3D visualisation and quantification of the fracture mechanisms in sparse fibre networks using multiscale X-ray microtomography

Victoria Krasnoshlyk[1,2], Sabine Rolland du Roscoat[1], Pierre J.J. Dumont[3], Per Isaksson[4], Edward Ando[1], Anne Bonnin[5]

[1]Univ. Grenoble Alpes, Grenoble INP, CNRS UMR 5521, 3SR, F-38000, Grenoble, France

[2]Univ. Grenoble Alpes, CNRS UMR 5518, LGP2, F-38000, Grenoble, France

[3]Univ. Lyon, INSA-Lyon, CNRS UMR 5259, LaMCoS, F-69621, Lyon, France

[4]Uppsala University, Ångström Laboratory, Sweden

[5]Paul Scherrer Institut, CH-5232, Villigen PSI, Switzerland

Matlab routine for the Measurement of flocs and antiflocs in 2D images of papers

```
clear all
close all
filename='12_6_0_s_before_8bits_small'; % Name of the image
foldername='12_6_0_s';

%image of the article

% filename='IP-ConvertImage-06';
% foldername='IP-ConvertImage-06';
all_data_saved=[]; %FLOCS: to save mean major and minor sizes of ellipses and mean distance
between flocs centroids
all_data_saved2=[]; %NONFLOCS: to save mean major and minor sizes of ellipses and mean
distance between nonflocs centroids
 i=5
chaine = sprintf('mkdir %s_%s',datestr(now,'yymmdd'),foldername);
unix (chaine);
chaine = sprintf('cd %s_%s',datestr(now,'yymmdd'),foldername);
unix(chaine);
chaine=sprintf('./%s_%s/',datestr(now,'yymmdd'),foldername);

%read image
im1=imread(sprintf('%s_%d.tif',filename,i));
% im1=imread(sprintf('%s.tif',filename));
im1 = imcomplement(im1);
im1_inverted=figure, imshow(im1);

%save image inverted
filename_inverted_origin=sprintf('%s_%d_inverted_origin',filename,i);
name_inverted_origin=strcat(chaine,filename_inverted_origin);
saveas(im1_inverted,name_inverted_origin,'tif');
procedure_treat=figure; subplot(2,3,1);
imshow(im1); title('original, opt microscope');
```

```
%threshold with median value
im1_size=size(im1,1)*size(im1,2);
B= reshape(im1,im1_size,1);
M = median(B)
% M = mean(B)
im0=im1;
im0(im1<M)=0; figure, imshow(im0); title('after threshold with median value');

% subplot(2,3,2);
figure, imshow(im0); title('after threshold with median value');
im1=im0;

%binarisation of the image
im1_bw=im2bw(im1);
% subplot(2,3,3);
figure, imshow(im1_bw); title('binary image');

%analysis of the flocs
%eroding
seDi = strel('disk',3); %disk-shaped struc element

% seSq = strel('square',5); %square-shaped struc element
im1_bw_eroded = imerode(im1_bw,seDi);

% subplot(2,3,4);
figure, imshow(im1_bw_eroded); title('after erosion, flocs');

%remove small areas
pmt=500;
im1_bw2=bwareaopen(im1_bw_eroded,pmt);   %removes all the connected components that have fewer
than P pixels
%subplot(2,3,5);
figure, imshow(im1_bw2); title('after removing small areas <500pxls, flocs');

%save procedure of treatment
filename_procedure_treat=sprintf('%s_%d_%drem_area_procedure_treat',filename,i,pmt);
name_procedure_treat=strcat(chaine,filename_procedure_treat);
saveas(procedure_treat,name_procedure_treat,'tif');
%floc phase measurements
