1989

Algorithm To Estimate Cell Biovolume Using Image Analyzed Microscopy

ME Sieracki
Virginia Institute of Marine Science

CL Viles
Virginia Institute of Marine Science

KL Webb
Virginia Institute of Marine Science

Cytometry 10:551–557 (1989)

HETEROGENEITY OF SIZE/ABUNDANCE

Algorithm to Estimate Cell Biovolume Using Image Analyzed Microscopy[1]

Michael E. Sieracki, Charles L. Viles, and Kenneth L. Webb
School of Marine Science, College of William and Mary, Virginia Institute of Marine Science, Gloucester Point, Virginia 23062

Received for publication November 16, 1988; accepted April 17, 1989

This paper describes an algorithm for calculating the biovolume of cells with simple shapes, such as bacteria, flagellates, and simple ciliates, from a 2-dimensional digital image. The method can be adapted to any image analysis system which allows access to the binary cell image—(i.e., the pixels, or (x,y) points, composing the cell. The cell image is rotated to a standard orientation (horizontal), and a solid of revolution is calculated by digital integration. Verification and a critical assessment of the method are presented. The algorithm accounts for irregularities in cell shape that conventional methods based on length, width, and geometrical formulas do not.

Key terms: Bacteria, protist, biomass, digital image analysis

Accurate measurements of the biomass of bacteria and protists from environmental samples depend primarily upon visual microscopic methods for enumeration and cell sizing. Epifluorescence microscopy is the method of choice in aquatic sciences, since algal pigments fluoresce specific colors and the use of fluorochromes allows the discrimination of living cells from detrital particles (9,11,13,17). Since these methods are tedious and subjective when conducted visually, there is much interest in automation by computerized image analysis (4,19,20). This technique allows numerous, more precise, and more detailed cell measurements to be made. For the purpose of ecological energy and nutrient flow modelling, cell biovolume must be converted into biomass units, usually carbon or nitrogen. Accurate measures of both cell biovolume and carbon or nitrogen cell content are necessary for good conversion factors. Since these volume estimates are based on cubed linear measurements, their errors can equal or exceed those associated with the carbon or nitrogen determinations (4). The variation in estimated cell volume due to linear measurement error can easily exceed the variation of different biovolume-to-carbon conversion factors, which for bacteria are currently controversial (6,7,16) and, until recently (5), were unmeasured for protozoa. Previous methods of estimating the biovolume of microscopic organisms have generally involved measuring overall cell size in two dimensions and applying a geometric formula to infer a three-dimensional structure (1,3,10,15). A typical method would be to measure length and width and use the formulas for a sphere and cylinder to calculate volume.

We have developed a method using an image analyzed epifluorescence microscopy system. The method is similar to one used by Brownlee (8) to measure ciliate volumes, but is more flexible and automated. We feel that the implicit assumptions of this method are met and that it will be of benefit wherever there is difficulty obtaining direct measurements of the object in question.

MATERIALS AND METHODS

Algorithm Description

A prerequisite to implementing this algorithm is the ability to obtain a digital image of the object in question. The main assumption of the integration algorithm is that the shape of the object to be measured is symmetrical in the 3rd dimension we cannot see—that

[1]This is VIMS contribution No. 1552.

