**Program report:**

**Contents**

**1. Introduction**

**2. Technical approach**

A. Graphical User Interface (GUI)

i. Part 1:Annotate coordinates

ii. Part 2:Examination

**3. Appendix**

A. GUI user manual

B. Program code with instructions

**1.Introduction**

Intensity-modulated radiation therapy (IMRT) is a targeted treatment currently widely used in curing cancer. It could target at the cancer area and thus bringing less harm to surrounding normal areas. However, with the interference of involuntary motion, including respiration, cardiac activity, we have to target at the whole areas that cancer may move around, which also harms the normal tissues during treatments. What doctors can do now to decrease the negative impacts on normal cells is holding patients’ breath by a specific kind of machine or using x-ray computerized tomography (CT) to track the trace of cancer, but both methods have their own limitations. Holding patients’ breath is a complex work as the treatment has to repeat several times and it not applies to the patients who have disease of lungs. In addition, using CT to assist treatments creates radiation to the whole body and the cost of the equipments is extremely expensive. Thus, our project wants to develop a treatment method that is much cheaper and could apply to all the patients. We plan to use ultrasound to track the movement of cancer and then guide the radiation therapy to follow and target at the cancer. Our project could help users to manually annotate the coordinates of cancers and examine the correctness of annotated coordinates in a more convenient way.

**2. Technical approach**

**Graphical User Interface (GUI)**

A graphical user interface(GUI) is developed to annotate the coordinates of vessels for helping examine the correctness of template\_matching used to track the moving cancer. Detailed information on the structure of the program and on how to use the program is in the Appendix A.

**i. Part 1:Annotate coordinates**

Part 1 is developed to help annotate and save the coordinates of patients’ vessels during free breathing directly on the interface.

**ii. Part 2:Examination**

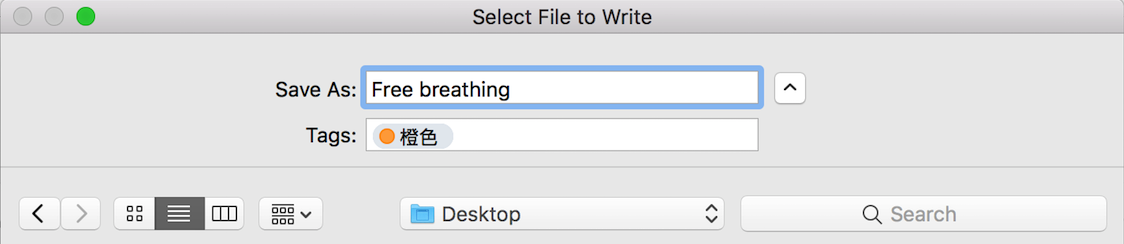
Part 2 is used to plot the x- or y-coordinates figures to help find the abnormal parts of coordinates, which may possibly contain errors when annotating them.

**3.Appendix**

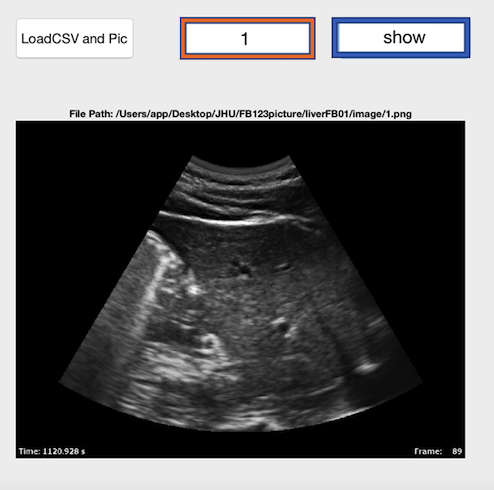
**A.GUI user manual**

**1. Part 1: Annotate coordinates**

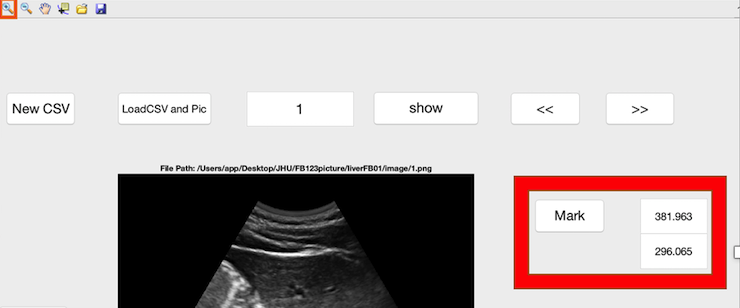
**a)**First Key in the total number of frames in the edit text ‘picture num’, and click the button ‘New CSV’ to create a new .csv file.

