
MorphCol Supplement 1 (23.6.2005)

3.1. Simple analysis of size and shape

Outline extraction: Program Trace33_batch.out (Cycle I)

The first step of any morphometric analysis is outline extraction with program Trace33_batch.out (see Figure 1). The program is calibrated for the Leica MZ6 binocular and allows for output of x,y coordinates either in pixel values or in microns.

Input files:

Input files are digital images of 640x480 pixels in raw format, and with all grey-level information removed (application of LUT changes to a binary black and white image, see Figures 2 and 3).

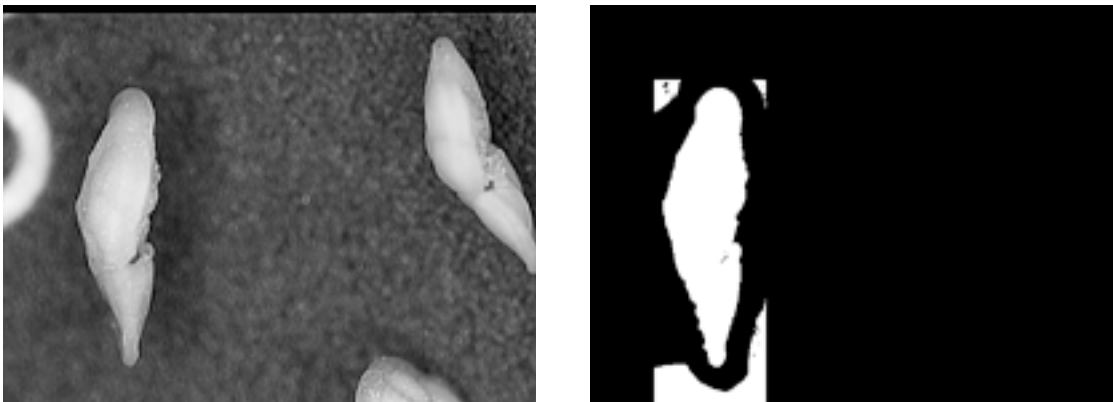


Figure 2: Original grey-level image (Name: 502_0100CCK0501, at Mag=2x) (G. menardii cultrata).
Figure 3: Black and white image in raw format (Name: 0502r).

All area to the left side of the shell should be in black, especially along horizontal line with $y=240$ pixels (e.g. the middle of the image). White areas around the object can occur as long as they have more than 2 pixel difference to the outline.

The file names of the raw images are of the following format:

xxxxr

where xxxx is a four digit integer designating the specimen in a particular cell of a microslide, and r indicates, that the file is a raw file. Other input files, that are necessary is a text file File_list containing the names of the image files, and the microscope magnification. The format is

xxx1r,3.20
xxx2r,1.25
xxx3r,0.63

where the first five characters indicate the file names of the raw images, and the following digits (separated by a comma) indicate the magnification read at the magnification changer on the microscope. The number of input image files is unlimited, depending only on the memory of the computer. The program reads the image names and the magnifications from file File_list, and then treats every file in sequence (batch mode of operation). If an image is missing from this list, a message is sent to the screen and the

Output files:

For each image an output file abcdefghijkKxxxx_T is written to disc containing the xy-coordinates of the object outline. The sample name abcdefghijk is 11 characters long and must be entered at launch of the program. K stands for a letter indicating the orientation of the specimen, usually K for keel view, U for umbilical view and S for spiral view. The four integers xxxx indicate the image file name xxxxr, and the suffix _T indicates that the file contains Traced data.

Other output files are missing_images and Image_errors (see further below). Figure 4 illustrates an outline consisting of 708 points, which was generated from the image shown in Figure 3.

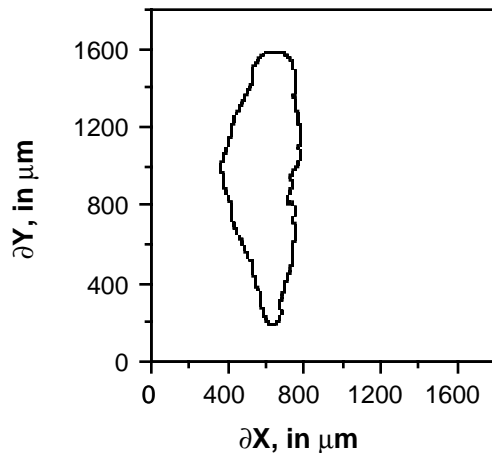


Figure 4. Traced file 502_0100CCK0501_T at magnification 2.00x (e.g. the record in file-list was 0501r,2.00).

Application:

After launching Trace33_batch.out the user is prompted for the sample name (11 characters long, see above). After entry, the program asks whether outline coordinates in pixel values or micrometers are desired. Enter 1 for pixel values and 2 for microns. Thereafter, the list containing the names of the images and the magnifications are requested. Enter the name containing these data (maximum 20 characters).

Calibration:

Trace33_batch.out has been calibrated for an imaging system consisting of a Kappa CF 11/2 color camera, a Leica MZ6 stereo microscope and a PowerMacintosh 8500 (G3) under system 9, and using Nih-Image 1.61 from Wayne Rasband. The framegrabber is a built-in framegrabber from QuickTime. The camera was mounted on the MZ6 with a standard 1x C-mount without any lense. For measurements a 1mm scale was used with 0.01 mm subdivisions. Conversion of x-pixels and y-pixels into micrometer values is performed with the two equations

$$\begin{aligned} XPREC &= 0.11829 * MAG + 0.00201 && \text{(in pixel/}\mu\text{m)} \\ YPREC &= 0.11792 * MAG + 0.00221 && \text{(in pixel/}\mu\text{m)} \end{aligned}$$

where MAG is the magnification read from the File_list.