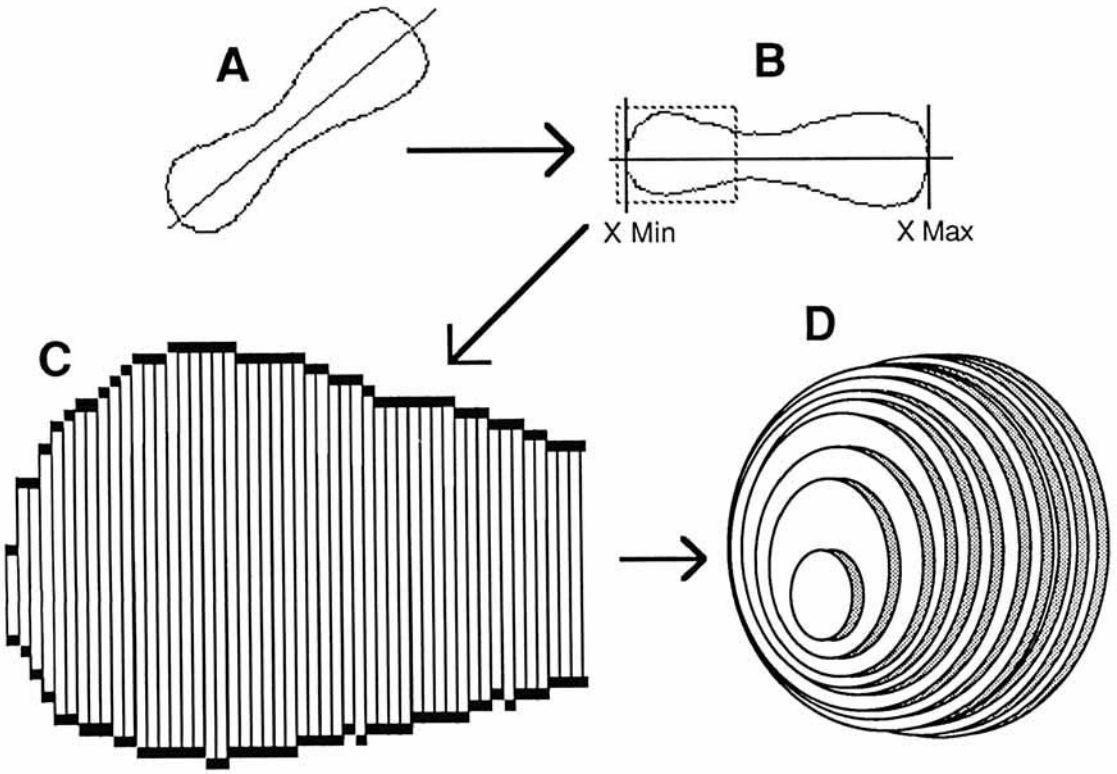

FIG. 1. The calculation of cell biovolume from a digitized cell image is made from **A)** the digitized cell perimeter and cell orientation by **B)** rotating the perimeter to horizontal, **C)** measuring the length of each 1-pixel-wide cross section, **D)** calculating the volume of each resulting thin cylinder, and then summing these volumes for the whole cell (reprinted from ref. 20).

is, the plane formed by the viewing axis and the minor axis of the object image. Since images are 2-dimensional projections of 3-dimensional objects, the shape in the viewing axis is not known. For prokaryotic and simple eukaryotic cells, the assumption of symmetry is commonly made and seems reasonable (1,15).

We used a standard edge-following algorithm (e.g., 2,18) to generate an array of (x,y) pairs representing the perimeter of the cell. The first and last entries in this array are the same point, ensuring a closed traverse of the object edge. The orientation of an object is often expressed as a line going through the center of the object such that the sum of the squares of the perpendicular distances from all points in the object to the line is minimized. This line can be considered the axis of maximum elongation. We are particularly interested in the angle that this line makes with one of the coordinate axes. Given this orientation angle, we can rotate the object so that its elongation axis is parallel to the x-axis, thus simplifying subsequent computations. Orientation is calculated from the first and second moments of the array of (x,y) points making up the entire object (see Horn [14]).

Figure 1 is a graphical representation of what the biovolume algorithm does. Our program is written in C language and the main parts of the program are presented in pseudocode in the Appendix. Complete source code is available from the authors. Our imaging system uses a coordinate system with the origin at top left and x and y positive to the right and down, respectively. The first step is to rotate the cell perimeter around its centroid so the elongation axis is parallel to the x-axis (Appendix, step 1). This is accomplished by basing the rotation angle upon the axis of elongation, and it allows the subsequent solids of revolution to be easily generated around the axis of elongation. This choice of the axis of revolution is consistent with previous volume estimation techniques (15). We used the trigonometric equations of Hearn and Baker (12), modified for our coordinate system, and a clockwise rotation (Fig. 1B; Appendix, step 1).

