Delineating microscopic objects and computing their principal parameters

Vossepoel, A.M.; Smeulders, A.W.M.

Publication date
1980

Published in
Medinfo 80: proceedings of the Third world conference on medical informatics, Tokyo, September 29-October 4, 1980

Citation for published version (APA):
Vossepoel, A. M., & Smeulders, A. W. M. (1980). Delineating microscopic objects and computing their principal parameters. In D. A. B. Lindberg, & S. Kaihara (Eds.), *Medinfo 80: proceedings of the Third world conference on medical informatics, Tokyo, September 29-October 4, 1980* (pp. 121-125). North-Holland.

A078327059
NCC/IBL AANVRAAGBON
KOPIE MONOGRAFIE EGB
(14)
14-04-2005

Datum indienen : 13-04-2005 20:29 5493-1 Clearing House
Datum plaatsen : 13-04-2005 20:29
Aanvrager : 0004/9998
Aanvraagident :
Aanvragerident : 0004/9999
Eindgebruiker : 041631433

Telefoonnummer : 050-3635057
Cooperatiecode : R

Leverwijze : Elektronisch
Fax :
Ftp :
E-Mail : m.s.van.delden@rug.nl
Ariel :

Plaatscode : 802289193 ; MB 347 e007 ; ;
802289193 ; MB 347 e008 ; ;

(1) [ ] Origineel gestuurd
(2) [ ] Copie gestuurd
(3) [ ] Overige
(4) [ ] Nog niet aanwezig
(5) [ ] Niet aanwezig
(6) [ ] Niet beschikbaar
(7) [ ] Uitgeleend
(8) [ ] Wordt niet uitgeleend
(9) [ ] Bibliografisch onjuist
(0) [ ] Bij de binder

A078327059
Verzamelnota volgt.
NCC/IBL AANVRAAGBON
KOPIE MONOGRAFIE EGB
14-04-2005

Datum indienen : 13-04-2005 20:29 9560-1 UvA Keur
Datum plaatsen : 13-04-2005 20:29
Aanvrager : 0004/9998
Aanvraagident :
Aanvragerident : 0004/9999
Eindgebruiker : 041631433

UB Groningen
Broerstraat 4
9700 AN Groningen

tav

Aantal

PPN Titel : 802289193
Titel : Medinfo 80 : proceedings of the Third world conference on
: medical informatics, Tokyo, September 29-October 4, 1980
Auteur : Lindberg, Donald A.B. (ed.)
Deel/Supplem. : Pt. I: ISBN 0-444-86030-4. - Pt. II: ISBN 0-444-86031-2
Corporatie : Externe database:
Jaar/Editie : 1980 Extern nummer :
Uitgave : Amsterdam [etc.] North-Holland
Serie/Sectie : IFIP world conference series on medical informatics. IFIP world
Pag-ISSN/ISBN : series on medical 0-444-86029-0 v.3

Plaatscode : 802289193 ; MB 347 e007 ; ;
802289193 ; MB 347 e008 ; ;

Jaar :
Volume :
Aflevering :
Eindgebruiker : 041631433 Aanvraagident. :
Auteur : Aanvragerident. : UVA KEUR (UB GRONINGEN)
Artikel :
Bladzijden : 121-125
Bron :
Opmerking : arno ID: 134002

MEDINFO 80, Lindberg/Kaihara, Editors
© IFIP, North-Holland Publishing Company (1980)

DELINEATING MICROSCOPIC OBJECTS AND COMPUTING THEIR PRINCIPAL PARAMETERS

Albert M. VOSSEPOEL and Arnold W. M. SMEULDERS

Dept. of Biomedical Information Processing and Laboratory for Pathology,
University Medical Center, Wassenaarseweg 62, 2333 AL LEIDEN
The Netherlands

An algorithm is presented for light-microscopic measurement of quantitatively stained cells. Delineation of the object in the scan array and discrimination of close lying other objects is done by optimization of an isodensity traced contour. Metrication errors of finite length line segments are estimated for the perimeter. Introducing the fall-set of the object permits accurate computation of the object's content by definition of the background as a rim around it. A program version is operational for several years on various minicomputers, rendering results quickly and rarely requiring operator intervention.

INTRODUCTION

Quantitative measurements of the amount of staining material in microscopic objects are becoming increasingly important, for scientific as well as for diagnostic purposes. A variety of staining, imaging and scanning techniques is presently available to visualize cells or cellular components such as nuclei, proteins or chromosomes. Those techniques should preferably be used that yield a high-resolution image of the object concerned, in order to avoid severe errors in the measurement caused by the optical system [1].