Pixel to micrometer conversion is done by multiplying the x,y (pixels) values with the corresponding inverse of Xprec or yprec. If a different camera, a different type of computer, or another microscope are used this calibration must first be verified and the system must eventually be recalibrated.

Extraction algorithm:

Because each pixel of an image is 8 bit in depth, a particular pixel and its grey value is recognized by one ascii character, which is also 8 bit long. A black pixel in Adobe Photoshop files has a grey-level of 0, a white pixel has a grey-level value of 255. Note, that this relationship can be reverse in other imaging software (such as Nih-Image, for example).

The grey level is recognized by the standard character to integer conversion function ichar(ch), where ch stands for one character. The method of outline extraction is by checking all grey-levels in the neighbouring eight pixels. If a certain level is met the center point is stored and its x,y coordinate (in pixels) can be calculated via the record number of each pixel (direct access). The algorithm is described in more detail in Knappertsbusch (1998).

Known difficulties:

The program works fine in almost every case of an outline. Nevertheless, outline coordinates should be checked before using them in statistical analyses (this can be done with program XYPlot2.out, see section 3.2.1 further below). Problems have been encountered if the (white) object has embayments, that are 1 pixel wide. In that case the program continues calculating but does not advance to the next point in the outline. Because of this difficulty, an interruption is implemented, which stops outlining that file and the program advances to the next image. In that case an error is reported to the screen and a message write to the external file called Image_Errors, designating the name of the failed image.

Other problems (like odd outliers in the outline) are not entirely excluded. In addition, if the left margin of the input images contain non-black pixels at the y coordinate of 240 pixels, a program stop is enforced, and only a runtime error is reported (reporting a text warning directly informing on this particular problem would be better).

Other versions of Trace:

There exist more versions of Trace, e.g.

Trace32_batch.out (has an older calibration for pixel to micrometer conversion at various magnifications than Trace33_batch.out).

Trace34_batch.out: Is an adaptation to MacOS X for handling file extensions, that are necessary under System MacOS X.

All versions of Trace up to Trace34_batch.out are calibrated for Leica's Achromat 1x and Achromat 2x objective lenses. The versions Trace35_batch.out and Trace36_batch.out are extended for the option to use Leica's Planapo 1x objective lense (with the polarizer plate inserted into the light path).

Trace35_batch.out: Has a calibration for Leica's Planapo 1x objective lens, and is the version for usage on MacOS X (was adapted from Trace34_batch.out)

Trace36_batch out: Has the same additional calibration for Leica's Planapo 1x objective lens than Trace35_batch.out, but runs under MacOS 9.x (was adapted from Trace33_batch.out).

In the following, the calibration details that is implemented in Trace35_batch.out and Trace36_batch.out for the Planapo 1x objective lens, the Kappa CF11/2 camera on a Macintosh 8500, is given:

Table S1:
Measurements for the horizontal calibration for the Achromat 1x and 2x objective lenses:

<u>Setting</u>	<u>Mag</u>	<u>XPrec. pixel/μm</u>
4.0x, 1x Achromat	4,00	0,4770
3.2x, 1x Achromat	3,20	0,3800
2.5x, 1x Achromat	2,50	0,3000
2.0x, 1x Achromat	2,00	0,2370
4.0x, 2x Achromat	8,00	0,9508
3.2x, 2x Achromat	6,40	0,7542
2.5x, 2x Achromat	5,00	0,5950
2.0x, 2x achromat	4,00	0,4740

Table S2:
Measurements for the horizontal calibration for the Planapo 1x objective lens (*):