The second step is to fill in "holes" in the perimeter caused by rotation (see Appendix, step 2). When the perimeter array is rotated, the original integer values of x and y are mapped into real numbers. To step in the x-direction in equal increments, we must round the x-values back to integers. Rounding, however, will cause some integer x-values to be unrepresented; i.e., there will be holes in the perimeter array. For example, if the (x,y) coordinates of two adjacent perimeter points are (10.3, 43.5) and (11.7, 46.5) then the x values get "binned" into 10 and 12, respectively, causing a missed value in the perimeter at 11. There are at least two obvious ways to fill this hole. One is to simply extend

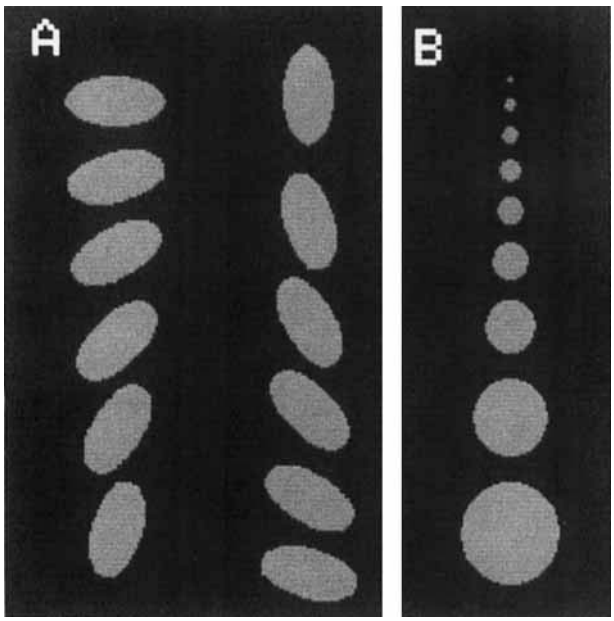

FIG. 2. The computer-generated graphical shapes used to test the effects of **A)** orientation and **B)** size on biovolume calculation.

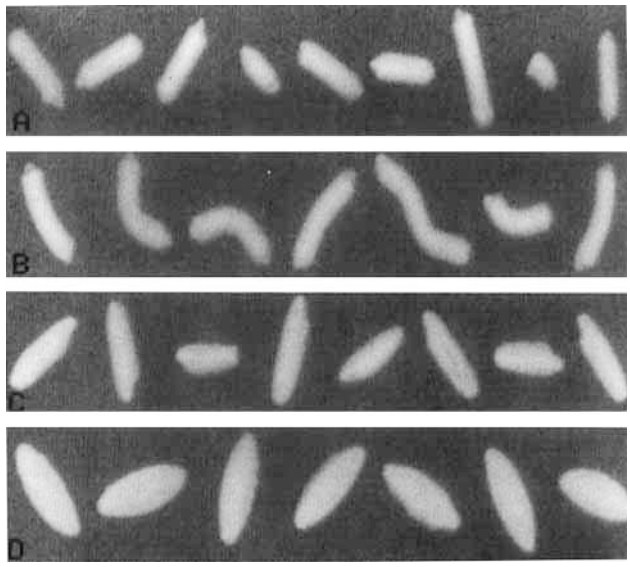

FIG. 3. Example images of the 4 objects used to test the biovolume algorithm: **A)** straight solder pieces, **B)** bent solder pieces, **C)** rice grains, and **D)** pasta.

one or the other of the two points adjacent to the hole, i.e., to add a point to the perimeter that is either (11.0, 43.5) or (11.0, 46.5). The other is to average the two y-values of the adjacent points and use the averaged value to add a point (11.0, 45.0) to the perimeter. The latter is the method we chose to implement. We constructed a new array of perimeter points that contained all of the old points as well as any points added because of holes. Adding these points does not cause an increase in the volume of the object. It only means that the perimeter points are closer together in the area where the point has been added. Both the length of the object and the size of the array needed for the next step is determined from the maximum and minimum x extent of the object perimeter. These are extracted from the perimeter array as shown in step 3 of the Appendix.

