

## EE492 Senior Design II - Weekly Report 12

Group Number: May1634	Date: 3/24/16 - 3/31/16
Project Name: Studying cell behaviors in 3D microtissues using a LabChip	
Advisor: Long Que	
Client: Long Que	

### The team

Role	Group Member
Group leader	Jonathan Yatkoske
Team Webmaster	Yaxiong Zhang
	Chun-Hao Lo
Team Communication Leader	Yuqian Hu
Team Key Concept Holder	Kaiyu Xu

### Attendance (meeting date: Mar. 27<sup>th</sup> 2015)

Jonathan Yatkoske	In person
Chun-Hao Lo	In person
Yaxiong Zhang	In person
Kaiyu Xu	In person
Yuqian Hu	In person

### Accomplishments of past week

1. Finish the PowerPoint for presentation.
2. Meet with Senior Design instructor. Give presentation and receive feedback.
  - a. Fasten the speed of our presentation;
  - b. Include our challenges throughout the process of the presentation.
  - c. Get rid of the time-line.
3. Improve cell tracking code and GUI

### Plan for coming week

1. Meet with our advisor and see if there is unsolved problem remains.
2. Improve our presentation. Prepare for the final presentation.

## Pending issues

Plans for group meeting to rework on the final presentation before dead week.

## Individual contributions

Jonathan Yatckoske	Work on coding; prepare for presentation
Chun-Hao Lo	website maintenance; code testing; prepare for presentation
Yaxiong Zhang	website maintenance; improve GUI; prepare for presentation
Kaiyu Xu	Take down meeting notes; prepare for presentation
Yuqian Hu	work on weekly report; prepare for presentation

## Individual hourly contributions

Name	Week Hours	Cumulative Hours
Jonathan Yatckoske	6	81.5
Chun-Hao Lo	5	67.5
Yaxiong Zhang	5	67
Kaiyu Xu	2	36
Yuqian Hu	4	57.5

## Appendix(Code)

### 1. findDroplets.m:

```
function [ centers, radii ] = findDroplets( image, min_radius, max_radius )
%findDroplets finds chambers with complete droplets on the LabChip device
% Uses the imfindcircles function to find the droplets within a radius
% range. Because imfindcircles sorts output by a metric that is useless
% for our purposes, this function then resorts the circles found by
% position in the image.
[centers_local, radii] = imfindcircles(image, [min_radius max_radius], 'Method', 'TwoStage');

[y_co,y_index] = sort(centers_local(:,2));

temp_i = sort(y_index);

temp = centers_local;

temp(temp_i) = centers_local(y_index); %sorts x-coordinate by ascending order of y-coordinates
temp(temp_i,2) = centers_local(y_index,2);

centers = temp;

end
```

## 2. findCells.m

```

function [ stats ] = findCells(X, centers, radii, radius, k, i)
%findCells using edge detection and image processing to locate the cells within the frame of the droplets
% final version of the function must iterate through the droplets
% identified by centers array

    rect = [centers(i,1)-radius centers(i,2)-radius 2*radius 2*radius];
    X2{1} = imresize(imcrop(X, rect),2.9,'bilinear');

    [-, threshold] = edge(X2{1}, 'canny');
    fudgeFactor = 0.9;
    Bws = edge(X2{1}, 'canny', threshold*fudgeFactor);

    se90 = strel('line',3,90);
    se0 = strel('line',3,0);

    BWsdil = imdilate(Bws, [se90,se0]);

    BWdfill = imfill(BWsdil, 'holes');

    BWnobord = imclearborder(BWdfill, 4);

    seD = strel('diamond',1);
    BWsmooth = imerode(BWnobord,seD);
    BWsmooth = imerode(BWsmooth,seD);

    BW_final = bwareaopen(BWsmooth, 300);
    figure(8), subplot(2,2,1), subimage(X), viscircles(centers,radii);
    subplot(2,2,2), subimage(X2{1});
    subplot(2,2,3), subimage(BW_final);

    stats = regionprops(BW_final, 'Centroid');

end

