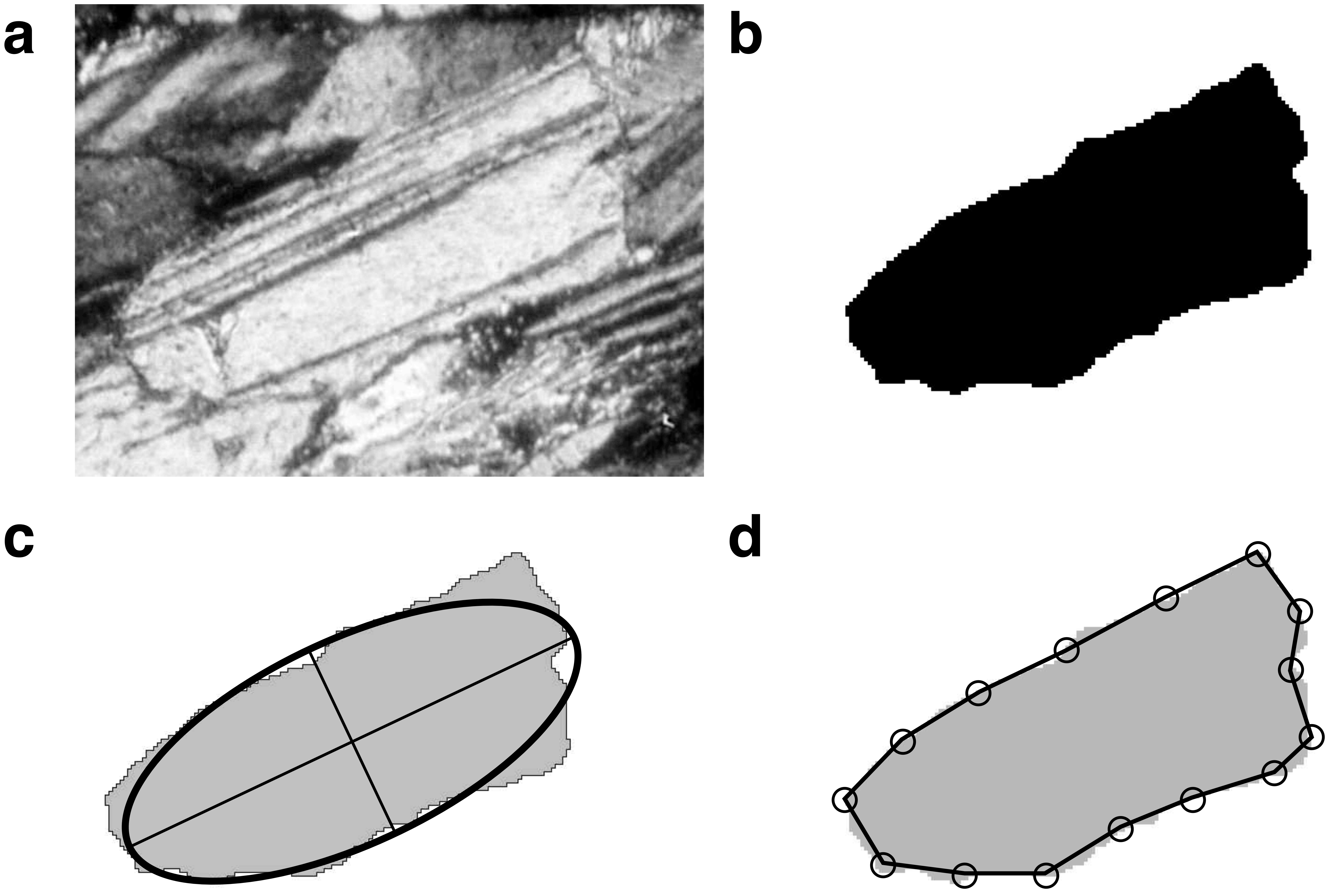
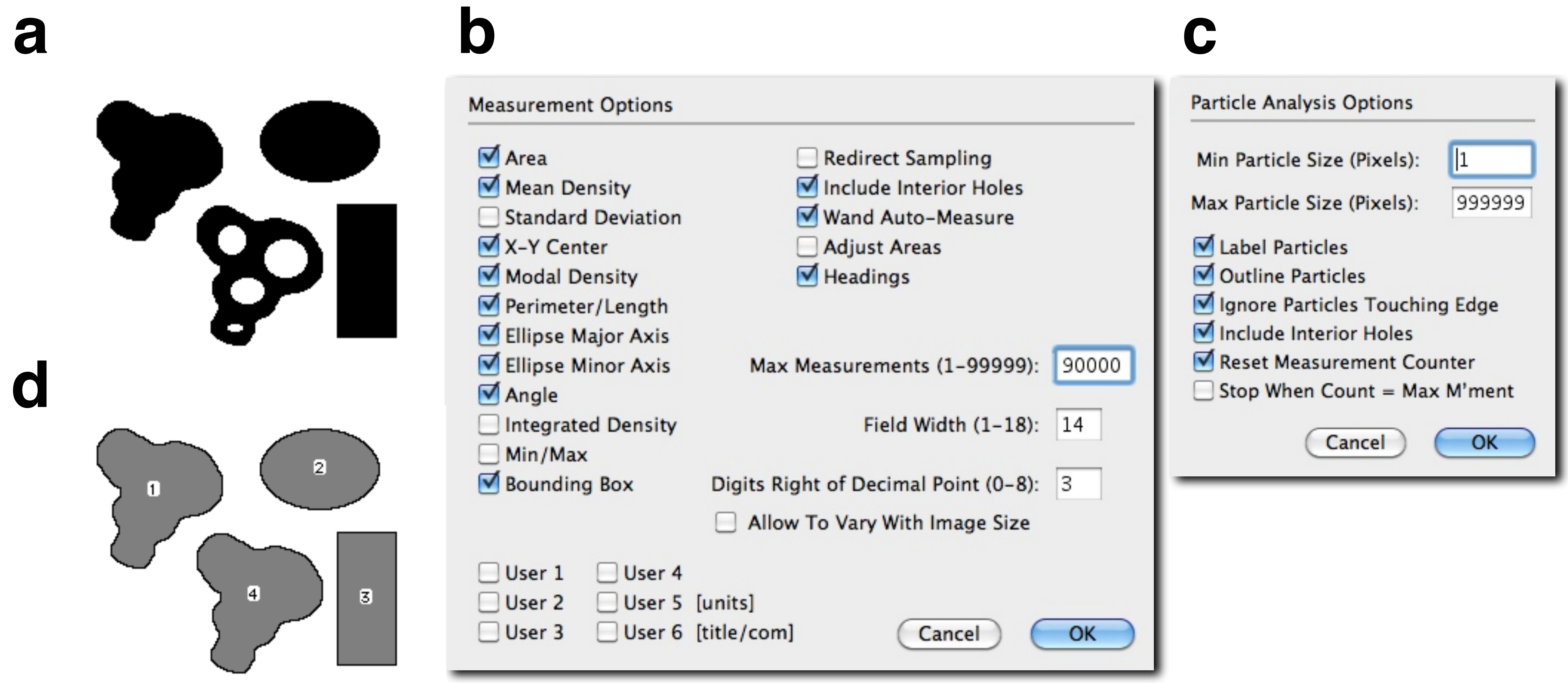


Figure 8.1
The objects of image analysis.
(a) Micrograph of twinned grain in deformed marble: grayscale image;
(b) segment, representing cross sectional area of the grain;
(c) outline and axes of best-fit ellipse superposed on segment;
(d) digitized outline of segment: polygonal chain connecting 15 vertices along segment boundary.



e

#	Area	Mean	X	Y	Mode	Len	Majr	Minr	Angle	xBB	yBB	wBB	hBB
1	4664	255.00	50.17	54.88	255	303.81	81.37	72.98	70.80	13	15	82	91
2	3258	255.00	157.50	41.00	255	217.48	78.05	53.15	0.00	119	15	78	53
3	3393	255.00	188.00	125.00	255	249.66	98.17	44.01	90.00	169	82	39	87
4	4664	191.03	115.17	122.88	255	303.81	81.37	72.98	70.80	78	83	82	91

Figure 8.2
Using the Analyze menu of Image SXM.
(a) Bitmap with 4 shapes (segments);
(b) dialog window for Measurement Options;
(c) dialog window for Particle Analysis Options;
(d) same as (a), after analysis;
(e) list showing results:
\# = number of particle;
Area = area (number of pixels);
Mean = mean density of segment;
X = x-coordinate of geometric center;
Y = y-coordinate of geometric center;
Mode = modal gray value;
Len = length of outline of segment;
Majr = major diameter of best-fit ellipse;
Minr = minor diameter of best-fit ellipse;
Angle = angle of Majr with respect to positive x-axis (CCW positive);
xBB = x-coordinate off top left corner of bounding box;
yBB = y-coordinate off top left corner of bounding box;
wBB = width of bounding box;
hBB = height of bounding box.