**b)**Write down the name of the new created file and choose the path of new file in the pop-up window.

**c)**Click the edit text ‘LoadCSV and Pic’. First choose the new created file in the first pop-up window, and then in the second pop-up window choose the images that will be annotated.

**d)**Key in which picture you want to start with in edit text ‘picture num’, and then click the button ‘show’ to load the picture you choose. The picture path will be shown on top of the picture.

**e)** Click the button “load template”, and then annotate the vessel precisely to form a “template”, which will be shown on the figure below the button “load template”. The template is used to examine the correctness of following annotated coordinates, and the correctness is shown in the edit text on the right of button “load template”.

**f)**Click the button “Mark” and mark the coordinate. You could also use “Zoom In” to see one part of the picture more clearly. The coordinates will be shown on the two edit text next to “Mark”, and these two coordinates will be saved in the created file automatically.

**g)**Clicking “<<” or “>>” button could load the last or next frame.

**2. Part 2: Examination**

**a)**Click the edit text ‘LoadCSV and Pic’. First choose the file containing the annotated coordinates of one patient in the first pop-up window, and then choose the images of the same patient in the second pop-up window.

**b)**Click the ‘plot X’ or ‘plot Y’ button at the left bottom corner to get the figure of x- or y-coordinates of one patient.

**c)**Observe the abnormal part of the figure, and the abnormality may be caused by the errors when annotating or other normal influence from the patient, but we should examine all of them to guarantee the coordinates are completely correct.

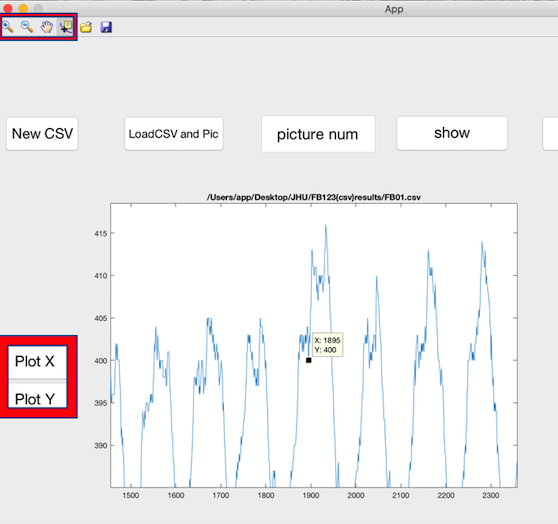
**d)**Click the “Zoom In” or “Zoom Out ” button to see the abnormalities clearly.

**e)**Using “Data Cursor” to figure out the coordinates of the frames which seem problematic.

**f)**Key in the frames which seem problematic in the edit text “picture num”, and click “show” the get the image and original annotated coordinates.

**g)**If you want to change the original coordinate, click the button “Mark” and mark the new coordinate. You could also use “Zoom In” to see one part of the picture more clearly. The new coordinates will be shown on the two edit text next to “Mark”, and these two coordinates will be updated automatically.

**h)**Clicking “<<” or “>>” button could load the last or next frame.



**B. Program code with instructions**

(Instructions are highlighted in orange)

function varargout = App(varargin)

% APP MATLAB code for App.fig

% APP, by itself, creates a new APP or raises the existing

% singleton\*.

%

% H = APP returns the handle to a new APP or the handle to

% the existing singleton\*.

%

% APP('CALLBACK',hObject,eventData,handles,...) calls the local

% function named CALLBACK in APP.M with the given input arguments.

%

% APP('Property','Value',...) creates a new APP or raises the

% existing singleton\*. Starting from the left, property value pairs are

% applied to the GUI before App\_OpeningFcn gets called. An

% unrecognized property name or invalid value makes property application

% stop. All inputs are passed to App\_OpeningFcn via varargin.

%

% \*See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one

% instance to run (singleton)".

%

% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help App

% Last Modified by GUIDE v2.5 28-Jul-2017 09:54:01

% Begin initialization code - DO NOT EDIT

gui\_Singleton = 1;

gui\_State = struct('gui\_Name', mfilename, ...

'gui\_Singleton', gui\_Singleton, ...

'gui\_OpeningFcn', @App\_OpeningFcn, ...

'gui\_OutputFcn', @App\_OutputFcn, ...

'gui\_LayoutFcn', [] , ...