To calculate the object diameters at each x-value, the maximum and minimum y extent must be found for each x-value (Fig. 1C, Appendix, step 4). It is likely that multiple pixels will occur in the perimeter array at certain rounded x-values, such as where the perimeter approaches vertical. We must decide which of these pixels to use to calculate the diameter of the cell at that x-value. We chose to use the maximum and minimum y-values for each x-value so that the cell diameter at each x-value is the distance between these y-values. The choice of how to handle these multiple pixels must be made with regard to the object shapes which may be encountered. For simple, closed shapes, the distance between the maximum and minimum y-values seems reasonable.

The final operation is to calculate a solid of revolution for each x step and sum them (Fig. 1D). The algorithm steps down the midline of the cell, calculating a solid of revolution at each step using the equation for the volume of a cylinder. Biovolume is calculated by summing all these individual cylinder volumes. Ideally, the solid would be generated perpendicular to the midline, but this direction is not necessarily parallel to one of the coordinate axes and so makes computation more difficult. To facilitate computation, we chose to generate the solid perpendicular to the x-axis, a direction that is parallel to a coordinate axis. Although this compromise introduces some error in the volume calculation it will be relatively small as long as the midline is essentially parallel to the x-axis. This biovolume algorithm is referred to below as the "integration method."

A binary image (silhouette) would be sufficient to implement our algorithm on other systems. The perimeter array alone could be used, although we used all the pixels of each object image to determine orientation for rotating the perimeter.

Algorithm Verification

Initial tests of the integration method were conducted using graphically generated images. The effect of object image size on the resulting volumes estimates was tested by drawing circles of known radii on our monitor and then running our biovolume program on these. Potential error introduced by the rotation step of the algorithm was tested by measuring ellipses drawn with different initial orientations. These test images are shown in Figure 2. The measured volumes were then scaled against the theoretical volumes (V =

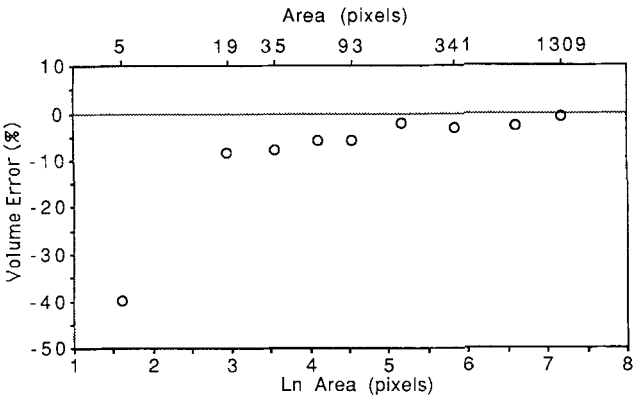

FIG. 4. Effect of object size on the calculation of biovolume using the integration method and computer-generated circles.

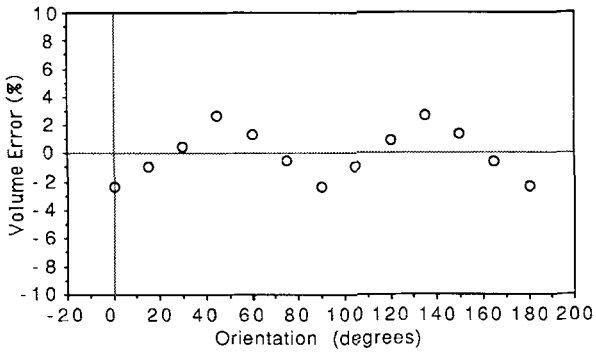

FIG. 5. Effect of object orientation on the calculation of biovolume using the integration method and computer-generated ellipses.

$4/3\ \pi r^3$ for circles, where r is radius and $V = (\pi/6)W^2L$ for ellipses, where L and W are the longest and shortest axes, respectively) of the appropriately sized objects.

