Cell identification in Differential Interference Contrast microscope images using edge detection

D. Young and A.J. Gray

Department of Statistics and Modelling Science, University of Strathclyde,
26 Richmond Street, Glasgow G1 1XH, Scotland, UK
email: davidy@stams.strath.ac.uk
alison@stams.strath.ac.uk

Abstract

Some issues are considered which arise when edge detection is used to identify cells in differential interference contrast (DIC) microscope images. A method for outlining the cells in such images is developed and a simple practical example illustrated. Edge detection is complex due to the nature of DIC images, in which an apparent light effect is captured across the image. Further complications arise due to differing cell sizes and orientations. The method presented compensates for these problems.

1 Motivation

The work described here results from a need in research into high rate algal ponds, an environmentally important development in applied microbiology. These are simple, energy efficient, low-technology waste treatment systems (Fallowfield and Martin [1]). Achieving optimum efficiency of such systems relies on knowledge of the biomass of algae and bacteria in the mixed microbial populations of the pond. This is determined by viewing pond samples under a microscope, counting the number and measuring the size of cells and using standard formulae to estimate biomass from these measurements.

The current methods for identifying, counting and measuring cells are at best semi-automatic and slow. No fully automatic method (e.g. edge detection algorithms) has so far proved successful, due to the complex nature of the microscope images e.g. the presence of different cell types and shapes, blurring from out-of-focus cells, presence of detritus material and cells typically being clustered and/or overlapping. Developing a method using image analysis for accurately separating, identifying and counting individual cells in a sample, while ignoring noise, would be the first step towards an automatic method for estimating biomass directly from the microscope image. Methods using edge detection will be considered as a means of achieving this for DIC images (Cogswell and Sheppard [2], Holmes and Levy [3]).

BMVC 1996 doi:10.5244/C.10.66

2 Introduction

Direct application of edge detection algorithms to DIC images has so far proved unsatisfactory due to noise and typical transitions from light to dark pixels within cells (see Fig. 1). Output from simple procedures like the gradient or Laplacian filters requires post-processing with edge following algorithms to connect edge elements and form smooth contour outlines. In order to detect meaningful edges in noisy images, edge detection algorithms may perform better by incorporating global information about edges in the image and the structure of the objects they represent. This approach is considered here.

Figure 1: Yeast cell DIC image

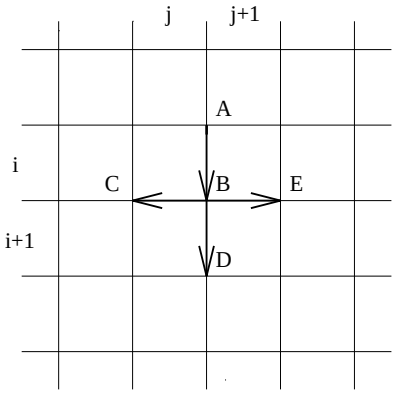

Figure 2: Boundary segments

3 Edge Detection Algorithm

The edge detection algorithm is based on the method of Martelli [4] for detecting edges and contours in noisy images. Two pixels, (i, j) and (h, k), within an image are defined as neighbours if

$$|i - h| + |j - k| = 1.$$

An *edge element* is defined as an ordered pair of pixels (P, Q) such that P and Q are neighbours. If P and Q correspond to the edge element $((i, j), (i, j + 1))$ then the associated *boundary segment* is AB (see Fig. 2). The direction of this segment is obtained with the convention of moving clockwise around the first square. A is called the *tail* and B the *head* of the segment AB. An edge element (P', Q') is said to be a *successor* of the edge element (P, Q) if the head of the boundary segment of (P, Q) coincides with the tail of the boundary segment of (P', Q'). Therefore, each edge element has exactly three successors, e.g. the successors of the edge element $((i, j), (i, j + 1))$ are the edge elements $((i, j), (i + 1, j))$, $((i + 1, j), (i + 1, j + 1))$, and $((i + 1, j + 1), (i, j + 1))$, corresponding respectively to the three boundary segments BC, BD and BE shown in Fig. 2. An *edge* is then defined as a sequence

of edge elements $x_1 = (P_1, Q_1), x_2 = (P_2, Q_2), .., x_n = (P_n, Q_n)$ such that (P_i, Q_i) is a successor of (P_{i-1}, Q_{i-1}), $i = 2, 3, .., n$.

