## Contents



```
clc;
close all;
clear all;
```

### folder direc

```
folder='/data1/nacere/Processing/XG01/Binary_images';
files=dir(fullfile(folder, 'binary*.png'));
widths_all=cell(length(files),1);
```

### cleaning binary images

```
a=imread('binary4.png');
cl=imfill(a,'holes');
figure;
imshow(cl);
%for loop
```

```
Error using imread>get_full_filename
File "binary4.png" does not exist.

Error in imread (line 372)
        fullname = get_full_filename(filename);

Error in widthsextractionandplots (line 9)
a=imread('binary4.png');
```

### strat looping

```
for imgNum=1:length(files)
```

```
    imgpath=fullfile(folder,files(imgNum).name);
    img=imread(imgpath);
    [height,width]=size(img);
    img=imfill(img,'holes');
    % height of white channel
    height_channel=regionprops(img,'BoundingBox');
    boundingbox=[height_channel.BoundingBox];
    heights=boundingbox(4:4:end);

    %figure; imshow(img);
    %getting x and y coordinates
    [y,x]=find(img);
    xy=[x,y]; %here column1=x and column 2=y; combine into a matrix
```

### then lets get average of y values

```
%first get unique x

unique_x=unique(x);
for i=1:length(unique_x)
    avg_y(i)=mean(y(x==unique_x(i))); %get vertical pos for each horiz to define smooth centerline
end
```

```
p=0.01;
fine_x=linspace(min(unique_x),max(unique_x),500);
spline_ft=csaps(unique_x,avg_y,p); %cui smoothline
smoothy=fnval(spline_ft, fine_x);%evaluation @x to get smoth y
figure;
imshow(img);
hold on;
%plot(unique_x,avg_y);
plot(fine_x,smoothy,'b-','LineWidth',4);
hold off;
```

### get the widths : here we need to get a tangent line then where it intersects the top and bottom will be the lines of the widths

equation of tangent lines : point slope from eqaution of a straight line why tang slope? because it gives direction of centerline @ specific point use this formula: y-y1=mt(X-X1) mT= f'(x10=f(x1+h)-f(x1)/h

```
%slope change in y/change in x for middle line
slopes = diff(smoothy) ./ diff(fine_x); %FDM to get slope of tang
slopes = [slopes, slopes(end)]; %for dimensions so slops same ffinex
perpen_slopes = -1 ./ slopes; %as slopes should be 90degrewes
```

### widths should match dimensions of x array

```
widths = zeros(1, length(fine_x));
figure;
imshow(img);
hold on;
plot(fine_x, smoothy, 'r-', 'LineWidth', 2);
tang_length = 45; % length of tan
for i = 1:length(fine_x)  %looping trhough points on certline to get widths
    slope = perpen_slopes(i);
    x0 = fine_x(i); %here we get x and y coord of centerline in red
    y0 = smoothy(i);
    % maths done by hands to see if it makes sense to get xchange in x an y
 % in tangent line
    dx = tang_length / sqrt(1 + slope^2);%using pythagorean theorem (used to wonder after middle school when ill use it) %to get horizontal of tan line
    dy = slope * dx;
    x_tan = [x0 - dx, x0 + dx];
    y_tan = [y0 - dy, y0 + dy];
    tangent_x = linspace(x_tan(1), x_tan(2), 100);
     % tangent should be from top and bottom of the image if 25 length is
     % too long then getting the length of tan to not crss the image size
    tangent_y = slope * (tangent_x - x0) + y0;
    tangent_x = max(1, min(size(img,2), tangent_x));
    tangent_y = max(1, min(size(img,1), tangent_y));
%we need to interpolate the tang as in the code given by professor where y-values were interpolated for both lines on the common x-range
    tangent_values = interp2(double(img), tangent_x, tangent_y, 'linear', 0);
% yo het the white channel is should be everything greater than 0 as
% white is 1 in binary
    white_channel = find(tangent_values == 1);
    if ~isempty(white_channel)
        top_idx = white_channel(1); %first white pixel on tan lin
        bottom_idx = white_channel(end); %last
        x_top = tangent_x(top_idx); %coord of x and y
        y_top = tangent_y(top_idx);
        x_bottom = tangent_x(bottom_idx);
        y_bottom = tangent_y(bottom_idx);
    % Plotting top and bottom to see
        %plot(x_top, y_top, 'bo', 'MarkerSize', 6, 'MarkerFaceColor', 'b');
        %plot(x_bottom, y_bottom, 'ro', 'MarkerSize', 6, 'MarkerFaceColor', 'r');
    % ok now we have top and bottom values, we can use them to get widths as the distance line
        widths(i) = sqrt((x_bottom - x_top)^2 + (y_bottom - y_top)^2);
    else
        widths(i) = 0; % No width if no white  found
    end
% Plotting tangent line
    plot([x_tan(1), x_tan(2)], [y_tan(1), y_tan(2)], 'g-', 'LineWidth', 1);
end
hold off;
```

```
    widths_all{imgNum}=widths;
    %histogram(widths);
    %%figure;
    %%hold on;
    %for k=1:length(files)
        %widths=widths;
        %plot(1:length(widths), widths, 'r');
    %end
    %hold off;

    avg_y=[];
```

```
end
```