In order to verify the integration method we tested it with images of several macroscopic objects, the volume of which could be independently determined. In addition to the integration method, we used two formulas based on length (L) and width (W) commonly used to measure bacterial (15) and protozoan (10) biovolume. The first method assumes a prolate spheroid shape and is calculated as

$$V = \frac{\pi}{6} W^2 L$$

The second method assumes the cell is a cylinder with hemispherical ends and is calculated as

$$V = \frac{\pi(L - W)W^2}{4} + \frac{\pi W^3}{6}$$

The objects we used were solid soldering wire pieces (specific gravity = 7.279), rice grains, and grain-shaped pasta (orzo). All were spray-painted with flat white enamel to ensure uniform reflectivity. The wire pieces varied in length from 0.5 cm to 1.4 cm and the ends were melted to simulate the rounded ends of cells. Such wire is a good choice because: 1) it does not violate the algorithm assumptions outlined above (in particular, it is circular in cross section); 2) it has uniform reflectivity; and 3) the actual volume of individual pieces can be accurately determined by weighing each piece and applying the known specific gravity of the wire.

The rice and pasta provided other shapes with which to test the algorithm. Average specific gravities of these were determined by weighing and measuring the volume displacement in water of a set of individuals ($n > 30$). Volumes of individuals could then be determined by weighing each and multiplying by the average specific gravity. These objects were not circular in cross section, so the estimates of volume from the 2-dimensional image would be larger than the true volume. We measured the ratio of minimum to maximum cross-sectional diameters of the rice and pasta in order to correct for this asymmetry and compare methods. The spray paint was considered to be part of the object in all calculations. Example images of the four types of objects used in the analysis are shown in Figure 3.

Our imaging system consisted of a black-and-white video camera (Cohu Electronics, San Diego, CA) with a macro lens attached. Objects were placed on a black background and illuminated from above with a dual fiber optics light source. Images were digitized into a Dual 83/20 (Dual Systems Corp., Berkeley, CA) computer using a frame-grabber and imaging system (Digital Graphics Systems Inc., Mountain View, CA), and thresholded using an automatic technique (21) that chooses the edge of the object to be where the pixel gray level gradient is strongest (i.e., the maximum of the first derivative of the image profile). The length and width of each object were taken to be the maximum and minimum projections of the object image, respectively.

Percentage errors for each method based on measurements of individual objects were calculated as:

$$\%\ \text{error} = \left(\frac{\text{estimate}}{\text{direct}} - 1\right) \times 100$$

where "estimate" is the volume of the individual measured by the various image analysis methods and "direct" is the volume calculated using weight and average specific gravity.

RESULTS

The effect of object image size on our biovolume estimate indicated that error was less than 10% of theoretical for circles above about 20 pixels in area (Fig. 4).

Table 1
Measured Volumes of the Four Test Objects Using Direct Measurement (By weight and specific gravity) and Three Image Analysis Methods Including the Biovolume Algorithm (Integration) and Simple Shape Formulas (Prolate spheroid and cylinder and hemispherical ends)

Object	n	Mean symmetry[2]	Mean volumes (mm^3) (±95% CI)[1]			
			Direct measure	Image analysis methods		
				Integration	Prolate spheroid	Cylinder
Straight solder	30	1.000	73.6 (29.6–98.8)	68.6 (31.8–88.2)	64.8 (34.1–84.0)	86.9 (38.5–115.4)
Bent solder	10	1.000	100.3 (79.8–135.0)	98.7 (77.9–130.6)	158.5 (94.6–265.8)	208.5 (127.0–358.7)
Pasta	30	0.820	29.1 (± 1.6)	27.3* (± 0.9)	31.3* (± 1.1)	40.7* (± 1.4)
Rice	30	0.768	12.7 (± 0.5)	12.3* (± 0.7)	12.8* (± 0.7)	17.1* (± 0.9)

[1]For solder pieces, ranges are shown rather than CI, since they were not necessarily normally distributed.
[2]Direct measure of ratio of minimum to maximum cross sectional diameters (width to depth ratio).
*For comparison these estimates have been corrected for asymmetry.