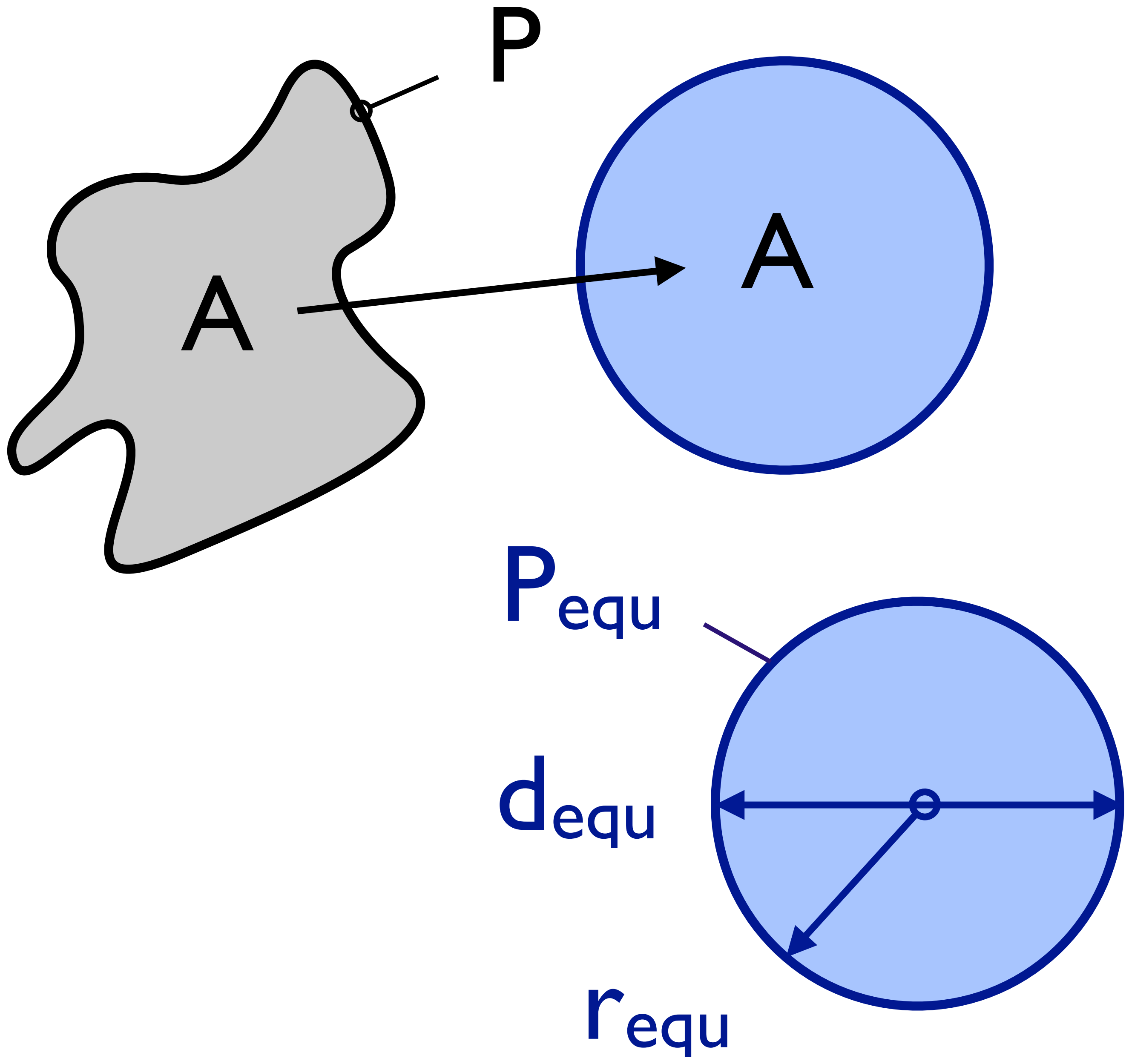


Figure 8.3
Shape and size measures from segments.
In black: measured; in blue: derived;
A = area of segment;
P = length of perimeter;
r_{equ} = equivalent radius;
d_{equ} = equivalent diameter;
P_{equ} = equivalent perimeter.

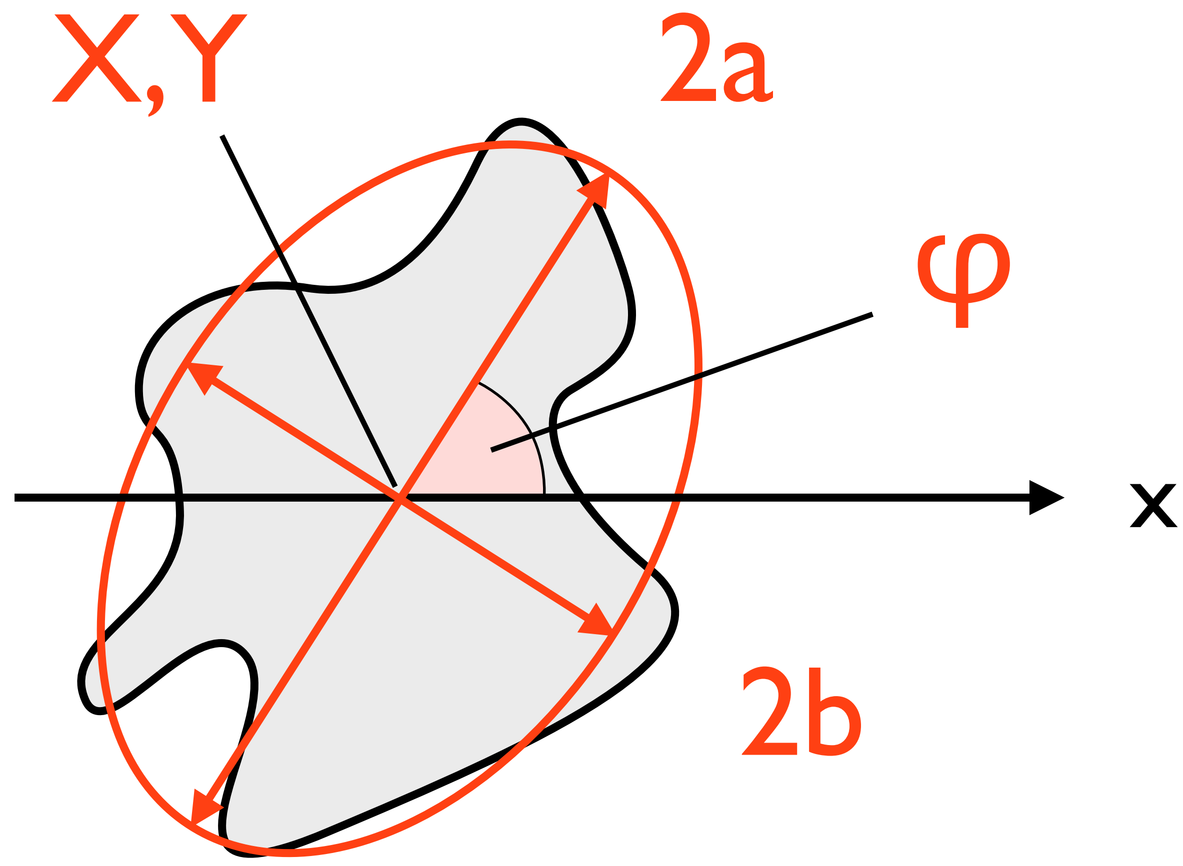


Figure 8.4
Shape and size measures from best-fit ellipses.
X,Y = center point;
2a (=Majr) = major diameter;
2b (=Minr) = minor diameter;
φ = orientation of major diameter, positive, CCW from positive x-axis.

measure	describing:
r_{equ}, a, b	size (linear)
wBB, hBB	caliper diameters, (projection lengths)
P / P_{equ}	circularity, ("fractal dimension")
b / a	axial ratio, (roundness)
a / b	aspect ratio, (elongation)

Table 8.1
Shape and size descriptors derived from the analysis of segments and best-fit ellipses.

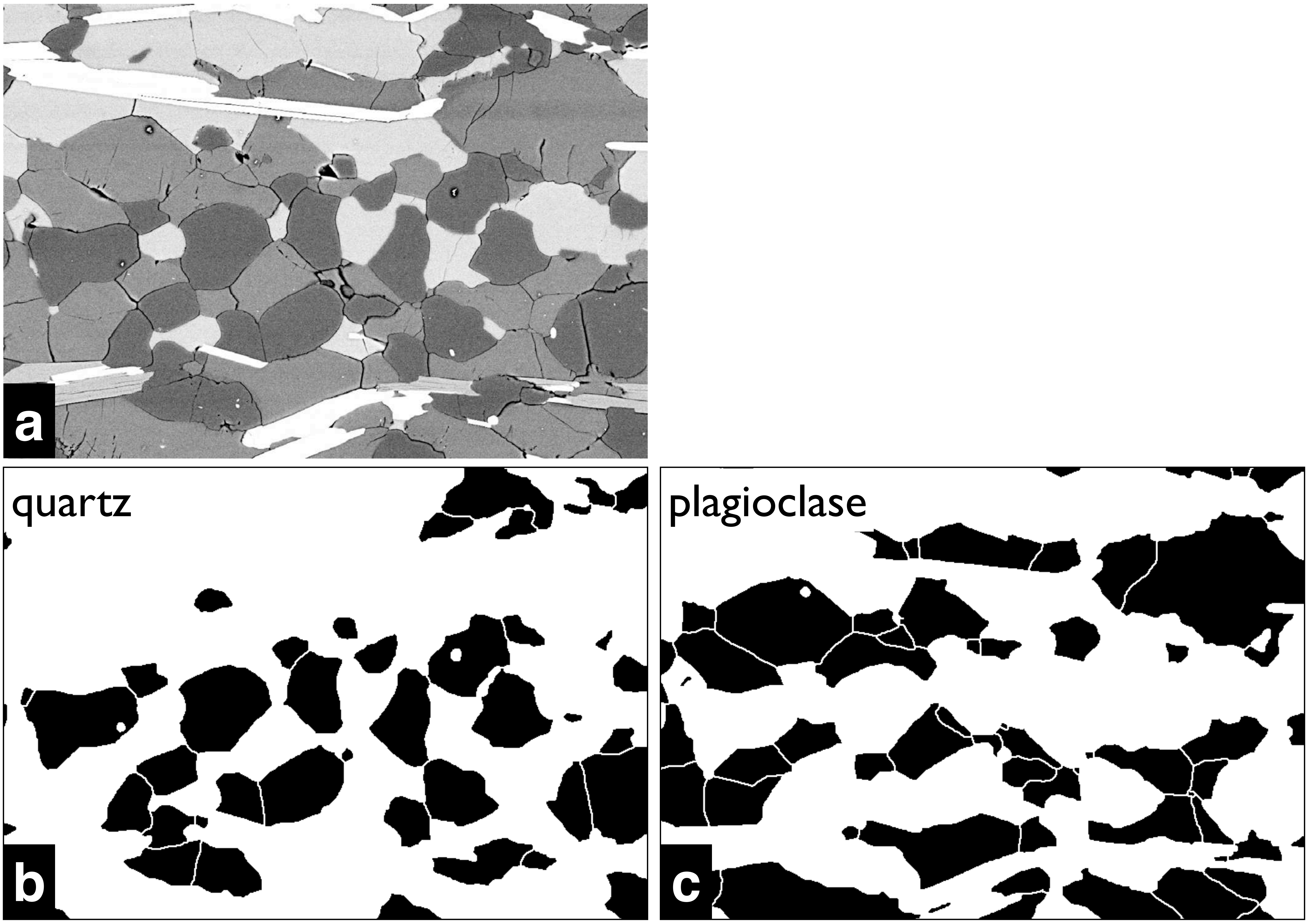


Figure 8.5
Segmented input for image analysis.
(a) Original SEM micrograph of granitoid rock;
(b) bitmap of quartz;
(c) bitmap of plagioclase.