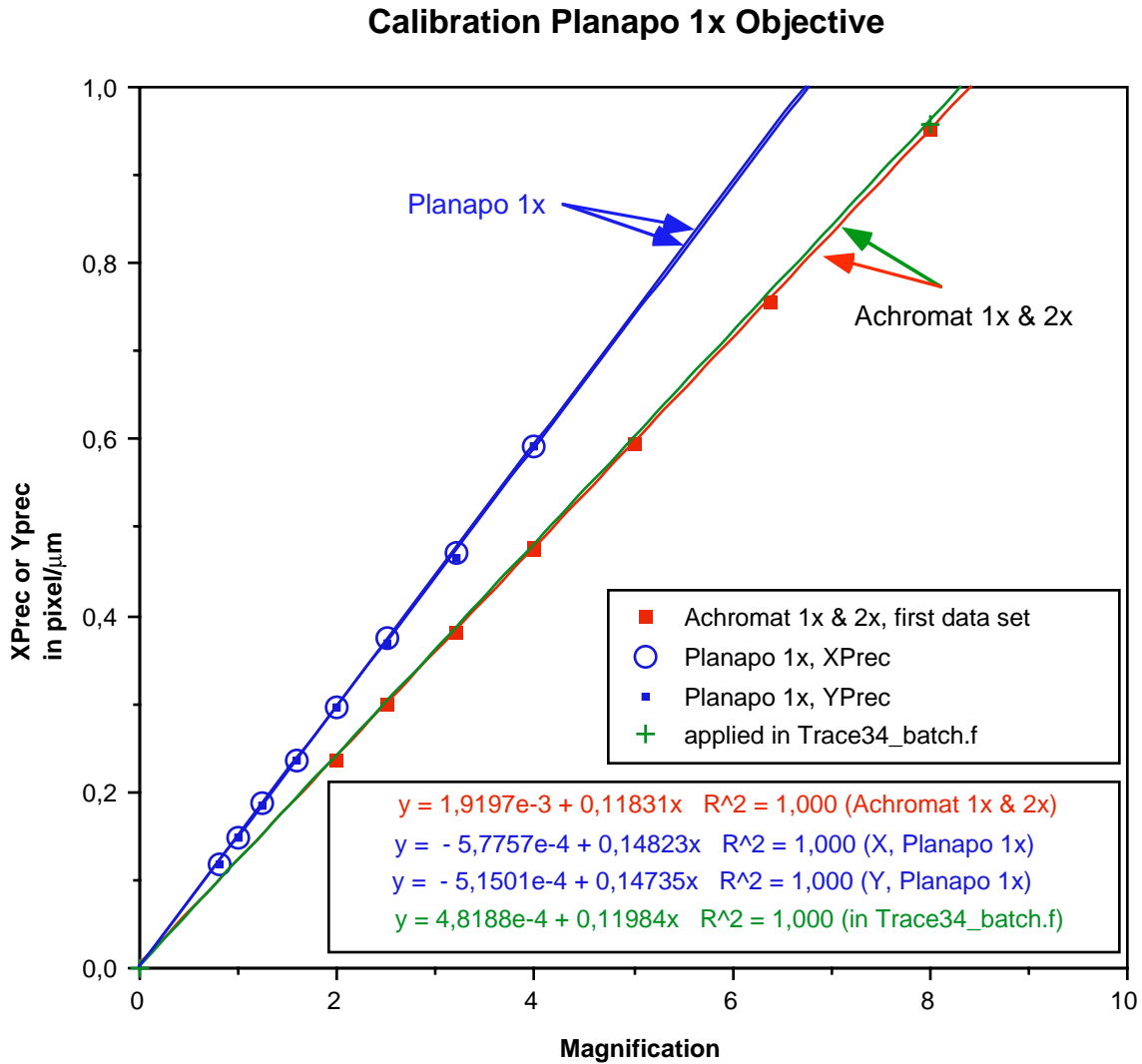
<u>Setting</u>	<u>Mag</u>	<u>XPixel</u>	<u>Distance. μm</u>	<u>XPrec. pixel/μm</u>
0.8x Planapo	0,80	117	1000	0,117
1.0x Planapo	1,00	147	1000	0,147
1.25x Planapo	1,25	186	1000	0,186
1.6x Planapo	1,60	235	1000	0,235
2.0x Planapo	2,00	296	1000	0,296
2.5x Planapo	2,50	375	1000	0,375
3.2x Planapo	3,20	471	1000	0,471
4.0x Planapo	4,00	592	1000	0,592

Table S3:
Measurements for the vertical calibration for the Planapo 1x objective lens (*):

<u>Setting</u>	<u>Mag</u>	<u>YPixel</u>	<u>Distance. μm</u>	<u>YPrec. pixel/μm</u>
Planapo 1x, 4x	4,00	444	750	0,592
Planapo 1x, 3.2x	3,20	372	800	0,465
Planapo 1x, 2.5x	2,50	333	900	0,370
Planapo 1x, 2x	2,00	295	1000	0,295
Planapo 1x, 1.6x	1,60	235	1000	0,235
Planapo 1x, 1.25x	1,25	186	1000	0,186

Planapo 1x, 1.0x	1,00	148	1000	0,148
Planapo 1x, 0.8x	0,80	117	1000	0,117

(*) A polarizer plate is inserted on top of the Planapo 1x objective.



21.6.2005

Figure S1:
Calibration sets for Achromat 1x and 2x lenses (red and green line), and for the Planapo 1x objective lens, with inserted polarizer plate (blue lines, applied in Trace35_batch.out and Trace36_batch.out).

