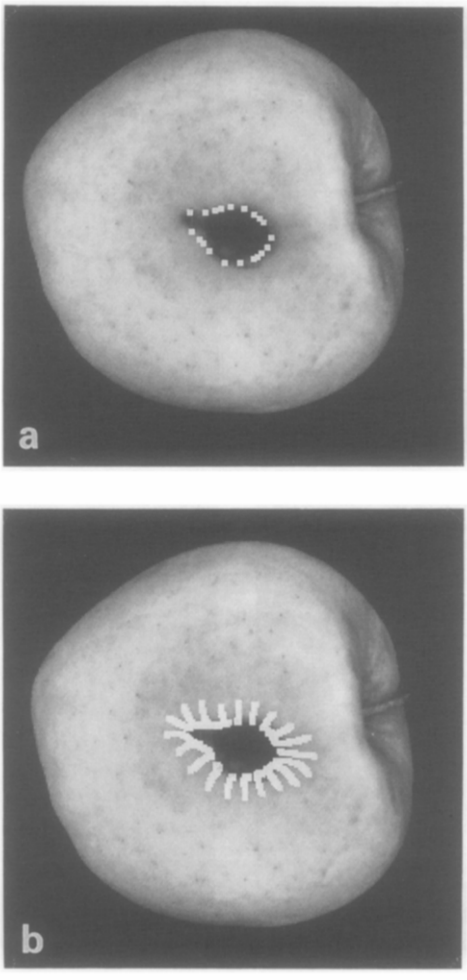

Fig. 3. Result of boundary refinement for the apple image in Fig. 1. (a) The final boundary of the detected blemish, shown as the contour indicated by the thick dots; (b) the motion path of the snake elements from the initial position shown in Fig. 1b to the final boundary.

can see that the final boundaries indicated by the thick dots are more accurate. The improvement in boundary localization varies according to the output from the first-step segmentation. If the initial boundary localization is very poor, the improvement is significant, as shown in Figs. 3, 4c and 4d.

The proposed method works well in the cases where a simple thresholding method will fail to detect blemishes. This can be seen in Fig. 2b in which the boundary of the bruise is not well defined and the left part of the bruise area has the same grey-level values as the surfaces near the fruit contour. It is not possible to set a fixed threshold value in this case. With our system the bruise is detected and the final result accords well to our subjective perception.

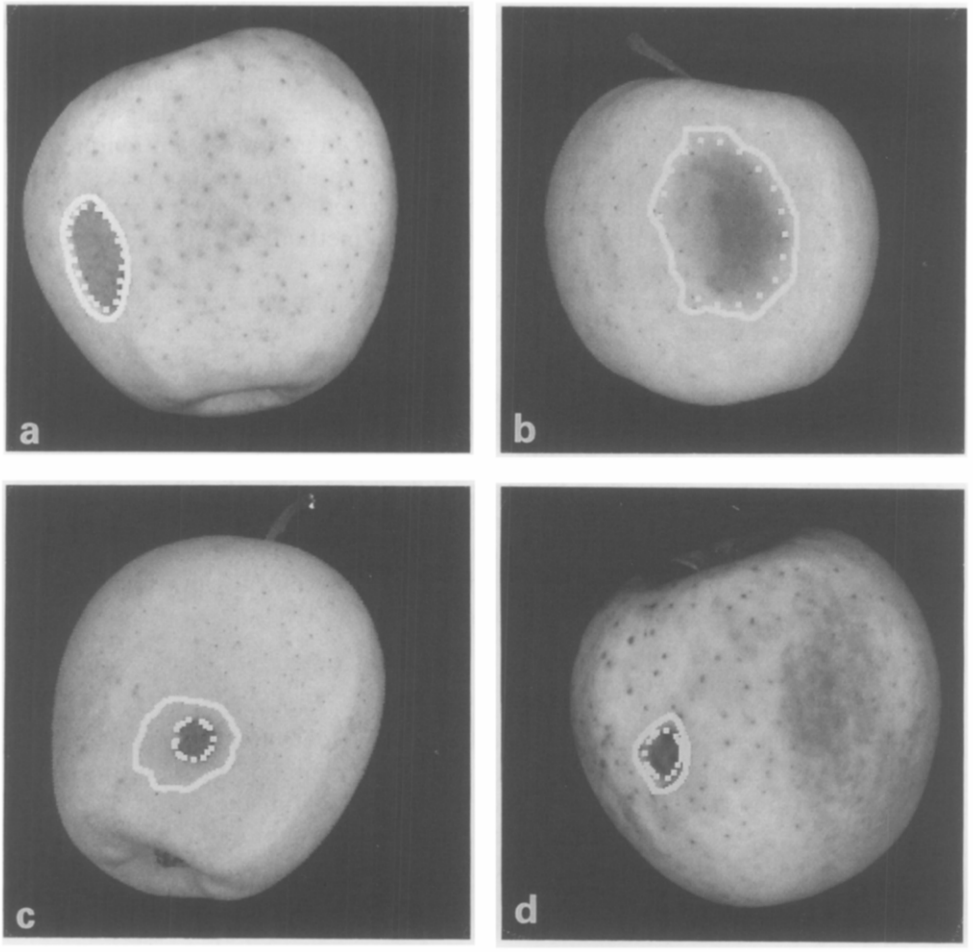

Fig. 4. System output for the apple images in Fig. 2. The bright lines are the results from the first-step segmentation and the dots show the final boundary locations of the detected blemishes.