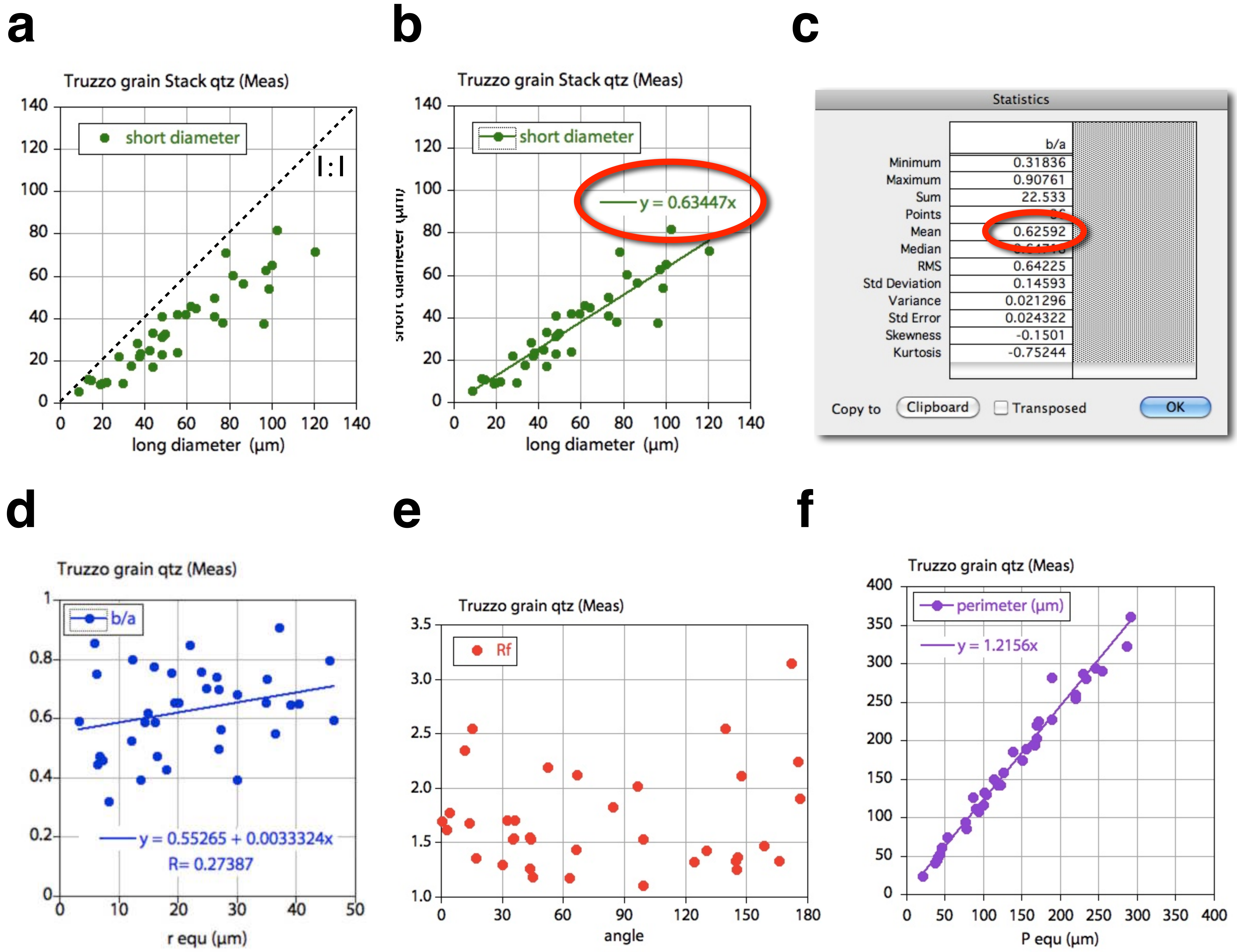


Figure 8.6
Results for quartz.
Bitmap shown in Figure 8.5.b has been analyzed.
(a) Plot of short versus long diameter;
(b) same as (a) with linear curve fit (through zero): slope = 0.634;
(c) statistics of axial ratio (b/a): average = 0.626;
(d) (b/a) versus equivalent radius, r_{equ};
(e) R_f - φ plot: a/b versus orientation;
(f) measured perimeter, P, versus equivalent perimeter, P_{equ}: 'fractal dimension' = 1.22.

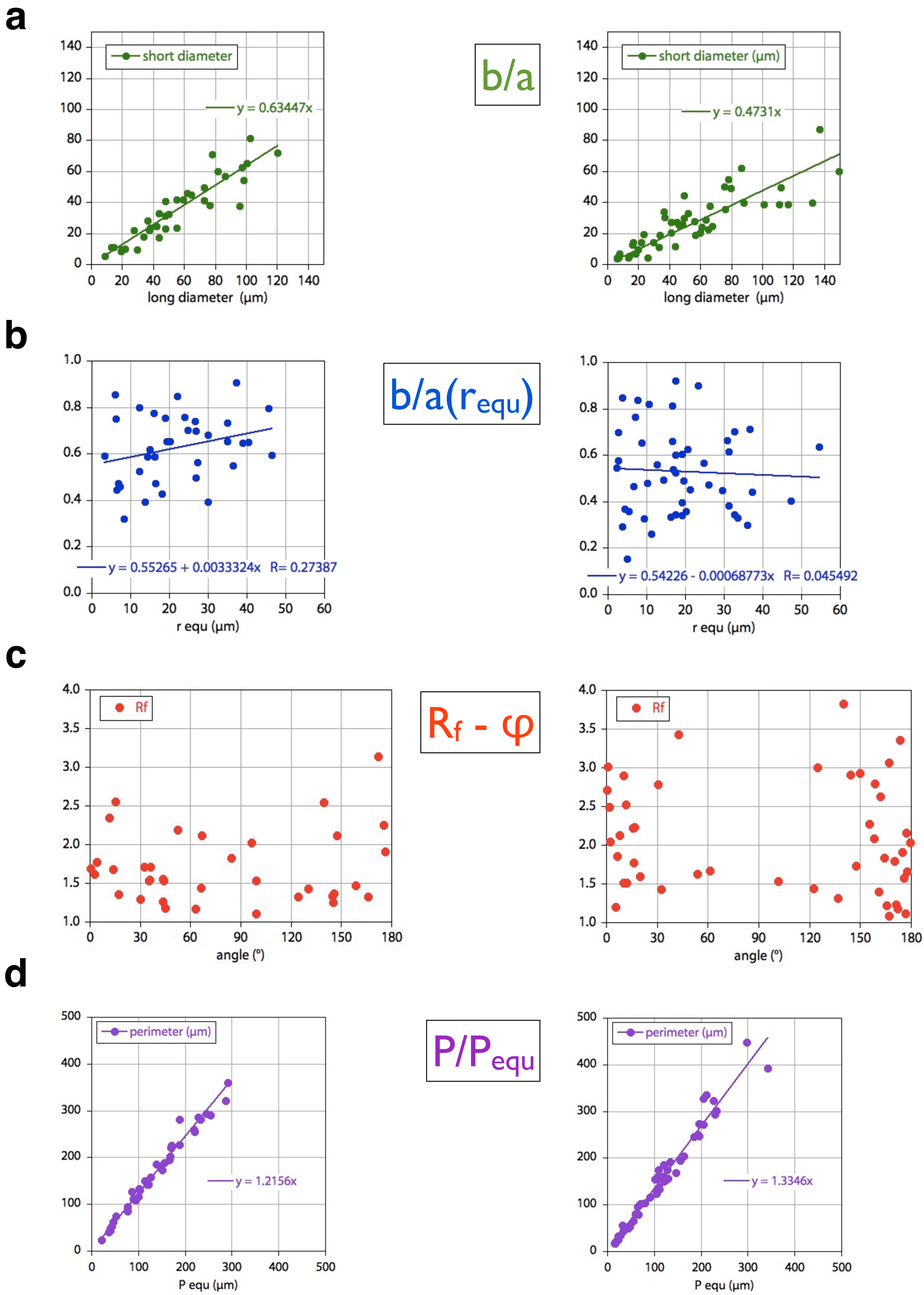


Figure 8.7

Comparative analysis of quartz and plagioclase. Bitmaps shown in Figure 8.5.b and 8.5.c have been analyzed. (a) average axial ratio: quartz b/a = 0.63, plagioclase b/a = 0.47; (b) dependence of shape on size; (c) preferred orientation, R_f - φ plot, quartz ± random, plagioclase preferred orientation at 0° (=180°); (d) shape of boundary, 'fractal dimension': quartz P/P_{equ} = 1.22, plagioclase P/P_{equ} = 1.33.