s1= regionprops(im1_bw2,'Area','MajorAxisLength','MinorAxisLength','Eccentricity',
'Orientation','Centroid');

%% analysis of the particles of opt image
%%analysis of the flocs
figure, h1_ellipse=imshow(im1_bw2); title('after removing small areas <500pxls,flocs');
%save the image with removed areas
filename_rem_ar_flocs=sprintf('%s_%d_%drem_areas_flocs',filename,i,pmt);
name_rem_ar_flocs=strcat(chaine,filename_rem_ar_flocs);
saveas(h1_ellipse,name_rem_ar_flocs,'tif');
%draw the ellipses around detected regions
hold on
phi = linspace(0,2*pi,50);
cosphi = cos(phi);
sinphi = sin(phi);

for k = 1:length(s1)
    xbar = s1(k).Centroid(1);
    ybar = s1(k).Centroid(2);

    a = s1(k).MajorAxisLength/2;
    b = s1(k).MinorAxisLength/2;

    theta = pi*s1(k).Orientation/180;
    R = [ cos(theta)   sin(theta)
        -sin(theta)   cos(theta)];

    xy = [a*cosphi; b*sinphi];
    xy = R*xy;

    x = xy(1,:) + xbar;
    y = xy(2,:) + ybar;

    plot(x,y,'r','LineWidth',5);
end
hold off

min_major=min(cat(1,s1.MajorAxisLength))
```

```
max_major=max(cat(1,s1.MajorAxisLength))

min_minor=min(cat(1,s1.MinorAxisLength))
max_minor=max(cat(1,s1.MinorAxisLength))

mean_ellipse_minor_opt=7.3*mean(cat(1,s1.MinorAxisLength))
mean_ellipse_major_opt=7.3*mean(cat(1,s1.MajorAxisLength))

median_ellipse_minor_opt=7.3*median(cat(1,s1.MinorAxisLength))
median_ellipse_major_opt=7.3*median(cat(1,s1.MajorAxisLength))

%%%distance between floc centroids measurement
%find the centres of the flocs
centroids = cat(1, s1.Centroid);
hold on
plot(centroids(:,1), centroids(:,2), 'g*')

%save image with ellipse
filename_ellipse=sprintf('%s_%d_%dremoved_area_flocs',filename,i,pmt);
name_ellipse=strcat(chaine,filename_ellipse);
saveas(h1_ellipse,name_ellipse,'tif');

%the distance between the centroids
for k=1:(size(cat(1, s1.Centroid),1)-1);
    for k2=k+1:size(cat(1, s1.Centroid),1);
    d(k,k2)=sqrt((centroids(k2,1) - centroids(k,1))^2 + (centroids(k2,2) - centroids(k,2))^2);
end
end

mean_distance_between_centr_flocs_opt=7.3*mean(d(d>0))
median_distance_between_centr_flocs_opt=7.3*median(d(d>0))
max_d_b_flocs=max(max(d));

%plot histograms of major and minor sizes of the ellipses
hist1=figure; hist(7.3*(cat(1,s1.MajorAxisLength)),max_major); title('Major ellipse length
distribution, flocs'); xlabel('length, µm'); ylabel('frequency');
hist2=figure; hist(7.3*cat(1,s1.MinorAxisLength),max_minor);  title('Minor ellipse length
distribution, flocs'); xlabel('length, µm'); ylabel('frequency');

%plot histo of distance between floc centroids distribution
hist3=figure; hist(7.3*(d(d>0)),max_d_b_flocs); title('distance between centroids of flocs');
xlabel('distance, µm'); ylabel('frequency');

%saving images
filename_hist=sprintf('%s_%d_%dremoved_major_lngt_fl',filename,i,pmt);
name_hist=strcat(chaine,filename_hist);
saveas(hist1,name_hist,'tif');

filename_hist=sprintf('%s_%d_%dremoved_minor_lngt_fl',filename,i,pmt);
name_hist=strcat(chaine,filename_hist);
saveas(hist2,name_hist,'tif');

filename_hist=sprintf('%s_%d_%dremoved_dist_btw_centr_fl',filename,i,pmt);