Provided such a digitized image of sufficient resolution and precision, we may compute parameters like, e.g., the perimeter (P), area (A) and total amount of staining material or "content" (T). Especially the last parameter may be diagnostically important. But before T can be evaluated one needs to know which set of pixels corresponds with the object, and which set contains pixels with values representative of a local background. The first set, the object region, provides the object perimeter and area. If the pixel values are proportional to the local amount of staining material S, an average background value B may be derived from the second set, the background region, which usually doesn't adjoin the object region. The content T may then be found by subtracting B from each original object pixel value S, before taking the sum of the latter over the object region. It is essential that the value B is evaluated with high accuracy, because any error in B is reflected n-fold in T, if n is the number of object pixels.

The most simple approach for segmentation of the digital image into regions is to assign the object and background regions to predefined parts of the image. The background may, e.g., be assigned to two (rectangular) regions of predefined width at the beginning and at the end of each scan line. The remaining intermediate pixels are then assigned to the object region. Obviously, this approach is restricted to images that contain nothing but one complete object, surrounded by a clean and flat background. Any alien object within the frame of the scanned image will either be added to the object concerned, or lead to an excessive value of B, in both cases biasing the resulting value of T. Instead of assessing the background value B by simply averaging over the background region, it can also be derived from the data histogram. To this end, one could (manually) select a value at or near the highest histogram peak. Alternatively, one could average the values below a threshold computed from the location and width of the highest histogram peak.

The approaches described always require visual inspection of the image, to make sure that no alien object is present, neither in the object region, nor in the background region. If this condition is not met, regions may be delineated in the image manually by means of a cursor or light-pen on a visual display unit. The regions delineated in this way can be assigned to or eliminated from the object and background regions, respectively. With the addition of this manual procedure, almost any microscopic image can be managed, as long as there is no overlap between the object concerned and any other object. Therefore, these approaches have found general acceptance, e.g. [2,3], but they require much time and attention. The only effective way to reduce this operator effort is to have the object and background regions delineated automatically. Besides, the procedure will then become more objective.

For the object delineation, several methods are known [4]. The most obvious one is thresholding the image with a distinct grey-level, usually derived from the data histogram. This method is simple, but uses only global information on a local level. A second group of methods does the thresholding in the gradient transformed image. The resulting boundaries are better proof against global variations in background value, but a high resolution is required to reduce the influence of high-frequency noise. Other apparently attractive methods cluster the data according to their local similarity, but the setting of the clustering algorithm turns out to be rather problem-dependent. Most image segmentation methods may be improved by introducing relaxation, i.e. optimizing the segmentation through iterative optimization of a distinct parameter considered as a quality measure.

A natural choice for a local background region around the object, or at least close to it, would be a rim of a few pixels wide, at a relatively short distance away from the object boundary [5]. However, the presence of any part of an object in such a region biases the value B derived from it. So the pixels concerned require scrutinizing to assess whether they should be assigned to the background, or not.

The algorithm presented in this paper has been developed to meet the following requirements, assuming an input pixel array containing at least one object of interest, possibly among others, with an area A greater than a distinct value:

- it should delineate the object nearest to a selected pixel;
- it should separate two objects as long as there exists a "valley" of at least one pixel wide between the objects;
- it should compute accurate values for the perimeter (P), area (A) and total content (T) of the object;
- it should have both high sensitivity and high specificity for scene segmentation errors;
- it should not require display or operator intervention, unless an error condition is indicated;
- it should run as a program on a small minicomputer, rendering results in a few seconds.

OBJECT DELINEATION

We have adopted the simple thresholding method for the object delineation. But the thresholding is repeated for several delineation levels in order to minimize the tortuosity of the resulting contour. This property can be measured by the contour ratio Q, a dimensionless quantity that is defined as the perimeter squared divided by 4π-times the object area:

$$Q = P^2/4\pi A$$

The objects considered visually show smooth-edged contours, so minimizing the contour ratio may be assumed equivalent to optimizing the contour quality.

The actual procedure starts by searching in the four principal directions for a starting point, from a selected pixel, by default in the middle of the array. A suitable starting point for a common contour-following algorithm consists of a pixel pair with values less than and not less than the delineation level. Both elements of the pair should not be isolated and therefore should have at least one adjacent element meeting the same condition. By imposing this condition, object or background regions consisting of only one pixel are prevented.

Contour following is accomplished by testing the neighbours of a contour pixel in a counterclockwise direction until a pixel is found with a value also less than the contour level. Then the neighbours of the pixel found are tested in turn, starting at the first neighbour that has not yet been tested, and so on, until the starting point is reached again. The method can be varied according to the number of neighbours, 4 or 8, assigned to each pixel, preference being mostly a matter of taste. If the contour bends back on itself, the resulting protrusion, or penninsula, should be cut off, because our goal is to minimize the tortuosity of the resulting contour. Besides, there is a better chance of separating two close lying objects, by doing so.