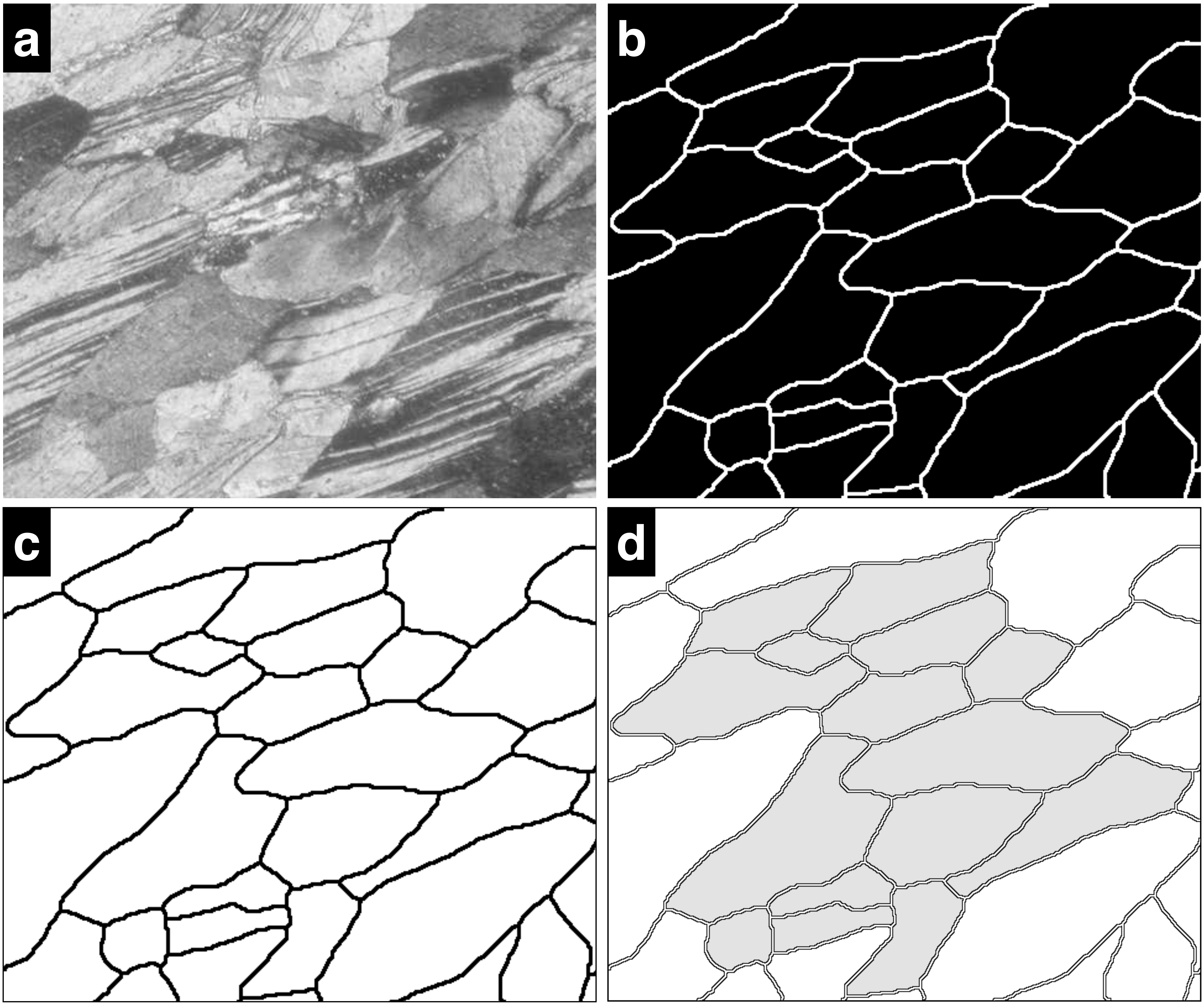


Figure 8.8
Segment boundaries and outlines.
(a) Original grayscale image (micrograph of deformed marble);
(b) segmented bitmap of (a): grain map;
(c) grain boundary map of (a);
(d) outlines of segments: outlines of gray segments are closed; all others are open.

digitize xy [1]
end 9999 [2]
fill area [3]

```
macro 'digitize xy [1]';
var
   i,j,x,y:integer;
XCount:integer;

begin
  SetUser1Label('x');
  SetUser2Label('y');
  SetOptions('User1 User2 ');
  XCount:=rCount+1;
GetMouse(x,y);
rUser1[XCount]:=x;
rUser2[XCount]:=y;
Measure;
ShowResults;

end;
```

```
macro 'end 9999 [2]';
var
   i,j,x,y:integer;
XCount:integer;

begin
  SetUser1Label('x');
  SetUser2Label('y');
  SetOptions('User1 User2 ');
  XCount:=rCount+1;
rUser1[XCount]:=9999;
rUser2[XCount]:=9999;
Measure;
ShowResults;

end;
```

```
macro 'fill area [3]';
var
   i,j,x,y:integer;
XCount:integer;

begin
  SetUser1Label('x');
  SetUser2Label('y');
  SetOptions('User1 User2 ');
  XCount:=rCount+1;
GetMouse(x,y);
SetForeGroundColour(128);
AutoOutline(x,y);
Fill;

end;
```

Figure 8.9
Digitizing macro.
[1] Digitize mouse location;
[2] write end coordinate (9999,9999);
[3] fill segment under cursor (works only for bitmaps).

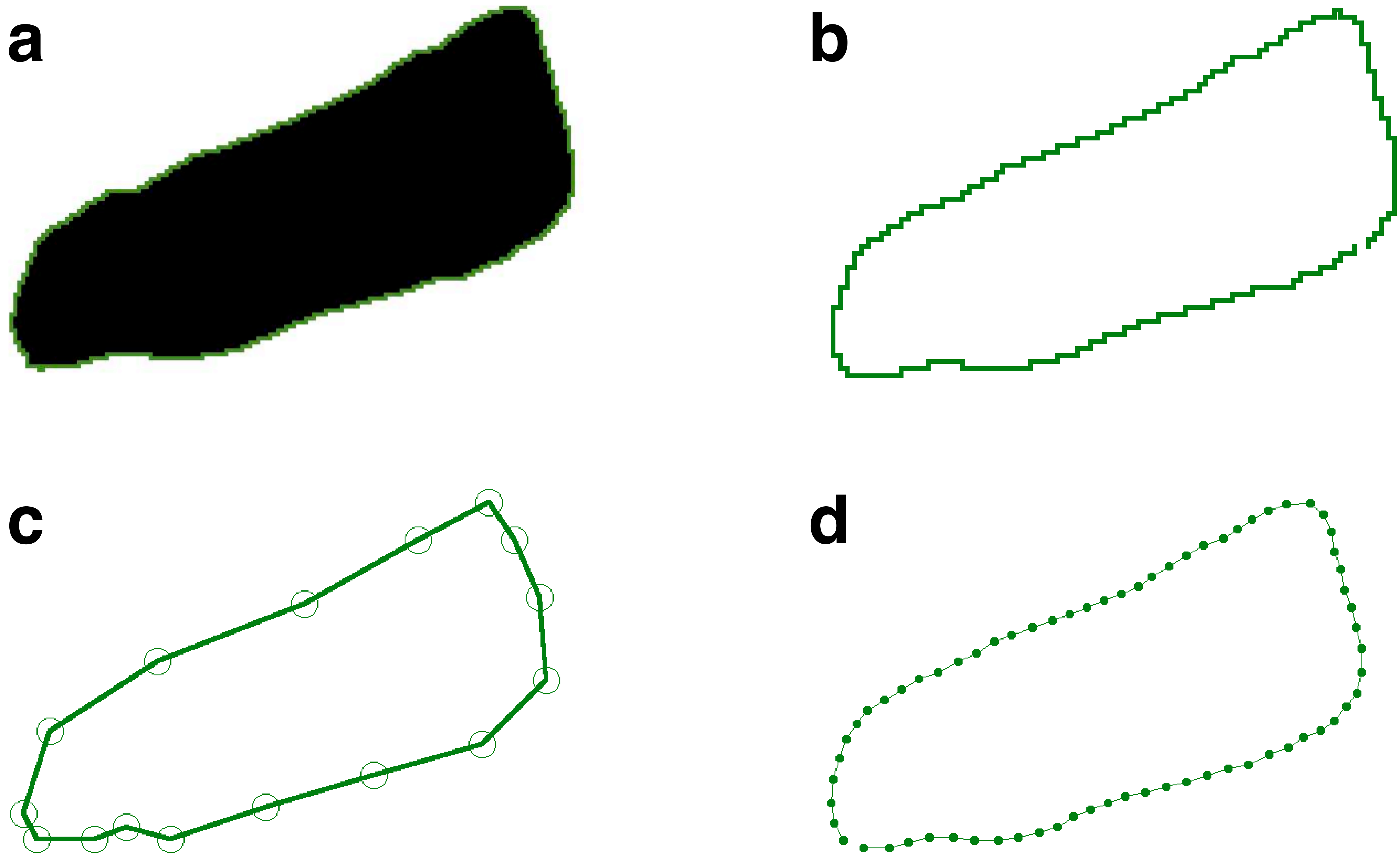


Figure 8.10
Digitizing outlines.
(a) Bitmap of segment; boundary pixels highlighted;
(b) 253 line segments connecting 254 boundary pixels; last and first are not connected;
(c) inscribed polygon; 16 vertices marked by circles;
(d) using high resolution digitizing tablet: 69 straight line segments connect 69 digitized points; loop is closed.

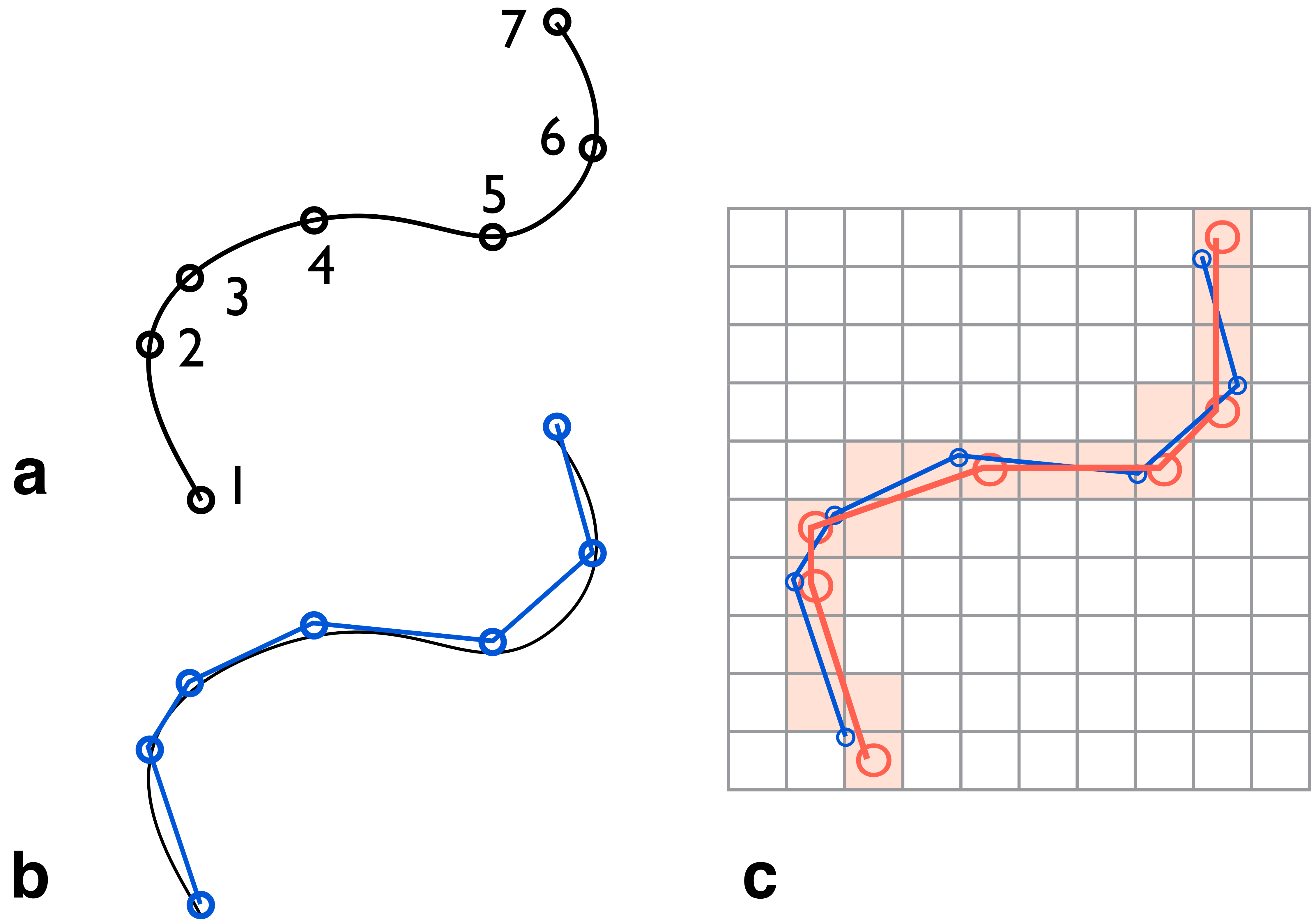


Figure 8.11
Curved lines on a grid.
(a) Original continuously curved line, connecting 7 points;
(b) same as (a) with straight lines (polygonal chain) connecting points (vertices) in blue;
(c) polygonal chain of (b) placed on a digitizing grid (blue circles and lines); grid points closest to vertices are connected to form digitized version of (a) (red circles and lines); boundary pixels are highlighted in light red.