'gui\_Callback', []);

if nargin && ischar(varargin{1})

gui\_State.gui\_Callback = str2func(varargin{1});

end

if nargout

[varargout{1:nargout}] = gui\_mainfcn(gui\_State, varargin{:});

else

gui\_mainfcn(gui\_State, varargin{:});

end

% End initialization code - DO NOT EDIT

% --- Executes just before App is made visible.

function App\_OpeningFcn(hObject, eventdata, handles, varargin)

% This function has no output args, see OutputFcn.

% hObject handle to figure

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% varargin command line arguments to App (see VARARGIN)

% Choose default command line output for App

handles.output = hObject;

% Update handles structure

guidata(hObject, handles);

% UIWAIT makes App wait for user response (see UIRESUME)

% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.

function varargout = App\_OutputFcn(hObject, eventdata, handles)

% varargout cell array for returning output args (see VARARGOUT);

% hObject handle to figure

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure

varargout{1} = handles.output;

function Picture\_Callback(hObject, eventdata, handles)

% hObject handle to Picture (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Picture as text

% str2double(get(hObject,'String')) returns contents of Picture as a double

% --- Executes during object creation, after setting all properties.

function Picture\_CreateFcn(hObject, eventdata, handles)

% hObject handle to Picture (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

% See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

% --- Executes on button press in show.

function show\_Callback(hObject, eventdata, handles)

%%This function is used to load the picture and plot the annotated coordinate.

% hObject handle to show (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

num =str2num(get(handles.Picture,'String'));

guidata(hObject,handles);

ImagePath = [handles.PicFolderPath,num2str(num),'.png'];

csvMatrix=csvread(handles.CSVFilePath);

x=csvMatrix(num,1);

y=csvMatrix(num,2);

I = imread(ImagePath);

axes(handles.axes5);

imshow(I);

hold on;

plot(x,y,'r+','MarkerSize',5);

display = ['File Path: ', ImagePath];

title(display, 'interpreter','none')

hold off;

% --- Executes on button press in pushbutton2.

function pushbutton2\_Callback(hObject, eventdata, handles)

%%This function is used to load the next picture.

% hObject handle to pushbutton2 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

P=str2num(get(handles.Picture,'String'))+1;

set(handles.Picture,'String',P);

ImagePath = [handles.PicFolderPath,num2str(P),'.png'];

csvMatrix=csvread(handles.CSVFilePath);

x=csvMatrix(P,1);

y=csvMatrix(P,2);

I = imread(ImagePath);

axes(handles.axes5);

imshow(I);

hold on;

plot(x,y,'r+','MarkerSize',5);

display = ['File Path: ', ImagePath];

title(display, 'interpreter','none')

hold off;

% --- Executes on button press in pushbutton3.

function pushbutton3\_Callback(hObject, eventdata, handles)

%%This function is used to load the last picture.

% hObject handle to pushbutton3 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

D=str2num(get(handles.Picture,'String'))-1;

set(handles.Picture,'String',D);

set(handles.Picture,'String',D);

ImagePath = [handles.PicFolderPath,num2str(D),'.png'];

csvMatrix=csvread(handles.CSVFilePath);

x=csvMatrix(D,1);

y=csvMatrix(D,2);

I = imread(ImagePath);

axes(handles.axes5);

imshow(I);

hold on;

plot(x,y,'r+','MarkerSize',5);

display = ['File Path: ', ImagePath];

title(display, 'interpreter','none')

hold off;

% --- Executes on button press in pushbuttonX.

function pushbuttonX\_Callback(hObject, eventdata, handles)

%%This function is used to plot all the annoatated x-coordinates of one patient

% hObject handle to pushbuttonX (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

csvMatrix=csvread(handles.CSVFilePath);

x=csvMatrix(:,1);

y=csvMatrix(:,2);

axes(handles.axes5);

plot(x);

title(handles.CSVFilePath);

% --- Executes on button press in pushbuttonY.

function pushbuttonY\_Callback(hObject, eventdata, handles)

%%This function is used to plot all the annoatated y-coordinates of one patient

% hObject handle to pushbuttonY (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

csvMatrix=csvread(handles.CSVFilePath);

x=csvMatrix(:,1);

y=csvMatrix(:,2);

axes(handles.axes5);

plot(y);

title(handles.CSVFilePath);

% --- Executes on button press in pushbuttonMark.

function pushbuttonMark\_Callback(hObject, eventdata, handles)

%%This function is used to mark and save the coordinates as well as show the marked coordinates in the edit texts.