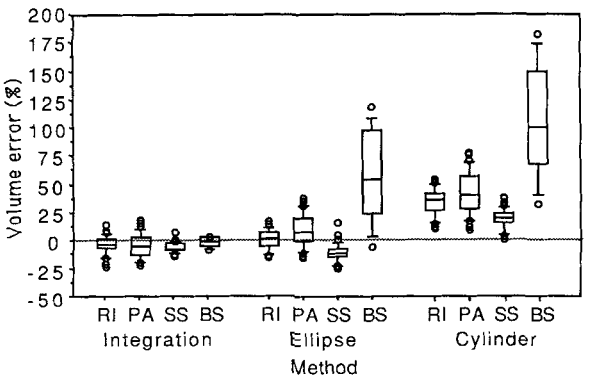

FIG. 6. The percent error in volume for each of the 3 image analysis methods and for the 4 test objects: RI = rice grain, PA = pasta, SS = straight solder, and BS = bent solder. The boxes enclose the 25th to 75th percentiles, with the horizontal line in the box at the median (50th percentile). The whiskers show the 10th and 90th percentiles and individual points outside these are shown. Positive error indicates overestimation.

For objects smaller than this, error increased rapidly. For example, the volume of the "circular" object composed of 5 pixels was underestimated by 40%. Initial orientation had a smaller effect on the biovolume estimation (Fig. 5). Error varied periodically with orientation angle, but estimated volume was always within 3% of theoretical volume.

Measured mean volumes and ranges (solder) or 95% confidence intervals (pasta and rice) are shown in Table 1. The average specific gravities of rice and pasta were 1.470 and 1.460, respectively. The object images (Fig. 3) were composed of from 381 to 1,509 pixels. The straight solder images averaged 833 pixels (range, 381–1075). The bent solder images averaged 1,122 pixels (range, 929–1490), and the pasta and rice images averaged 1,354 ± 24 and 877 ± 33 (95% CI) pixels. For the purpose of comparison, the image analysis methods for the pasta and rice have been corrected for asymmetry (Table 1). The distributions of the individual errors by the 3 methods for the 4 object types show that the integration method had consistently less error (Fig. 6) and performed particularly well on the bent solder pieces in comparison with the other methods. Although the integration method tended to slightly underestimate biovolume, the error averaged less than 7%. The method based on assuming a prolate spheroid shape performed well on the straight objects but poorly on the bent solder. The method based on assuming a cylinder with hemispherical ends overestimated the volumes of all the experimental objects.

DISCUSSION

Our integration method consistently underestimated the true volume of objects on average from 3.5% to 6.5%. This bias is most likely due to our maximum gradient threshold selection method. In other words, the true edge of the object in the image may occur somewhere other than where the grey level gradient is greatest. This problem has been addressed elsewhere (21). Choosing a slightly lower grey level threshold would have made slightly larger images and thus unbias the estimate, but because this error is relatively small we chose to use the fully automatic method of choosing a threshold.

Error in the volume estimates also arises from two other sources. Firstly, there is a loss of accuracy when obtaining a digital image from a continuous scene for a digital image is only a sample of the real object. Variability in estimated volume due to image size and orientation can be explained within the context of this source. Secondly, an unavoidable loss of accuracy arises when inferring a 3-dimensional structure based upon a single 2-dimensional description of that structure. However the relatively low errors of the integration method for the bent solder pieces illustrate the

advantage of this algorithm over methods using geometric formulas. Because the integration method steps along the midline in small increments, large or small undulations do not contribute to error. The poor performance of the ellipsoid and cylinder methods underscores the problems of using a simple shape factor to estimate cell volume.

The shapes that we chose to analyze for verifying the algorithm approximate the shapes of bacteria and many simple protists found in the marine environment. More complicated cell shapes are found in these populations and the algorithm would not be as accurate for these cells. However, the assumptions about shape that we have used are the same used by most marine microbiologists for calculating cell volume visually (3,5,10). Another assumption we and others make is that the cells are oriented with their longest axis parallel to the image plane. The appeal of the integration method lies in its ability to handle many different shapes automatically and with reasonable accuracy. At its worst, when cell images are perfect circles, or perfect ellipses, it does as well as any formula. At its best, it gives a significantly more accurate estimate of cell volume than could be obtained using a geometric formula.

