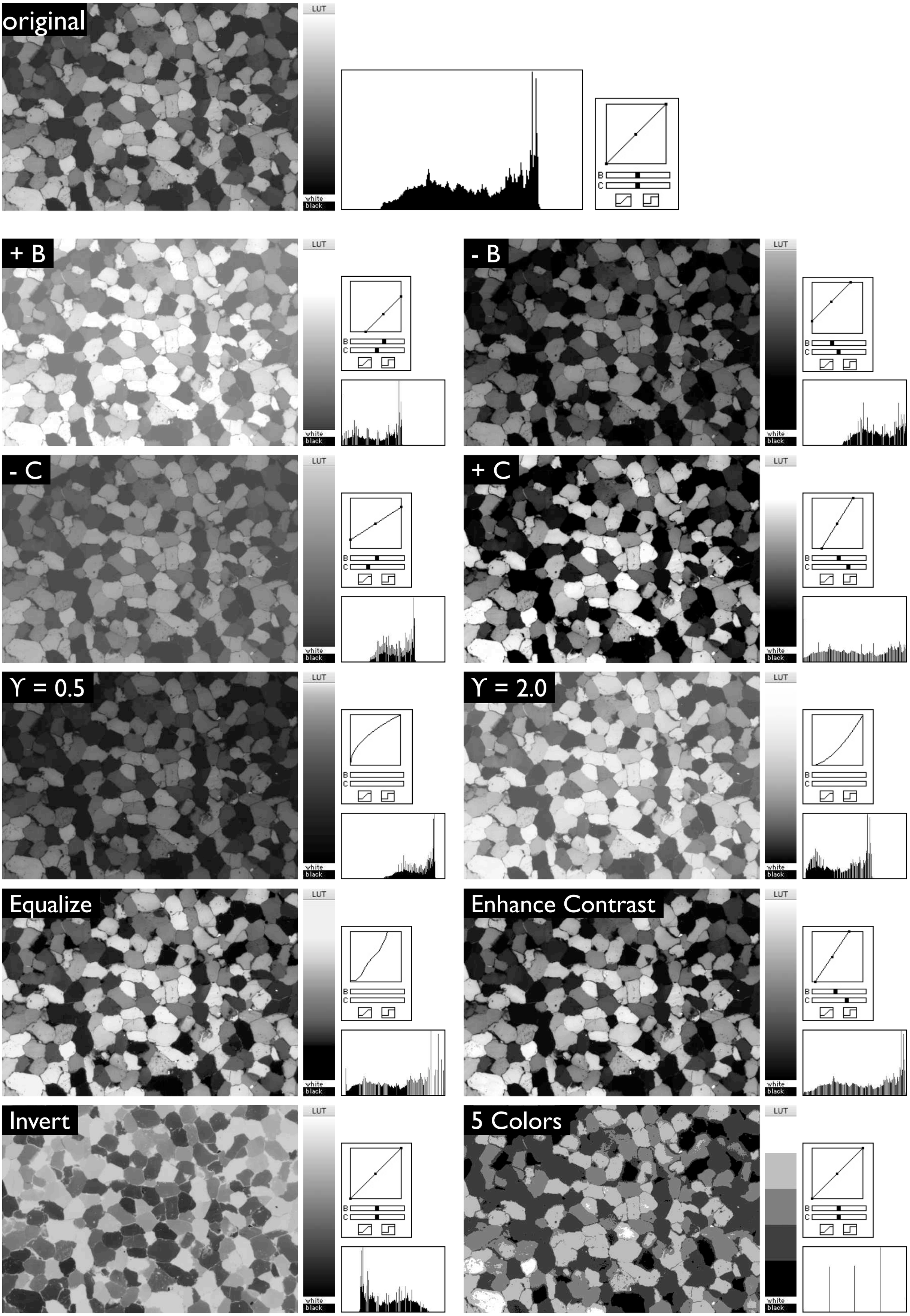


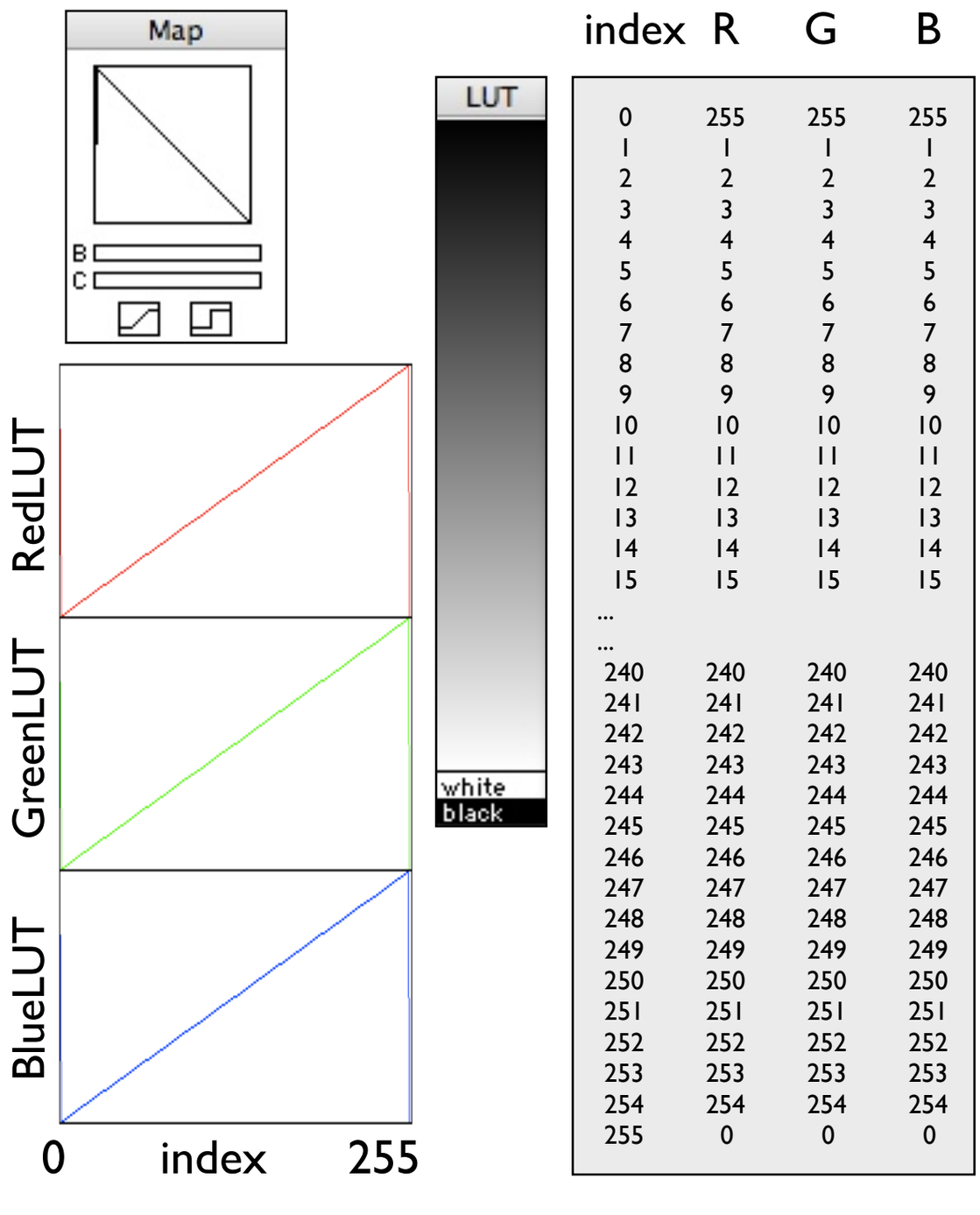
Figure 5.1 Contrast modifications. Left: Original with histogram and Map; (a) Map window: +B, -B: adding or reducing brightness (offset of curve); (b) Map window: -C, +C: decreasing or increasing contrast (slope of curve); (c) Map window (use control key): γ = 0.5, γ = 2.0: two different gamma settings; (d) Process menu: Equalize: using the sum histogram as transfer function, histogram on left; Enhance Contrast: stretching contrast to maximum; (e) Edit Menu: Invert LUT: inverted look-up table; (f) LUT Options (Options menu): 5 Colors: choosing five equally spaced gray levels.

a

b

```
macro '-';

macro 'invert LUT [I]';
var
  i:integer;
begin
  for i:=0 to 255 do begin
    RedLUT[i]:=255-RedLut[i];
    GreenLUT[i]:=255-GreenLut[i];
    BlueLUT[i]:=255-BlueLut[i];
  end;
  UpdateLUT;
end;

macro 'log transform';
var
  i,v:integer;
  scale:real;
begin
  scale := 255.0 / ln(255.0);
  for i:=1 to 254 DO begin
    v := 255-round(ln(i) * scale);
    RedLUT[i]:=v;
    GreenLUT[i]:=v;
    BlueLUT[i]:=v;
  end;
  UpdateLUT;
end;

...
..
```

