

## Appendix A – Dielectrophoresis Data Processing

This appendix provides an example of the custom script generated in MATLAB to detect the presence of oocysts within the microchannel of the DEP chip (Chapters 8 and 9). In the script, text of green colour indicates comments, which are given so that the reader can follow the data processing procedure. Thus, black text constitutes the custom script. An example of this script is as follows:

```
%CODE BEGINS%

%%% clear variables, commands and close figures
clear all
close all
clc

%%% specify file location
path = 'E:\DEP data Sept
16\060916_S0.2\060916_VP_S0.2_B0.6_30MHz_100mV_RMS\photo_1609
06_';

%%% define variables
no_im=2:10;          %%% number of images saved in folder
counter = 1;        %%% start of loop
threshold = 15;     %%% lower probability limit for detected
                    %%% differences
thresh2_high=50;   %%% upper pixel limit for detected
events

                    (n)
thresh2_low=10;    %%% lower pixel limit for detected
events
```

```

                                (n)
oocyst_check = 1;    %%% TURN ON to see detected oocysts

%%
    %%% read first image and calibrate
    Aref=imread(strcat(path,'001.bmp'));

    figure('units','normalized','outerposition',[0 0 1 1]);
    image(Aref);
    title('For calibration')    ;
    [x,y]=ginput(2);
    dist_um=400;

    %%% calibrate using known difference between top and
bottom
        walls
    cal=floor(max(y)-min(y));
    factor=dist_um/cal;

    %%% detect top wall in reference image
    figure('units','normalized','outerposition',[0 0 1 1]);
    image(Aref);
    title('For top wall detection');
    [x2,y2]=ginput(2);
    a1=(y2(2)-y2(1))/(x2(2)-x2(1));
    b1=y2(2)-a1*x2(2);
    hold on
    z=1:1360;
    plot(z,a1*z+b1);

    %%% detect bottom wall in reference image
    figure('units','normalized','outerposition',[0 0 1 1]);
    image(Aref);

```

```

title('For bottom wall detection');
[x3,y3]=ginput(2);
a2=(y3(2)-y3(1))/(x3(2)-x3(1));
b2=y3(2)-a2*x3(2);
hold on
z2=1:1360;
plot(z2,a2*z2+b2);

close figure 1;
close figure 2;
close figure 3;

%%

%%% read second image and subtract %%%

for i=2:length(no_im)

if no_im(i)<10
A=imread(strcat(path,'00',num2str(no_im(i)),'.bmp'));
elseif no_im(i)<100
A=imread(strcat(path,'0',num2str(no_im(i)),'.bmp'));
else
A=imread(strcat(path,num2str(no_im(i)),'.bmp'));
end

%%% subtracts each file from ref image
B = Aref-A;
B2=mean(B,3);

%%% identifies events/ changes with probability >
threshold
B3=B2>threshold;

```

```

%%% plots sub-image showing difference between two images
if oocyst_check == 1
figure(500+counter)
subplot (1,2,1)
imagesc(B2);

% plots sub-image showing detections > probability
threshold
subplot(1,2,2)
imagesc(B3);
end

%%% Converts image to binary and differences
with prob greater than threshold marked as 1
L=bwlabel(B3,8);
N=max(L(:));
C_crypto(:,:)=zeros(N, 2); % identifies events

%%% measures properties / gets stats of image regions
stats = regionprops(L, 'All');

%%% detects events with area (in pixels) < thresh2_low & >
thresh2_high
for n=1:N
if stats(n).Area>thresh2_low &&
stats(n).Area<thresh2_high;
C_crypto(n,1)=stats(n).Centroid(1);% gets x co-
ordinates
C_crypto(n,2)=stats(n).Centroid(2);% gets y co-
ordinates

%%% plot detected events

```

```

if oocyst_check == 1
figure(500+counter);
hold on
plot(C_crypto(n,1),C_crypto(n,2), 'x');
end

Cx = C_crypto(:,1);
Cy = C_crypto(:,2);

%%% remove events which are above upper boundary of
microchannel
for j = 1:length(Cy)
if Cy(j,1) <= b1
Cy(j,1) = NaN;
Cx(j,1) = NaN;
end

%%% remove events which are below lower boundary of
microchannel
if Cy(j,1) >= b2
Cy(j,1) = NaN;
Cx(j,1) = NaN;
end
end
end

end

%%% stores data from every loop iteration to new column
C_x(counter,1:length(Cx)) = Cx;
C_y(counter,1:length(Cy)) = Cy;

%%% clears variable and ensures continuation of loop

```

```

        iterations
clear C_crypto;
counter = counter + 1;

end

%%

%%% convert pixels to microns

C_x(C_x == 0) = NaN;    % firstly remove zeros from matrix
C_y(C_y == 0) = NaN;
C_x(isnan(C_x))=[];
C_y(isnan(C_y))=[];

XX_c = C_x(:);        % form single column of data
YY_c = C_y(:) - b1;   % make upper channel boundary data
origin

X_data = XX_c*factor; % convert x data to microns
Y_data = YY_c*factor; % convert y data to microns

% plot data and flips y-axis to imitate device design
figure(1100)
plot(X_data, Y_data, 'bo')
ylim([0 400]);
set(gca, 'Ydir', 'reverse')

%CODE ENDS%

```