In a grey level image, an edge is the boundary between two regions of different grey levels. A good edge will be such that the sum of the difference in grey levels between the first pixel (P_i) and the second pixel (Q_i) of its edge element x_i is large. To choose edge elements in an image, a cost is assigned to every possible edge. The cost associated with the edge element $x = (P, Q)$ is

$$c(x) = Max - (int(P) - int(Q)),$$

where $int(P)$ and $int(Q)$ are the intensities of the pixels P and Q, and Max is the maximum difference of grey levels between any two adjacent pixels in the image. This cost is low if $int(P)$ is high and $int(Q)$ is low, and is always non-negative.

The cells in Fig. 1 are approximately elliptical and in order to find the cell outlines, elliptical objects like the simplified artificial one shown in Fig. 3 would have to be identifiable.

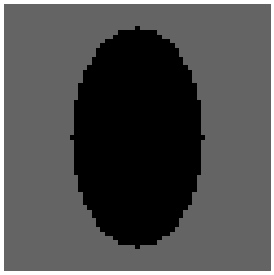

Figure 3: Elliptical object

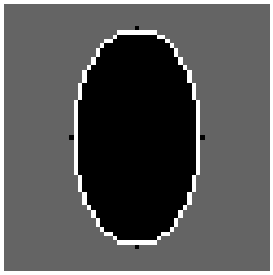

Figure 4: Outline of ellipse

The starting point is obtained by calculating the difference in intensity between adjacent pixels along the rows in the image. For each row, moving from right to left across the image, the sum of grey levels of two adjacent pixels is subtracted from the sum of the next two. This gives a positive value at transitions from light to dark areas across the image and negative for dark to light. Choosing the maximum such (positive) difference will identify a starting point on a cell boundary and avoid choosing a point inside the cell at the transition from light to dark as an initial edge pixel. The algorithm then generates further edge pixels in a clockwise direction (by minimising the cost function for each pair of edge pixels) until the starting point is recovered and the outline traced.

The first edge element identified in Fig. 3 was at the right hand side of the cell and the initial direction of search taken as downwards, corresponding to the segment AB in Fig. 2. Fig. 4 shows the result of tracking the cell outline as described above. This simple application of Martelli's algorithm is adapted in stages to suit real DIC images, which are much more complex.

3.1 Development of Algorithm

The edges of objects within a DIC image are, at best, slightly blurred. Blurring Fig. 3 using a 3×3 mean filter gives Fig. 5.

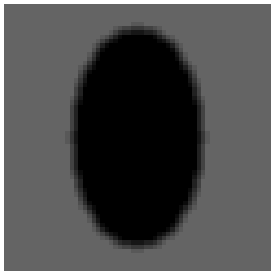

Figure 5: Blurred elliptical object

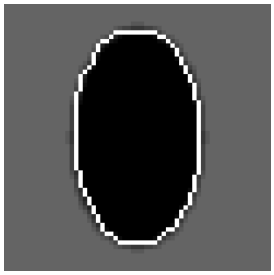

Figure 6: Outline of blurred ellipse

3.1.1 Generation of Edge Contour

To allow for blurring, the cost function is modified to incorporate differences in intensity between two pairs of edge pixels rather than single pixels. The edge element now used consists of four adjacent pixels (P', P'', Q', Q'') which all lie in the same row or column and is such that P' and P'' belong to the region of highest grey level (see Martelli [4]). At each step two further edge elements are chosen using the new cost function,

$$c(x) = Max - (int(P') + int(P'') - int(Q') - int(Q'')),$$

where Max is now the maximum difference associated with the starting point.

For a smooth contour, sequences of three edge elements whose first two or last two pixels coincide are not allowed. Certain sequences of two edge elements (see Fig. 7) which follow on from the starting point with an initial downward direction AB are considered.