ACKNOWLEDGMENTS

The authors thank D. Evans for useful discussions of the algorithm and B. Sherr for discussions concerning this approach for measuring protozoans from natural waters. This work was partially funded by NSF grant OCE-88-13356.

LITERATURE CITED

1. Baldwin WW, Pankston PW: Measurement of live bacteria by Nomarski interference microscopy and stereologic methods as tested with macroscopic rod-shaped models. Appl Environ Microbiol 54:105–109, 1988.
2. Ballard DH, Brown CM: Computer Vision. Prentice-Hall, Englewood Cliffs, New Jersey, 1982.
3. Beers JR, Reid FMH, Stewart GL: Microplankton of the North Pacific Central Gyre. Population structure and abundance, June 1973. Int Rev Ges Hydrobiol 60:607–638, 1975.
4. Bjørnsen PK: Automatic determination of bacterioplankton biomass by image analysis. Appl Environ Microbiol 51:1199–1204, 1986.
5. Borsheim KY, Bratbak G: Cell volume to cell carbon conversion factors for a bacterivorous *Monas* sp. enriched from seawater. Mar Ecol Progr Ser 36:171–175, 1987.
6. Bratbak G: Bacterial biovolume and biomass estimations. Appl Environ Microbiol 49:1488–1493, 1985.
7. Bratbak G, Dundas I: Bacterial dry matter content and biomass estimations. Appl Environ Microbiol 48:755–757, 1984.
8. Brownlee DC: Measuring the secondary production of marine planktonic tintinnine ciliates. Ph.D. Dissertation. University of Maryland, College Park, MD, 1982.
9. Caron DA: Technique for enumeration of heterotrophic and phototrophic nanoplankton, using epifluorescence microscopy, and comparison with other methods. Appl Environ Microbiol 46:491–498, 1985.
10. Geider RJ, Leadbeater BSC: Kinetics and energetics of growth of the marine choanoflagellate *Stephanoeca diplocostata*. Mar Ecol Prog Ser 47:169–177, 1988.
11. Haas LW: Improved epifluorescence microscopy for observing planktonic microorganisms. Ann Instit Oceanogr 58:940–946, 1982.
12. Hearn D, Baker MP: Computer Graphics. Prentice-Hall, Englewood Cliffs, New Jersey, 1986.
13. Hobbie JE, Daley RJ, Jasper S: Use of Nuclepore filters for counting bacteria by fluorescence microscopy. Appl Environ Microbiol 33:1225–1228, 1977.
14. Horn BKP: Robot Vision. MIT Press. Boston, Massachusetts, 1986.
15. Krambeck C, Krambeck HJ, Overbeck J: Microcomputer-assisted biomass determination of plankton bacteria on scanning electron micrographs. Appl Environ Microbiol 42:142–149, 1981.
16. Lee S, Fuhrman JA: Relationships between biovolume and biomass of naturally derived marine bacterioplankton. Appl Environ Microbiol 53:1298–1303, 1987.
17. Porter KG, Feig YS: The use of DAPI for identifying and counting aquatic microflora. Limnol Oceanogr 25:943–948, 1980.
18. Rosenfeld A, Kak AC: Digital Picture Processing. Academic Press, New York, 1976.
19. Sieracki ME, Johnson PW, JMcN Sieburth: The detection, enumeration and sizing of aquatic bacteria by image analyzed epifluorescence microscopy. Appl Environ Microbiol 49:799–810, 1985.
20. Sieracki ME, Webb KL: Applications of image analyzed fluorescence microscopy for quantifying and characterizing planktonic protistan communities. In: Protozoa and their Role in Marine Processes. Reid C, Burkhill P (eds). Springer-Verlag, New York, in press.
21. Sieracki ME, Reichenbach SE, Webb KL: An evaluation of automated threshold selection methods for accurate sizing of microscopic fluorescent cells by image analysis. Appl Environ Microbiol, submitted.

APPENDIX: PSEUDOCODE FOR CALCULATING CELL BIOVOLUME

Variable representations and conventions are as follows:

p	= array of unrotated perimeter points.
p′	= array of rotated perimeter points.
p^{new}	= array of rotated perimeter points with holes filled.
i	= index into p, p′, and p^{new} representing the current point.
θ	= angle of orientation.
$p^{new}_{5.x}$	= x coordinate of the 5th point in p^{new}

This algorithm assumes the array p and orientation angle θ have already been generated.