### %% save widths for each xantham gum

```
save(fullfile(folder,'avgr_widthslast04thresh.mat'),'avgr_widthsthreshxg4');
```

### find average

before finding the average, we need to not consider the zeros (where the missing data are located) so we can use the non zero function

```
%for each xanthan gum take the non zeros widths then apply the mean on them

for i=1:length(widths_all)
    avgr_widths(i)=mean(nonzeros(widths_all{i}));
end
```

### save widths for each xantham gum

```
save(fullfile(folder,'avgr_widthslast05.mat'),'avgr_widths');
```

### conert to c m

```
%read ruler img
img_ruler=imread('binary3.png');
figure;imshow(img_ruler);
[x,y]=ginput(2);
ruler=sqrt((x(2)-x(1))^2 +(y(2)-y(1))^2);
```

### converyt to cm

see conversion .m file

```
%then after converting our different metrics, they were saved into a folder
%and uploaded below
```

### Plotting

```
clc;
close all;
clear;
```

```
data0=load('widthscmxg0.mat');
data1=load("widthscmxg01.mat");
data2=load("widthscmxg02.mat");
data3=load("widthscmxg03.mat");
data4=load("widthscmxg04thresh.mat");
data5=load("widthscmxg05thresh.mat");

widths0=data0.widthsxg0;
widths1=data1.widthsxg1;
widths2=data2.widthsxg2;
widths3=data3.widthsxg3;
widths4=data4.widthsxg4thrsh;
widths5=data5.widthsxg5thresh;
```

```
all_data={widths0;widths1;widths2;widths3;widths4;widths5};
```

### time array

```
time0=load('Timexg0.mat');
time1=load('Timexg01.mat');
time2=load('Timexg02.mat');
time3=load('Timexg03.mat');
time4=load('Timexg04.mat');
time5=load('Timexg05.mat');

timexg0=time0.time_array;
timexg1=time1.time_array;
timexg2=time2.time_array;
timexg3=time3.time_array;
timexg4=time4.time_array;
timexg5=time5.time_array;
```