```

## 3. Code for cell-tracking:

close all

clear all

```
filename = uigetfile('*.tif');
```

```
info = imfinfo(filename);
```

```
num_images = numel(info);
```

```
bit_depth = info.BitDepth;
```

```
test = 0;
```

```
data={};
```

```
radius = 57;
```

```
for k = 1:num_images
```

```
    X = imread(filename, k);
```

```
    %if the image is RGB instead of grayscale, convert it before doing
```

```
    %anything else
```

```
    if bit_depth == 24
```

```
        X = rgb2gray(X);
```

```
    end
```

```
    %find the droplets
```

```
[centers, radii] = findDroplets(X,40,120);
```

```
%ignore any droplets with nothing in them
```

```
%do this by deleting the rows from centers and radii that have no cells
```

```
%found in the first frame
```

```
%TODO: write a function for this part
```

```
%find the cells in remaining droplets
```

```
if not(isempty(centers))
```

```
    for i = 1:length(centers)
```

```
        stats = findCells( X, centers, radii, radius, k, i);
```

```
        data{k,i} = stats;
```

```
    end
```

```
else
```

```
    data{k} = [];
```

```
end
```

```
end
```

#### 4.improved GUI code

```
1 function varargout = CellTrackerGUI(varargin)
2 % CELLTRACKERGUI MATLAB code for CellTrackerGUI.fig
3 % CELLTRACKERGUI, by itself, creates a new CELLTRACKERGUI or raises the existing
4 % singleton*.
5 %
6 % H = CELLTRACKERGUI returns the handle to a new CELLTRACKERGUI or the handle to
7 % the existing singleton*.
8 %
9 % CELLTRACKERGUI('CALLBACK',hObject,eventData,handles,...) calls the local
10 % function named CALLBACK in CELLTRACKERGUI.M with the given input arguments.
11 %
12 % CELLTRACKERGUI('Property','Value',...) creates a new CELLTRACKERGUI or raises the
13 % existing singleton*. Starting from the left, property value pairs are
14 % applied to the GUI before CellTrackerGUI_OpeningFcn gets called. An
15 % unrecognized property name or invalid value makes property application
16 % stop. All inputs are passed to CellTrackerGUI_OpeningFcn via varargin.
17 %
18 % *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
19 % instance to run (singleton)".
20 %
21 % See also: GUIDE, GUIDATA, GUIHANDLES
22
23 % Edit the above text to modify the response to help CellTrackerGUI
24
25 % Last Modified by GUIDE v2.5 27-Mar-2016 23:25:07
26
27 % Begin initialization code - DO NOT EDIT
28 gui_Singleton = 1;
29 gui_State = struct('gui_Name', mfilename, ...
30 'gui_Singleton', gui_Singleton, ...
31 'gui_OpeningFcn', @CellTrackerGUI_OpeningFcn, ...
32 'gui_OutputFcn', @CellTrackerGUI_OutputFcn, ...
33 'gui_LayoutFcn', [], ...
34 'gui_Callback', []);
35 if nargin && ischar(varargin{1})
36     gui_State.gui_Callback = str2func(varargin{1});
37 end
38
39 if nargin
40     [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
41 else
42     gui_mainfcn(gui_State, varargin{:});
43 end
44 % End initialization code - DO NOT EDIT
45
46
47 % --- Executes just before CellTrackerGUI is made visible.
48 function CellTrackerGUI_OpeningFcn(hObject, eventdata, handles, varargin)
49 % This function has no output args, see OutputFcn.
50 % hObject handle to figure
51 % eventdata reserved - to be defined in a future version of MATLAB
```

```

52 % handles    structure with handles and user data (see GUIDATA)
53 % varargin   command line arguments to CellTrackerGUI (see VARARGIN)