c



d

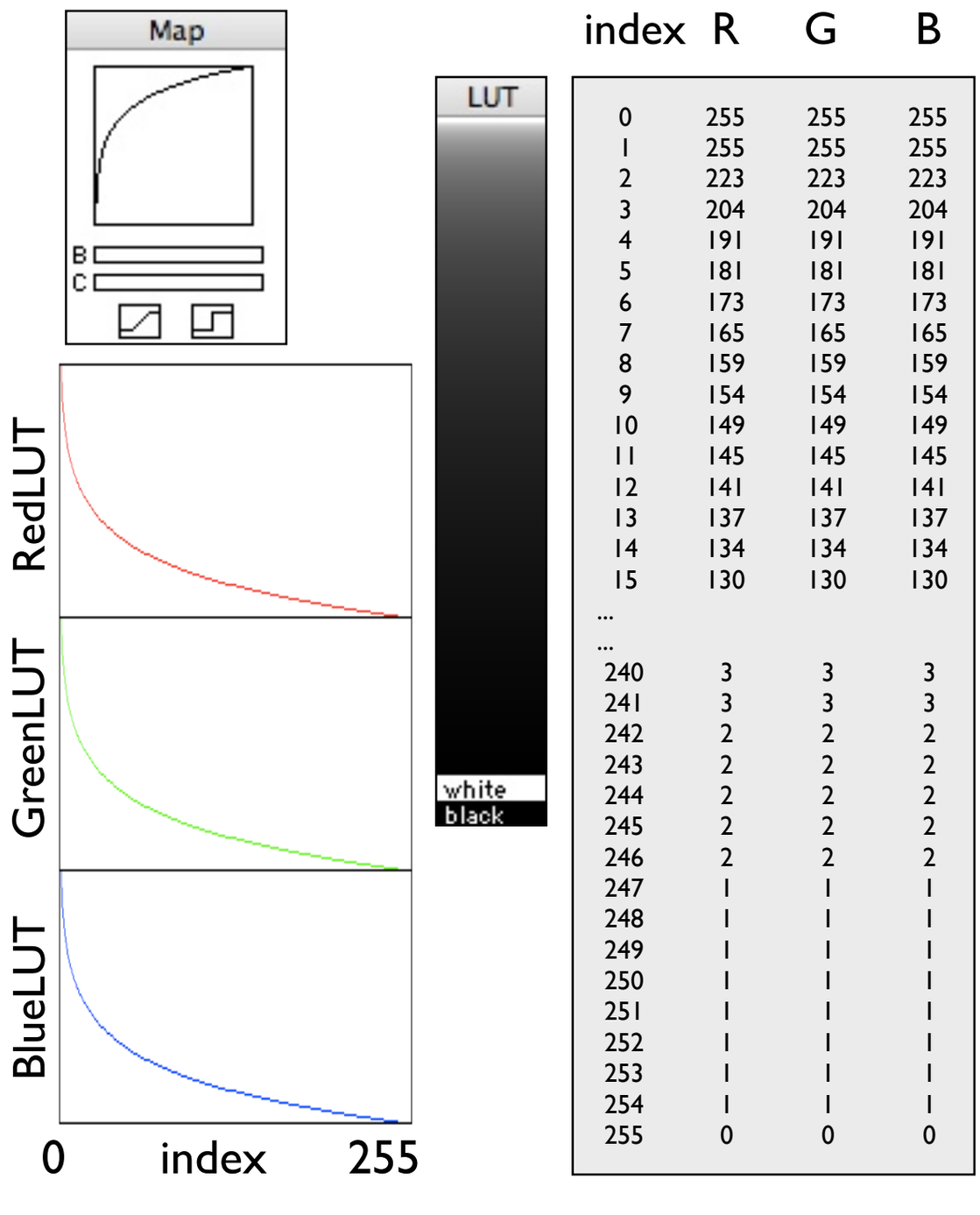
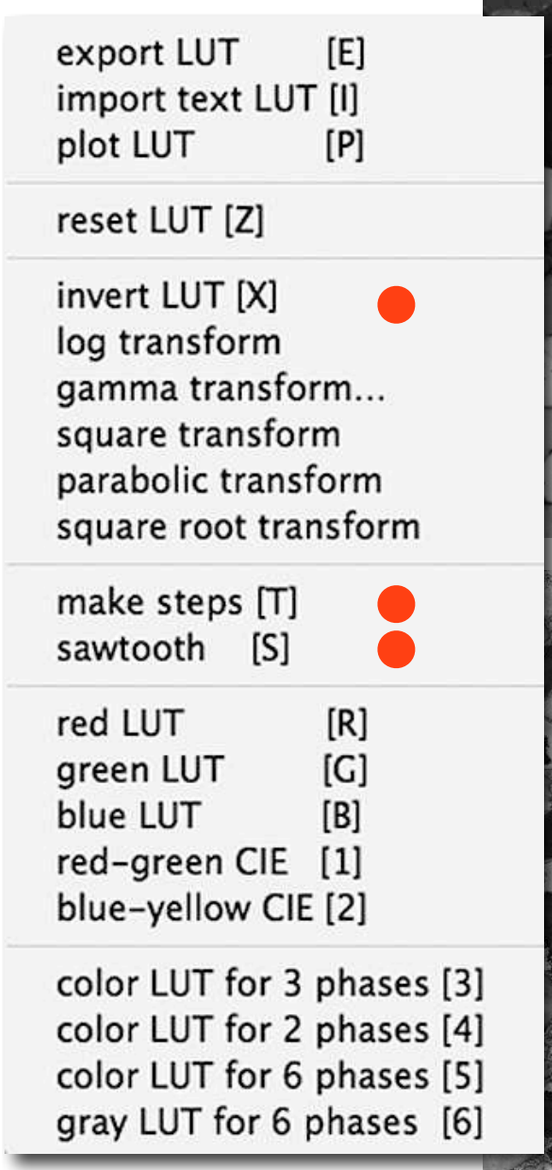
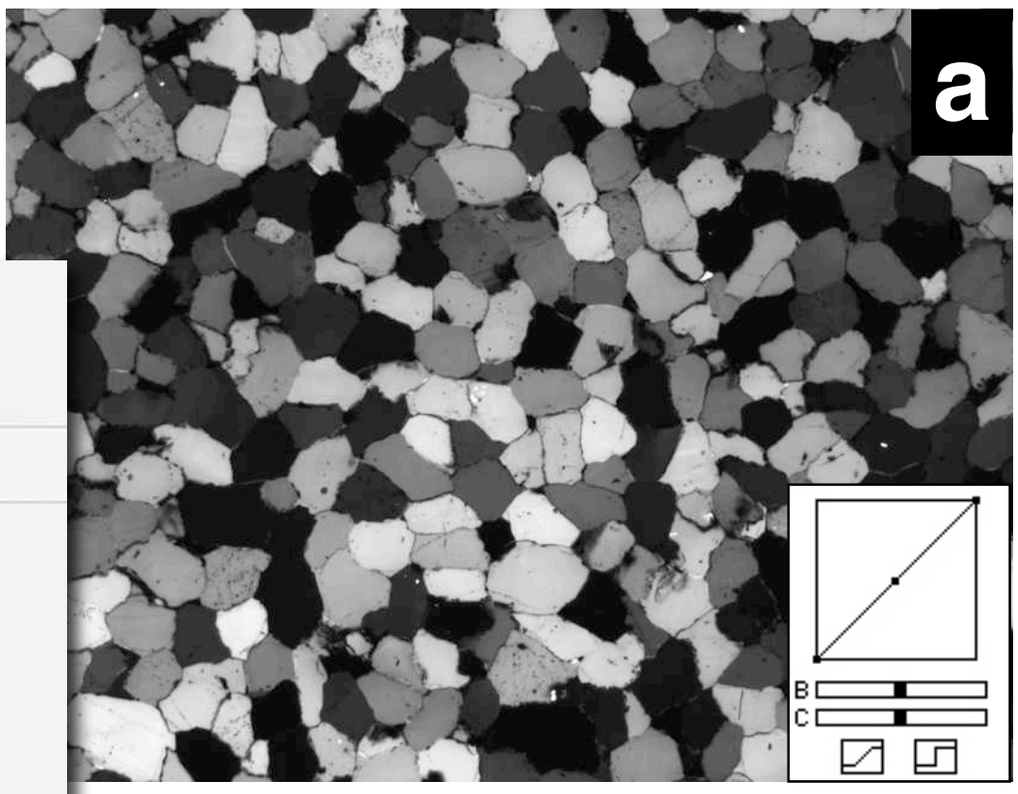
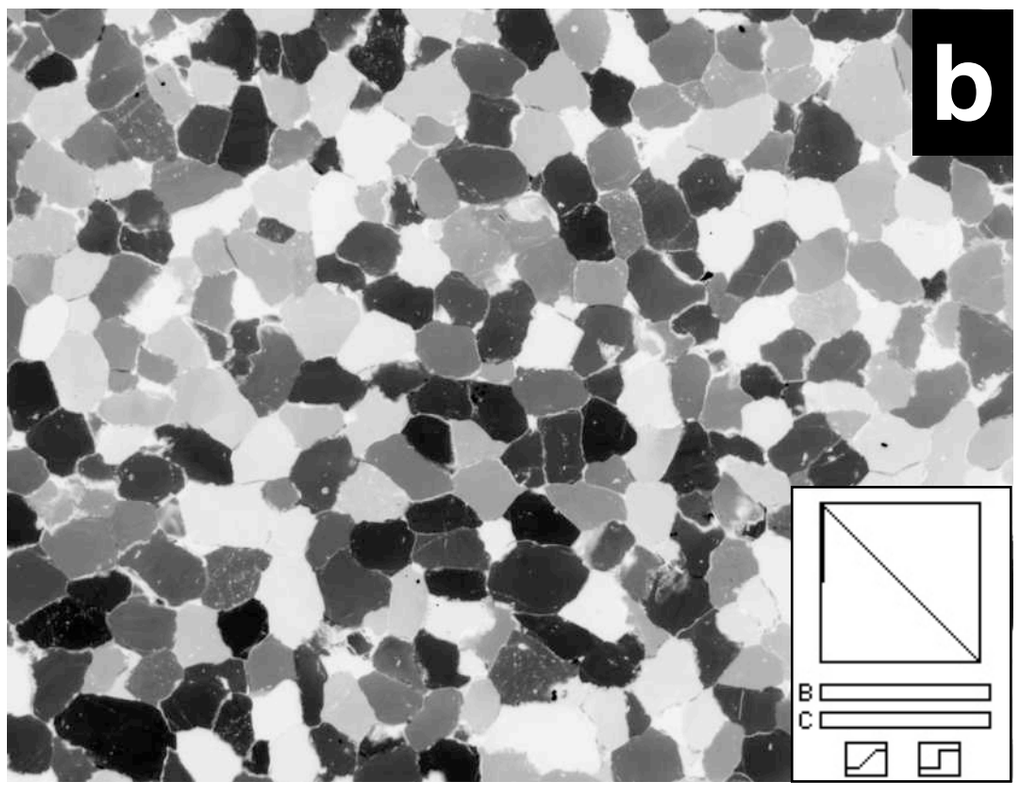
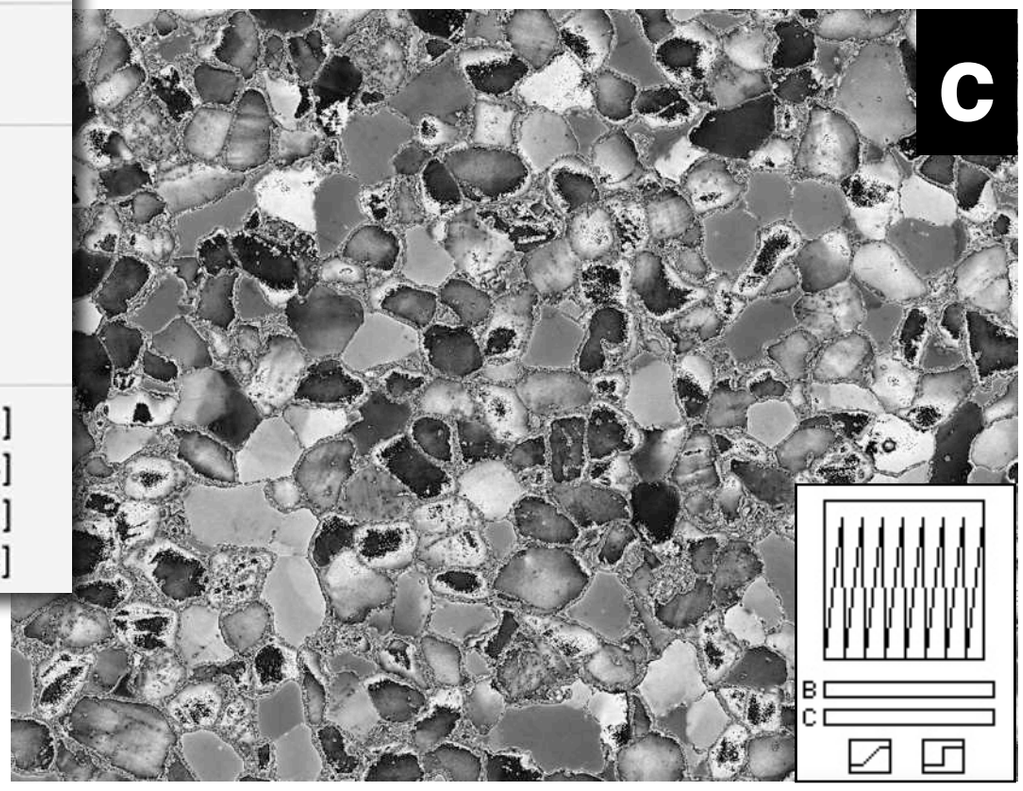


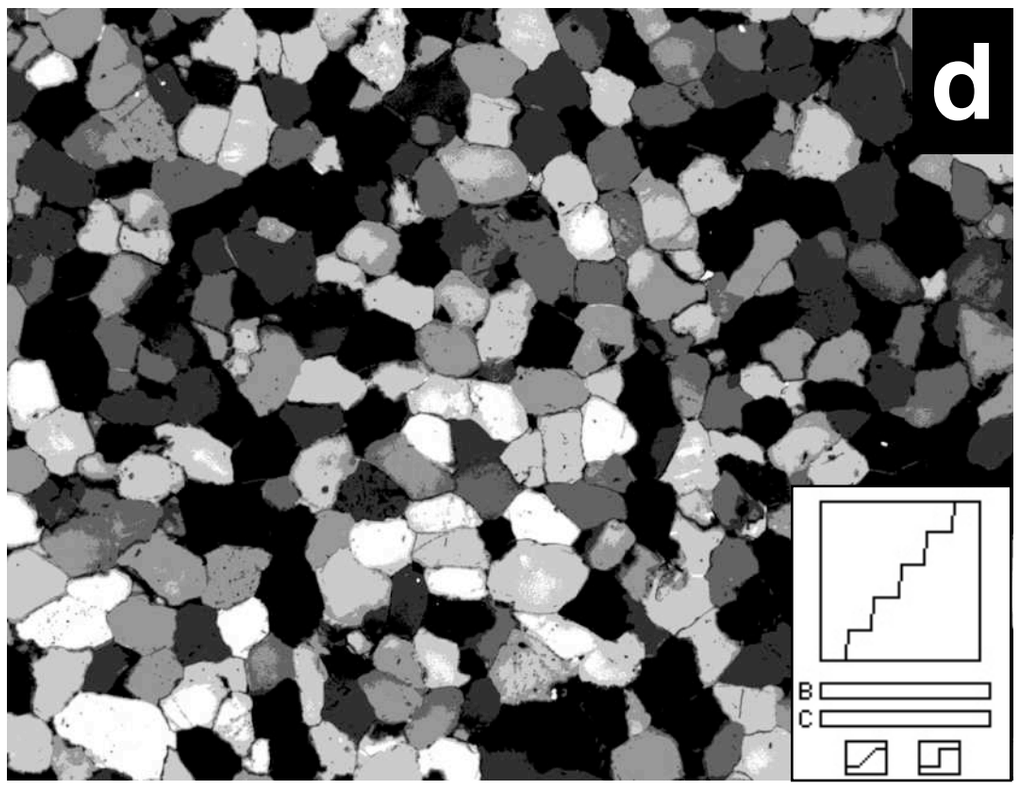
Figure 5.2
Using macros.
(a) Macros menu with Lazy LUTs macro loaded;
(b) Lazy LUTs macro: section of text file describing the 'invert LUT' and the 'log transform' command;
(c) Map window, LUT window, plotted LUT (macro command [P]) and text file of LUT (macro command [E]) to show effect of 'invert LUT' command;
(d) same as (c) for command 'log transform' command.