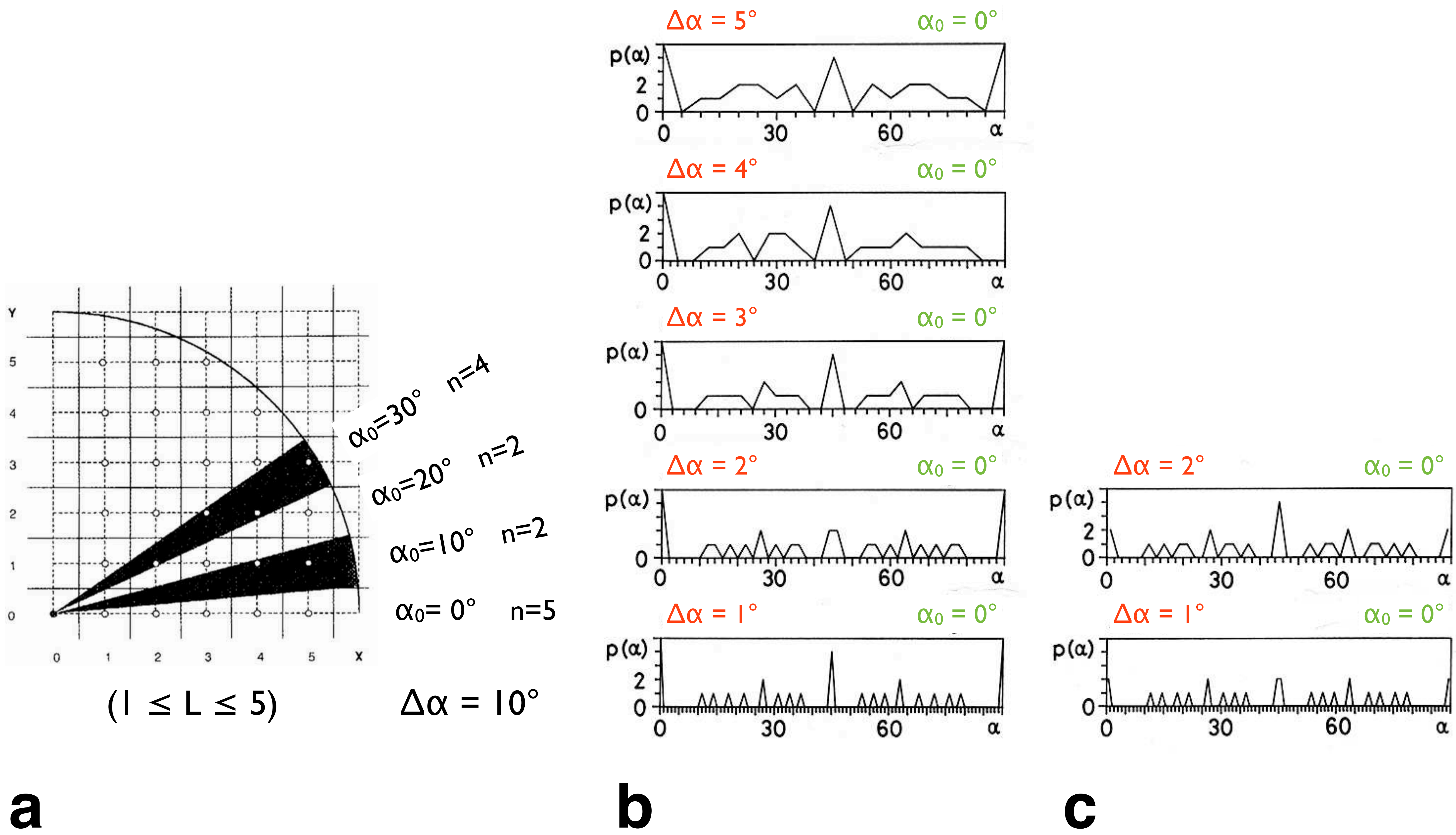


Figure 8.12
Digitizing artifacts.
(a) Digitizing grid with four 10° sectors; n = number of possible end points for lines (1 ≤ L ≤ 5) starting at origin;
(b) number of possible lines, p(α), as function of orientation, for lines (1 ≤ L ≤ 5) and for different angular resolutions, Δα; the first bin is centered at 0°;
(c) same as (b); the first bin is centered at Δα/2;
Δα = angular resolution;
α_0 = center of first bin;
L = length of line segments in grid units;
p(α) = grid points within sector.

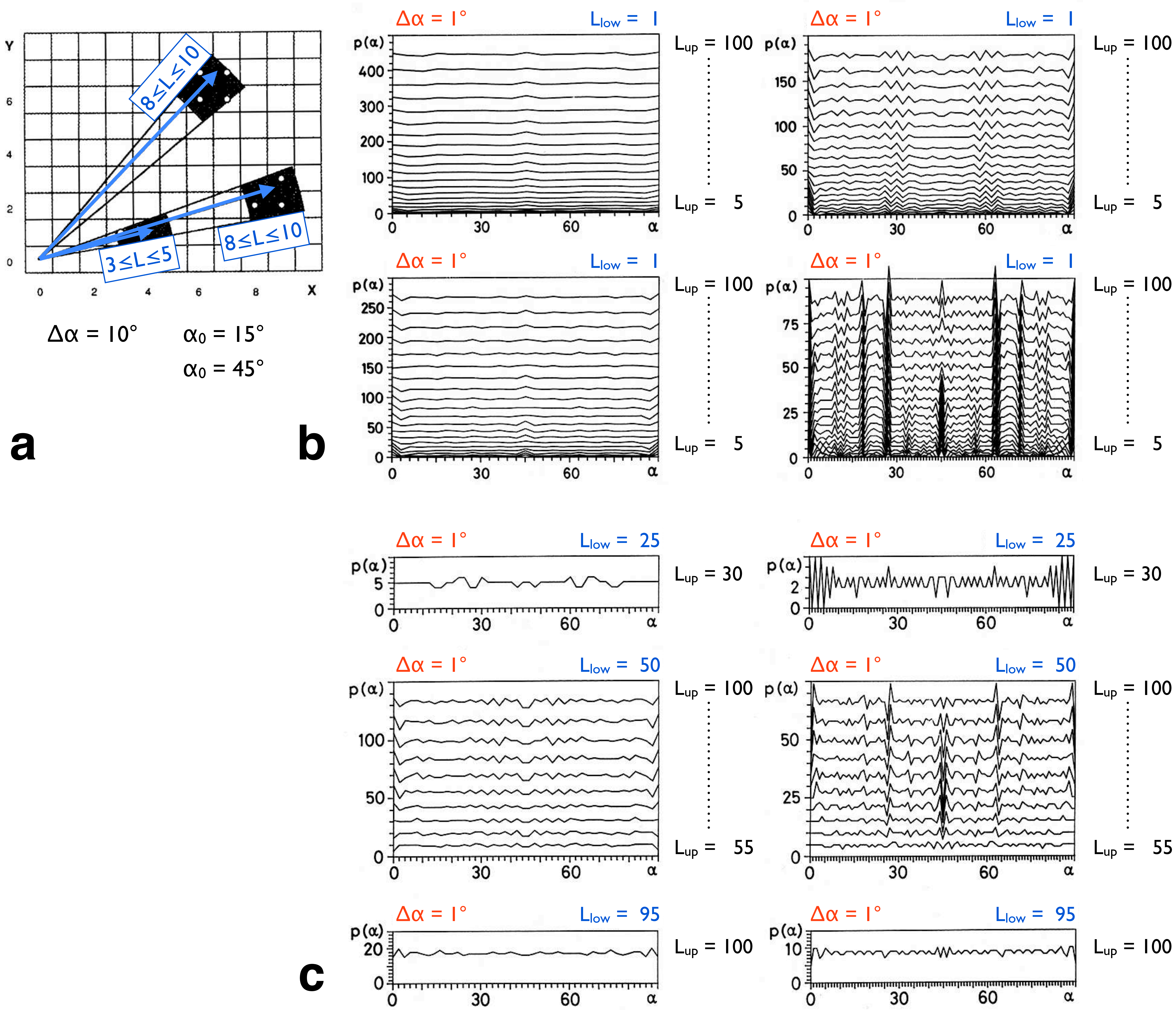


Figure 8.13

Density distribution of grid points.

(a) Three ring sectors with $\Delta L = 2$ and $\Delta\alpha = 10°$;

(b) number of possible lines, $p(\alpha)$, as function of orientation, for lines with $L_{low} = 1$ and $L_{up} = 5$ to 100, for different angular resolutions, $\Delta\alpha$;

(c) same as (b), for $L_{low} = 25$, 50 and 95;

$\Delta\alpha$ = angular resolution;

α_0 = center of first bin = 0°;

L = length of line segments in grid units; L_{low} = lower bound; L_{up} = upper bound;

$p(\alpha)$ = grid points within sector $\Delta\alpha$.