54
55 % Choose default command line output for CellTrackerGUI
56 handles.output = hObject;
57
58 % Update handles structure
59 guidata(hObject, handles);
60
61 % UIWAIT makes CellTrackerGUI wait for user response (see UIRESUME)
62 % uiwait(handles.figure1);
63
64
65 % --- Outputs from this function are returned to the command line.
66 function varargout = CellTrackerGUI_OutputFcn(hObject, eventdata, handles)
67 % varargout   cell array for returning output args (see VARARGOUT);
68 % hObject    handle to figure
69 % eventdata  reserved - to be defined in a future version of MATLAB
70 % handles    structure with handles and user data (see GUIDATA)
71
72 % Get default command line output from handles structure
73
74
75
76 % --- Executes on button press in loadButton.
77 function loadButton_Callback(hObject, eventdata, handles)
78 % hObject    handle to loadButton (see GCBO)
79 % eventdata  reserved - to be defined in a future version of MATLAB
80 % handles    structure with handles and user data (see GUIDATA)
81 global info;
82 global filename;
83 global num_images;
84 global test;
85 global data;
86 global radius;
87
88 filename = uigetfile('*.tif');
89 info = imfinfo(filename);
90 num_images = numel(info);
91 set(handles.StaticText,'string',filename);
92
93 test = 0;
94 data={};
95
96 radius = 57;
97
98
99
100 function StartPage_Callback(hObject, eventdata, handles)
101 % hObject    handle to StartPage (see GCBO)
102 % eventdata  reserved - to be defined in a future version of MATLAB
103 % handles    structure with handles and user data (see GUIDATA)
104
105 % Hints: get(hObject,'String') returns contents of StartPage as text
106 %        str2double(get(hObject,'String')) returns contents of StartPage as a double
107
108
109 % --- Executes during object creation, after setting all properties.
110 function StartPage_CreateFcn(hObject, eventdata, handles)
111 % hObject    handle to StartPage (see GCBO)
112 % eventdata  reserved - to be defined in a future version of MATLAB
113 % handles    empty - handles not created until after all CreateFcns called
114
115 % Hint: edit controls usually have a white background on Windows.
116 %        See ISPC and COMPUTER.
117 if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
118     set(hObject,'BackgroundColor','white');
119 end
120
121
122
123 function EndPage_Callback(hObject, eventdata, handles)
124 % hObject    handle to EndPage (see GCBO)
125 % eventdata  reserved - to be defined in a future version of MATLAB
126 % handles    structure with handles and user data (see GUIDATA)
127
128 % Hints: get(hObject,'String') returns contents of EndPage as text
129 %        str2double(get(hObject,'String')) returns contents of EndPage as a double
130
131
132 % --- Executes during object creation, after setting all properties.
133 function EndPage_CreateFcn(hObject, eventdata, -)
134 % hObject    handle to EndPage (see GCBO)
135 % eventdata  reserved - to be defined in a future version of MATLAB
136 % handles    empty - handles not created until after all CreateFcns called
137
138 % Hint: edit controls usually have a white background on Windows.
139 %        See ISPC and COMPUTER.
140 if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
141     set(hObject,'BackgroundColor','white');
142 end