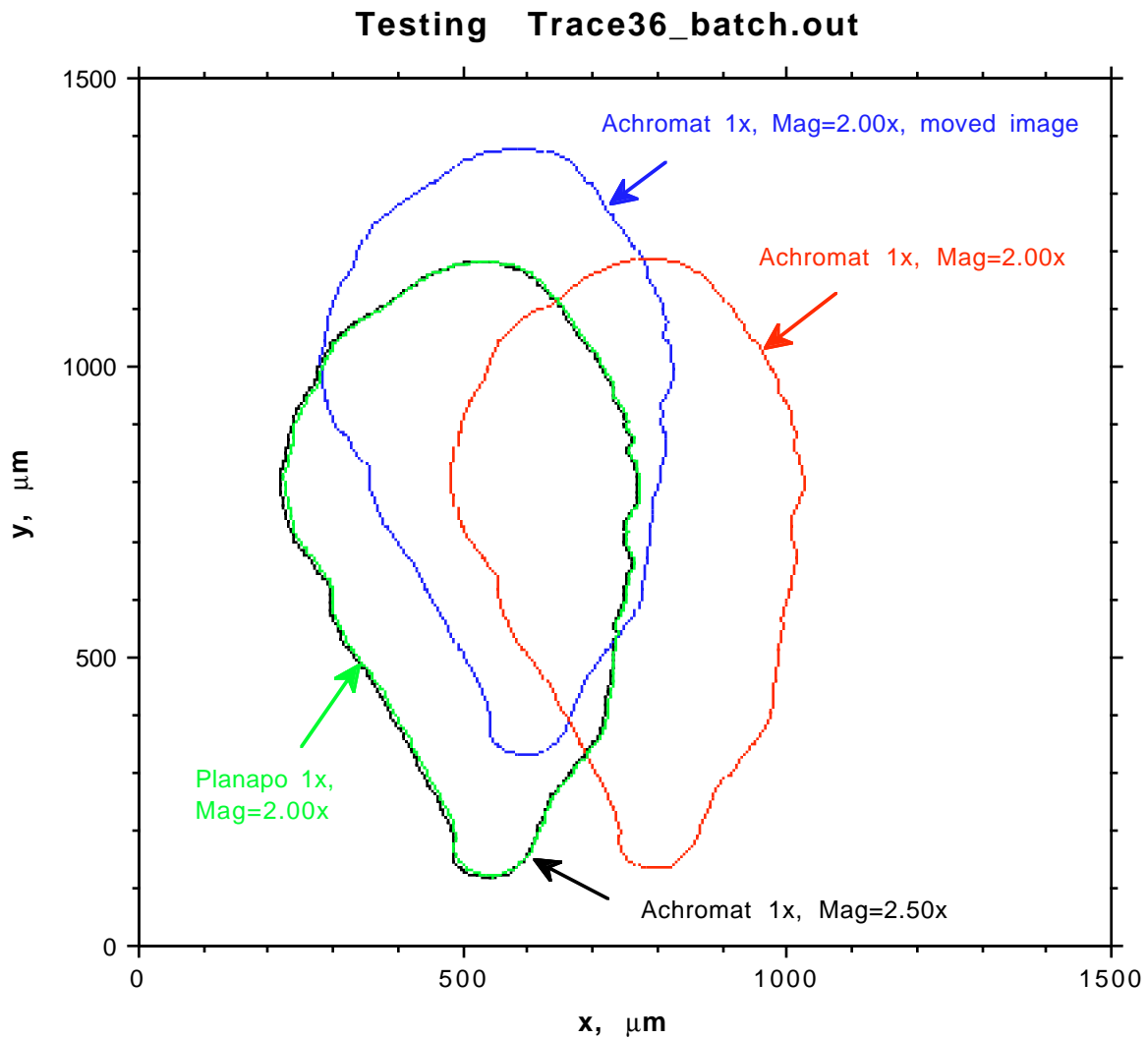


Figure S2:
Imaging the same specimen (503A091125K0202) by using first the Achromat 1x objective at magnification set at 2.0x and 2.5x (black, red and blue curves), and then by using the Planapo 1x objective at Mag=2.0x. All outlines have the same size and coincide when shifted by the appropriate amount (shifting occurs, because the optical axis and the image origin are not identical).