At each stage, 11 different routes are considered in order to add on a further 2 edge elements. Of these, the route with minimum cost is chosen. To give comparable costs for each route, the cost function has to be averaged along the route since adding on 2 edge elements involves calculating the difference in intensity across 3, 4 or 5 boundary segments, i.e. the cost is divided by the number of boundary segments. For example, the cost associated with route 9 in Fig. 7 is

$$\begin{aligned} cost = 0.25 \times \{&Max - (int(i, j+2) + int(i, j+1) - int(i, j) - int(i, j-1)) \\ &+ Max - (int(i-1, j+1) + int(i, j+1) - int(i+1, j+1) - int(i+2, j+1)) \\ &+ Max - (int(i+1, j+3) + int(i+1, j+2) - int(i+1, j+1) - int(i+1, j)) \\ &+ Max - (int(i+2, j+3) + int(i+2, j+2) - int(i+2, j+1) - int(i+2, j))\} \end{aligned}$$

where $int(i, j)$ is the grey level intensity at location (i, j) (indicated by the darker square in Fig. 7, where row labels i increase from top to bottom and column labels j increase from left to right).

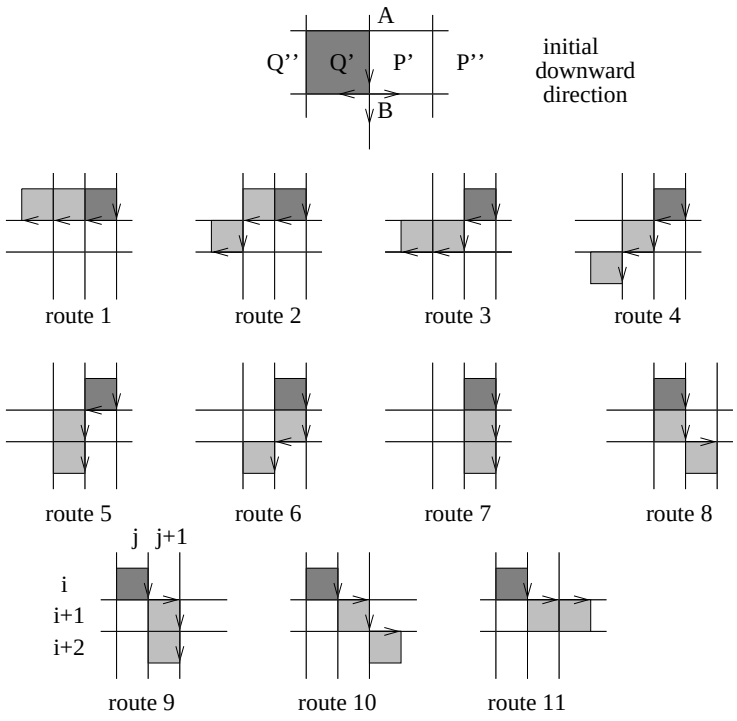

Figure 7: Possible sequences of edge elements

3.2 Results

Fig. 6 shows the apparently promising result of applying the method of Section 3.1.1 to Fig. 5. The main body of the object is clearly outlined and the blurred part round the edges is left as background.

3.3 DIC Images

A test image was then constructed to represent a DIC image of a yeast cell, by assuming cells are ellipsoidal in shape. Grey levels were assigned to pixels within an ellipse according to their distance from the edge, the centre pixel receiving intensity of 1, points at the edge and beyond, intensity of 0. A 63×63 cell with major axis length 25 and minor axis length 15 was constructed with grey levels

$$f(i,j) = \sqrt{1 - \left\{ \left(\frac{i}{\sigma_x}\right)^2 + \left(\frac{j}{\sigma_y}\right)^2 \right\}}$$

where $\sigma_x = 15$, $\sigma_y = 25$, and if $f(i,j) < 0$ then $f(i,j)$ is set to 0. The grey levels were then scaled to the full range $(0, 255)$. To mimic the apparent light effect in the DIC image, first-order differencing was applied to this image at the angle at which light appears to hit the specimen, namely from the top left, so differencing was done at 45^0 in this direction, taking $int(i,j) - int(i-1, j-1)$ as the differenced

image value at location (i, j) (a more sophisticated modelling approach is currently being considered). The resulting image is shown in Fig. 8 which also shows the initial part of the contour, generated as described above.