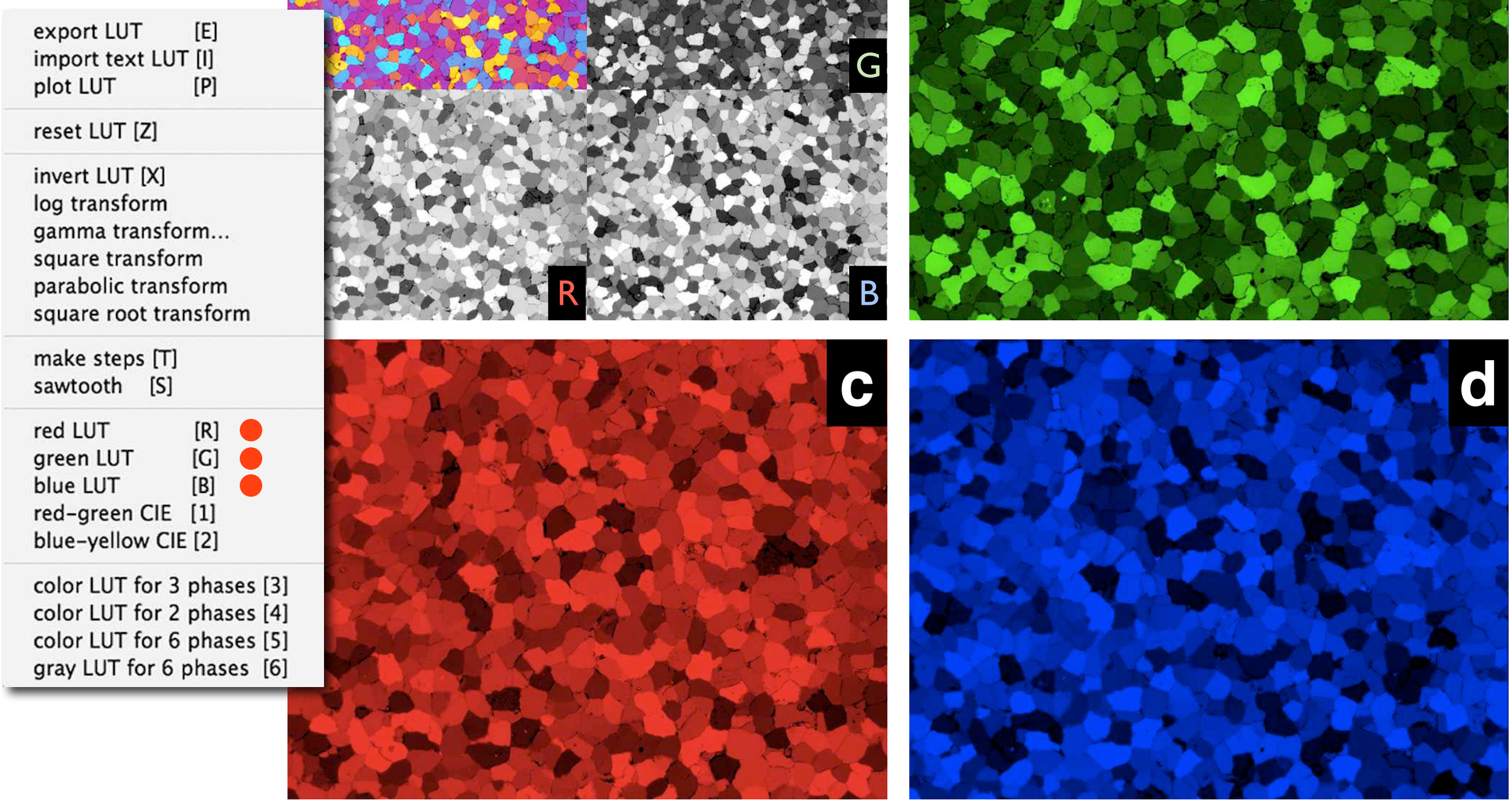
e

```
macro 'make steps [T]';

VAR
  delta,steps,StepSize,NextStep:real;
  level,i:integer;

BEGIN

  steps := GetNumber('no. of steps', 10);

  StepSize:=256/steps;
  delta:=256/(steps-1);
  NextStep:=trunc(StepSize);
  level:=255;

  for i:=0 to 255 do begin
    if i>=NextStep then begin
      NextStep:=trunc(NextStep+StepSize);
      level:=level-delta;
      UpdateLUT;
    end;
    if level<0 then level:=0;
    RedLUT[i]:=level;
    GreenLUT[i]:=level;
    BlueLUT[i]:=level;
  end;

end;
```

f

```
macro 'sawtooth [S]';

VAR
  delta,steps,StepSize,NextStep:real;
  level,i,sigbit,sloplen,nslopes:integer;
  j,istart,iend,grayinc:integer;

BEGIN

  sigbit:= GetNumber
      ('no. of significant bits (1-8)', 4);
  if sigbit = 8 then sloplen:=256;
  if sigbit = 7 then sloplen:=128;
  if sigbit = 6 then sloplen:=64;
  if sigbit = 5 then sloplen:=32;
  if sigbit = 4 then sloplen:=16;
  if sigbit = 3 then sloplen:=8;
  if sigbit = 2 then sloplen:=4;
  if sigbit = 1 then sloplen:=2;

  nslopes:=256/sloplen;
  grayinc:=256/sloplen;

  for j:= 1 to nslopes do begin
    istart:= (j-1)*sloplen;
    for i:=1 to sloplen do begin
    level:=255-((i-1)*grayinc);
    RedLUT[i+istart-1]:=level;
    GreenLUT[i+istart-1]:=level;
    BlueLUT[i+istart-1]:=level;
    end;
  end;

UpdateLUT;

end;
```

Figure 5.3
Macro commands for special transfer functions.
(a) Original with 1:1 Map;
(b) 'invert LUT' command with with Map window;
(c) 'sawtooth' command using 5 significant bits (yields 8 ramps) with Map window;
(d) 'make steps' command for 5 step coloring with Map window;
(e) text file of the 'sawtooth' command ;
(f) text file of the 'make steps' command.



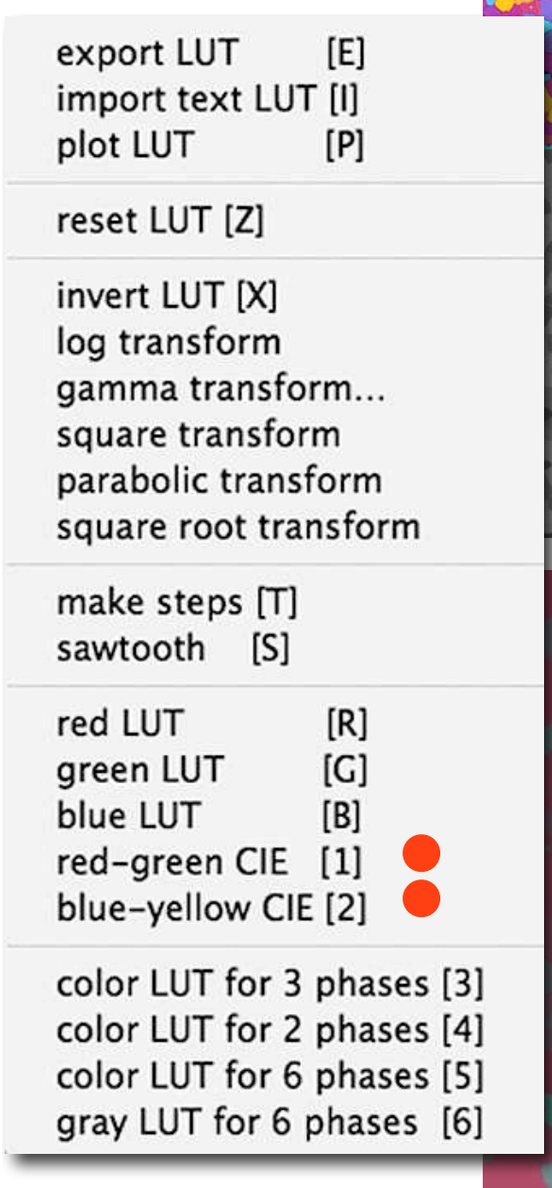
e

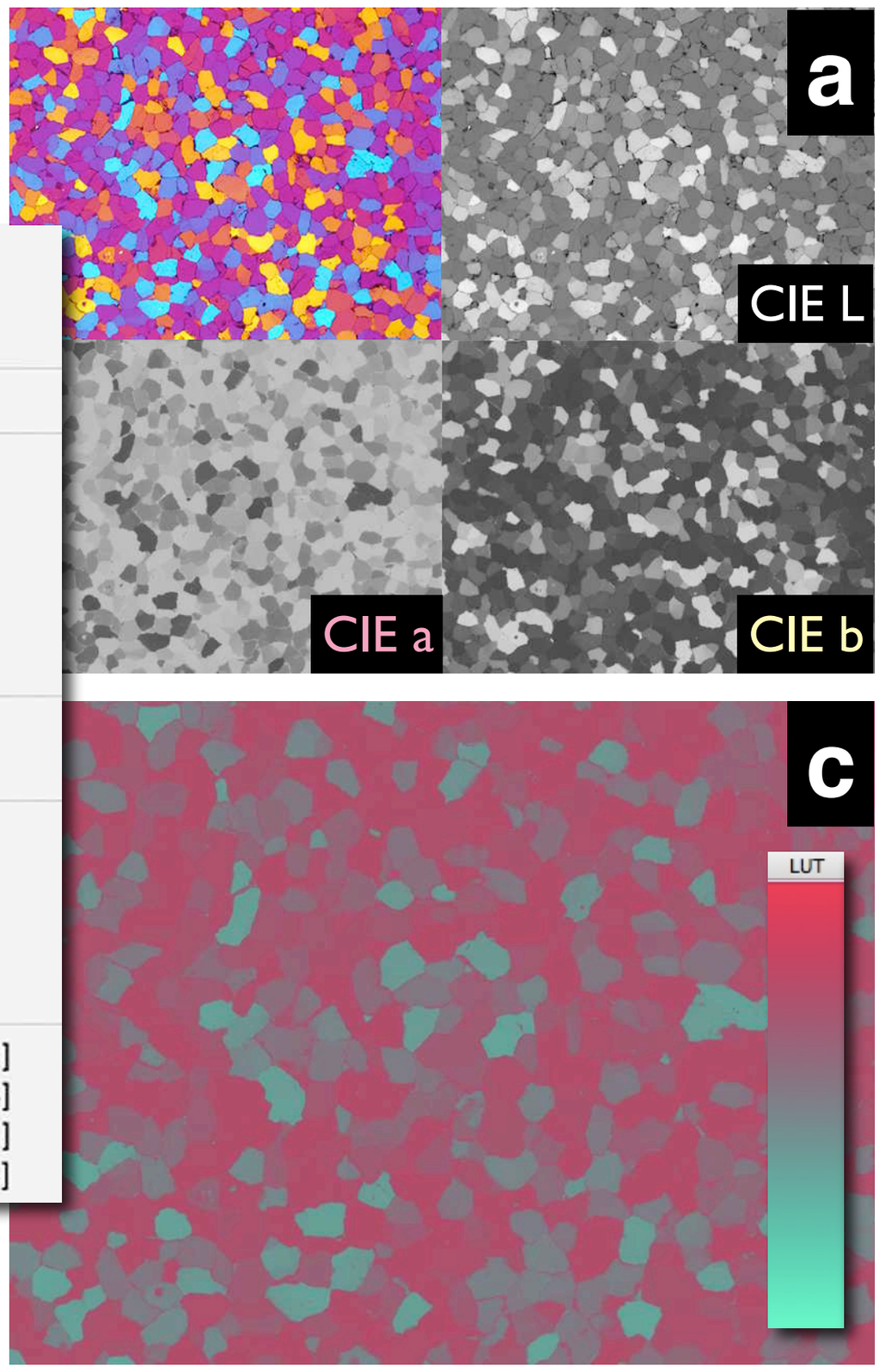
```
macro 'Red LUT [R]';
var
  i:integer;
begin
  for i:=1 to 254 do begin
    RedLUT[i]:=RedLUT[i];
    GreenLUT[i]:=0;
    BlueLUT[i]:=0;
  end;
  UpdateLUT;
end;
```

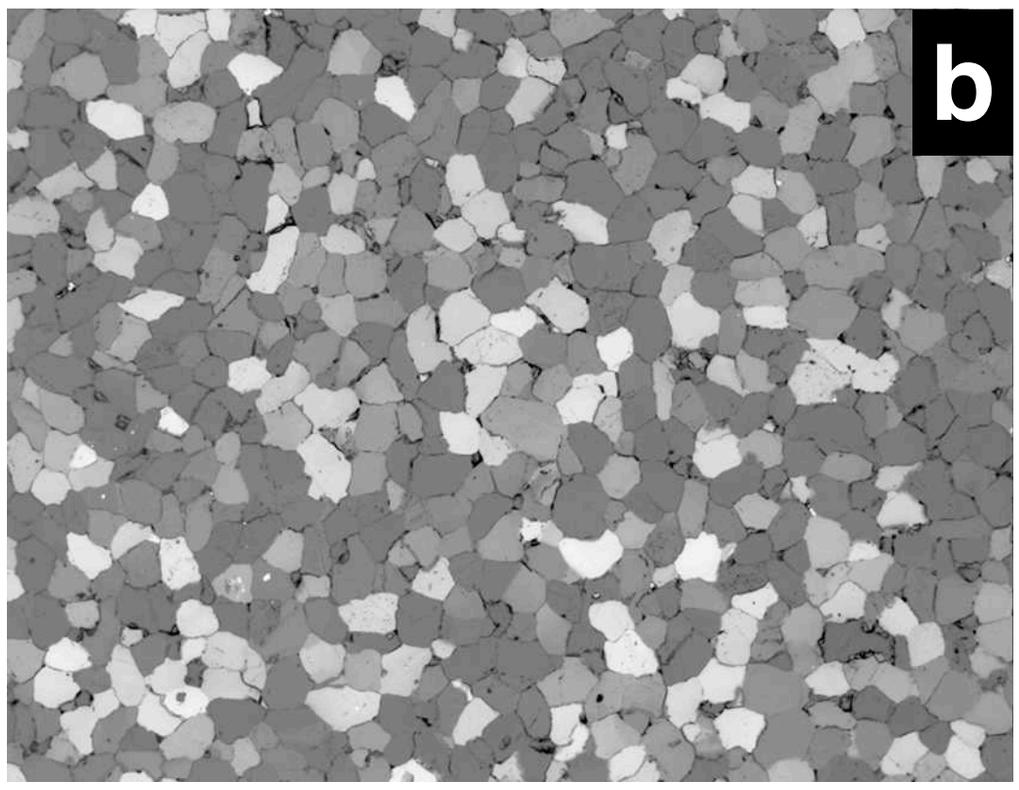
```
macro 'Green LUT [G]';
var
  i:integer;
begin
  for i:=1 to 254 do begin
    RedLUT[i]:=0;
    GreenLUT[i]:=GreenLut[i];
    BlueLUT[i]:=0;
  end;
  UpdateLUT;
end;
```