name_hist=strcat(chaine,filename_hist);
saveas(hist3,name_hist,'tif')

all_data_saved=[all_data_saved;i,mean_ellipse_minor_opt,median_ellipse_minor_opt,mean_ellipse_
major_opt,median_ellipse_major_opt,mean_distance_between_centr_flocs_opt,median_distance_betwe
en_centr_flocs_opt];

%nonflocs
%invert
im1_invert=imcomplement(im1_bw);
figure, imshow(im1_invert); title('inverted image to analyse nonflocs');

%eroding
seDi = strel('disk',3); %disk-shaped struc element
% seSq = strel('square',5); %square-shaped struc element
im1_bw_eroded_nonfloc = imerode(im1_invert,seDi);
figure,imshow(im1_bw_eroded_nonfloc); title('after erosion, nonfloc');

%remove small areas
pmt=500;
im1_bw2_nonfloc=bwareaopen(im1_bw_eroded_nonfloc,pmt);   %removes all the connected components
that have fewer than P pixels
rem_areas_nonfloc=figure,imshow(im1_bw2_nonfloc); title('small areas are removed <500pxls,
nonfloc');
```

```
%save this image
%save image with ellipse
filename_rem_areas_nonfloc=sprintf('%s_%d_%dremoved_nonfl',filename,i,pmt);
name_rem_areas_nonfloc=strcat(chaine,filename_rem_areas_nonfloc);
saveas(rem_areas_nonfloc,name_rem_areas_nonfloc,'tif');

%nonfloc phase measurements
s2= regionprops(im1_bw2_nonfloc,'Area','MajorAxisLength','MinorAxisLength','Eccentricity',
'Orientation','Centroid','PixelIdxList');
h1_nonfloc=figure, imshow(im1_bw2_nonfloc); title('nonfloc, image after removing small areas
<500pxls');
saveas(rem_areas_nonfloc,name_rem_areas_nonfloc,'tif');
save vic.mat im1_bw2_nonfloc s2

%draw the ellipses around detected regions
hold on
phi = linspace(0,2*pi,50);
cosphi = cos(phi);
sinphi = sin(phi);

for k = 1:length(s2)
    xbar = s2(k).Centroid(1);
    ybar = s2(k).Centroid(2);

    a = s2(k).MajorAxisLength/2;
    b = s2(k).MinorAxisLength/2;

     theta = pi*s2(k).Orientation/180;
%      theta = pi*s2(k).Orientation;
    R = [ cos(theta)   sin(theta)
        -sin(theta)   cos(theta)];

    xy = [a*cosphi; b*sinphi];
    xy = R*xy;

    x = xy(1,:) + xbar;
    y = xy(2,:) + ybar;

    plot(x,y,'y','LineWidth',5);
end
hold off

min_major=min(cat(1,s2.MajorAxisLength))
max_major=max(cat(1,s2.MajorAxisLength))

min_minor=min(cat(1,s2.MinorAxisLength))
max_minor=max(cat(1,s2.MinorAxisLength))

mean_ellipse_minor_opt_nonfloc=7.3*mean(cat(1,s2.MinorAxisLength))
mean_ellipse_major_opt_nonfloc=7.3*mean(cat(1,s2.MajorAxisLength))

median_ellipse_minor_opt_nonfloc=7.3*median(cat(1,s2.MinorAxisLength))
median_ellipse_major_opt_nonfloc=7.3*median(cat(1,s2.MajorAxisLength))

%%%distance between nonfloc centroids measurement
%find the centres of the nonflocs
centroids = cat(1, s2.Centroid);
hold on
plot(centroids(:,1), centroids(:,2), 'b*');

%the distance between the centroids
for k=1:(size(cat(1, s2.Centroid),1)-1);
    for k2=k+1:size(cat(1, s2.Centroid),1);
    d1(k,k2)=sqrt((centroids(k2,1) - centroids(k,1))^2 + (centroids(k2,2) -
centroids(k,2))^2);
end
end

mean_distance_between_centr_nonflocs=7.3*mean(d1(d1>0));