% hObject handle to pushbuttonMark (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

Z=ginput(1);

x1=Z(1,1);

y1=Z(1,2);

set(handles.X,'String',x1);

set(handles.Y,'String',y1);

N=str2num(get(handles.Picture,'string'));

csvMatrix=csvread(handles.CSVFilePath);

csvMatrix(N,1) = x1;

csvMatrix(N,2) = y1;

csvwrite(handles.CSVFilePath, csvMatrix);

ImagePath = [handles.PicFolderPath,num2str(N),'.png'];

csvMatrix=csvread(handles.CSVFilePath);

x=csvMatrix(N,1);

y=csvMatrix(N,2);

I = imread(ImagePath);

axes(handles.axes5)

imshow(I);

hold on;

plot(x,y,'r+','MarkerSize',5);

display = ['File Path: ', ImagePath];

title(display, 'interpreter','none')

hold off;

yr\_half = 40;

xr\_half = 40;

xcent(1) = round(x);

ycent(1) = round(y);

im1 = double(I);

%im\_roi = im(ycent(1)-yr\_half:ycent(1)+yr\_half,xcent(1)-xr\_half:xcent(1)+xr\_half);

im\_nt = im1(ycent(1)-yr\_half:ycent(1)+yr\_half,xcent(1)-xr\_half:xcent(1)+xr\_half);

%imshow(im\_nt), caxis auto;

im\_roi=handles.im\_roi;

C = template\_matching2(im\_roi,im\_nt);

K=num2str(C(xr\_half,yr\_half));

set(handles.edit4,'String',K);

function X\_Callback(hObject, eventdata, handles)

% hObject handle to X (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of X as text

% str2double(get(hObject,'String')) returns contents of X as a double

% --- Executes during object creation, after setting all properties.

function X\_CreateFcn(hObject, eventdata, handles)

% hObject handle to X (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

% See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

function Y\_Callback(hObject, eventdata, handles)

% hObject handle to Y (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Y as text

% str2double(get(hObject,'String')) returns contents of Y as a double

% --- Executes during object creation, after setting all properties.

function Y\_CreateFcn(hObject, eventdata, handles)

% hObject handle to Y (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

% See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

% --- Executes on button press in pushbutton9.

function pushbutton9\_Callback(hObject, eventdata, handles)

%%This function is used to create a new .csv file.

num =str2num(get(handles.Picture,'String'));

[file, path] = uiputfile('\*.csv');

name = fullfile(path,file);

emptyMatrix = zeros(num,2);

csvwrite(name,emptyMatrix);

% hObject handle to pushbutton9 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% --- Executes on button press in pushbutton10.

function pushbutton10\_Callback(hObject, eventdata, handles)

%%This function is used to choose the file path as well as the image path.

%this function is to load csv first and then the pic to get the file path

% hObject handle to pushbutton10 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

num =1;

[file,path, index]=uigetfile('\*.csv');

handles.CSVFilePath = fullfile(path,file);

[file,path, index]=uigetfile({'\*.jpg;\*.tif;\*.png;\*.gif','All Image Files';...

'\*.\*','All Files' },'mytitle',...

'C:\Work\setpos1.png');

handles.PicFolderPath = path;

guidata(hObject,handles);

display = ['Pic Path: ', handles.PicFolderPath, '\n', 'CSV Path', handles.CSVFilePath];

% --- Executes on button press in button\_template.

function button\_template\_Callback(hObject, eventdata, handles)

%%This function is used to select the template.

% hObject handle to button\_template (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

yr\_half = 40;

xr\_half = 40;

D=str2num(get(handles.Picture,'String'));

ImagePath = [handles.PicFolderPath,num2str(D),'.png'];

[x,y] = ginput(1); % read in coordinates of POI

xcent(1) = round(x);

ycent(1) = round(y);

%I = imread(ImagePath);

im = double(imread(ImagePath));

%im1 = double(I);

im\_roi = im(ycent(1)-yr\_half:ycent(1)+yr\_half,xcent(1)-xr\_half:xcent(1)+xr\_half);

handles.im\_roi = im\_roi;

guidata(hObject,handles);

axes(handles.axes4);

imshow(handles.im\_roi), caxis auto;

function edit4\_Callback(hObject, eventdata, handles)

% hObject handle to edit4 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text

% str2double(get(hObject,'String')) returns contents of edit4 as a double

% --- Executes during object creation, after setting all properties.

function edit4\_CreateFcn(hObject, eventdata, handles)

% hObject handle to edit4 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

% See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end