```

```
145 % --- Executes on selection change in listBox2.
146 function listBox2_Callback(hObject, eventdata, handles)
147 % hObject    handle to listBox2 (see GCBO)
148 % eventdata  reserved - to be defined in a future version of MATLAB
149 % handles    structure with handles and user data (see GUIDATA)
150
151 % Hints: contents = cellstr(get(hObject,'String')) returns listBox2 contents as cell array
152 %         contents{get(hObject,'Value')} returns selected item from listBox2
153
154
155 % --- Executes during object creation, after setting all properties.
156 function listBox2_CreateFcn(hObject, eventdata, handles)
157 % hObject    handle to listBox2 (see GCBO)
158 % eventdata  reserved - to be defined in a future version of MATLAB
159 % handles    empty - handles not created until after all CreateFcns called
160
161 % Hint: listBox controls usually have a white background on Windows.
162 %       See ISPC and COMPUTER.
163 if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
164     set(hObject,'BackgroundColor','white');
165 end
166
167
168 % --- Executes on key press with focus on listBox2 and none of its controls.
169 function listBox2_KeyPressFcn(hObject, eventdata, handles)
170 % hObject    handle to listBox2 (see GCBO)
171 % eventdata  structure with the following fields (see MATLAB.UI.CONTROL.UICONTROL)
172 %             Key: name of the key that was pressed, in lower case
173 %             Character: character interpretation of the key(s) that was pressed
174 %             Modifier: name(s) of the modifier key(s) (i.e., control, shift) pressed
175 % handles    structure with handles and user data (see GUIDATA)
176
177
178 % --- Executes on button press in DoitButton.
179 function DoitButton_Callback(hObject, eventdata, handles)
180 % hObject    handle to DoitButton (see GCBO)
181 % eventdata  reserved - to be defined in a future version of MATLAB
182 % handles    structure with handles and user data (see GUIDATA)
183
184
185 % --- Executes on selection change in popupmenu1.
186 function popupmenu1_Callback(hObject, eventdata, handles)
187 % hObject    handle to popupmenu1 (see GCBO)
188 % eventdata  reserved - to be defined in a future version of MATLAB
189 % handles    structure with handles and user data (see GUIDATA)
190 global type;
191 contents=get(handles.popupmenu1,'value');
192 switch contents
193     case 1
194         type=0;
195     case 2
196         type=1;
```

```

197 end
198
199
200
201 % Hints: contents = cellstr(get(hObject,'String')) returns popupmenu contents as cell array
202 %       contents{get(hObject,'Value')} returns selected item from popupmenu
203
204
205 % --- Executes during object creation, after setting all properties.
206 function popupmenu1_CreateFcn(hObject, eventdata, handles)
207 % hObject    handle to popupmenu1 (see GCBO)
208 % eventdata  reserved - to be defined in a future version of MATLAB
209 % handles    empty - handles not created until after all CreateFcns called
210
211 % Hint: popupmenu controls usually have a white background on Windows.
212 %       See ISPC and COMPUTER.
213 if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
214     set(hObject,'BackgroundColor','white');
215 end
216

```

```

217
218 % --- Executes on button press in pushbutton3.
219 function pushbutton3_Callback(hObject, eventdata, handles)
220 % hObject    handle to pushbutton3 (see GCBO)
221 % eventdata  reserved - to be defined in a future version of MATLAB
222 % handles    structure with handles and user data (see GUIDATA)
223 global info;
224 global type;
225 global filename;
226 global num_images;
227 global test;
228 global data;
229 global radius;
230 global pausetime;
231 bit_depth = info.BitDepth;
232 for k = 1:num_images
233     X = imread(filename, k);
234     if bit_depth == 24
235         X = rgb2gray(X);
236     end
237     [centers, radii] = findDroplets(X,40,120);
238     %figure(1), imshow(X), viscircles(centers, radii);
239     if not(isempty(centers))
240         [X2{k}, BW_final, stats] = findCells( X, centers, radii, radius, k, 1);
241         data{k,1} = stats;
242     else
243         data{k} = [];
244         X2{k} = X2{k-1};
245     end