It is essential that the resulting contour encloses an object, and not an open space within an object or between clustering objects, as both may happen, given the algorithm used. This condition can be checked by examining the contour's direction of rotation, which is found by adding the angles between the individual vector elements that connect the contour pixels. If the contour appears not to enclose an object, the search for a starting point should be continued, preferably in the same direction as before, towards the pixel array boundary. It is understood that neither the starting point, nor any contour pixel may lie on the pixel array boundary. If a contour comes out at a distinct delineation level, we need to compute its length and the area it encloses for the calculation of the contour ratio, which is to be minimized through various delineation levels.

Given the contour as a chain of vector elements, the most obvious estimate of its length would be to assign the inter-pixel distance G to each orthogonal vector element, and $G\sqrt{2}$ to each diagonal one. On second thoughts, the estimate of the contour length turns out to be not so trivial, because each discrete vector element represents a variety of possible segments of the original continuous contour, even if the latter is assumed piecewise linear. Different assumptions about the length of the straight segments of the original contour lead to different values for the length of the digitized contour. If 8 neighbours are assigned to each pixel, there are two more sophisticated methods to estimate the length of straight segments of an original contour corresponding with distinct numbers of orthogonal (n) and diagonal (m) vector elements:

- without corner correction:
  assign an average length B to each orthogonal vector element, and an average length D to each diagonal one;
- with corner correction:
  keep $D = B\sqrt{2}$, and correct the resulting contour length by the amount C for each corner between consecutive vector elements.

The methods can be formulated as follows:

$$P = nB + mD - kC +/- (n+m)E$$

in which k represents the corner count, and E the average error in the length estimate per vector element. The coefficients B and C or D are evaluated through a least square error approach, also rendering the value of E. The first method has been developed assuming the original contour to be straight between two pixels only (see Table, entry "1", left column) [6]; the second method has been developed assuming the original contour to be straight over infinite length (see Table, entry "inf", right column). Otherwise, in reference [7], 4 neighbours are assigned to each pixel. We have generalized both methods assuming the original contour to be straight over the length of a variable integer number (n+m) of vector elements [8]. Some results are given in the following table, assuming the physical inter-pixel distance G = 1:

	without	corner correction		with	corner correction	
n+m	B	D	E	B	C	E
1	1.059	1.183	0.095	0.955	0.017	0.163
2	1.004	1.262	0.071	0.972	0.056	0.082
3	0.977	1.299	0.056	0.978	0.070	0.055
4	0.965	1.317	0.046	0.981	0.077	0.042
5	0.958	1.327	0.041	0.983	0.080	0.034
6	0.954	1.332	0.037	0.984	0.083	0.029
7	0.951	1.336	0.035	0.984	0.084	0.025
8	0.950	1.338	0.033	0.985	0.085	0.022
9	0.949	1.340	0.032	0.985	0.086	0.020
10	0.948	1.341	0.031	0.985	0.086	0.019
20	0.945	1.345	0.027	0.986	0.088	0.012
30	0.945	1.345	0.027	0.986	0.088	0.010
inf	0.944	1.346	0.026	0.986	0.089	0.009

From this table it is apparent that the first approach only renders more accurate results if the original contour is assumed to be straight over the length of less than 3 vector elements, otherwise the second approach, with corner correction C, gives better results. The table also shows that for contours assumed to be straight over the length of 30 or more vector elements, one single set of coefficients B and C, or B and D, may be used. If linearity of the original contour is assumed over a length of only a few vector elements, it evidently requires special attention.

The area enclosed by the contour is computed by a method that bears analogy with integrating around a closed curve: following the contour clockwise, pixels below and including the contour pixel are added to the area A while going from left to right; pixels below and excluding the contour pixel are subtracted from A while going from right to left. The actual implementation of this principle requires detailed analysis of all situations that can result from different combinations of incoming and outward vector elements of each contour pixel [9].

Now that the perimeter P and the area A of the object are known at a distinct delineation level, the contour ratio Q can be computed. In order to find a minimum value for Q, the process is repeated for a number of delineation levels. This number, and the corresponding level values, can be chosen within wide margins, as long as 2 or 3 values are situated between the background peak and the object peak (or shoulder) of the data histogram. The smaller the number of level values in this range, the greater the risk of not finding any contour that doesn't touch the pixel array boundary and also encloses more than a predefined number of pixels.