The difficulty in applying the technique to DIC images becomes evident here. The contour follows a path through the cell at the transition from light to dark. At this part of the cell, there is a greater difference in intensity within the cell than there is between the edge of the cell and the background, the grey level of pixels within the cell being close to that of the background.

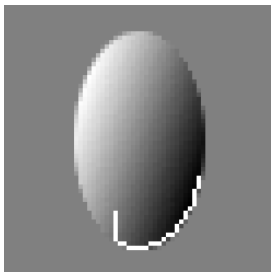

Figure 8: Yeast cell test image

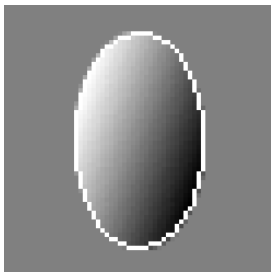

Figure 9: Result of modified algorithm

3.3.1 Gain Function

The differences in intensity between pixels, from right to left across the image, go from positive (at the transition from the background into the dark part of the cell) to negative (within the cell at the transition from dark to light) then positive again (from the light part of the cell into the background). This information is now used in the cost function to prevent the contour tracing a path within the cell. All costs are composed by summing the intensity difference between two pairs of adjacent pixels either along a row (summing the grey levels of two adjacent pixels and subtracting from this the sum of the two pixels to the left) or down a column (taking the absolute difference in intensity). This should always give a positive value at the edge of the cell and a negative value within the cell. Of the possible routes, the one with the maximum 'cost' or gain is taken as the best path around the cell.

For example, the gain function associated with route 9 in Fig. 7 is now

$$\begin{aligned} gain = 0.25 \times \{&int(i, j+2) + int(i, j+1) - int(i, j) - int(i, j-1)) \\ &+ABS(int(i-1, j+1) + int(i, j+1) - int(i+1, j+1) - int(i+2, j+1)) \\ &+(int(i+1, j+3) + int(i+1, j+2) - int(i+1, j+1) - int(i+1, j)) \\ &+(int(i+2, j+3) + int(i+2, j+2) - int(i+2, j+1) - int(i+2, j))\} \end{aligned}$$

where ABS is the absolute value. The result of applying this to the test image of Fig. 8 is shown in Fig. 9. The cell outline is now accurately traced.

4 Touching/Overlapping Cells

Fig. 10 is used to investigate splitting touching and overlapping cells, using two slightly overlapping yeast cells (each constructed as before on a uniform background of grey level 128).

The starting point is located between the two cells and the contour then generated identifies one cell in the image, as shown. This area of the image is then ignored, and the maximum intensity difference is again used to locate other possible cells. A point on the right hand side of the other cell is located and its contour generated as before.

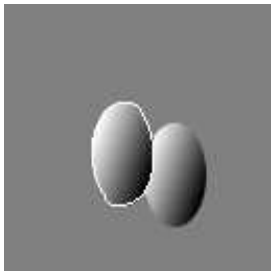

Figure 10: Touching yeast cells with first cell identified

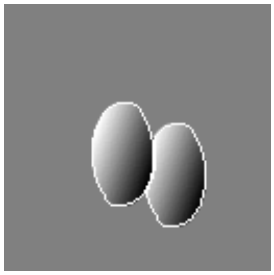

Figure 11: Outlines superimposed

The rest of the image is then searched for other possible cells. If there is a maximum difference in intensity not less than 30% of the original maximum difference, then another starting point is identified and the contour traced. (The 30% cut-off value was chosen to detect differences in intensity large enough to indicate a cell rather than a point in the background). For this image there is no such difference and the algorithm stops. The final result, which successfully identifies the two cells, is shown in Fig. 11 . The cell sizes (not shown) are then recorded (i.e. the number of pixels enclosed by the cell contour). The size of the second cell is under-estimated as it is slightly covered by the first cell.

4.1 Application to real images

Fig. 12 shows a 50×50 pixel sub-image of one of the cells from the bottom of the cell cluster in Fig. 1. The edges of three of the surrounding cells to the top, left and right hand side are visible. Fig. 13 shows part of the contour traced by the algorithm, which initially is very accurate but then begins to follow the outline of the cell above.