MorphCol Supplement #1 - Program listings

6.1. Listing of program Trace35 batch.f

```
PROGRAM TRACE
C
C   !!! Processing in batch mode. !!!
C   Version 1.2.3, April 28, 1997 by M. Knappertsbusch,
C   modified to batch mode on 21.12.1999
C   modified to write erroneous images to external file. 21.1.2000
C   modified to record missing images, that are read from FILE_LIST.
24.1.2000
C   modified by changing MAG from INTEGER to REAL. 18.12.2000
C   modified by changing the calibration (New XPREC and YPREC).
22.12.2000
C   Adapted for MacOS X (file extension raw for input images added).
9.8.2004.
C   Adapted from Trace34_batch.f: Calibration for Planapo 1x objective
added. 22.6.2005.
C
C   Program to trace the outline of a series of digitized light images
C   in batch mode. Input images are grey-level pictures with 480 lines
C   and 680 pixels per line.
C
C
C   !!!! Output are the X,Y coordinates of the outline
C   in COUNTER-CLOCKWISE direction !!!!
C   Program includes conversion of pixels in micrometers.
C   Program calibrated to the Leica MZ6 binocular stereo microscope.
C
C   Input files are:
C   - A File with name FILE_LIST containing a list of all
C     names of the input images (filename) to be traced,
C     and the magnifications, that were applied to each image.
C     FILE_LIST is a character variable and can bear any name
C     up to 20 characters in length.
C
C   - All image files to be traced. The names of these raw images
C     are interpreted through the character variable INPUT.
C     INPUT is exactly 9 characters long.
C     Example: 0101r.raw
C
C   - The program calls for the sample name through the
C     character variable SAMPLE. SAMPLE is exactly 11 characters
C     long (including orientation codes K or U for Keel or
Umbilical).
C
```

```
C          INPUT and SAMPLE are used to compose the name of the
corresponding
C          output file (suffix _T).
C
C      Output files are:
C      - One file OUTPUT for every traced sample, containing the x,y
C        coordinates of the outline. The character variable OUTPUT is
C        17 characters long (including the suffix _T).
C
C      Note: All file handling for morphometric analyses follows a name
convention:
C
C      filename      = Name of sample, 15 characters long.
C      filename_T    = Input file for raw curves with cartesian coordinates, 17
characters long.
C      filename_POL  = Polar coordinates of filename,          19
characters long.
C      filename_INT  = Interpolated (cartesian) coordinates of filename,    19
characters long.
C      filename_NORM = Normalized curve,                          20 characters
long.
C
C
C*** Note: MAG must be real, otherwise results may be wrong !
C
C      INTEGER I,N,N0,X,Y,THRES,NPTS,NF
C      INTEGER GREY(1:8), K(1:8), A(1:8,1:8)
C      LOGICAL IMAGEX
C      REAL MAG
C      DOUBLE PRECISION XPREC,YPREC,XRES,YRES
C      CHARACTER*1 CHAR(1:8),ANSWER,COMMA
C      CHARACTER*1 OBJ
C      CHARACTER*9 INPUT
C      CHARACTER*17 OUTPUT
C      CHARACTER*11 SAMPLE
C      CHARACTER*20 FILE_LIST
C      PARAMETER (THRES=49)
C
C      In A(I,J) is the local numbering of points:
C
C      DATA ((A(I,J),I=1,8),J=1,8)/1,
* 2,3,4,5,6,7,8,
* 2,3,4,5,6,7,8,1,
* 3,4,5,6,7,8,1,2,
* 4,5,6,7,8,1,2,3,
* 5,6,7,8,1,2,3,4,
* 6,7,8,1,2,3,4,5,
* 7,8,1,2,3,4,5,6,
* 8,1,2,3,4,5,6,7/
C
C      Experimental setup and calibration:
C
C      A Leica MZ6 Stereo-Binocular equipped with a CF 11/2 colour
C      CCD-camera 795(H)x596(V) pixels from Kappa, and imaged using Nih-
Image 1.61
C      on a Power Macintosh 8500 computer. The camera is connected to the
C      microscope via a 1x Cmount.
```



```
IF (ANSWER.EQ.'1') THEN
  WRITE(9,*) 'Output is in pixels'
ELSE
  WRITE(9,*) 'Output is in μm'
  WRITE(9,*) ' '
END IF
C
C
C   Definition of the microscope objective used:
C   Option 1: Achromat 1x or Achromat 2x
C   Option 2: Planapo 1x (with polarizer plate inserted)
C
  WRITE(9,*) 'Indicate which microscope objective is used'
  WRITE(9,*) 'Enter 1 for Achromat 1x or Achromat 2x'      !
Option 1
  WRITE(9,*) 'Enter 2 for Planapo 1x (with polarizer plate)'  !
Option 2
  READ(9,40) OBJ
  IF (OBJ.EQ.'1') THEN
    WRITE(9,*) 'Achromat 1x or 2x objectives'
  ELSE IF (OBJ.EQ.'2') THEN
    WRITE(9,*) 'Planapo 1x objective'
  END IF
40  FORMAT(A1)
C
C
C
C   Get filenames of images and magnifications from FILE_LIST, and start
opening
C   and reading for first image:
C
  WRITE(9,*) ' . . Enter list containing image
1 names (20 chars) and MAG. . .'
  WRITE(9,*) 'Format: 9 chars, comma ,1 real value (2 decimals)'
  WRITE(9,*) 'Note: MAG=Objective lens x zoom factor'
  WRITE(9,*) 'Example: 0101r.raw,3.20 (specimen=0101r,
2 magnification=3.20x)'
  READ(9,101) FILE_LIST
101  FORMAT(A20)
C
C
C
C   Initialization of NF (NF= number of files traced):
C
  NF=0
C
C   Recording errors during program run:
C
  OPEN(21,FILE='Image_Errors',STATUS='NEW')
  WRITE(21,96) SAMPLE
96  FORMAT('Errors in Images of Sample: ',A11)
  OPEN(17,FILE='Missing_images',STATUS='NEW')
C
C   Now reading filenames etc:
C
  -----
```

```
100 READ(20,110,END=999) INPUT,COMMA,MAG
110 FORMAT(A9,A1,F4.2)
C
C   Check first, whether image with name INPUT exists or not. If it does
not exist:
C   print a message and continue reading next image.
C
   INQUIRE(FILE=INPUT,EXIST=IMAGEX)
   IF (.NOT.IMAGEX) THEN
       WRITE(9,*) 'Error: image ',INPUT,' is missing'
       WRITE(17,97) INPUT
97   FORMAT(A9)
       WRITE(9,*) ' . . Program continues. . .'
       GOTO 100
   END IF
C
C   Counting the number of images (=NF) treated:
C
   NF=NF+1
C
C   Initialization of NPTS: (=Number of points in outline):
C
   NPTS=1
C
C   Determination of the names of the OUTPUT files (17 chars long), and
C   writing information to screen:
C
   OUTPUT=(SAMPLE(1:11)//INPUT(1:4))//'_T'
   WRITE(9,111) OUTPUT,MAG
111 FORMAT(' Specimen: ',A17,' at magnification: ',F4.2,'x')
C
C
C   Determination of XRES and YRES (has been calibrated using the NTSC
C   (640x480 pixel) format in NIH-Image:
C
   IF (OBJ.EQ.'1') THEN                                ! Option 1 (Achromat 1x or 2x
objectives)
       XPREC=0.11984*MAG+0.00048188
       YPREC=0.11926*MAG+0.00074223
   ELSE IF (OBJ.EQ.'2') THEN                            ! Option 2 (Planapo 1x objective)
       XPREC=0.14823*MAG-0.00057757
       YPREC=0.14735*MAG-0.00051501
   END IF
C
   XRES=1/XPREC
   YRES=1/YPREC
C
C
C   WRITE(9,*) ' . . calculating. . .'
C
   OPEN(15,FILE=INPUT,ACCESS='DIRECT',
* RECL=1,FORM='FORMATTED',STATUS='OLD')
   OPEN(16,FILE=OUTPUT,STATUS='NEW')
C
C   Find first point of outline in the middle of the
C   TV-screen (this is line 240; its record is calculated
C   as 152960=(240-1)*640):
```

```
DO 1, I=1,640
  N=152960+I
  READ(15,50,REC=N) CHAR(1)
  GREY(1)=ICHAR(CHAR(1))
  IF (GREY(1).GE.THRES) THEN
    N0=N
    GOTO 5
  END IF
1  CONTINUE
C
5  CALL COORD(N,X,Y)
C  WRITE(9,*) X,Y
C
C  Output of results to file:
C
C  IF (ANSWER.EQ.'1') THEN
C    Output of X and Y in pixels:
C    WRITE(16,*) X,',',Y
C  ELSE
C    Output of X and Y in micrometers:
C    WRITE(16,*) X*XRES,',',Y*YRES
C  END IF
C
C
C  STOP Criterium if there is an error in the image:
C  (NPTS indicates the number of points written to the outline file;
usually,
C  an outline of forams at a magnification of  $\leq 4x$  has less than 2500
points).
C  The erroneous INPUT images are written into external file with name
C  "Image_Errors", that can be used as input for subsequent runs.
C
  NPTS=NPTS+1
  IF (NPTS.GT.2500) THEN
    WRITE(9,119) OUTPUT
    WRITE(21,110) INPUT,COMMA,MAG
119  FORMAT('Error in image ',A17, '. Reading next image')
    GOTO 100
  END IF
C
C
C
C
C
C
C  Local initialization: At each point (pixel) of the outline (record
number N)
C  the eight surrounding pixels are read, then their grey-levels
C  are determined, and the pixels are numbered from 1 to 8:
C
C
C      2      3      4
C
C      1      N      5
C
C      8      7      6
C
C
```

```

READ(15,50,REC=N-1) CHAR(1)
READ(15,50,REC=N-641) CHAR(2)
READ(15,50,REC=N-640) CHAR(3)
READ(15,50,REC=N-639) CHAR(4)
READ(15,50,REC=N+1) CHAR(5)
READ(15,50,REC=N+641) CHAR(6)
READ(15,50,REC=N+640) CHAR(7)
READ(15,50,REC=N+639) CHAR(8)
50 FORMAT(A1)
C
C Determination of the grey-levels:
C
DO 10, I=1,8
  GREY(I)=ICHAR(CHAR(I))
10 CONTINUE
C
C Assignment of the local pixel number (i.e. 1 through 8)
C to the record number N for each pixel.
C (i.e. in K are the record numbers of the pixels 1 to 8 stored,
C with K having the indices from i to 8):
C
K(1)=N-1
K(2)=N-641
K(3)=N-640
K(4)=N-639
K(5)=N+1
K(6)=N+641
K(7)=N+640
K(8)=N+639
C
C Search algorithm for the next point of the outline: The algorithm
C searches first for the local point to start with: This first
C point has the grey-level  $\geq$  Threshold. From this initial
C local point it searches the next point, where the greylevel is
C lower than Threshold. The pixel, which is previous to that
C point is the next point of the outline (i.e. the next center for
C the search algorithm).
C
C Look for first point of lokal environment-search:
C
DO 20, I=1,8
  IF (GREY(I).GE.THRES) THEN
C
C First point of search path found. Check every next surrounding
point in
C clockwise direction:
C
DO 30, J=1,8
  IF (GREY(A(I,J)).LT.THRES) THEN
    IF (A(I,J).EQ.1) THEN
      N=K(8)
    ELSE
      N=K(A(I,J)-1)
    END IF
C
C Stop-Criterium:
C
C -----
```

```
C          WRITE(9,*) 'Outline closed'
          CLOSE(16)
          CLOSE(15)

C
C          Reading next file:
          GOTO 100

C
          END IF

C
          GOTO 5
          END IF
30      CONTINUE
          END IF
20      CONTINUE
C
999     WRITE(9,*) ' '
        WRITE(9,*) NF,' images traced and written to files'
        WRITE(9,*) 'One file "Image_Errors" written'
        WRITE(9,*) 'One file "Missing_images" written'
        WRITE(9,*) 'Enter a key to end the program'
        PAUSE 888
888     CONTINUE
C
        STOP
        END
```