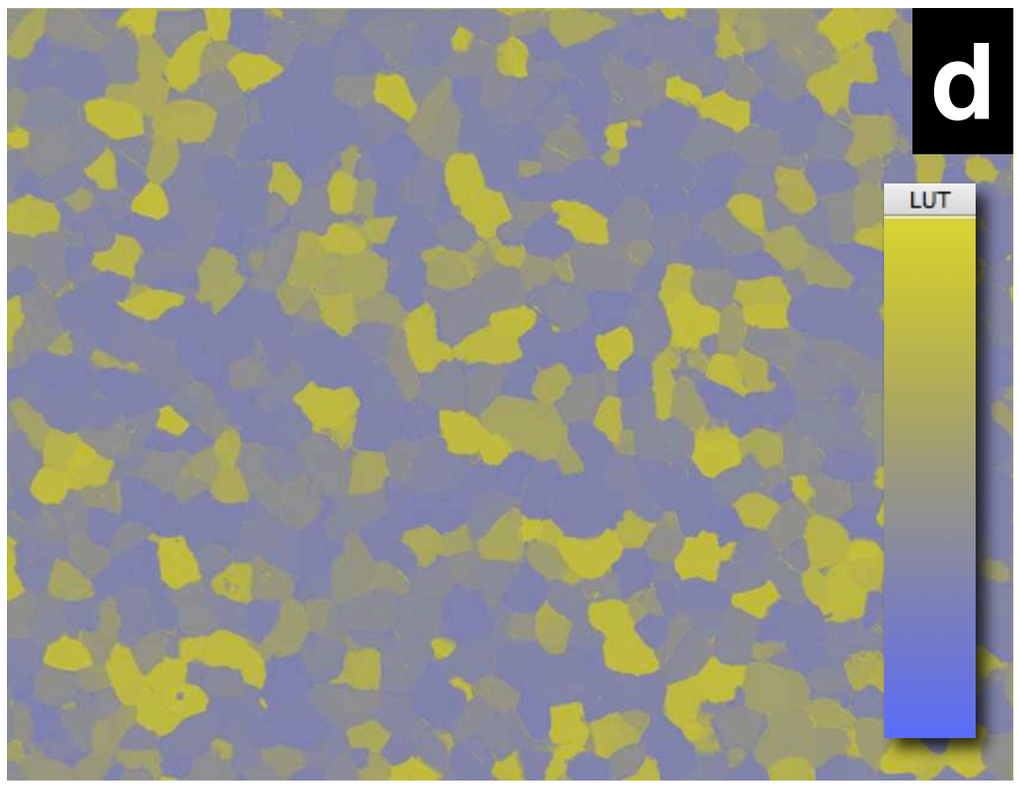
```
macro 'Blue LUT [B]';
var
  i:integer;
begin
  for i:=1 to 254 do begin
    RedLUT[i]:=0;
    GreenLUT[i]:=0;
    BlueLUT[i]:=BlueLut[i];
  end;
  UpdateLUT;
end;
```

Figure 5.4
RGB color channels in color.
(a) Mosaic of color image and its three (monochromatic) channels;
(b) green channel in green;
(c) red channel in red;
(d) blue channel in blue;
(e) text files of 'red LUT', 'green LUT' and 'blue LUT' commands of Lazy LUTs macro.







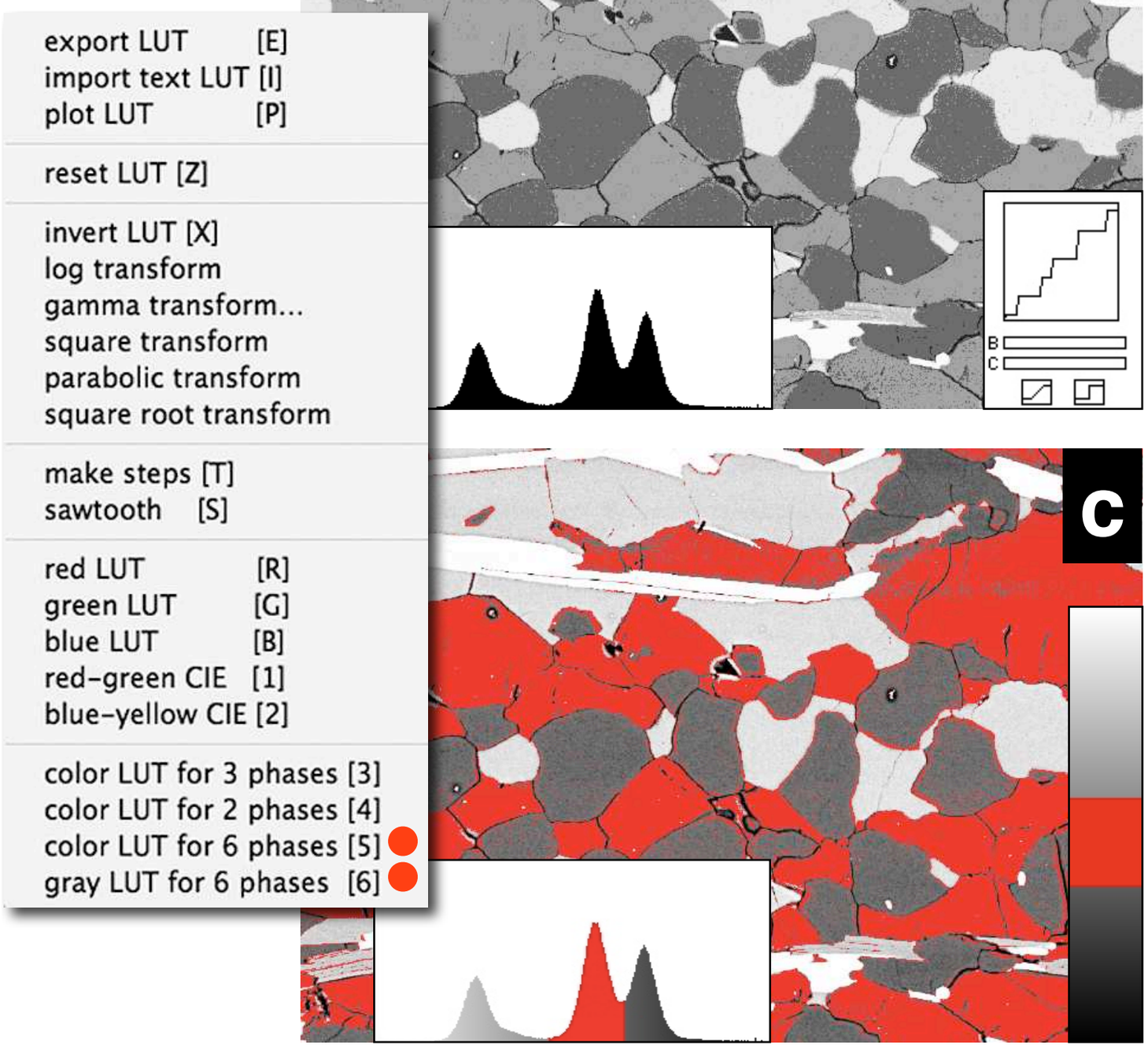


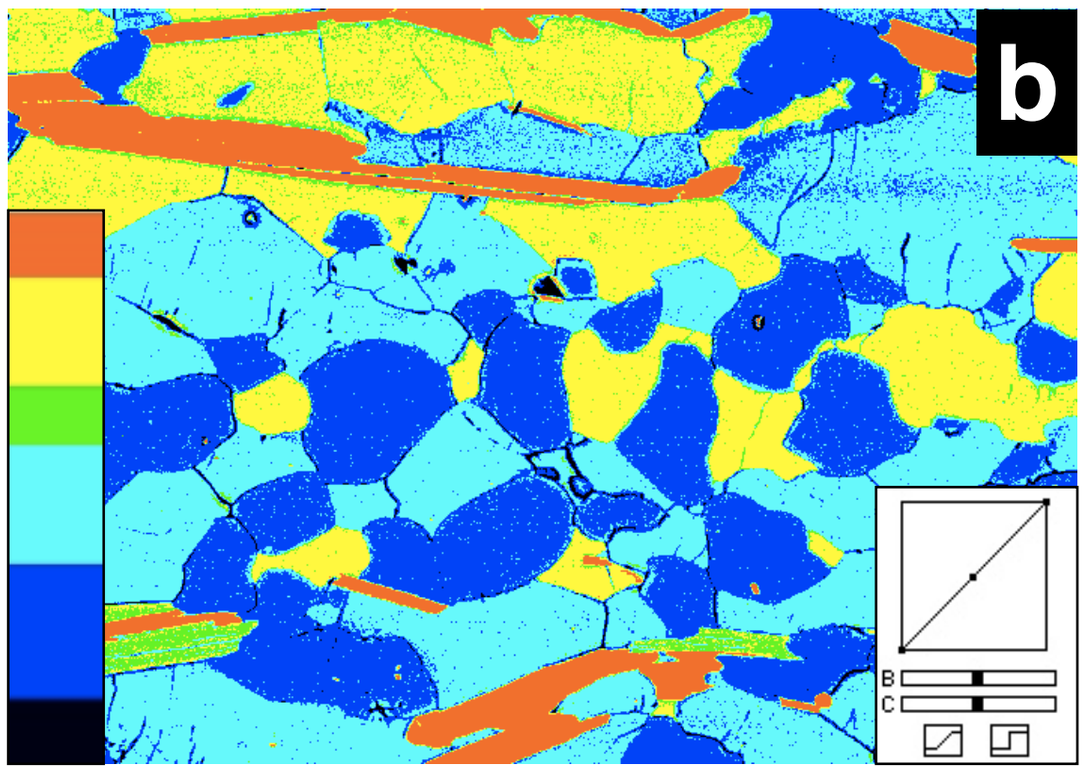
e

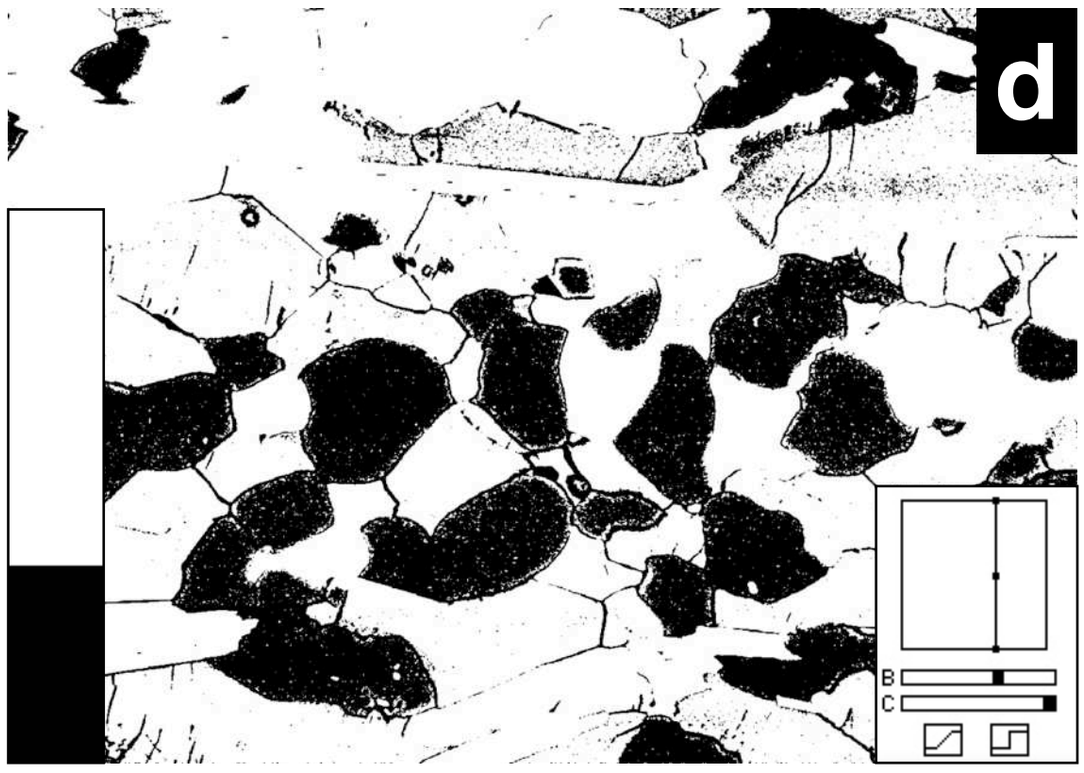
```
macro 'red-green CIE [1]';
var
  i:integer;
begin
  for i:=1 to 254 do begin
    RedLUT[i]:=RedLUT[i];
    GreenLUT[i]:=255-GreenLUT[i];
    BlueLUT[i]:=200-(BlueLut[i]/2);
  end;
  UpdateLUT;
end;
```

```
macro 'blue-yellow CIE [2]';
var
  i:integer;
begin
  for i:=1 to 254 do begin
    RedLUT[i]:=90+(RedLUT[i]/2);
    GreenLUT[i]:=90+(GreenLUT[i]/2);
    BlueLUT[i]:=255-BlueLut[i];
  end;
  UpdateLUT;
end;
```