From these results, it can be concluded that the snake algorithm is effective in refining the initial segmentation obtained from the flooding algorithm, and the combination of the two algorithms forms a system that can accurately detect blemishes on apples.

6. Discussion

Once the boundary of a blemished area has been located, the measurement for size of the area can be made. However, the detected blemish areas are accurate only in the two-dimensional projection plane. To obtain accurate three-dimensional area measurement, an adjustment considering the geometric surface shape of a fruit is

required. In this respect, any viable geometric technique such as the one developed by Wolfe and Bhatt (1989) for tomatoes will do.

Snakes are not autonomous. They need a pre-attention mechanism to place them somewhere near the desired contours. To this sense, the flooding algorithm in the first step of the blemish detection can be viewed as a pre-attention method. Incorporating this into the snake algorithm, we have autonomous snakes, which can be a generic image segmentation method for irregular contours. Therefore the snake algorithm can be regarded as a segmentation method on its own instead of a post-processing for refinement.

In the application of machine vision to fruit blemish inspection, an individual fruit under consideration need firstly to be separated out from the image background and from other fruits when they are touching and not oriented. The snake algorithm, as a generic method, can be used for this separation task as well. Simple thresholding can yield the initial position of a fruit contour. Then the snake algorithm can be applied to get the boundary of the fruit.

It is worth pointing out that the proposed system is applicable to detecting patch-like blemishes on other apple varieties and other types of fruits. For fruits with significant skin colour variations such as peaches, the image contrast needs to be enhanced before the initial segmentation. The enhancement can be achieved by working in the appropriate spectral regions (Yang et al., 1994).

7. Conclusion and future work

An image analysis scheme is proposed for accurate detection of fruit blemishes. The scheme is a combination of a region-based segmentation method, the flooding algorithm, and a contour-based segmentation method, the snake algorithm. Blemishes are first segmented out with the flooding algorithm. From this initial segmentation the snake algorithm refines the boundary localization so to improve the accuracy of the segmented areas. A simple closed loop snake is implemented, which mainly utilizes image gradient information. To allow snakes to move large distances, the smoothed images available from the pre-processing in the flooding algorithm are used, which is an alternative to the scale-space continuation. The scheme has been tested with apple images and compared with the segmentation obtained from the flooding algorithm alone. Initial results show that good improvement in boundary localization and so in the area size of a detected blemish is achieved, even when the boundary is poorly defined. The detection results accord well to our visual perceptive judgement.

The emphasis of the development has been on the methodology. The test results are presented as real images of system output. Future work will involve further quantitative testing of the system. This may involve the comparison of segmented areas with and without the refinement. Further tests can also be performed to compare between the blemish areas estimated by the system (with refinement) and the real areas on fruit surfaces, provided that a correction for surface projection is made.

In current work, a fixed number of 20 snake elements was used, which may not be ideal for very small blemish patches. A variable number of snake elements based on the perimeter of a blemish patch may be desirable.

Acknowledgement

The work is funded by the Office of Science and Technology and the EC (CAMAR: 8001-CT91-0206).

References

Davenel, A., Guizard, C.H., Labarre, T. and Sevila, F. (1988) Automatic detection of surface defects on fruit by using a vision system. J. Agric. Eng. Res., 41: 1–9.

Kass, M., Witkin, A. and Terzopoulos, D. (1988) Snakes: Active contour models. Int. J. Comput. Vision, 1: 321–331.

Marchant, J.A. and Schofield, C.P. (1993) Extending the snake image processing algorithm for outlining pigs in scenes. Comput. Electron. Agric., 8: 261–275.

Rehkugler, G.E. and Throop, J.A. (1989) Image processing algorithm for apple defect detection. Trans. ASAE, 32: 267–272.

Williams, D.J. and Shah, M. (1992) A fast algorithm for active contours and curvature estimation. CVGIP: Image understanding, 55 14–26.

Wolfe, R.R. and Bhatt, S. (1989) Compensation of projected image measurements for obtaining defect areas on tomatoes. ASAE Paper 89-6020.

Yang, Q. (1992) The potential for applying machine vision to defect detection in fruit and vegetable grading. Agric. Eng., 47(3): 74–79.

Yang, Q. (1994) An approach to apple surface feature detection by machine vision. Comput. Electron. Agric., 11: 249–264.

Yang, Q., Garcia-Pardo, E., Zwiggelaar, R., Bull, C.R. and Tillett, R.D. (1994) Machine vision for automatic fruit defect detection. In: Proceedings of the 6th International Symposium of the European Concerted Action Program, COST '94, on Current Status and Future Prospect, Oosterbeek, The Netherlands, 19–22 Oct., 1994.