SUBROUTINE COORD(M,U,V)

```
C
C      Determines the cartesian coordinates U and V as a function of
C      the record number M representing a pixel in the image.
C      The image has 480 lines and 640 pixels per line.
C
C      The record number starts with M=1 (representing the first
C      pixel in line 1).
C
C      INTEGER M,U,V

C
C      This determination of U and V is valid for all pixels except those
C      that are at the end of each line (i.e. pixel no. 640, 1280,...).
C
C      U=MOD(M,640)
C      V=479-INT(M/640)

C
        RETURN
        END
```

6.2. Listing of program Trace36 batch.f

```
PROGRAM TRACE
C
C   !!! Processing in batch mode. !!!
C   Version 1.2.3, April 28, 1997 by M. Knappertsbusch,
C   modified to batch mode on 21.12.1999
C   modified to write erroneous images to external file. 21.1.2000
C   modified to record missing images, that are read from FILE_LIST.
24.1.2000
C   modified by changing MAG from INTEGER to REAL. 18.12.2000
C   modified by changing the calibration (New XPREC and YPREC).
22.12.2000
C   modified from Trace33_batch.f: Calibration for Planapo 1x objective
added. 22.6.2005.
C
C   Program to trace the outline of a series of digitized light images
C   in batch mode. Input images are grey-level pictures with 480 lines
C   and 680 pixels per line.
C
C
C   !!!! Output are the X,Y coordinates of the outline
C   in COUNTER-CLOCKWISE direction !!!!
C   Program includes conversion of pixels in micrometers.
C   Program calibrated to the Leica MZ6 binocular stereo microscope.
C
C   Input files are:
C   - A File with name FILE_LIST containing a list of all
C     names of the input images (filename) to be traced,
C     and the magnifications that were applied to each image
```