Figure 5.5
CIE color channels in color.
(a) Mosaic of color image and its three (monochromatic) channels;
(b) L channel in gray;
(c) a channel in red-green (positive = green) with LUT window;
(d) b channel in blue-yellow (positive = yellow) with LUT window;
(e) text files of the 'red-green CIE' and 'blue-yellow CIE' commands of Lazy LUTs macro.







e

```
macro 'Color LUT for 6 phases [5]';

var
  i,ii,iii,iv,v,vi,j,l,k,min,max,mode,gv:integer;
  x,y,w,h,n,mean : integer;

begin
 ShowHistogram;

PutMessage('lower limit 1 is zero, determine
6 upper limits before you start, from upper
limit 6 to 255 will be inverted');

 i:= GetNumber('upper limit 1: ',30);
 ii:= GetNumber('upper limit 2: ',80);
 iii:= GetNumber('upper limit 3: ',107);
 iv:= GetNumber('upper limit 4: ',162);
 v:= GetNumber('upper limit 5: ',200);
 vi:= GetNumber('upper limit 6: ',220);

  for j:=0 to i do begin
    RedLUT[j]:=255;
    GreenLUT[j]:=100;
    BlueLUT[j]:=0;
  end;
  for j:= i+1 to ii do begin
    RedLUT[j]:=255;
    GreenLUT[j]:=255;
    BlueLUT[j]:=0;
  end;
  for j:= ii+1 to iii do begin
    RedLUT[j]:=0;
    GreenLUT[j]:=255;
    BlueLUT[j]:=0;
  end;
  for j:= iii+1 to iv do begin
    RedLUT[j]:=0;
    GreenLUT[j]:=255;
    BlueLUT[j]:=255;
  end;
  for j:= iv+1 to v do begin
    RedLUT[j]:=0;
    GreenLUT[j]:=0;
    BlueLUT[j]:=255;
  end;
  for j:= v+1 to vi do begin
    RedLUT[j]:=0;
    GreenLUT[j]:=0;
    BlueLUT[j]:=20;
  end;
  for j:= vi+1 to 255 do begin
    RedLUT[j]:=255-RedLUT[j];
    GreenLUT[j]:=255-RedLUT[j];
    BlueLUT[j]:=255-RedLUT[j];
  end;
 UpdateLUT;
end;
```

Figure 5.6
Macro commands for segmentation by point operations.
(a) 'Gray LUT for 6 phases': selecting 6 gray levels for 5 phases and rest (grain boundaries, holes, etc.), histogram and Map are shown;
(b) 'color LUT for 6 phases': selecting 6 colors for 5 phases and rest, LUT and Map are shown;
(c) 'Density Slice' command (Options menu): gray level slicing, histogram and LUT are shown;
(d) 'Threshold' command (Options menu): gray values > threshold level, LUT and Map are shown;
(e) text file of 'color LUT for 6 phases' command of Lazy LUTs macro.

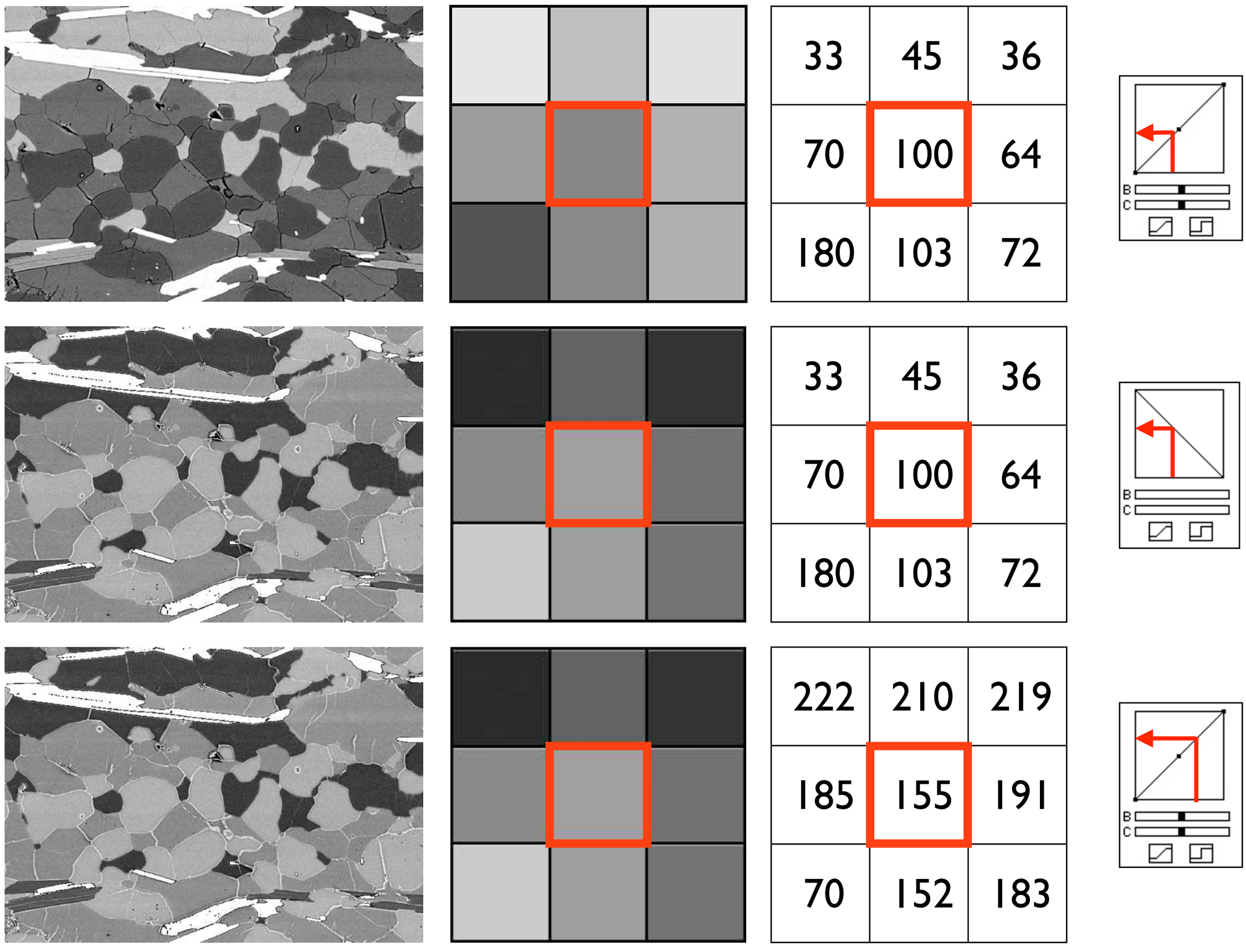


Figure 5.7
Look-up table and point operation.
From left to right: image; 3 · 3 pixel detail; corresponding 3 · 3 matrix; Map with red arrow representing highlighted pixel.
(a) 1:1 LUT;
(b) inverted LUT, values in image matrix are unchanged;
(c) inverted LUT 'applied' to image, pixe values are changed, LUT is 1:1 again.