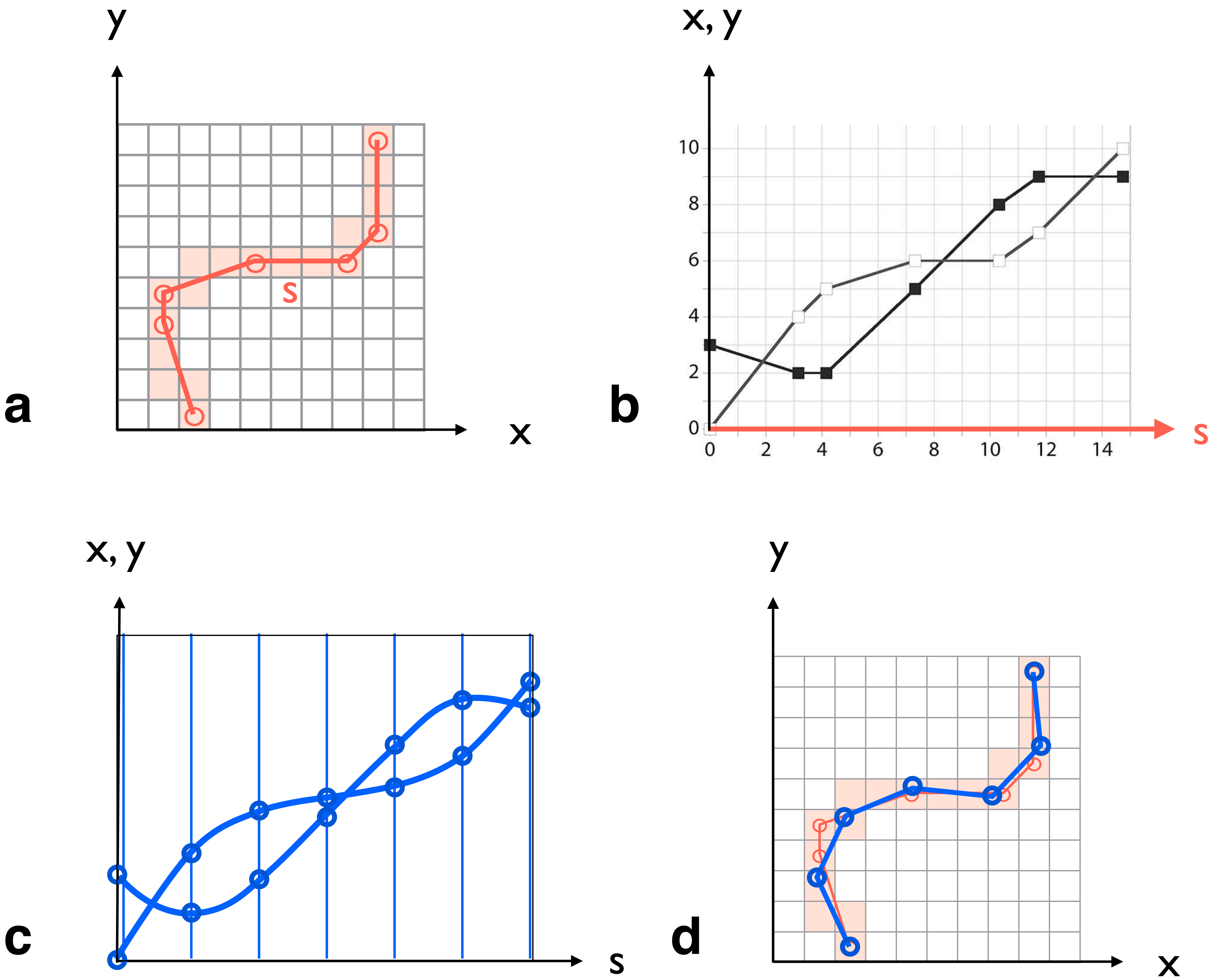
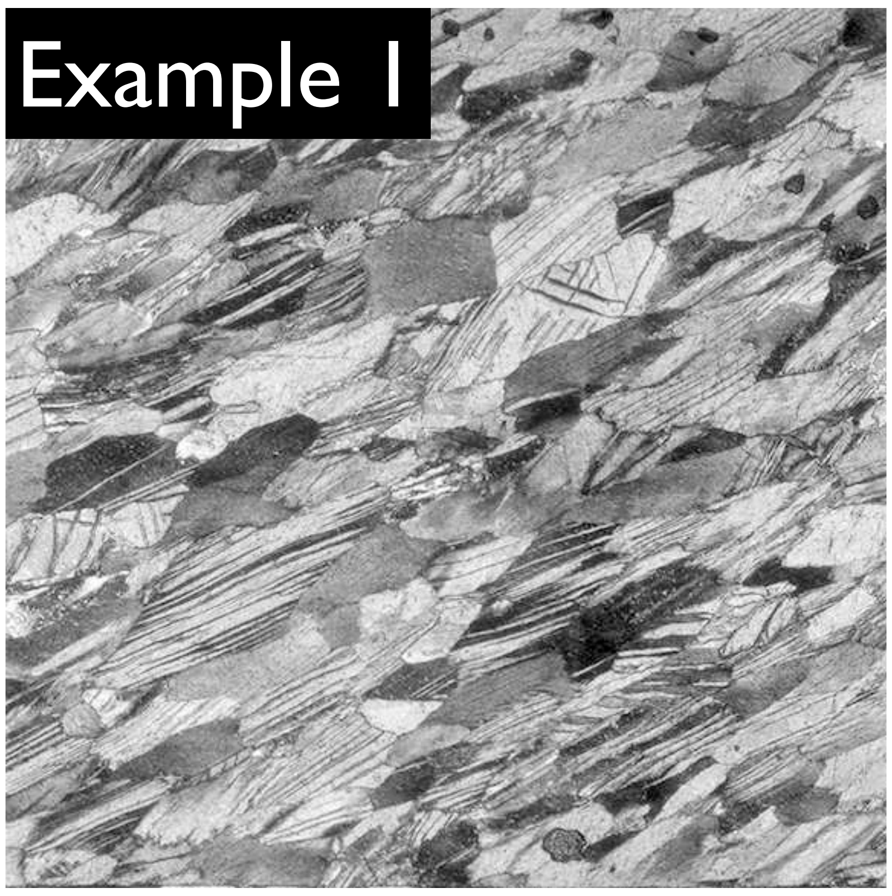
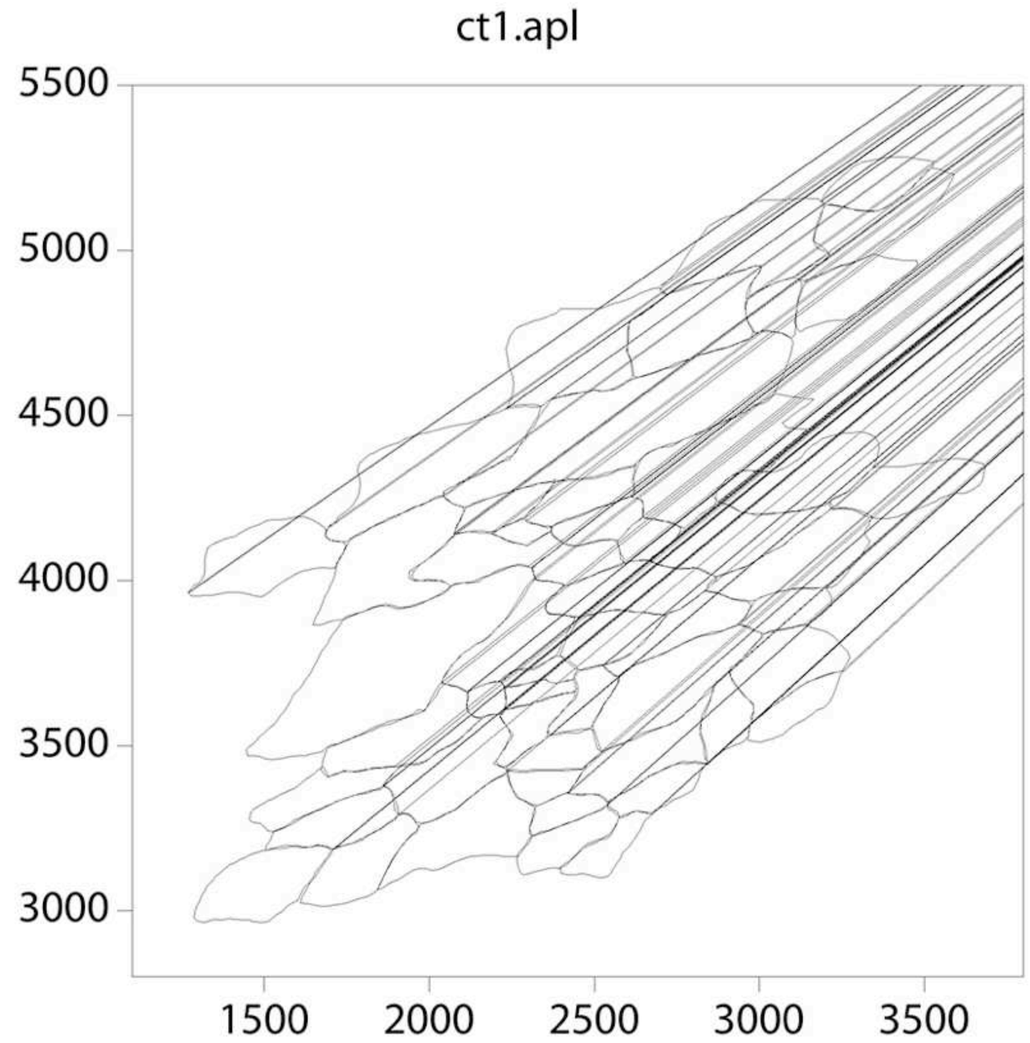
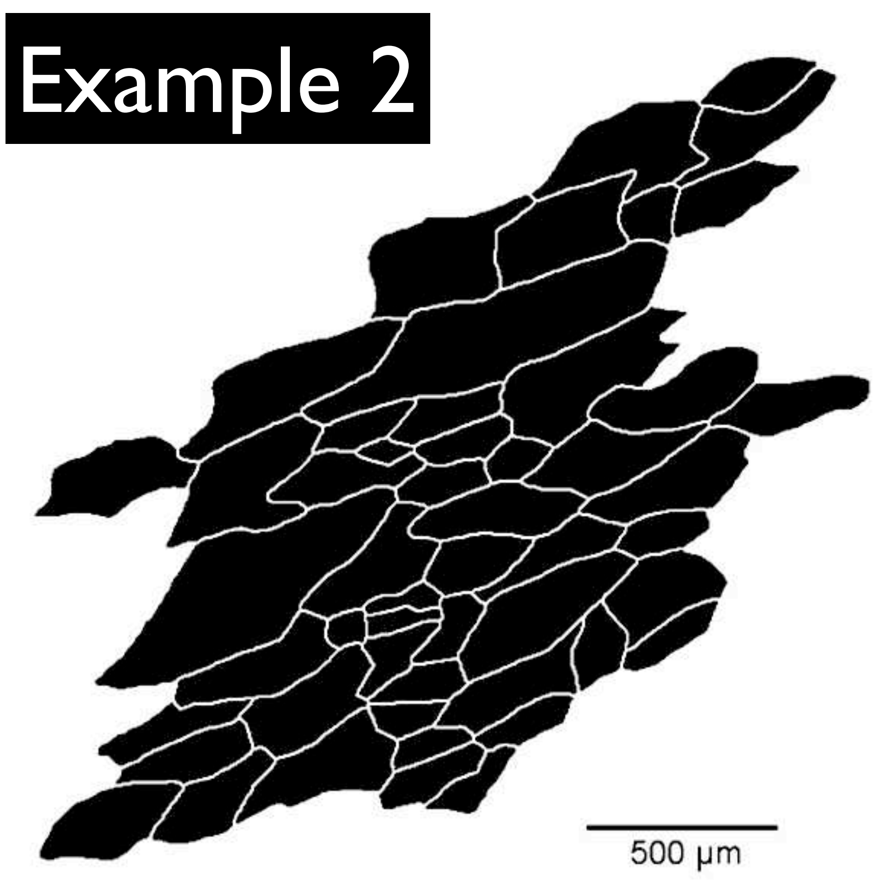