C FILE_LIST is a character variable and can bear any name
C up to 20 characters in length.
C
C - All image files to be traced. The names of these raw images
C are interpreted through the character variable INPUT.
C INPUT is exactly 5 characters long.
C
C - The program calls for the sample name through the
C character variable SAMPLE. SAMPLE is exactly 11 characters
C long (including orientation codes K or U for Keel or
Umbilical).
C
C INPUT and SAMPLE are used to compose the name of the
corresponding
C output file (suffix _T).
C
C Output files are:
C - One file OUTPUT for every traced sample, containing the x,y
C coordinates of the outline. The character variable OUTPUT is
C 17 characters long (including the suffix _T).
C
C Note: All file handling for morphometric analyses follows a name
convention:
C
C filename = Name of sample, 15 characters long.
C filename_T = Input file for raw curves with cartesian coordinates, 17
characters long.
C filename_POL = Polar coordinates of filename, 19
characters long.
C filename_INT = Interpolated (cartesian) coordinates of filename, 19
characters long.
C filename_NORM = Normalized curve, 20 characters
long.
C
C
C*** Note: MAG must be real, otherwise results may be wrong !
C
C INTEGER I,N,N0,X,Y,THRES,NPTS,NF
C INTEGER GREY(1:8), K(1:8), A(1:8,1:8)
C LOGICAL IMAGEX
C REAL MAG
C DOUBLE PRECISION XPREC,YPREC,XRES,YRES
C CHARACTER*1 CHAR(1:8),ANSWER,COMMA
C CHARACTER*1 OBJ
C CHARACTER*5 INPUT
C CHARACTER*17 OUTPUT
C CHARACTER*11 SAMPLE
C CHARACTER*20 FILE_LIST
C PARAMETER (THRES=49)
C
C In A(I,J) is the local numbering of points:
C
C

DATA ((A(I,J),I=1,8),J=1,8)/1,
* 2,3,4,5,6,7,8,


```
WRITE(9,*) '          Program Trace'
WRITE(9,*) '          Version 3.6'
WRITE(9,*) 'Achromat 1x and 2x: calibration from 21.12.2000'
WRITE(9,*) 'Planapo 1x:          calibration from 21.06.2005'
WRITE(9,*) ' '
WRITE(9,*) '          Processing in batch mode'
WRITE(9,*) ' '
WRITE(9,*) ' . . .Enter sample name (11 chars, with K or U). . . '
C
READ(9,102) SAMPLE
102 FORMAT(A11)
C
WRITE(9,*) ' . . .Output in Pixels (1) or Microns (2) ?. . . '
READ(9,4) ANSWER
4 FORMAT(A1)
IF (ANSWER.EQ.'1') THEN
  WRITE(9,*) 'Output is in pixels'
ELSE
  WRITE(9,*) 'Output is in µm'
  WRITE(9,*) ' '
END IF
C
C
C Definition of the microscope objective used:
C Option 1: Achromat 1x or Achromat 2x
C Option 2: Planapo 1x (with polarizer plate inserted)
C
WRITE(9,*) 'Indicate which microscope objective is used'
WRITE(9,*) 'Enter 1 for Achromat 1x or Achromat 2x'          !
Option 1
WRITE(9,*) 'Enter 2 for Planapo 1x (with polarizer plate)'  !
Option 2
READ(9,40) OBJ
IF (OBJ.EQ.'1') THEN
  WRITE(9,*) 'Achromate 1x or 2x Objectives'
ELSE IF (OBJ.EQ.'2') THEN
  WRITE(9,*) 'Planapo 1x objective'
END IF
40 FORMAT(A1)
C
C
C
C
C
C
C Get filenames of images and magnifications from FILE_LIST, and start
opening
C and reading for first image:
C
WRITE(9,*) ' . . .Enter list containing image
1 names (20 chars) and MAG. . . '
WRITE(9,*) 'Format: 5 chars, comma ,1 real value (2 decimals)'
WRITE(9,*) 'Note: MAG=Objective lens x zoom factor'
WRITE(9,*) 'Example: 0101r,3.20 (specimen=0101r,
2 magnification=3.20x)'
READ(9,101) FILE_LIST
101 FORMAT(A20)
```

```
C
C
C
C   Initialization of NF (NF= number of files traced):
C
C   NF=0
C
C   Recording errors during program run:
C
C   OPEN(21,FILE='Image_Errors',STATUS='NEW')
C   WRITE(21,96) SAMPLE
96  FORMAT('Errors in Images of Sample: ',A11)
C   OPEN(17,FILE='Missing_images',STATUS='NEW')
C
C   Now reading filenames etc:
C
C   OPEN(20,FILE=FILE_LIST,STATUS='OLD')
100  READ(20,110,END=999) INPUT,COMMA,MAG
110  FORMAT(A5,A1,F4.2)
C
C   Check first, whether image with name INPUT exists or not. If it does
not exist:
C   print a message and continue reading next image.
C
C   INQUIRE(FILE=INPUT,EXIST=IMAGEX)
C   IF (.NOT.IMAGEX) THEN
C       WRITE(9,*) 'Error: image ',INPUT,' is missing'
C       WRITE(17,97) INPUT
97  FORMAT(A5)
C       WRITE(9,*) ' . . .Program continues. . .'
C       GOTO 100
C   END IF
C
C   Counting the number of images (=NF) treated:
C
C   NF=NF+1
C
C   Initialization of NPTS: (=Number of points in outline):
C
C   NPTS=1
C
C   Determination of the names of the OUTPUT files (17 chars long), and
C   writing information to screen:
C
C   OUTPUT=(SAMPLE(1:11)//INPUT(1:4))//'_T'
C   WRITE(9,111) OUTPUT,MAG
111  FORMAT(' Specimen: ',A17,' at magnification: ',F4.2,'x')
C
C
C   Determination of XRES and YRES (has been calibrated using the NTSC
C   (640x480 pixel) format in NIH-Image:
C
C   IF (OBJ.EQ.'1') THEN                ! Option 1 (Achromat 1x or 2x
objectives)
C       XPREC=0.11984*MAG+0.00048188
C       YPREC=0.11926*MAG+0.00074223
C   ELSE IF (OBJ.EQ.'2') THEN          ! Option 2 (Planapo 1x objective with
```

```

XPREC=0.14823*MAG-0.00057757
YPREC=0.14735*MAG-0.00051501
END IF
C
XRES=1/XPREC
YRES=1/YPREC
C
C
C WRITE(9,*) ' . .calculating. . .'
C
OPEN(15,FILE=INPUT,ACCESS='DIRECT',
* RECL=1,FORM='FORMATTED',STATUS='OLD')
OPEN(16,FILE=OUTPUT,STATUS='NEW')
C
C Find first point of outline in the middle of the
C TV-screen (this is line 240; its record is calculated
C as  $152960=(240-1)*640$ ):
C
DO 1, I=1,640
  N=152960+I
  READ(15,50,REC=N) CHAR(1)
  GREY(1)=ICHAR(CHAR(1))
  IF (GREY(1).GE.THRES) THEN
    N0=N
    GOTO 5
  END IF
1 CONTINUE
C
5 CALL COORD(N,X,Y)
C WRITE(9,*) X,Y
C
C Output of results to file:
C
IF (ANSWER.EQ.'1') THEN
C Output of X and Y in pixels:
  WRITE(16,*) X,',',Y
ELSE
C Output of X and Y in micrometers:
  WRITE(16,*) X*XRES,',',Y*YRES
END IF
C
C
C STOP Criterium if there is an error in the image:
C (NPTS indicates the number of points written to the outline file;
usually,
C an outline of forams at a magnification of  $\leq 4x$  has less than 2500
points).
C The erroneous INPUT images are written into external file with name
C "Image_Errors", that can be used as input for subsequent runs.
C
NPTS=NPTS+1
IF (NPTS.GT.2500) THEN
  WRITE(9,119) OUTPUT
  WRITE(21,110) INPUT,COMMA,MAG
119 FORMAT('Error in image ',A17,'. Reading next image')
  GOTO 100
END IF