```

```

246     if k == 1
247         x_traj = [data{1}(1).Centroid(1)];
248         y_traj = [data{1}(1).Centroid(2)];
249     end
250     if not(isempty(data{k}))
251         x_traj = [x_traj data{k}(1).Centroid(1)];
252     end
253     if not(isempty(data{k}))
254         y_traj = [y_traj data{k}(1).Centroid(2)];
255     end
256
257     if type == 0
258         plot(handles.axes1,x_traj,300-y_traj,'ro');
259     end
260
261     if type == 1
262         plot(handles.axes1,x_traj,300-y_traj,'ro-');
263     end
264
265     axis(handles.axes1, [0 300 0 300]);
266     title(handles.axes1, 'Position');
267     xlabel(handles.axes1, 'Pixels')
268     ylabel(handles.axes1, 'Pixels')
269     imshow(X, 'Parent', handles.axes2), viscircles(handles.axes2, centers, radii);
270     label = sprintf('%.2f%% (%d/%d)', k/num_images*100, k, num_images);
271     set(handles.percentage, 'String', label);
272     imshow(BW_final, 'Parent', handles.axes3);
273     imshow(X2{k}, 'Parent', handles.axes4);
274     pause(pausetime);
275
276 end
277
278

```

```

279 function PauseTime_Callback(hObject, eventdata, handles)
280 % hObject    handle to PauseTime (see GCBO)
281 % eventdata  reserved - to be defined in a future version of MATLAB
282 % handles    structure with handles and user data (see GUIDATA)
283 global pausetime;
284 n=get(hObject,'string');
285 pausetime=str2double(n);
286
287
288 % Hints: get(hObject,'String') returns contents of PauseTime as text

```

```

289 %       str2double(get(hObject,'String')) returns contents of PauseTime as a double
290
291
292 % --- Executes during object creation, after setting all properties.
293 function PauseTime_CreateFcn(hObject, eventdata, handles)
294 % hObject    handle to PauseTime (see GCBO)
295 % eventdata  reserved - to be defined in a future version of MATLAB

```

```

296 % handles empty - handles not created until after all CreateFcns called
297
298 % Hint: edit controls usually have a white background on Windows.
299 % See ISPC and COMPUTER.
300 if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
301     set(hObject,'BackgroundColor','white');
302 end
303
304
305 function [ X2, BW_final, stats ] = findCells(X, centers, radii, radius, k, i)
306 %findCells using edge detection and image processing to locate the cells within the frame of the droplets
307 % final version of the function must iterate through the droplets
308 % identified by centers array
309
310     rect = [centers(i,1)-radius centers(i,2)-radius 2*radius 2*radius];
311     X2 = imresize(imcrop(X, rect),2.9,'bilinear');
312
313     [-, threshold] = edge(X2, 'canny');
314     fudgeFactor = 0.9;
315     BWs = edge(X2,'canny',threshold*fudgeFactor);
316
317     se90 = strel('line',3,90);
318     se0 = strel('line',3,0);
319
320     BWsdil = imdilate(BWs, [se90,se0]);
321
322     BWdfill = imfill(BWsdil, 'holes');
323
324     BWnobord = imclearborder(BWdfill, 4);
325
326     seD = strel('diamond',1);
327     BWsmooth = imerode(BWnobord,seD);
328     BWsmooth = imerode(BWsmooth,seD);
329
330     BW_final = bwareaopen(BWsmooth, 300);
331
332     stats = regionprops(BW_final, 'Centroid');
333
334
335
336 function [ centers, radii ] = findDroplets( image, min_radius, max_radius )
337 %findDroplets finds chambers with complete droplets on the LabChip device
338 % Uses the imfindcircles function to find the droplets within a radius
339 % range. Because imfindcircles sorts output by a metric that is useless
340 % for our purposes, this function then resorts the circles found by
341 % position in the image.
342 [centers_local, radii] = imfindcircles(image, [min_radius max_radius], 'Method', 'TwoStage');
343
344 if not(isempty(centers_local))
345     %sort by y
346     [y_co,y_index] = sort(centers_local(:,2));
347
348     temp_i = sort(y_index);
349
350     temp = centers_local;
351
352     temp(temp_i) = centers_local(y_index); %sorts x-coordinate by ascending order of y-coordinates
353     temp(temp_i,2) = centers_local(y_index,2);
354
355     %determine lengths of rows by finding first and last index of y within
356     %ranges
357
358     %then sort by x
359
360
361     centers = temp;
362 else
363     centers = [];
364 end

```