

EE492 Senior Design II - Weekly Report 14

Group Number: May1634	Date: 4/7/16 - 4/14/16
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 Yatcoske
Team Webmaster	Yaxiong Zhang
	Chun-Hao Lo
Team Communication Leader	Yuqian Hu
Team Key Concept Holder	Kaiyu Xu

Attendance (meeting date: Apr. 7th 2015)

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

Accomplishments of past week

1. Improved celltrackGUI code
2. Finished poster for final presentation

Plan for coming week

1. Meeting with adviser. Go through the presentation before the official final presentation
2. Finalize the powerpoint for final presentation.

Pending issues

Finalize the poster:

1. Remark the flowchart to match the color scheme of the poster

2. Change the color and fonts used in the GUI to match the aesthetic of the poster

Individual contributions

Jonathan Yatckoske	Work on the poster; improve the code
Chun-Hao Lo	website maintenance; improve the layout of the poster
Yaxiong Zhang	website maintenance; change the timeline and flow chart in he poster
Kaiyu Xu	Take down meeting notes
Yuqian Hu	work on weekly report; work on the poster

Individual hourly contributions

Name	Week Hours	Cumulative Hours
Jonathan Yatckoske	10	95.5
Chun-Hao Lo	6	76.5
Yaxiong Zhang	6	76
Kaiyu Xu	1	38
Yuqian Hu	4	64.5

Appendix:

(CellTrackerGUI.m Updated)

```

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 17-Apr-2016 17:58:16
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

```

```

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 assignin('base','filename',filename);
90 info = imfinfo(filename);
91 assignin('base','info',info);
92 num_images = numel(info);
93 assignin('base','num_images',num_images);
94 set(handles.StaticText,'string',filename);
95
96 test = 0;
97 data={};
98
99 radius = 57;
100
101
102
103 function StartPage_Callback(hObject, eventdata, handles)
104 % hObject    handle to StartPage (see GCBO)
105 % eventdata  reserved - to be defined in a future version of MATLAB
106 % handles    structure with handles and user data (see GUIDATA)
107
108 % Hints: get(hObject,'String') returns contents of StartPage as text
109 %        str2double(get(hObject,'String')) returns contents of StartPage as a double

```

```

113 % --- Executes during object creation, after setting all properties.
114 function StartPage_CreateFcn(hObject, eventdata, handles)
115 % hObject    handle to StartPage (see GCBO)
116 % eventdata  reserved - to be defined in a future version of MATLAB
117 % handles    empty - handles not created until after all CreateFcns called
118
119 % Hint: edit controls usually have a white background on Windows.
120 %         See ISPC and COMPUTER.
121 if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
122     set(hObject,'BackgroundColor','white');
123 end
124
125
126
127 function EndPage_Callback(hObject, eventdata, handles)
128 % hObject    handle to EndPage (see GCBO)
129 % eventdata  reserved - to be defined in a future version of MATLAB
130 % handles    structure with handles and user data (see GUIDATA)
131
132 % Hints: get(hObject,'String') returns contents of EndPage as text
133 %        str2double(get(hObject,'String')) returns contents of EndPage as a double
134
135
136 % --- Executes during object creation, after setting all properties.
137 function EndPage_CreateFcn(hObject, eventdata, ~)
138 % hObject    handle to EndPage (see GCBO)
139 % eventdata  reserved - to be defined in a future version of MATLAB
140 % handles    empty - handles not created until after all CreateFcns called
141
142 % Hint: edit controls usually have a white background on Windows.
143 %         See ISPC and COMPUTER.
144 if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
145     set(hObject,'BackgroundColor','white');
146 end

```

```

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

```

```

189 % --- Executes on selection change in popupmenu1.
190 function popupmenu1_Callback(hObject, eventdata, handles)
191 % hObject    handle to popupmenu1 (see GCBO)
192 % eventdata  reserved - to be defined in a future version of MATLAB
193 % handles    structure with handles and user data (see GUIDATA)
194 global type;
195 contents=get(handles.popupmenu1,'value');
196 switch contents
197     case 1
198         type=0;
199     case 2
200         type=1;
201 end
202
203
204
205 % Hints: contents = cellstr(get(hObject,'String')) returns popupmenu1 contents as cell array
206 %         contents{get(hObject,'Value')} returns selected item from popupmenu1
207
208
209 % --- Executes during object creation, after setting all properties.
210 function popupmenu1_CreateFcn(hObject, eventdata, handles)
211 % hObject    handle to popupmenu1 (see GCBO)
212 % eventdata  reserved - to be defined in a future version of MATLAB
213 % handles    empty - handles not created until after all CreateFcns called
214
215 % Hint: popupmenu controls usually have a white background on Windows.
216 %         See ISPC and COMPUTER.
217
217 if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
218     set(hObject,'BackgroundColor','white');
219 end