median_distance_between_centr_nonflocs=7.3*median(d1(d1>0));
max_d_b_flocs=max(max(d1));

%plot histograms of major and minor sizes of the ellipses
hist1_1=figure; hist(7.3*(cat(1,s2.MajorAxisLength)),max_major); title('Major nonfloc length
distribution'); xlabel('length, µm'); ylabel('frequency');
```

```
hist2_2=figure; hist(7.3*cat(1,s2.MinorAxisLength),max_minor);  title('Minor nonfloc length
distribution'); xlabel('length, µm'); ylabel('frequency');

%plot histo of distance between floc centroids distribution
hist3_3=figure; hist(7.3*(d1(d1>0)),max_d_b_flocs); title('distance between centroids of
nonflocs'); xlabel('distance, µm'); ylabel('frequency');

%%
%saving images
%save image with ellipse
filename_ellipse1=sprintf('%s_%d_%dremoved_area_nonflocs',filename,i,pmt);
name_ellipse1=strcat(chaine,filename_ellipse1);
saveas(h1_nonfloc,name_ellipse1,'tif');

%histograms
filename_hist1=sprintf('%s_%d_%dremoved_major_lngt_nonfl',filename,i,pmt);
name_hist1=strcat(chaine,filename_hist1);
saveas(hist1_1,name_hist1,'tif');

filename_hist1=sprintf('%s_%d_%dremoved_minor_lngt_nonfl',filename,i,pmt);
name_hist1=strcat(chaine,filename_hist1);
saveas(hist2_2,name_hist1,'tif');

filename_hist1=sprintf('%s_%d_%dremoved_dist_btw_centr_nonfl',filename,i,pmt);
name_hist1=strcat(chaine,filename_hist1);
saveas(hist3_3,name_hist1,'tif')

%%
%%%final images with flocs and nonflocs on it
%read image
im1=imread(sprintf('%s_%d.tif',filename,i));
im1_inverted=figure, imshow(im1);

%draw the ellipses around flocs
hold on
phi = linspace(0,2*pi,50);
cosphi = cos(phi);
sinphi = sin(phi);

for k = 1:length(s1)
    xbar = s1(k).Centroid(1);
    ybar = s1(k).Centroid(2);

    a = s1(k).MajorAxisLength/2;
    b = s1(k).MinorAxisLength/2;

    theta = pi*s1(k).Orientation/180;
    R = [ cos(theta)   sin(theta)
        -sin(theta)   cos(theta)];

    xy = [a*cosphi; b*sinphi];
    xy = R*xy;

    x = xy(1,:) + xbar;
    y = xy(2,:) + ybar;

    plot(x,y,'r','LineWidth',4);
end

%draw the ellipses around nonflocs
hold on
phi = linspace(0,2*pi,50);
cosphi = cos(phi);
sinphi = sin(phi);

for k = 1:length(s2)
    xbar = s2(k).Centroid(1);
    ybar = s2(k).Centroid(2);

    a = s2(k).MajorAxisLength/2;
    b = s2(k).MinorAxisLength/2;

    theta = pi*s2(k).Orientation/180;
    R = [ cos(theta)   sin(theta)
        -sin(theta)   cos(theta)];

    xy = [a*cosphi; b*sinphi];
    xy = R*xy;

    x = xy(1,:) + xbar;
    y = xy(2,:) + ybar;

    plot(x,y,'y','LineWidth',4);
end
hold off

%saving images
filename_floc_nonfloc_final=sprintf('%s_%d_%darea_floc_nonfloc_final',filename,i,pmt);
name_floc_nonfloc_final=strcat(chaine,filename_floc_nonfloc_final);
saveas(im1_inverted,name_floc_nonfloc_final,'tif');


all_data_saved2=[all_data_saved2;i,mean_ellipse_minor_opt_nonfloc,median_ellipse_minor_opt_non
floc,mean_ellipse_major_opt_nonfloc,median_ellipse_major_opt_nonfloc,mean_distance_between_cen
tr_nonflocs,median_distance_between_centr_nonflocs];
% end

save all_data_saved
save all_data_saved2
```