### normalized time

```
times_norxg0=(timexg0-min(timexg0))/(max(timexg0)-min(timexg0));
times_norxg1=(timexg1-min(timexg1))/(max(timexg1)-min(timexg1));
times_norxg2=(timexg2-min(timexg2))/(max(timexg2)-min(timexg2));
times_norxg3=(timexg3-min(timexg3))/(max(timexg3)-min(timexg3));
times_norxg4=(timexg4-min(timexg4))/(max(timexg4)-min(timexg4));
times_norxg5=(timexg5-min(timexg5))/(max(timexg5)-min(timexg5));
```

```
set(0, 'DefaultAxesFontName', 'TimesNewRoman');
figure;
hold on;
colors=orderedcolors('meadow');
markers = {'o', 's', 'd', '^', 'x', '*'};
plot(timexg0, widths0, 'Color', colors(1,:), 'Marker', markers{1}, 'LineWidth', 1,'LineStyle', 'none', 'MarkerSize', 5, 'DisplayName', '0% XG');
plot(timexg1, widths1, 'Color', colors(2,:), 'Marker', markers{2}, 'LineWidth', 1,'LineStyle', 'none', 'MarkerSize', 5, 'DisplayName', '0.1% XG');
plot(timexg2, widths2, 'Color', colors(3,:), 'Marker', markers{3}, 'LineWidth', 1, 'LineStyle', 'none','MarkerSize', 5, 'DisplayName', '0.2% XG');
plot(timexg3, widths3, 'Color', colors(4,:), 'Marker', markers{4}, 'LineWidth', 1, 'LineStyle', 'none','MarkerSize', 5, 'DisplayName', '0.3% XG');
plot(timexg4, widths4, 'Color', colors(5,:), 'Marker', markers{5}, 'LineWidth', 1,'LineStyle', 'none', 'MarkerSize', 5, 'DisplayName', '0.4% XG');
plot(timexg5, widths5, 'Color', colors(6,:), 'Marker', markers{6}, 'LineWidth', 1, 'LineStyle', 'none','MarkerSize', 5, 'DisplayName', '0.5% XG');

set(gca,'FontSize',12);
xlabel('Time (s)', 'FontSize', 12, 'FontWeight', 'normal');
ylabel('Spatially Averaged Active Channel Widths (cm)', 'FontSize', 11, 'FontWeight', 'normal');
%title('Average Channel Widths vs Time for Different Xanthan Gum Concentrations', 'FontSize', 8, 'FontWeight', 'bold');
legend('show', 'Location', 'best');
grid on;
grid minor;
box on;
saveas(gcf, 'avgchannelwidthscm_time_figurenolegend.png');
hold off;
```

```
figure;
hold on;
%colors = {'b', 'r', 'g', 'm', 'c', 'k'};
colors=lines(6);
markers = {'o', 's', 'd', '^', 'x', '*'};
plot(timexg0, widths0, [colors markers{1} ], 'LineWidth', 1, 'MarkerSize', 5, 'DisplayName', '0% Xanthan Gum');
plot(timexg1, widths1, [colors markers{2} ], 'LineWidth', 1, 'MarkerSize', 5, 'DisplayName', '0.1% Xanthan Gum');
plot(timexg2, widths2, [colors markers{3} ], 'LineWidth', 1, 'MarkerSize', 5, 'DisplayName', '0.2% Xanthan Gum');
plot(timexg3, widths3, [colors markers{4} ], 'LineWidth', 1, 'MarkerSize', 5, 'DisplayName', '0.3% Xanthan Gum');
plot(timexg4, widths4, [colors markers{5} ], 'LineWidth', 1, 'MarkerSize', 5, 'DisplayName', '0.4% Xanthan Gum');
plot(timexg5, widths5, [colorsmarkers{6} ], 'LineWidth', 1, 'MarkerSize', 5, 'DisplayName', '0.5% Xanthan Gum');

set(gca, 'FontSize', 8, 'LineWidth', 1);
xlabel('Time (s)', 'FontSize', 8, 'FontWeight', 'bold');
ylabel('Spatially Averaged Active Channel Widths (cm)', 'FontSize', 8, 'FontWeight', 'bold');
title('Average Channel Widths vs Time for Different Xanthan Gum Concentrations', 'FontSize', 8, 'FontWeight', 'bold');
legend('show', 'Location', 'best');
%grid on;
box on;
saveas(gcf, 'avgchannelwidthscm_time_figure.png');
hold off;
```