```

```

222 % --- Executes on button press in pushbutton3.
223 function pushbutton3_Callback(hObject, eventdata, handles)
224 % hObject    handle to pushbutton3 (see GCBO)
225 % eventdata  reserved - to be defined in a future version of MATLAB
226 % handles    structure with handles and user data (see GUIDATA)
227 global info;
228 global type;
229 global filename;
230 global num_images;
231 global test;
232 global data;
233 global radius;
234 global pausetime;
235 bit_depth = info.BitDepth;
236
237 colorsForTrajPlot = {'r','g','b','y','m','c'};
238
239 X = imread(filename, 1);
240 [centers, radii] = findDroplets(X,50,100);
241 [numDroplets, trash] = size(centers);
242
243 label = sprintf('Droplets(1/%d)', numDroplets);
244 set(handles.chamberprog, 'String', label);
245
246 percentageChange = 1/(numDroplets*num_images);
247 percentage = 0;
248
249     x_traj = cell(150,1); xtrajreal = cell(150,1);
250     y_traj = cell(150,1); ytrajreal = cell(150,1);
251     lastx = cell(10,1);
252     lasty = cell(10,1);
253     droppedframes = cell(10,1);
254
255 for i = 1:numDroplets
256     i
257     empty = 0;
258     TotCells = 0;

```

```

260 %Check if the droplet has any cells
261 for k = 1:5
262     X = imread(filename, k);
263
264     if bit_depth==24
265         X = rgb2gray(X);
266     end
267
268     [centers, radii] = findDroplets(X,50,100);
269     [X2{k}, BW final, stats, centers_loc, rads] = findCells( X, centers, radii, radius, k, i);
270     if isempty(centers_loc)
271         empty = empty + 1;
272     end
273     imshow(X,'Parent',handles.axes2), viscircles(handles.axes2, centers, radii);
274     testlabel = sprintf('Testing(%d/5)', k);
275     set(handles.frameprog, 'String', testlabel);
276     label = sprintf('Droplet (%d/%d)', i, numDroplets);
277     set(handles.chamberprog, 'String', label);
278     imshow(BW final,'Parent',handles.axes3), viscircles(handles.axes3, centers_loc, rads);
279     imshow(X2{k},'Parent',handles.axes4);
280     label = sprintf('Progress: %.2f%%', percentage*100);
281     set(handles.percentage, 'String', label);
282     pause(pausetime);
283 end
284
285
286
287 %if the majority of the first 5 frames aren't empty, process all frames
288 %for trajectory plots and data

```

```

289     if (empty < 3)
290         for k = 1:num_images
291             X = imread(filename, k);
292
293             if bit_depth == 24
294                 X = rgb2gray(X);
295             end

```

```

297     [centers, radii] = findDroplets(X,50,100);
298     [numCenters trash] = size(centers);
299
300     if (numCenters==numDroplets)
301         [X2{k}, BW final, stats, centers_loc, rads] = findCells( X, centers, radii, radius, k, i);
302         data{k} = centers_loc;
303     else
304         data{k} = [];
305         if (k>1)
306             X2{k} = X2{k-1};
307         else
308             X2{k} = [];
309         end
310     end
311
312     [numCells, trash] = size(data{k});
313     if (numCells>TotCells)
314         TotCells = numCells;
315     end
316
317     if not(isempty(data{k}))
318         for j=1:numCells
319             testlast = max(find(~cellfun('isempty',lastx)));
320             assignin('base','k',k); assignin('base','lastx',lastx); assignin('base','lasty',lasty); assign
in('base','j',j); assignin('base','numCells',numCells);
321             if (k>1)&&(j<testlast)
322                 if ((abs(data{k}(j,1)-lastx{j}) < 5) & (abs(data{k}(j,2)-lasty{j}) < 5))
323                     x_traj{j} = [x_traj{j} data{k}(j,1)];
324                     y_traj{j} = [y_traj{j} data{k}(j,2)];
325                 end
326             elseif j==testlast
327                 x_traj{j} = [x_traj{j} data{k}(j,1)];
328                 y_traj{j} = [y_traj{j} data{k}(j,2)];
329             else
330                 droppedframes{j} = droppedframes{j} + 1;
331             end
332         end
333         lastx{j} = data{k}(j,1); lasty{j} = data{k}(j,2);
334     end

```

```

336         end
337
338         if numel(x_traj)>=1
339             if type==0
340                 for j=1:TotCells
341                     plot(handles.axes1,x_traj{j}.*(120/334),120.-(y_traj{j}.*(120/334)), 'o-', 'Color', colorsF
orTrajPlot{j});
342                 end
343             end
344
345             if type==1
346                 %edit this to make origin-centered plots
347                 for j=1:TotCells
348                     plot(handles.axes1,-334.+x_traj{j}.*(334/120),120.-(y_traj{j}.*(334/120)), 'o-', 'color', col
orsForTrajPlot{j});
349                 end
350             end
351         end
352
353
354         axis(handles.axes1, [0 120 0 120]);
355         title(handles.axes1, 'Position')
356         xlabel(handles.axes1, '\mum')
357         ylabel(handles.axes1, '\mum')

