ANNEXE 4 : Matlab code *Grey Scale Image Analyses* (Bjork et al, 2009)

```
% = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = %
% GRAY SCALE IMAGE ANALYSIS
%
% PHASE RECOGNITION AND PARTICLE IDENTIFICATION
%
% Torbj¨rn Bj¨rk, November 2005
% Updated June 2006
% = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = %

clear;
close all;
clc;

warning off;

% = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = %
% USER INPUT
% = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = %

% OUTPUT FILE NAME
output = 'output';

% READ IMAGE GRAPHICS FROM FILE
image   = imread('KO4-C21_g.jpg');
image   = image(:,1:floor(size(image,2)/2));


% MAGNIFICATION TO DISPLAY IMAGE GRAPHICS
mag = 50;             % of original size

% CONVERTION TO GRAYSCALE
convert2grayscale = false;

% INCREASE CONTRAST

% 1. Saturate low and high intensity values
adjust = false;

% 2. Stretch and adjust intensity values to full lenght of gray scale
stretch_scale = false;

% STRUCTURING ELEMENT
% Set shape and size of structuring element for morpohlogical opening and closing.
se = strel('square',2);    % See reference page in Help browser for other shapes


% SET CONNECTIVITY OF PARTICLES (3x3 ENVIRONMENT)
connectivity = 4;          % 4: Only nearest neighbours.
                           % 8: Nearest and next-nearest neighbours.

% MANUAL REMOVAL OF PRATICLES
% Manual removal of particles using a graphical interface
manually = false;

% MINIMUM DETECTABLE PARTICLE AREA
% Set minimum particle area (pixels)
min_area = 200; % 200

% POST-PROCESSING
% Plotting of particles
boundaries = false;        % true: super-impose particle boundaries
                           % (Note: Touching particles will be plotted
                           % with same colour regardless of particle labeling).
                           % false: Super-impose transparent particles.

% METHOD SELECTION
METHOD = {'Thresholding', 'Watershed'};
method_choice = 1;

% METHOD INPUT
switch METHOD{method_choice}
```

```
   case 'Thresholding'

      % Exagerate particle bounadries
      exaggerate = false;
      ex_number = 1; % Number of times to add tophat and subtract bothat from original image

      % Gray scale range (0 - 255) for thresholding
      lower_boundary = 150;
      upper_boundary = 200;

      % If particle particle density is high and more processing is
      % needed to identify particles
      heavy = false; % MUST BE FALSE; A LITTLE BUG.


      % Number of times to perform morphological closing (dilate + erode)
      % to make the particle boundaries more smooth
      close_factor = 2;

      % Number of times to perform morphological opening (erode + dilate)
      % to separate and identify particles (used if heavy = true)
      open_factor = 1;

   case 'Watershed'

      %Inverse image (particles need to be dark to function as catchment basins)
      inverse = true;

      % Exagerate particle bounadries
      exaggerate = true;
      ex_number = 1; % Number of times to add tophat and subtract bothat from original image

      % Detect intennsity valleys lower than specified threshold
      threshold = true;
      basin_min =20;

      % Oversegmentation
      oversegmentation = false;
      overseg_removal = 90; % Remove any catchment basin shallower than overseg_removal
end

% = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = %

% MAKE OUTPUT DIRECTORY
[SUCCESS,MESSAGE,MESSAGEID] = mkdir('OUTPUT');

%ADD PATH TO ADDITIONAL FUNCTIONS
addpath([pwd, dir_div, 'FUNCTIONS']);

disp('% = = = = = = = = = = = = = = = = = = = = = = = = = %')
disp('% GRAY SCALE IMAGE ANALYSIS                    %')
disp('%                                              %')
disp('% PHASE SEPARATION AND PARTICLE IDENTIFICATION  %')
disp('% = = = = = = = = = = = = = = = = = = = = = = = = = %')
disp(' ')

if convert2grayscale | stretch_scale | adjust
   disp('% = = = = = = = = = = = = = = = = = = = = = = = = = %')
   disp('PRE-PROCESSING')
   disp('% = = = = = = = = = = = = = = = = = = = = = = = = = %')
   disp(' ')
end

% = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = %
% DISPLAY IMAGE
% = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = %

% Convert to grayscale using the NTSC standard for luminance
if convert2grayscale
   disp('Converting to gray scale')
   image = .2989*image(:,:,1)...