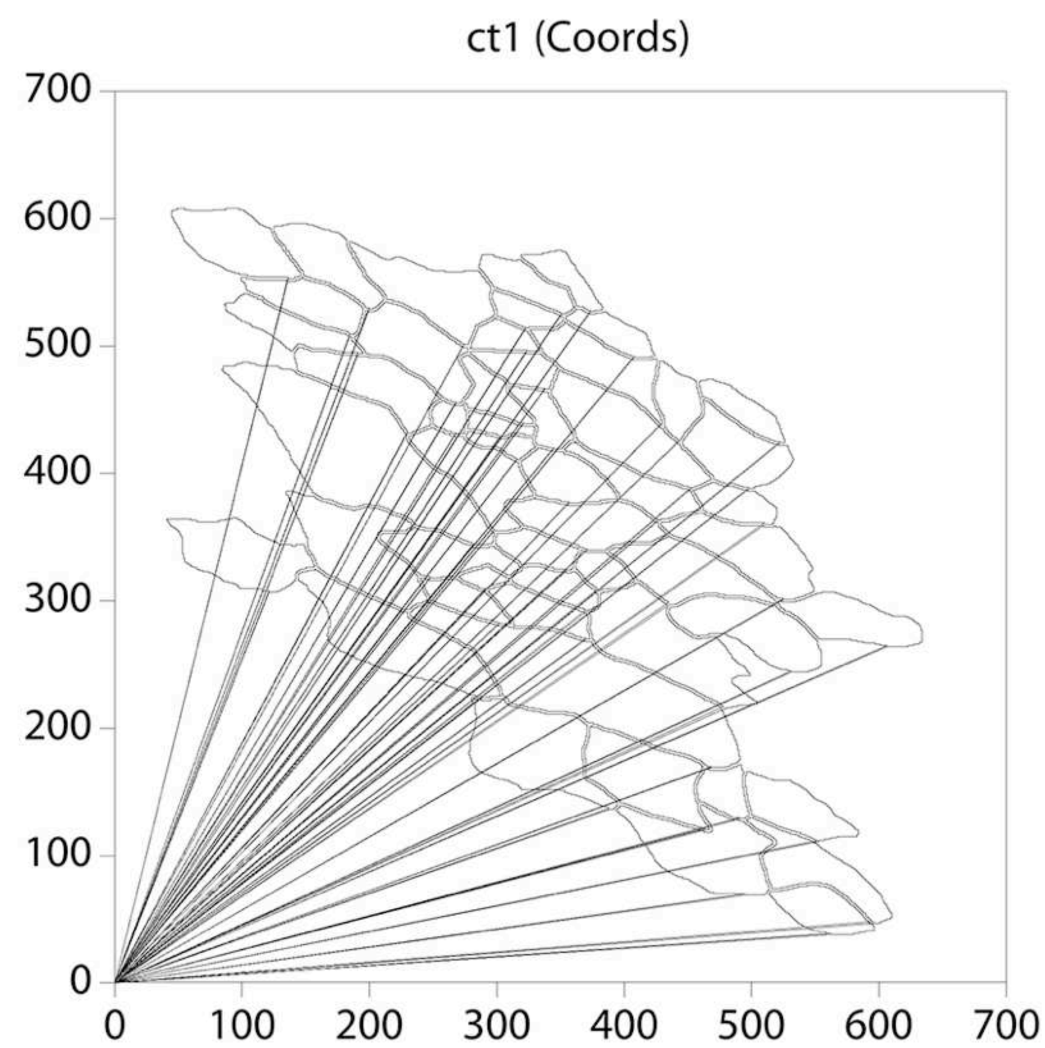
Figure 8.14
Smoothing digitized outlines.
(a) Polygonal chain with 7 integer coordinates (X,Y) defined on digitizing grid;
(b) X and Y coordinates plotted against integrated path length, s;
(c) smoothing spline functions, x(s) and y(s), fitted to discrete values of X and Y; smoothing error = 1; blue dots denote new (continuous) coordinates picked at regular intervals along s;
(d) new coordinates are plotted in x-y plane; integer coordinates of (a) are superposed for comparison.





9999	9999
3189	5153
3198	5168
3207	5180
3216	5192
3228	5204
3240	5216
....	....
3280	5126
3258	5129
3240	5132
3222	5137
3201	5138
3186	5147
9999	9999
9999	9999
3114	4925
3126	4937
3141	4946
....	....





0	0	
562	38	
580	38	
580	39	
585	39	
585	40	
589	40	
...	...	etc.
548	42	
548	41	
553	41	
553	40	
558	40	
558	39	
562	39	
0	0	
596	47	
601	47	
601	48	
...	...	etc.

Software Box 8.1
Input files for SCASMO.
Example 1: Grayscale image, plot of digitized contours, including separator coordinates (9999,9999); total number of points = 3185, magnification: 1.047 µm / pixel.
Example 2: segmented bitmap, plot of boundary coordinates, including separator coordiantes (0,0); total number of points = 9540, magnification: 3.6842 µm / pixel.
Text on gray background: source files for plot = input file for SCASMO.

```
---------------------------------------------------------------
***  scasmo  (full version)**                    2010-10-25, rh
---------------------------------------------------------------
converts digitized files to formatted input files
plus optional: scaling, smoothing, closing of outlines
plus optional: reduction of number of coordinate points
maximum number of points per particle =   4000
particles with less than 3 points are discarded
---------------------------------------------------------------
input file:
   for each particle: X,Y          integer x-y coordinates
   |                  ...          ...etc.
   |                  XE,YE        end coordinate (XE=YE)
output file:
   line 1:            bti          title
   line 2:            n            total number of points
   for each particle: x,y          floating x-y coordinates
   |                  ...          ...etc.
   |                  xe,ye        end coordinates
--------------------------------------------------------------

     name of input file:
1    ct1Coords.txt
     magnification (mm/inch/etc. per pixel):
2    3.6842
     end coordinates of input file (one number):
3    0
     end coordinate of output file (one number):
4    9999
     want to reduce number of digitized points ?
      1=yes, 0=no
5    1
     resolution (resampling between points):
     (1)fine (2)medium (3)coarse (4)manual
6    4
     indicate min.dist (same units as outlines):
7    20
      distmin =    20.000000
     want completed outlines ? (1=yes, 0=no):
8    1
     want smoothing ? 1=yes,0=no :
9    1
     smoothing error (in pixel units):
10   1
     want inverted axes? 0=no; 1=x-axis; 2=y-axis; 3=both :
11   2
     want  spacing... 1=regular, 0=as digitized :
12   1
     name of output file ? [ct1Coords.txt.scm] (return=default):
13
     type header (maximum length = 132 characters):
14   ct1 SXM Coords 1 px smooth
```

Software Box 8.2
Dialog with program SCASMO; answers are numbered and highlighted, see text for explanation.

```
ct1 1 px smooth
      3185
  3322.2859       5367.9067
  3329.9636       5382.5005
  3339.9858       5395.5610
  3350.3250       5408.0444
  3362.5251       5421.1118

  .........       .........

  3435.6853       5334.0098
  3416.6008       5338.5054
  3393.9631       5342.9556
  3375.1279       5346.1025
  3355.2957       5349.2959
  3333.4331       5354.2905
  3322.5149       5363.1323
  3322.2859       5367.9067
  9999.0000       9999.0000
  3246.3596       5128.7568
  3255.3682       5141.4678

  .........       .........
```

```
ct1 SXM Coords 1 px smoo
     1894
  2071.7583      -142.64511
  2101.1436      -140.12206
  2129.5110      -140.71396
  2155.8850      -144.78642
  2176.1084      -152.86475

  .........       .........

  1913.2555      -233.74568
  1929.3328      -215.58563
  1949.5594      -197.32109
  1971.2324      -180.48318
  1994.3207      -166.30144
  2018.8632      -155.35596
  2044.5809      -147.70366
  2071.7583      -142.64511
  9999.0000       9999.0000
  2199.2988      -181.43900
  2220.1973      -180.86613

  .........       .........
```

Software Box 8.3
Output files of SCASMO.