4.2 Curvature

The yeast cells are roughly elliptical and the algal cells shown in Fig. 15 are roughly circular. Information about the object shape is now used to ensure that the contour is not allowed to follow an object into the background or trace round another cell. This is done by estimating the curvature of the contour.

Figure 12: Extracted yeast cell

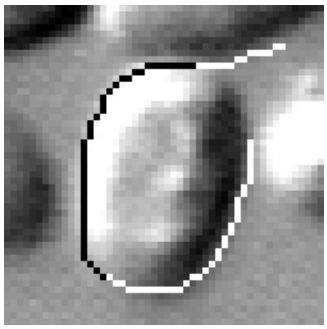

Figure 13: Contour tracing round touching cell

The smoothness and shape of the contour generated can be measured and constrained by assigning a cost to its curvature. This was done by considering 9 consecutive edge elements. The first edge element is connected to the fifth element by a straight line and the fifth element to the ninth. The angle between the two lines is then measured, and considered as proportional to the curvature of the contour.

4.2.1 Measuring Curvature

To measure the angle of curvature of a contour being traced round an object, it is necessary to calculate the angle internal to the object. This was done by considering the coordinate points of the edge elements and calculating the angle in the appropriate quadrant - see Fig. 14. This illustrates four of the possible contours within a circular curve. The direction of the contour is indicated by the numbering of edge elements 1, 5 and 9 as 1, 2 and 3, which are then joined by straight lines. Consider contour (a). Using the coordinate points (x_1, y_1), (x_2, y_2) and (x_3, y_3) say, of 1, 2 and 3, the values of α and β relative to the $x-y$ coordinate system are calculated as the obtuse angles

$$\alpha = tan^{-1}\frac{(y_2 - y_1)}{(x_2 - x_1)} \qquad \text{and} \qquad \beta = tan^{-1}\frac{(y_3 - y_2)}{(x_3 - x_2)}.$$

A value γ is then calculated as $\gamma = \alpha - \beta$, and used as a measure of the curvature. For the illustration shown in Fig. 14, the contour is approximately circular. To avoid the edge curving outwards or round another cell, the range of γ is chosen as $0^o \leq \gamma \leq 44^o$.

This would allow contours (a) and (b) to be accepted as part of the edge. Contours (c) and (d) would be rejected since they both produce values of γ which are negative. In this application there are 16 possible contour configurations for which the internal angle γ may need to be calculated.

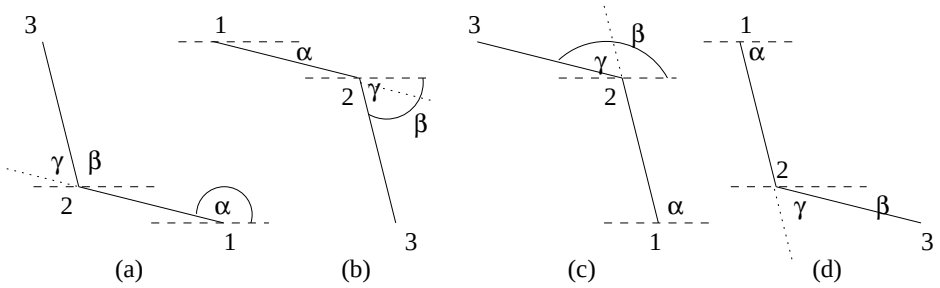

Figure 14: Measuring internal angles

4.2.2 Curvature Measuring Algorithm

At each stage, when two new edge elements are added to the contour, the angle of curvature is calculated. The edge elements are accepted as part of the outline of an algal or yeast cell if the angle of curvature, γ, lies between 0^o and 44^o inclusive. Otherwise the edge elements do not give the correct curvature for the required type of contour and are rejected. From the stored cost functions, the algorithm then chooses the sequence of two edge elements with the next lowest cost in terms of difference in intensity. If these give angles within the permitted range they are added to the contour, otherwise the next possibility is considered until a satifactory angle is obtained.