```

```

358         hold(handles.axes1, 'on')
359
360
361
362         imshow(X,'Parent',handles.axes2), viscircles(handles.axes2, centers, radii);
363         label = sprintf('Frames(%d/%d)', k, num_images);
364         set(handles.frameprog, 'String', label);
365         percentage = percentage+percentageChange;
366         label = sprintf('Progress: %.2f%%', percentage*100);
367         set(handles.percentage,'String',label);
368         imshow(BW_final,'Parent',handles.axes3), viscircles(handles.axes3, centers_loc, rads);
369         imshow(X2{k},'Parent',handles.axes4);
370         label = sprintf('Droplets(%d/%d)', i, numDroplets);

```

```

371         set(handles.chamberprog,'String',label);
372         pause(pausetime);
373
374         assignin('base','data',data);
375         assignin('base','x_traj',x_traj);
376         assignin('base','y_traj',y_traj);
377         assignin('base','X2', X2);
378     end
379 else
380     percentage = percentage + percentageChange*num_images;
381 end
382
383
384 end
385
386
387 function PauseTime Callback(hObject, eventdata, handles)
388 % hObject    handle to PauseTime (see GCBO)
389 % eventdata  reserved - to be defined in a future version of MATLAB
390 % handles    structure with handles and user data (see GUIDATA)
391 global pausetime;
392 n=get(hObject,'string');
393 pausetime=str2double(n);
394
395
396 % Hints: get(hObject,'String') returns contents of PauseTime as text
397 %       str2double(get(hObject,'String')) returns contents of PauseTime as a double
398
399
400 % --- Executes during object creation, after setting all properties.
401 function PauseTime CreateFcn(hObject, eventdata, handles)
402 % hObject    handle to PauseTime (see GCBO)
403 % eventdata  reserved - to be defined in a future version of MATLAB
404 % handles    empty - handles not created until after all CreateFcns called
405
406 % Hint: edit controls usually have a white background on Windows.
407 %       See ISPC and COMPUTER.
408 if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
409     set(hObject,'BackgroundColor','white');

```

```

410 end
411
412
413 function [ X2, BW final, stats, centers loc, rads ] = findCells(X, centers, radii, radius, k, i)
414 %findCells using edge detection and image processing to locate the cells within the frame of the droplets
415 % final version of the function must iterate through the droplets
416 % identified by centers array
417     rect = [centers(i,1)-radius centers(i,2)-radius 2*radius 2*radius];
418     X2 = imresize(imcrop(X, rect),2.9,'bilinear');
419
420     [-, threshold] = edge(X2, 'canny');
421     fudgeFactor = 1.2;
422     Bws = edge(X2,'canny',threshold*fudgeFactor);
423
424     se90 = strel('line',3,90);
425     se0 = strel('line',3,0);
426
427     Bwsdil = imdilate(Bws, [se90,se0]);
428
429     Bwdfill = imfill(Bwsdil, 'holes');
430
431     Bwnobord = imclearborder(Bwdfill, 4);
432
433     seD = strel('diamond',1);
434     BWsmooth = imerode(Bwnobord,seD);
435     BWsmooth = imerode(BWsmooth,seD);
436
437     BW_final = bwareaopen(BWsmooth, 500);
438
439     [centers loc, rads] = imfindcircles(BW_final, [20 50], 'Method', 'TwoStage');
440     %figure(1), imshow(BW_final), viscircles(centers_loc, rads);
441
442     stats = regionprops(BW_final, 'Centroid');
443     ...
444 function [ centers, radii ] = findDroplets( image, min_radius, max_radius )
445 %findDroplets finds chambers with complete droplets on the LabChip device
446 % Uses the imfindcircles function to find the droplets within a radius
447 % range. Because imfindcircles sorts output by a metric that is useless
448 % for our purposes, this function then resorts the circles found by
449 % position in the image.
450 [centers_local, radii] = imfindcircles(image, [min_radius max_radius], 'Method', 'TwoStage');
451
452 if not(isempty(centers_local))
453     %sort by y
454     [y_co,y_index] = sort(centers_local(:,2));
455
456     temp_i = sort(y_index);
457
458     temp = centers_local;
459
460     temp(temp_i) = centers_local(y_index); %sorts x-coordinate by ascending order of y-coordinates
461     temp(temp_i,2) = centers_local(y_index,2);
462
463     %then sort by x
464     if (length(temp)>2)
465         for j = 1:length(temp)-1
466             for i=1:length(temp)-j
467                 if (temp(i+1,2)-temp(i,2) < 100)
468                     if (temp(i+1,1) < temp(i,1))
469                         holdx = temp(i,1); holdy = temp(i,2);
470                         temp(i,1) = temp(i+1,1); temp(i,2) = temp(i+1,2);
471                         temp(i+1,1) = holdx; temp(i+1,2) = holdy;
472                     end
473                 end
474             end
475         end
476     end
477     centers = temp;
478 else
479     centers = [];
480 end
481
482
483
484
485
486 end

```