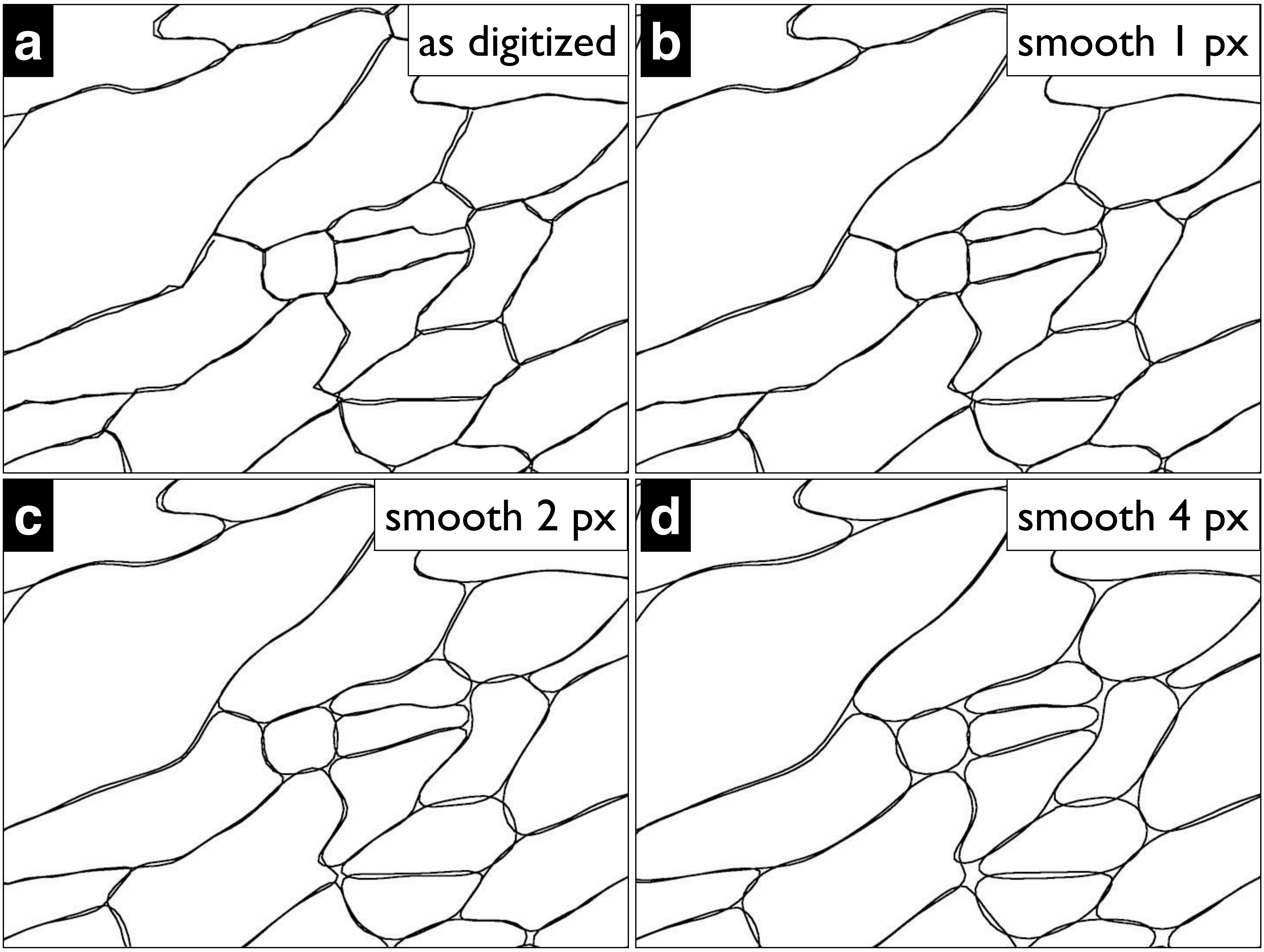


Figure 8.15
Effect of smoothing outlines.
(a) Outlines as digitized;
(b) outlines closed and smoothed with a smoothing error of 1 pixel;
(c) same as (b) with a smoothing error of 2 pixels;
(d) same as (b) with a smoothing error of 4 pixels.

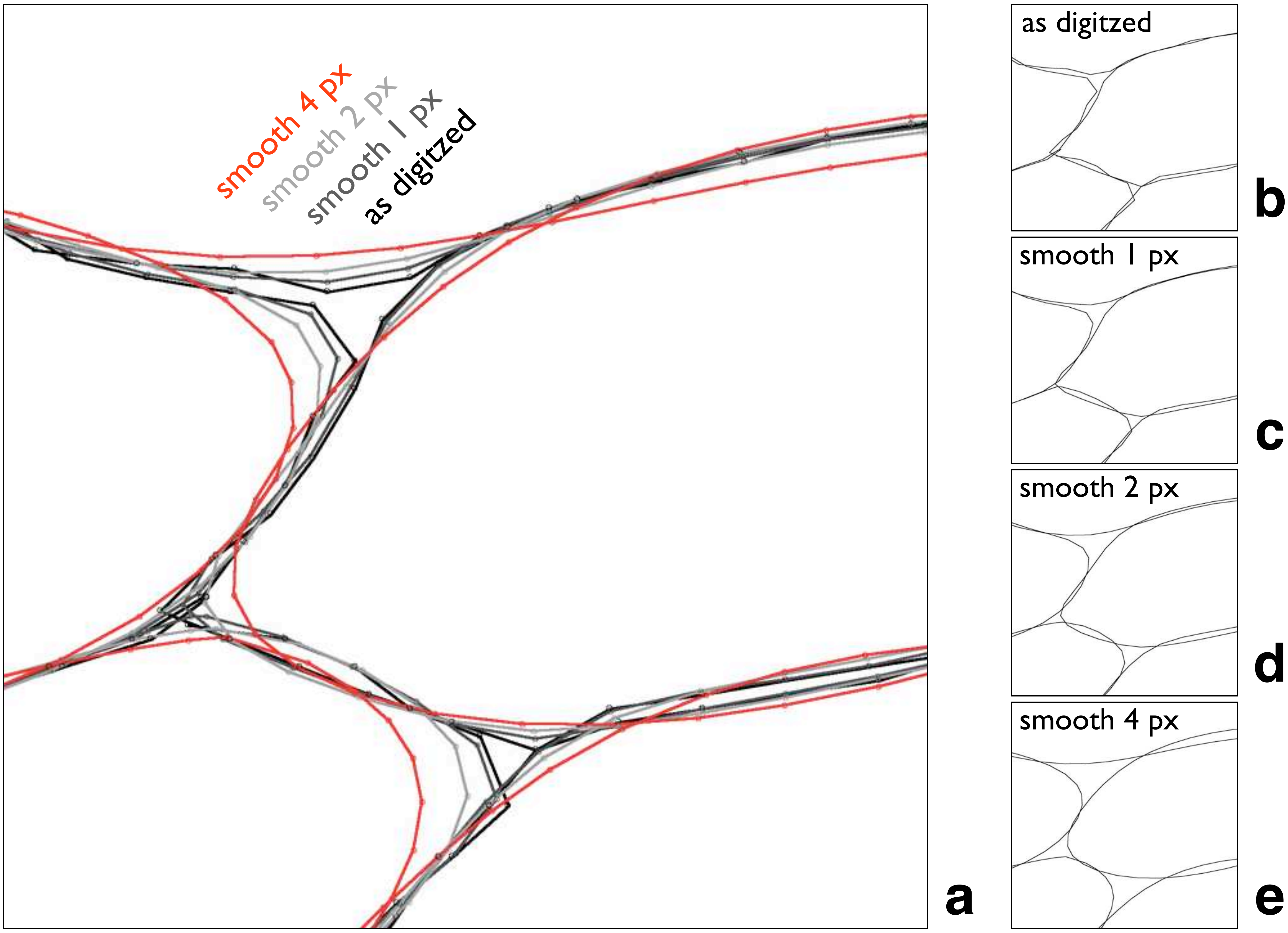


Figure 8.16
Shape distortion through smoothing.
(a) Superposed plots of differently smoothed outlines;
(b) to (e) separate views of outlines shown in (a);
(b) outlines as digitized;
(c) smoothed with a smoothing error of 1 pixel;
(d) smoothed with a smoothing error of 2 pixels;
(e) smoothed with a smoothing error of 4 pixels.

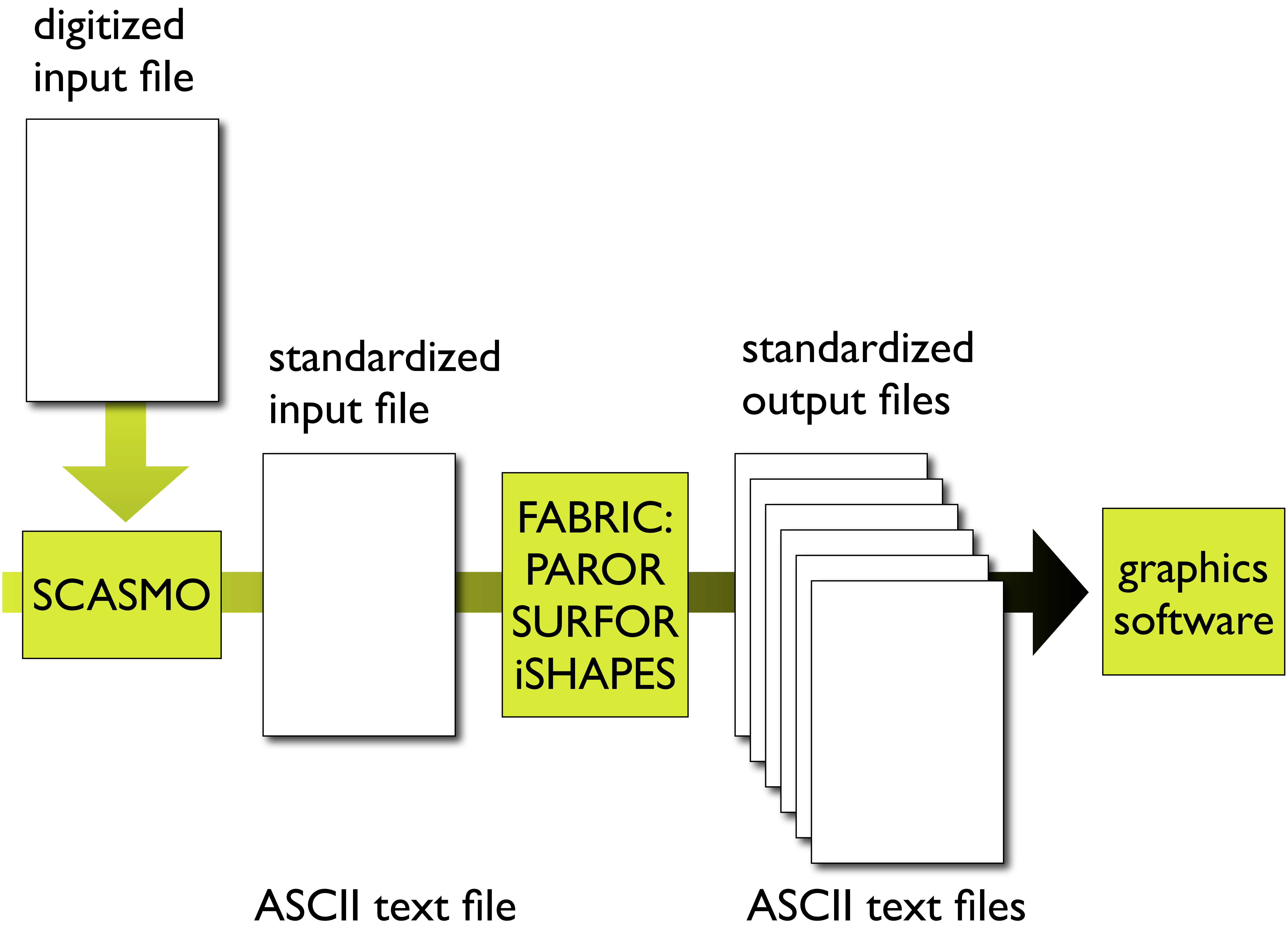


Figure 8.17
Flow chart for the FABRIC package.
The programs are written in Fortran; the format for all input and output files is ASCII text.