4.2.3 Results and Conclusion

This modified algorithm was applied to the algal cell DIC image shown in Fig. 15. This simple image illustrates how single cells are easily identified, and touching cells separated, using the curvature constraint. Although some outlines are rather irregular, the average squared error between the true and estimated cells sizes is 8301, a considerable improvement over a previous template matching approach (Young *et al.* [5]) which gave an error of 34028 and was very computationally intensive. Actual cell sizes were taken as the number of pixels enclosed by the cell outlines on an image scanned after tracing the cell outlines manually.

The major limitation of the method is in its being tailored to detect a certain shape of object, namely circular or elliptical. This arises from the introduction of curvature constraints to separate touching objects. For single cells, or for images containing distinct cells, the method is both quick and accurate at detecting the true outlines. For more complex images where cells are clustered or overlapping (e.g. Fig. 1) the results are poor and more constraints would be necessary to make the method successful.

Acknowledgements

We are grateful to Dr. Chris Glasbey of BioSS, Edinburgh, for his assistance in this research. The images were supplied by Dr. Nick Martin of the Scottish Agricultural College, Ayr. The first author was supported by an Earmarked Studentship from the Engineering and Physical Sciences Research Council.

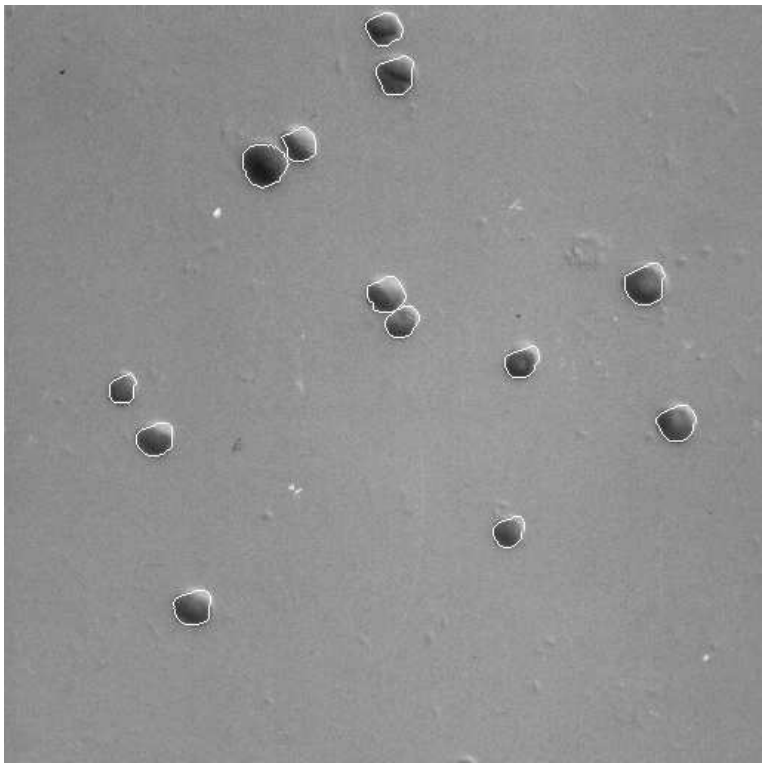

Figure 15: Algal cell outlines identified by the final algorithm

References

[1] H.J. Fallowfield and N.J. Martin. *The operation, performance and computer design of high rate algal ponds. Institute of Chemical Engineering Symposium Series*, vol. 111, 1990.

[2] C.J. Cogswell and C.J.R. Sheppard. *Confocal differential interference contrast (DIC) microscopy: including a theoretical analysis of conventional and confocal DIC imaging.* Journal of Microscopy, vol. 165, pp81-101, 1990.

[3] T.J. Holmes and W.J. Levy. *Signal-processing characteristics of differential interference contrast microscopy.* Applied Optics, vol. 26, pp3929-3939, 1987.

[4] A. Martelli. *An application of heuristic search methods to edge and contour detection.* Communications of the Association for Computing Machinery, vol. 19, pp73-83, 1976.

[5] D. Young, C.A. Glasbey, A.J. Gray and N.J. Martin. *Cell identification in differential interference contrast microscope images using template matching.* Proceedings of the 9*th* Scandinavian Conference on Image Analysis, Uppsala, Sweden, vol. 1, pp199-208, 1995.