```

C
C
C
C
C

C Local initialization: At each point (pixel) of the outline (record number N)

C the eight surrounding pixels are read, then their grey-levels are determined, and the pixels are numbered from 1 to 8:

C
C
C
C
C
C
C
C
C
C

2	3	4
1	N	5
8	7	6

```
READ(15,50,REC=N-1) CHAR(1)
READ(15,50,REC=N-641) CHAR(2)
READ(15,50,REC=N-640) CHAR(3)
READ(15,50,REC=N-639) CHAR(4)
READ(15,50,REC=N+1) CHAR(5)
READ(15,50,REC=N+641) CHAR(6)
READ(15,50,REC=N+640) CHAR(7)
READ(15,50,REC=N+639) CHAR(8)
50 FORMAT(A1)
```

C
C
C

Determination of the grey-levels:

```
DO 10, I=1,8
  GREY(I)=ICHAR(CHAR(I))
10 CONTINUE
```

C
C
C
C
C
C
C

Assignment of the local pixel number (i.e. 1 through 8) to the record number N for each pixel. (i.e. in K are the record numbers of the pixels 1 to 8 stored, with K having the indices from i to 8):

```
K(1)=N-1
K(2)=N-641
K(3)=N-640
K(4)=N-639
K(5)=N+1
K(6)=N+641
K(7)=N+640
K(8)=N+639
```

C
C
C
C
C
C
C
C
C

Search algorithm for the next point of the outline: The algorithm searches first for the local point to start with: This first point has the grey-level \geq Threshold. From this initial local point it searches the next point, where the greylevel is lower than Threshold. The pixel, which is previous to that point is the next point of the outline (i.e. the next center for the search algorithm).

```
C
      DO 20, I=1,8
        IF (GREY(I).GE.THRES) THEN
C
C          First point of search path found. Check every next surrounding
point in
C          clockwise direction:
C
          DO 30, J=1,8
            IF (GREY(A(I,J)).LT.THRES) THEN
              IF (A(I,J).EQ.1) THEN
                N=K(8)
              ELSE
                N=K(A(I,J)-1)
              END IF
C
C              Stop-Criterium:
C
              IF (N.EQ.N0) THEN
C                WRITE(9,*) 'Outline closed'
                  CLOSE(16)
                  CLOSE(15)
C
C                Reading next file:
                  GOTO 100
C
                  END IF
C
                  GOTO 5
                END IF
          30    CONTINUE
            END IF
          20    CONTINUE
C
          999  WRITE(9,*) ' '
              WRITE(9,*) NF,' images traced and written to files'
              WRITE(9,*) 'One file "Image_Errors" written'
              WRITE(9,*) 'One file "Missing_images" written'
              WRITE(9,*) 'Enter a key to end the program'
              PAUSE 888
          888  CONTINUE
C
          STOP
          END
```

SUBROUTINE COORD(M,U,V)

```
C
C   Determines the cartesian coordinates U and V as a function of
C   the record number M representing a pixel in the image.
C   The image has 480 lines and 640 pixels per line.
C
C   The record number starts with M=1 (representing the first
C   pixel in line 1).
C
C   -----
```

```
C
C   This determination of U and V is valid for all pixels except those
C   that are at the end of each line (i.e. pixel no. 640, 1280,...).
C
C   U=MOD(M,640)
C   V=479-INT(M/640)
C
C   RETURN
C   END
```