   +.5870*image(:,:,2)...
   +.1140*image(:,:,3);
```

```
end



% Stretch and adjust intensity values to full lenght of gray scale
if stretch_scale
   disp('Stretching and adjusting intensity values to full lenght of gray scale')
image = histeq(image);
end

% Saturate at low and high intensity values
if adjust
   disp('Saturating low and high intensity values')
   image = imadjust(image);
end

% ORIGINAL TOTAL SIZE OF IMAGE
original_total_area = size(image,1)*size(image,2);

% MAKE IMAGE DUPLICATE FOR PROCESSING

label = image;

% Display image
figure(1)
clf
imshow(image,'InitialMagnification',mag)
title('Original Image')

% = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = %
% METHOD SWITCH
% = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = %

disp(' ')
disp('% = = = = = = = = = = = = = = = = = = = = = = = = %')
disp('PROCESSING')
disp('% = = = = = = = = = = = = = = = = = = = = = = = = %')
disp(' ')

switch METHOD{method_choice}

   case 'Thresholding'
      fprintf(1, ' Method: Thresholding,      ');

      if heavy == false
         fprintf(1,'Option: Light processing')
         disp(' ')
         [label,num,area_particles,total_area,label_edges,area_edge_grains,area_matrix]=...
         threshold_light(label,image,mag,lower_boundary,upper_boundary,min_area,....
         original_total_area,close_factor,se,manually,boundaries,connectivity,exaggerate,ex_number)
      else
         fprintf(1,'Option: Heavy processing')
         disp(' ')
         [label,num,area_particles,total_area,label_edges,area_edge_grains,area_matrix] = ...
         threshold_heavy(label,image,mag,lower_boundary,upper_boundary,min_area,...
         original_total_area,close_factor,open_factor,se,manually,boundaries,connectivity,exaggerate,ex_number);
      end

   case 'Watershed'
      disp('Method: Watershed')
      disp(' ')
      [label,num,area_particles,total_area,label_edges,area_edge_grains,area_matrix] = ...
      watershedding(label,image,mag,min_area,original_total_area,se,oversegmentation,...
      overseg_removal,basin_min,exaggerate,ex_number,threshold,inverse,manually,boundaries,connectivity);
end

% = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = %
% SAVE DATA
% = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = %
disp('% = = = = = = = = = = = = = = = = = = = = = = = = %')
disp('POST-PROCESSING')
disp('% = = = = = = = = = = = = = = = = = = = = = = = = %')
disp(' ')
```

```
save(['OUTPUT',dir_div,output],'label','num','area_particles','total_area','label_edges','area_edge_grains','area_matrix','orig
inal_total_area');
disp(['Data saved as ',pwd,dir_div,'OUTPUT',dir_div,output,'.mat'])
disp(' ')
disp([output,'.mat contains the following matrices:'])
disp(' ')
disp('label:              Labeled matrix of particles*')
disp('num:                Number of particle')
disp('area_particles:     Area of particles')
disp('area_matrix:        Area of matrix')
disp('total_area:         The total area (area of particles + area of matrix)')
disp('label_edges:        Labeled matrix of removed edge particles')
disp('area_edge_grains:   Area of edgde particles')
disp('original_total_area:  Area of original image (toal area + area of edge particles')
disp(' ')
disp('*The "regionprops"-function can be used for calculation')
disp('of size and shape properties of the particles')
disp(' ')
disp('% = = = = = = = = = = = = = = = = = = = = = = = = %')

close(warndlg)

close(warndlg('FLOOD & FILL - REMOVE PARTICLES MANUALLY'))
```