### widths xanthan gum

```
set(0, 'DefaultAxesFontName', 'TimesNewRoman');
xg=[0;1;2;3;4;5];
allwidths_avg=[mean(widths0),mean(widths1),mean(widths2),mean(widths3),mean(widths4),mean(widths5)];
allwidths_std = [std(widths0), std(widths1), std(widths2),nanstd(widths3), std(widths4), std(widths5)];
```

```
% Plot
figure;
errorbar(xg, allwidths_avg, allwidths_std, 'k', 'LineStyle', 'none', 'LineWidth', 1,'HandleVisibility', 'off');
hold on
plot(xg(1),allwidths_avg(1),'ko','LineWidth',1,'MarkerSize',6,'DisplayName','0% Xanthan Gum');
plot(xg(2),allwidths_avg(2),'ks','LineWidth',1,'MarkerSize',6,'DisplayName','0.1% Xanthan Gum');
plot(xg(3),allwidths_avg(3),'kd','LineWidth',1,'MarkerSize',6,'DisplayName','0.2% Xanthan Gum');
plot(xg(4),allwidths_avg(4),'k^','LineWidth',1,'MarkerSize',6,'DisplayName','0.3% Xanthan Gum');
plot(xg(5),allwidths_avg(5),'kp','LineWidth',1,'MarkerSize',6,'DisplayName','0.4% Xanthan Gum');
plot(xg(6),allwidths_avg(6),'kx','LineWidth',1,'MarkerSize',6,'DisplayName','0.5% Xanthan Gum');
%grid on;
set(gca,'FontSize',12);
xlabel('% Xanthan Gum', 'FontSize', 12, 'FontWeight', 'normal');
ylabel('Time and Space Averaged Widths (cm)', 'FontSize', 11, 'FontWeight', 'normal');
%title('Average Channel Widths vs Time for Different Xanthan Gum Concentrations', 'FontSize', 8, 'FontWeight', 'bold');
%legend('show', 'Location', 'best');
grid on;
grid minor;
box on;
saveas(gcf, 'xanthanwidthscm_time_figurenolegend.png');
hold off;
```

## Width calculations to fix time gap

## widths data

```
widths0=data0.widthsxg0;
widths1=data1.widthsxg1;
widths2=data2.widthsxg2;
widths3=data3.widthsxg3;
widths4=data4.widthsxg4thrsh;
widths5=data5.widthsxg5thresh;
```

## times

```
data=load('flume_time_arrays_by_xg (1).mat');

time1=data.xg_0;
time2=data.xg_0_1;
time3=data.xg_0_2;
time4=data.xg_0_3;
time5=data.xg_0_4;
time6=data.xg_0_5;
```

```
set(0, 'DefaultAxesFontName', 'TimesNewRoman');
all_times = {time1, time2, time3, time4, time5, time6};
all_widths = {widths0, widths1, widths2, widths3, widths4, widths5};
labels = {'0% XG', '0.1% XG', '0.2% XG', '0.3% XG', '0.4% XG', '0.5% XG'};
colors = orderedcolors('meadow');
markers = {'o', 's', 'd', '^', 'x', '*'};

figure;
hold on;

for i = 1:length(all_times)
    t = all_times{i};
    w = all_widths{i};

    % Truncate both to the shorter length
    min_len = min(length(t), length(w));
    t = t(1:min_len);
    w = w(1:min_len);

    plot(t, w, 'Color', colors(i, :),'Marker', markers{i}, 'LineStyle', 'none','LineWidth', 1,'MarkerSize', 5,'DisplayName', labels{i});
end
set(gca,'FontSize',12);
xlabel('Time (min)', 'FontSize', 12, 'FontWeight', 'normal');
ylabel('Spatially Averaged Active Channel Widths (cm)', 'FontSize', 11, 'FontWeight', 'normal');
%title('Average Channel Widths vs Time for Different Xanthan Gum Concentrations', 'FontSize', 8, 'FontWeight', 'bold');
legend('show', 'Location', 'best');
grid on;
grid minor;
box on;
saveas(gcf, 'avgchannelwidthscm_timegap_figurenolegend.png');
hold off;
```

*Published with MATLAB® R2023b*