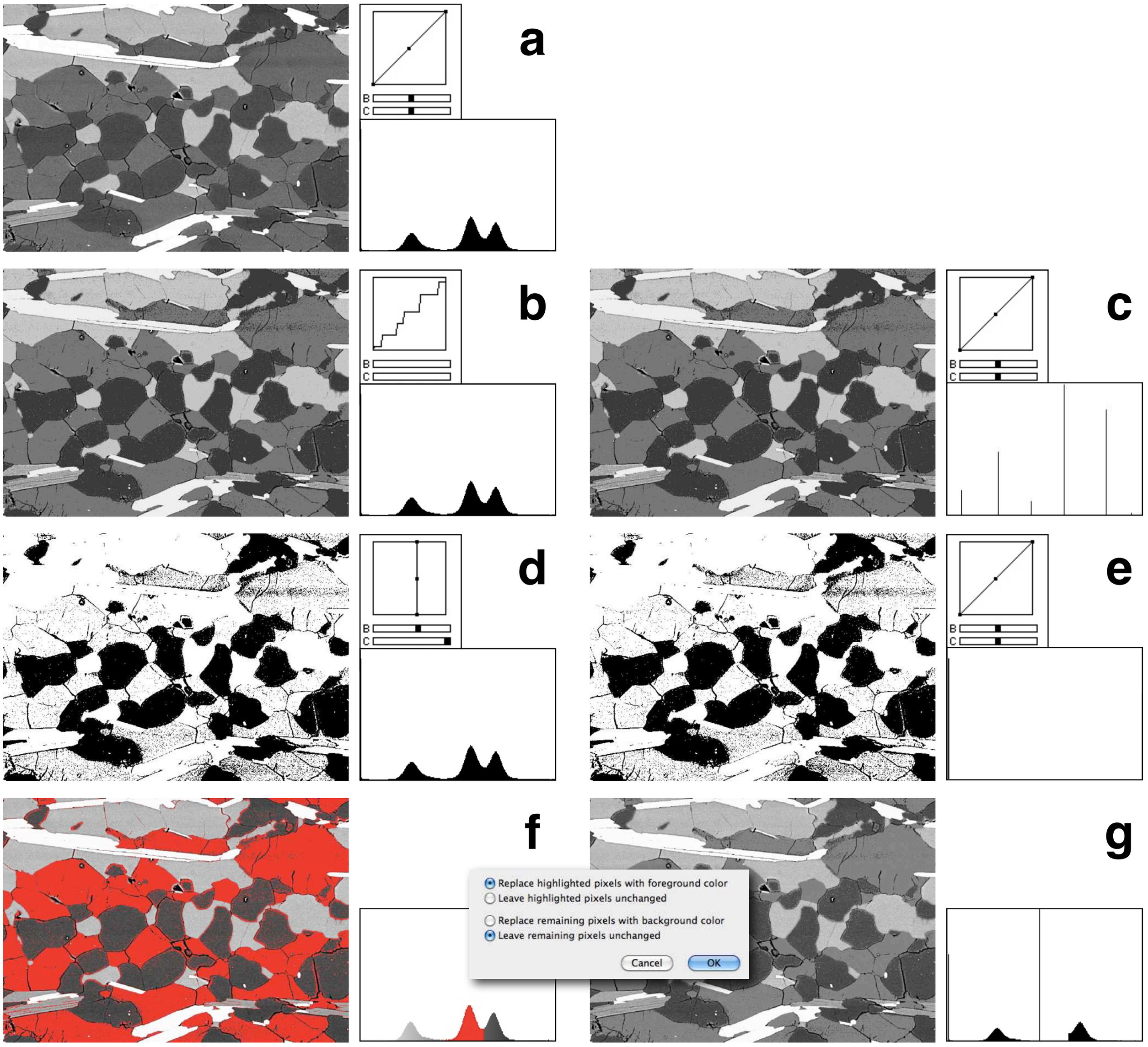
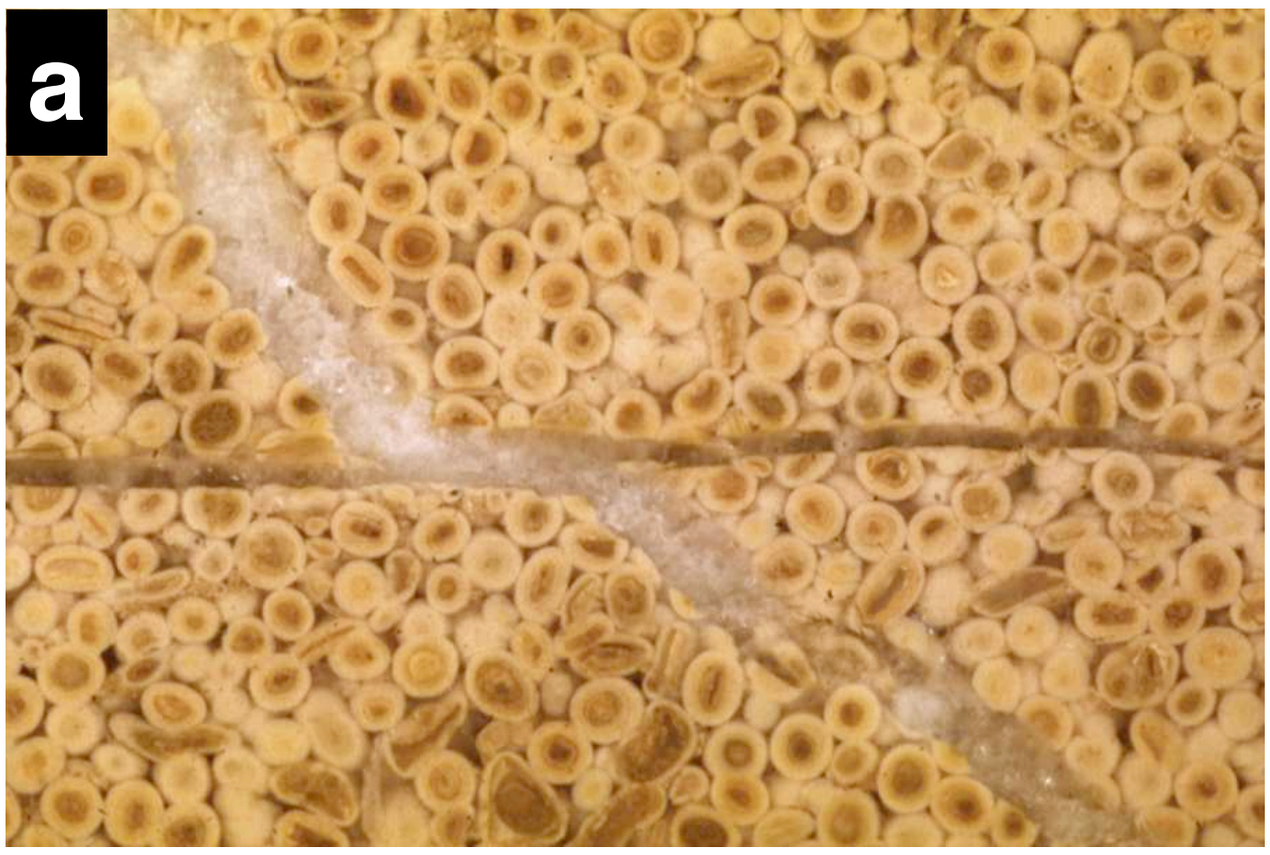
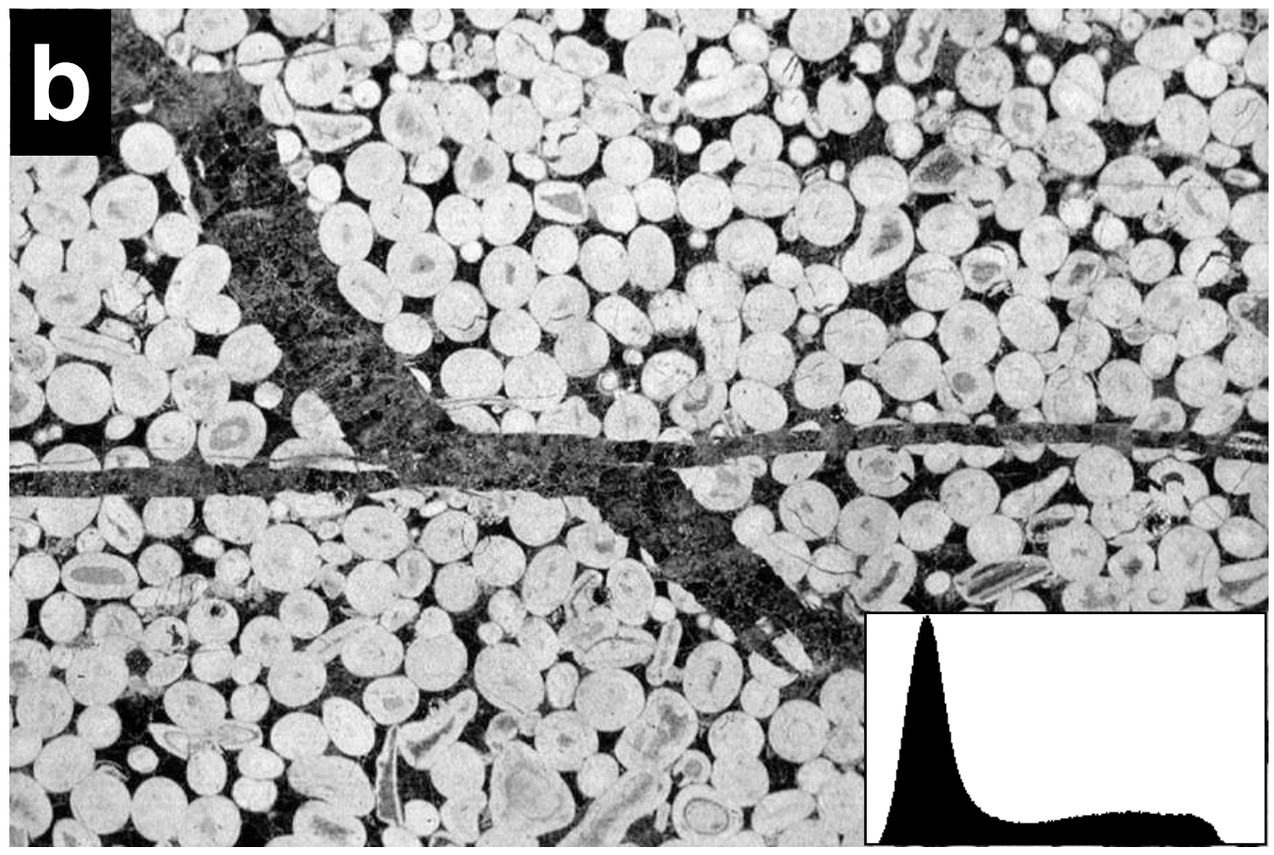
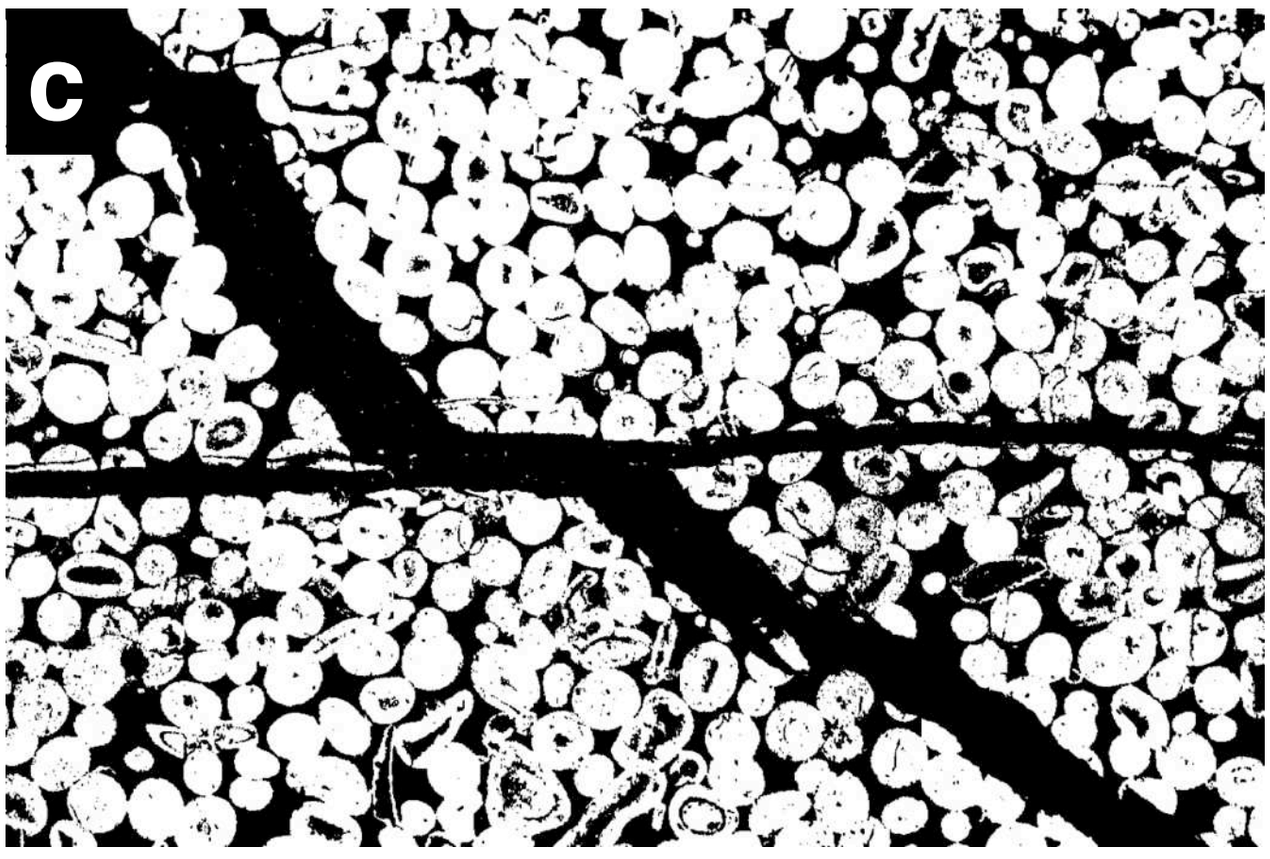
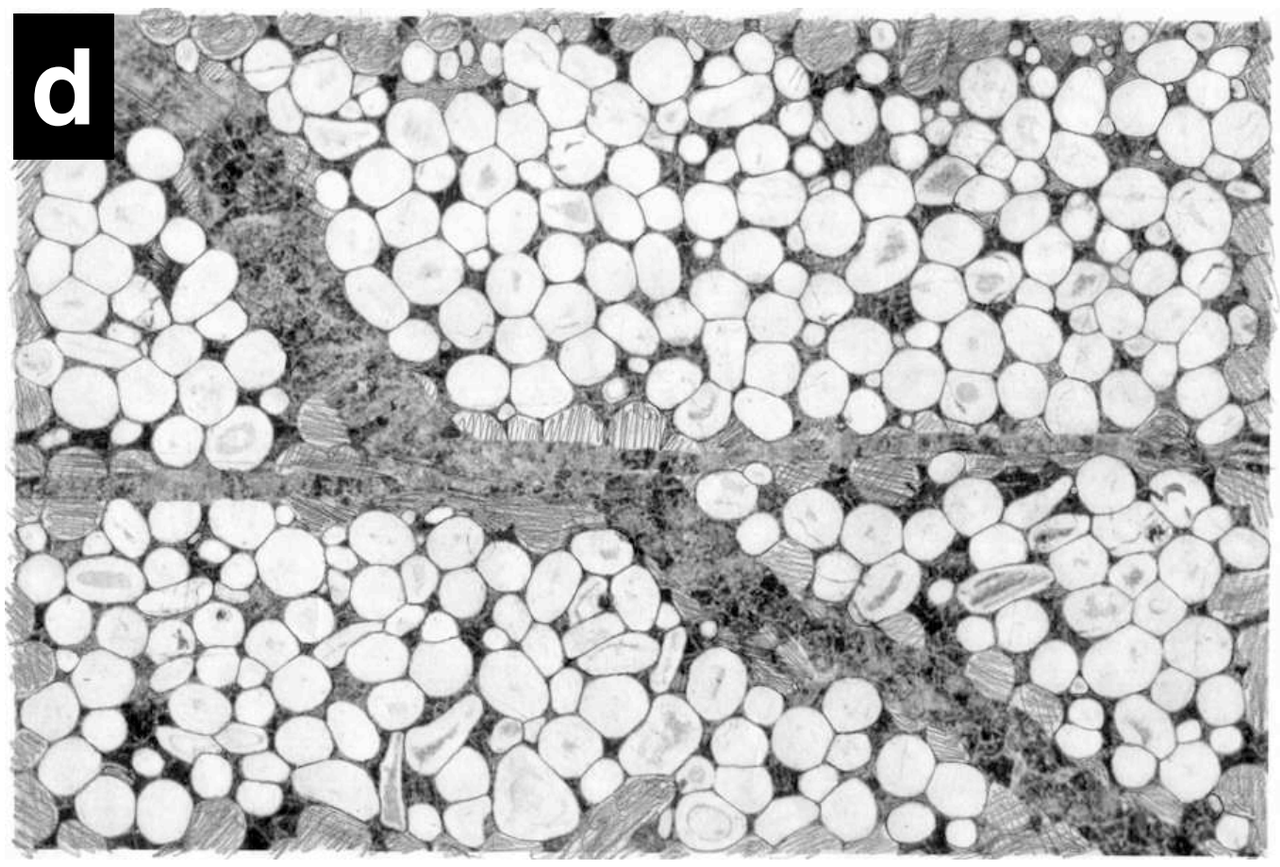


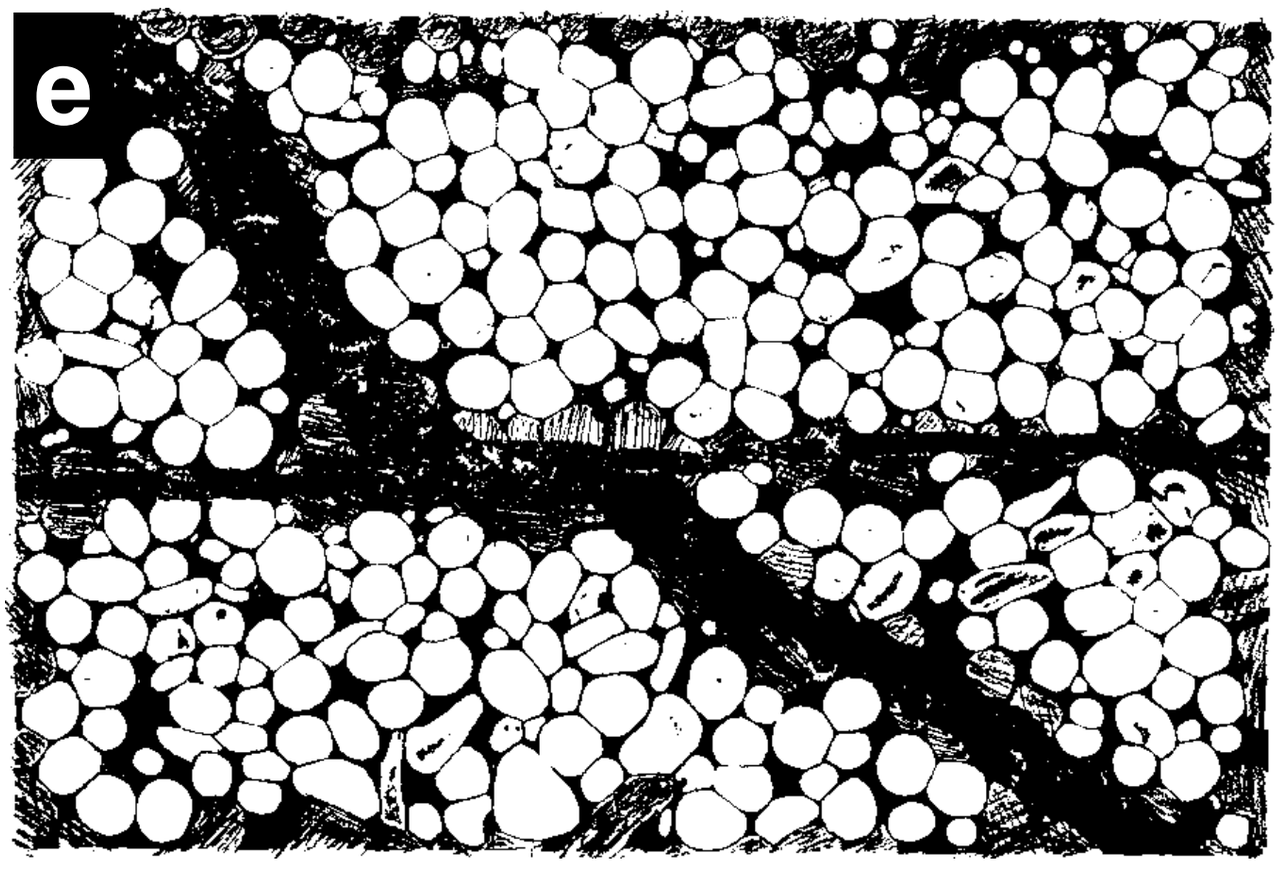
Figure 5.8
Applying look-up tables.
(a) 1:1 LUT, from left to right: image, histogram, Map;
(b) LUT with tone separation (compare Figure 5.8.a), image and histogram are unchanged;
(c) LUT 'applied' to image, image and histogram are changed, LUT is 1:1 again;
(d) threshold is set, image and histogram are unchanged;
(e) after 'Make Binary' command (Process > Binary menu), image and histogram are changed, LUT is 1:1 again;
(f) 'Density Slice' command (Options menu);
(g) with Foreground Color = 120, 'Apply LUT' command using setting shown in window, note histogram.