1) Rotate the object perimeter around its centroid so the elongation axis is parallel to the x-axis.

We are interested in the edge of the object, so only the perimeter needs to be rotated. The transformation for our coordinate system (origin top left, y positive going down, x positive to the right) is as follows:

$$x' = x_r + (x - x_r)\cos\phi + (y - y_r)\sin\phi$$
$$y' = y_r - (x - x_r)\sin\theta + (y - y_r)\cos\phi$$

where

ϕ	= −θ (for clockwise rotation of the array p).
(x_r, y_r)	= $(\bar{x}, \bar{y})$ = centroid of object = rotation point.
(x, y)	= unrotated perimeter point.
(x′, y′)	= rotated perimeter point.

The centroid coordinates are the average x and y

values for the entire object image, including interior points.

```
FOR each point i begin                    /*Rotate*/
p'i . x = xr + (pi . x − xr) * cos ϕ + (pi . y − yr) * sin ϕ;
p'i . y = yr − (pi . x − xr) * sin ϕ + (pi . y − yr) * cos ϕ;
    i = i + 1;
end;        /* FOR loop */
```

2) Fill the "holes" in the perimeter caused by rotation.

```
FOR each point i begin
  p^newj . x = p'i . x;        /* Retain old points.*/
  p^newj . y = p'i . y;
  cur = ROUND (p'i . x);       /* Rounded x value
                                  of current point.*/
  next = ROUND (p'i+1 . x);    /* Rounded x
                                  value of adjacent point.*/
  IF (ABS(cur − next) > 1) THEN begin
                                  /* Add a point.*/
    j = j + 1;
    p^newj . x = 0.5 * (cur + next);  /* Average the
                                         two */
    p^newj . y = 0.5 * (p'i . y + p'i+1 . y); /* adjacent
                                                 points. */
    end;      /* IF THEN */
  i = i + 1;
  j = j + 1;
end; /* FOR loop */
```

p^{new} is the new perimeter array containing all the old points as well as points added to fill the holes. J is the index into p^{new}.

3) Find the maximum and minimum x entent of the object perimeter.

```
xmax = p^new1 . x;          /* Initialize using
                               first point */
xmin = p^new1 . x;
For each point i begin      /* Find xmax and
                               xmin */
  If (p^newi . x > xmax) THEN
      xmax = p^newi . x;
  If (p^newi . x < xmin) THEN
      xmin = p^newi . x;
    i = i + 1;
end; /* FOR loop */
bins = xmax − xmin + 1;     /* Size of array
                               for step 4 and 5 */
offset = xmin;
```

Bins is used below (step 4) to size the arrays of maximum and minimum y-extent. Offset is used to index into these arrays.

4) Find the maximum and minimum y extent for each x value.

```
FOR each bin k begin
  ymink = BIG;              /* Initialize to a very
                               large number */
  ymaxk = −BIG;             /* Initialize to a very
                               small number */
  k = k + 1;
end;     /* FOR loop */
FOR each point i begin
  xtmp = ROUND(p^newi . x − offset);   /* cal-
                          culate the appropriate x-bin */
  ytmp = p^newi . y;
  IF (ytmp < ymin_xtmp) THEN
    ymin_xtmp = ytmp;
  IF (ytmp > ymax_xtmp) THEN
    ymax_xtmp = ytmp;
  i = i + 1;
end;     /* FOR loop */
```

Ymin and ymax are the arrays of minimum and maximum y-extent. K and xtmp are used as indices into ymin and ymax.

5) Calculate and sum solids of revolution for each x step.

```
sum = 0.0;                  /* Initialize to 0.0 */
FOR each bin k begin
  radius = (ymaxk − minyk + 1.0) * 0.5;
                            /* Calculate radius */
  sum = sum + radius^2;     /* Calculate and
                               sum volume */
  k = k + 1;
end;                        /* FOR loop */
volume = sum * π;
```