BACKGROUND EVALUATION

Firstly, we have to find the set of pixels whose values are representative of a local background around the delineated object, or at least close to it. To this end, every neighbour of a contour pixel, that has not yet been assigned to the object, is examined to see if it has a value less than that of the contour pixel. If the neighbour pixel indeed has a lower value, it will also be assigned to the object. After examination of the neighbours of all contour pixels, the process is repeated with the freshly assigned object pixels playing the role of the contour pixels. Iterating this process until no pixels can be assigned to the object any more, results in what is called the fall-set of the object [10]. Every pixel on the border of the fall-set represents a local minimum, so it seems attractive to use these border pixels as the background region. However, there are two reasons for not doing so:

- the propagation of the fall-set may have been halted by accident, e.g. by an object lying close to the delineated object, resulting in relatively high local pixel values;
- if the background is flat, with only noise superimposed, the border pixels being local minima biases their average value with respect to the average value of a slightly wider rim of pixels.

Because of the second reason, a better choice for the background region will be a rim of pixels around and adjoining the border of the fall-set, but then the first reason for not using it will count even more heavily. On the small scale of the direct neighbourhood of the rim pixels, the problem can be solved by median smoothing. This technique replaces the value of the central pixel of a 3 x 3 region by the median of the ordered 9 values (in this case the fifth), thereby avoiding local extrema. On a larger scale, the problem can be solved - for want of a better solution - by eliminating from the background all pixels with a value greater than a distinct level. This level must be chosen somewhere between the background peak and the object peak (or shoulder) of the data histogram.

The geometrical features of the fall-set depend much less on the delineation level than those of the delineating contour. If we imagine, e.g., the object as consisting of grey-values monotonically decreasing from an apex into a flat background, the area enclosed by various contours strongly depends on their respective levels, whereas the base area and outline does not. For this reason, we consider the entire fall-set as the object in the computation of the total content T, rather than the pixels enclosed by the delineating contour. The advantage of doing so is that the resulting value of T is highly independent of the preceding delineation procedure, allowing the use of a separate system for that purpose.

RESULTS

The algorithm described above has been implemented on various computers as a set of Fortran-IV subroutines [9]. Together, the subroutines require approximately 3.5k 16-bits words of computer memory, excluding the size of the (integer) pixel array. It has been tested on a set of 1343 typical cell images from cervical and mammarian specimens. The program required approximately 4 seconds for each image, with size between 2k and 8k pixels, using a DEC PDP-11/60 computer. Only 2 warnings resulted, both concerning the low number of available background pixels. Testing on a much smaller set of 43 chromosome images initially resulted in 3 errors. The errors apparently were caused by fragmentation of the chromosomes into objects - actually bands - with a less tortuous contour than the chromosomes themselves. The errors disappeared after increasing the lower limit of the object area in the delineation process. They also disappeared if the upper limit of the range of trial delineation levels was decreased.

The results show that application of the algorithm to elongated objects requires some extra attention, because the relaxation algorithm basically favours round objects over elongated ones. In conclusion, the algorithm meets the requirements specified in the Introduction.

REFERENCES:

[1] W.A.L. Duijndam, A.W.M. Smeulders, P. van Duijn, A.C. Verweij: Optical errors in scanning stage absorbance cytophotometry, I. Procedures for correcting apparent integrated absorbance values for distribution, glare and diffraction errors. J. Histochem. Cytochem., submitted.

[2] M.L. Mendelsohn, B.H. Mayall: Chromosome identification by image analysis and quantitative cytochemistry. In J.J. Yunis (ed.): Human Chromosome Methodology, 2nd ed. (Academic Press, New York, 1974) pp.311-346.

[3] E. Bengtsson, J. Holmquist, B. Olsen, B. Stenkvist: SCANCANS - An interactive scanning cell analysis system. Comput. Prog. Biomed. 6 (1976) 39-49.

[4] L.S. Davis: A survey of edge detection techniques. Comput. Graph. Image Proc. 4 (1975) 248-270.

[5] F.C.A. Groen, M. van der Ploeg: DNA cytophotometry of human chromosomes. J. Histochem. Cytochem. 27 (1979) 436-440.

[6] F.C.A. Groen, P.W. Verbeek: Freeman-code probabilities of object boundary quantized contours. Comput. Graph. Image Proc. 7 (1978) 391-402.

[7] D. Profitt, D. Rosen: Metrication errors and coding efficiency of chain-encoding schemes for the representation of lines and edges. Comput. Graph. Image Proc. 10 (1979) 318-332.

[8] A.M. Vossepoel, A.W.M. Smeulders: Metrication errors of discrete vector elements representing straight lines of finite length. Comput. Graph. Image Proc., submitted.

[9] A.M. Vossepoel, A.W.M. Smeulders, K. van den Broek: DIODA: Delineation and feature extraction of microscopical objects. Comput. Prog. Biomed. (1979) in the press.

[10] D. Rutovitz: Expanding picture components to natural density boundaries by propagation methods. The notions of fall-set and fall-distance. Proc. 4th Int. Joint Conf. Pattern Recognition, Kyoto, Japan, Nov. 7-10, 1978.