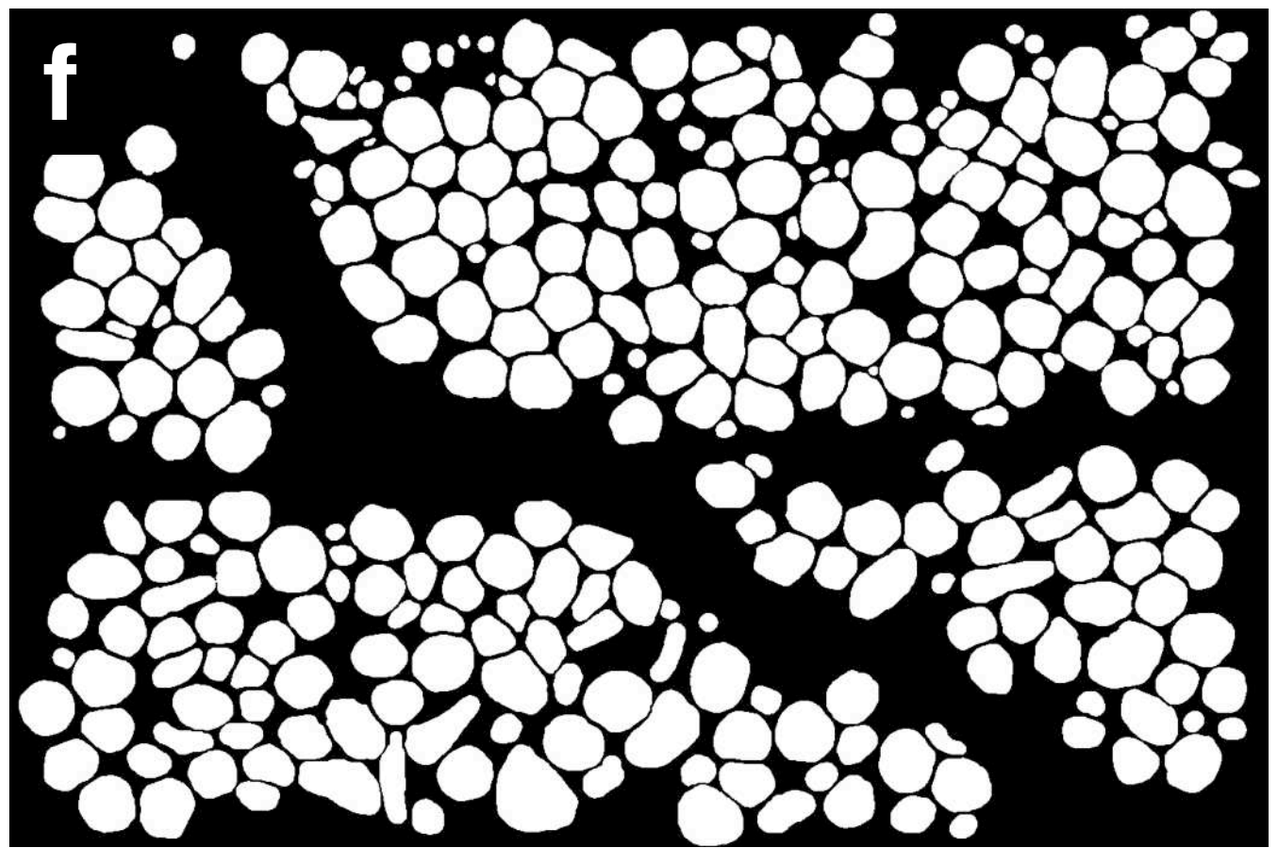
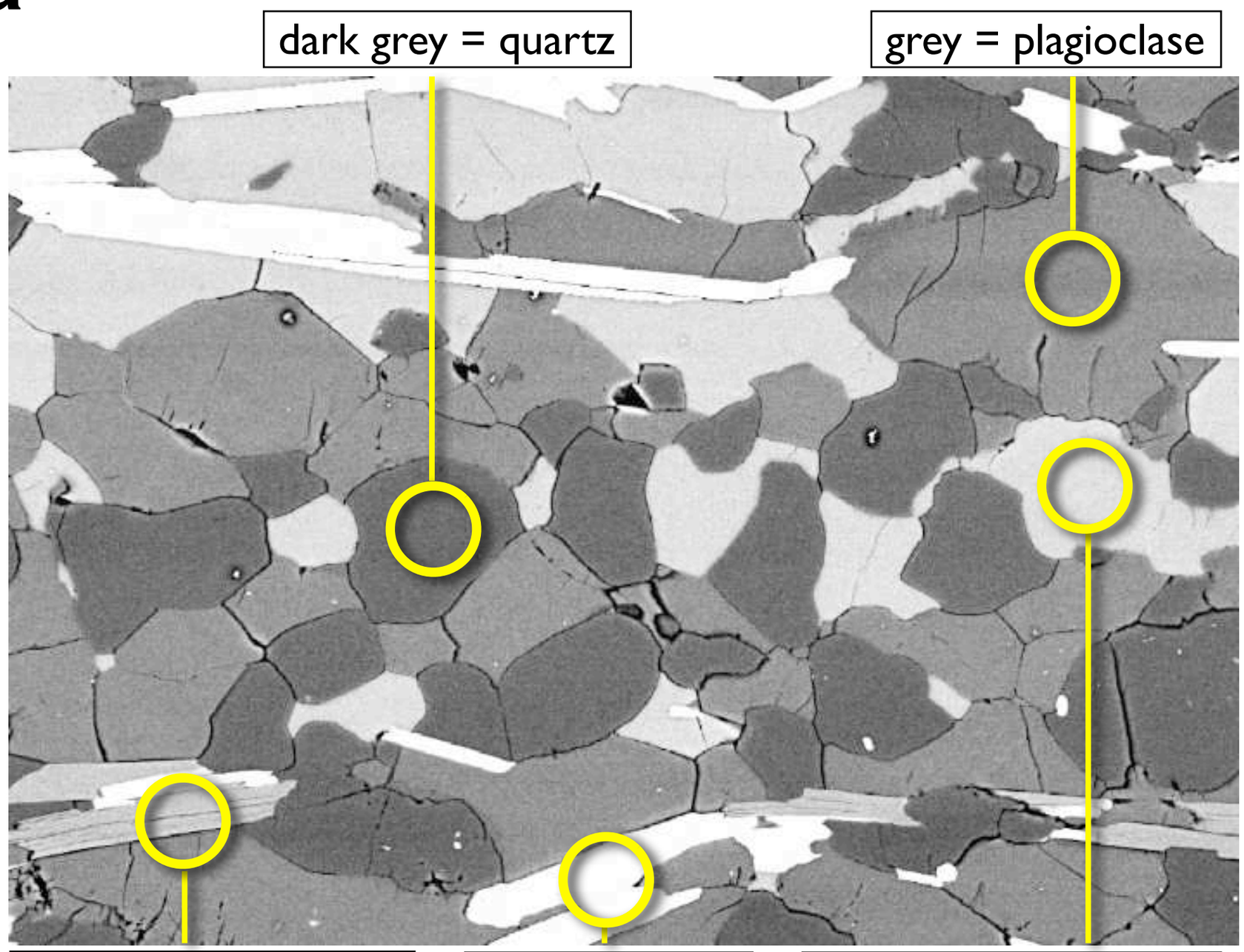
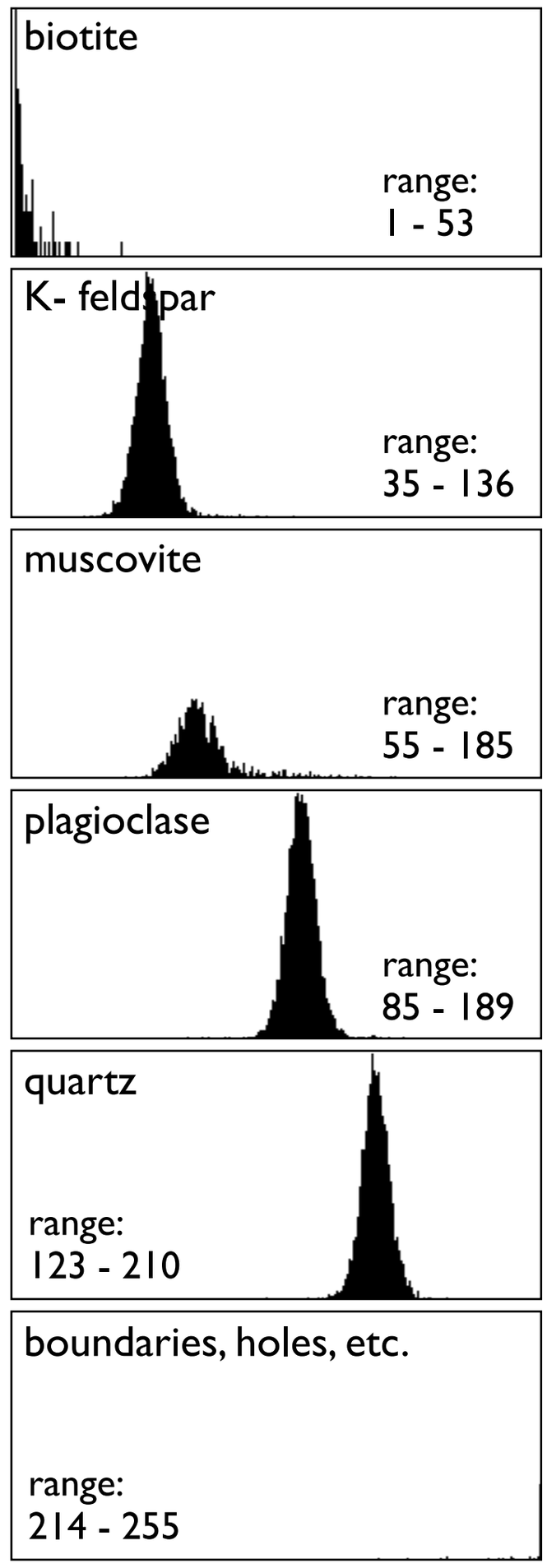


Figure 5.9
Segmentation by thresholding.
(a) Polished surface of oolithic limestone with histogram;
(b) acetate foil replica of oolithic limestone surface;
(c) first attempt to threshold image (b);
(d) high key print of image (b) with 'manual pre-processing' (pencil retouching);
(e) thresholding of (d);
(f) segmentation completed after post-processing.

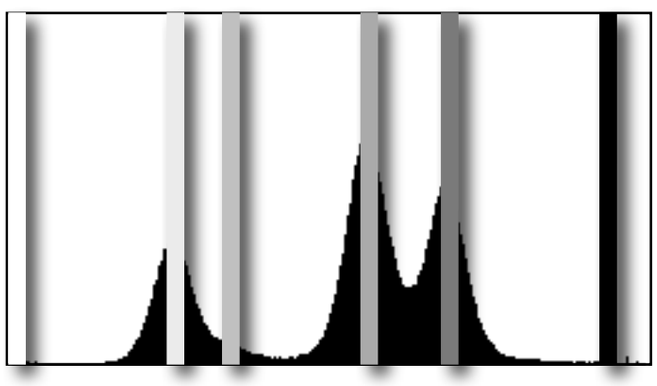
a



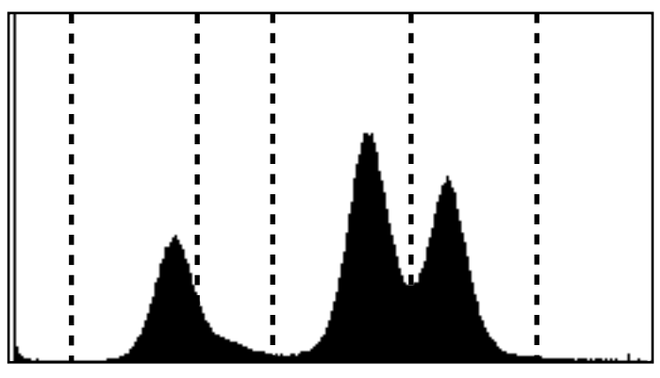
b



c



d



e

```
macro 'Gray LUT for 6 phases [6]';
var
  i,ii,iii,iv,v,vi,j:integer;
  l, k,min,max,mode,gv:integer;
  x,y,w,h,n,mean : integer;

begin
 ShowHistogram;
 PutMessage('lower limit 1 is zero,
 determine 5 upper limits before
 you start, upper limit 6 is 255');

 i:= GetNumber('upper limit 1: ',30);
 ii:= GetNumber('upper limit 2: ',80);
 iii:= GetNumber('upper limit 3: ',107);
 iv:= GetNumber('upper limit 4: ',162);
 v:= GetNumber('upper limit 5: ',200);
 vi:= 255;
```

```
for j:=0 to i do begin
  gv:=i/2;
  RedLUT[j]:=255-gv;
  GreenLUT[j]:=255-gv;
  BlueLUT[j]:=255-gv;
end;

for j:= i+1 to ii do begin
  gv:=(i+ii)/2;
  RedLUT[j]:=255-gv;
  GreenLUT[j]:=255-gv;
  BlueLUT[j]:=255-gv;
end;

for j:= ii+1 to iii do begin
  gv:=(ii+iii)/2;
  RedLUT[j]:=255-gv;
  GreenLUT[j]:=255-gv;
  BlueLUT[j]:=255-gv;
end;
```

```
  for j:= iii+1 to iv do begin
    gv:=(iii+iv)/2;
    RedLUT[j]:=255-gv;
    GreenLUT[j]:=255-gv;
    BlueLUT[j]:=255-gv;
  end;
  for j:= iv+1 to v do begin
    gv:=(iv+v)/2;
    RedLUT[j]:=255-gv;
    GreenLUT[j]:=255-gv;
    BlueLUT[j]:=255-gv;
  end;
  for j:= v+1 to vi do begin
    gv:=(v+vi)/2;
    RedLUT[j]:=255-gv;
    GreenLUT[j]:=255-gv;
    BlueLUT[j]:=255-gv;
  end;
 UpdateLUT;
end;
```

Figure 5.10
Distinguishing phases on the basis of gray values.
(a) SEM micrograph, areas of measurement (for histograms) are indicated;
(b) histograms of gray value distributions of individual phases;
(c) histogram of all phases with representative gray value for the six phases;
(d) histogram of all phases with limits between phases;
(e) text file of 'Gray LUT for 6 phases' macro.

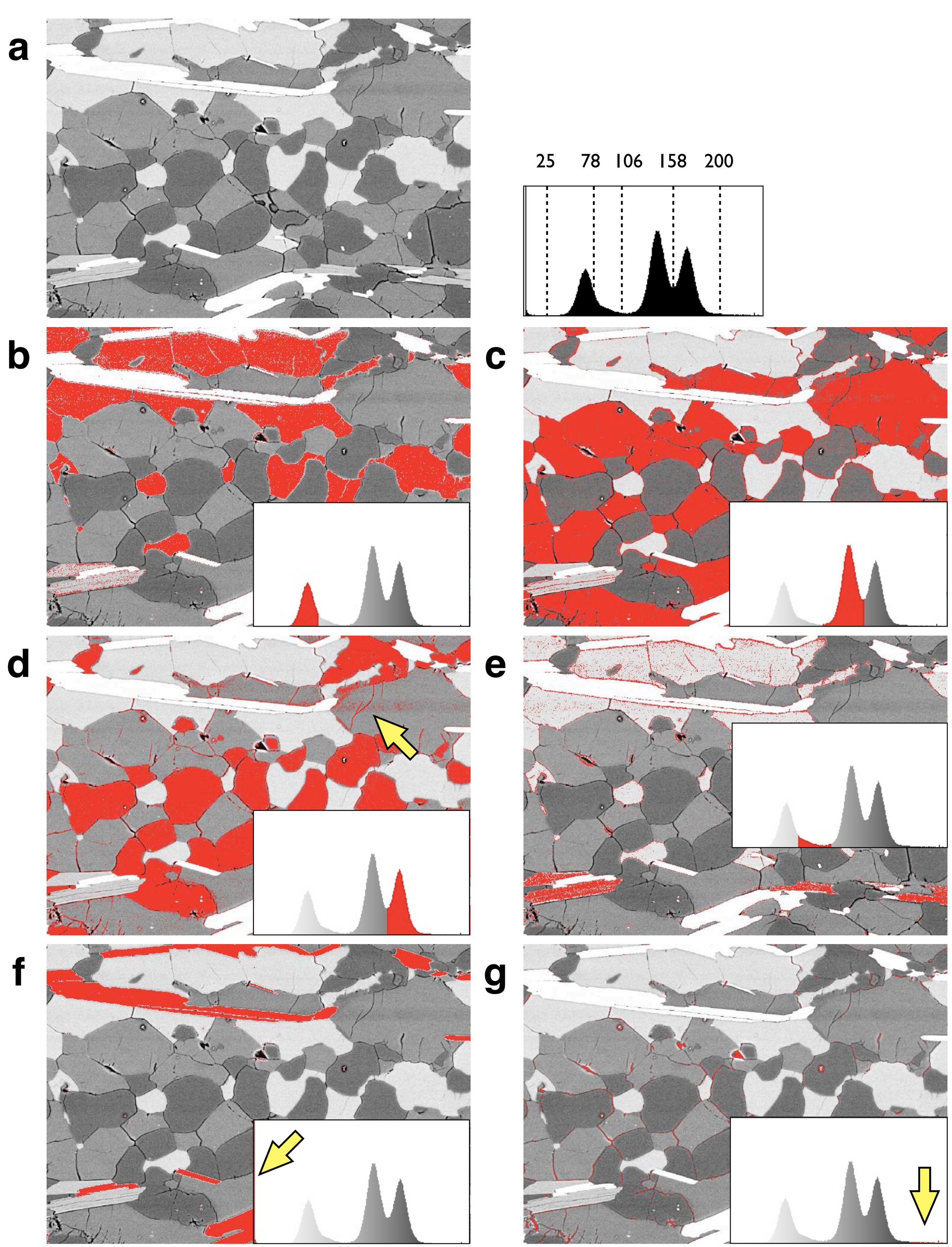
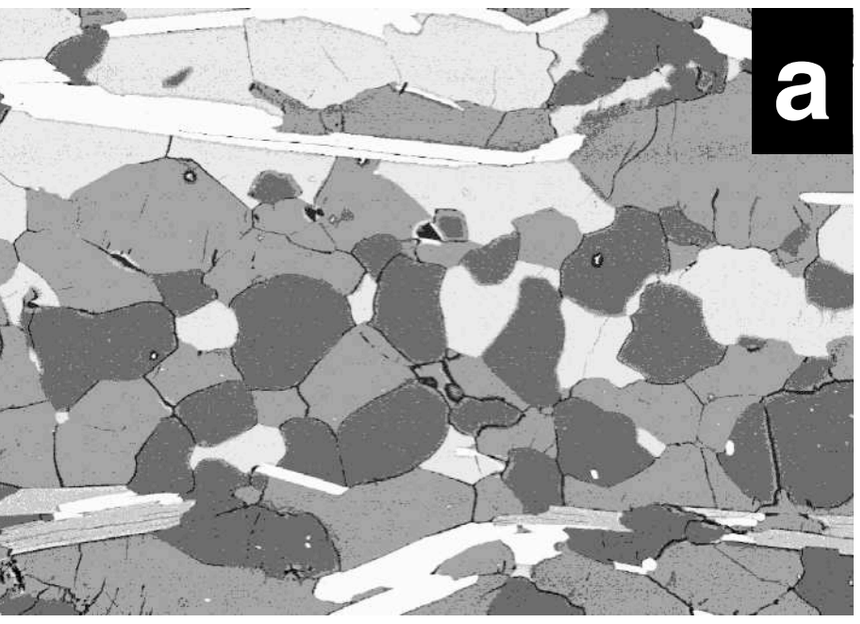
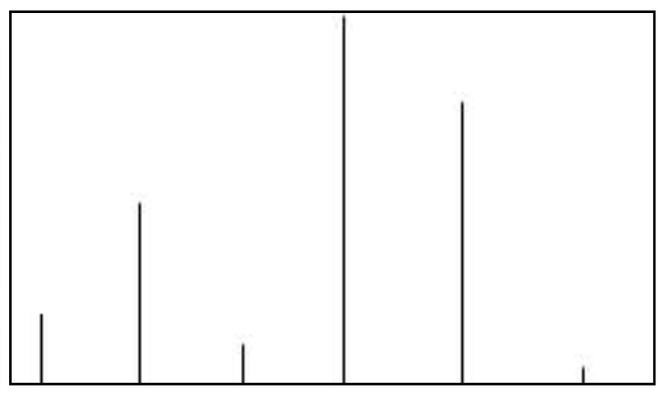


Figure 5.11
Contiguous gray level slicing.
(a) Original; (b) K-feldspar; (c) plagioclase;
(d) quartz, arrow pointing to scanning artefact;
(e) muscovite, note overlap with K-feldspar;
(f) biotite (arrow), with addition of 1 gray value to image;
(g) grain boundaries, holes, dust (arrow).





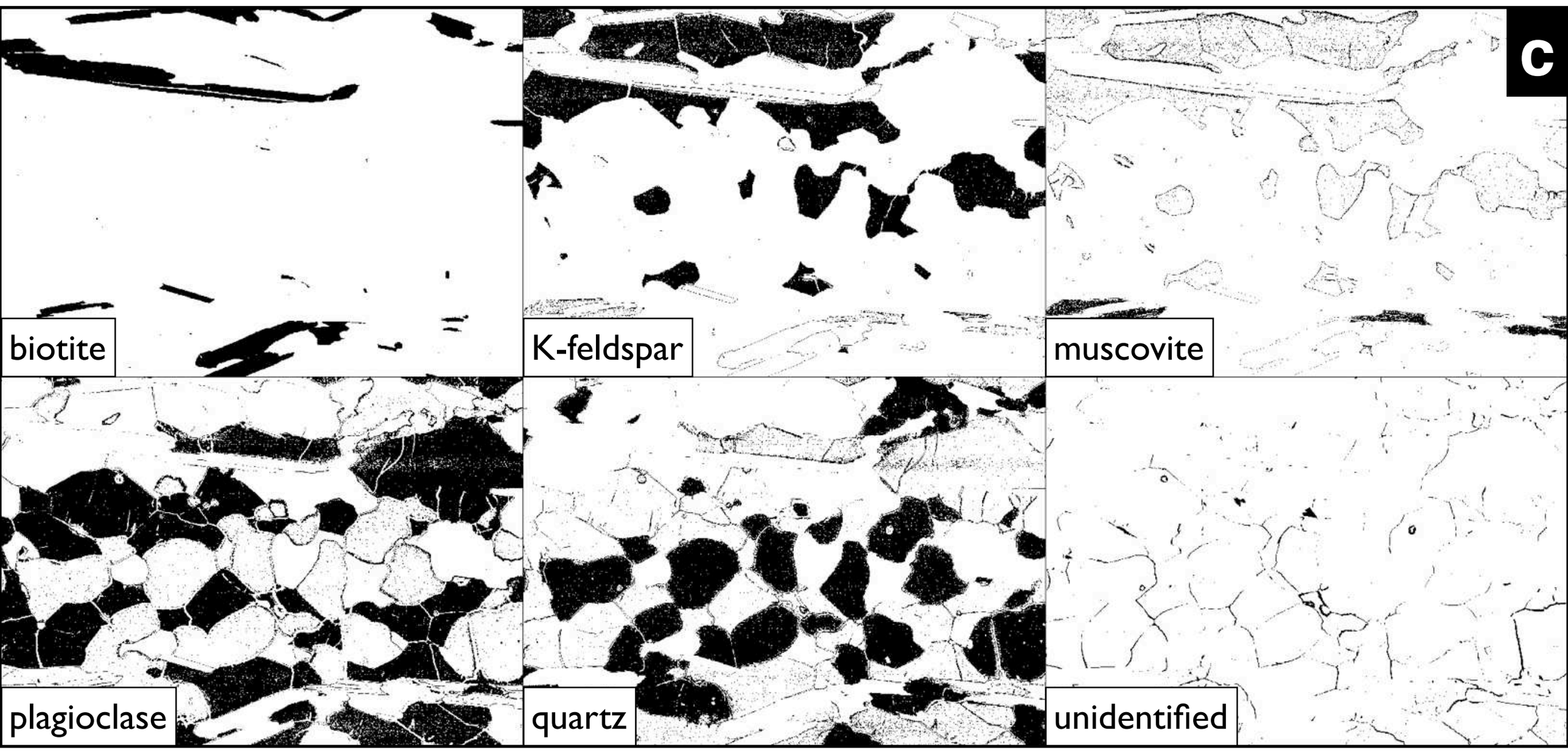


Figure 5.12
Segmentation by tone separation.
(a) Segmented granitoid rock with histogram (six gray levels);
(b) montage of stack with slices that are obtained by separating the gray values at 25, 78, 106, 158, and 200 (compare Figure 5.10.c and d);
(c) average of stack, note absence of overlap or missing pixels, tone separation is perfectly contiguous.