生物医学工程实验

课程设计

基于 Matlab GUI 的医学图像处理及分析系统

作者：MingyangXue

邮箱：1537361071@qq.com

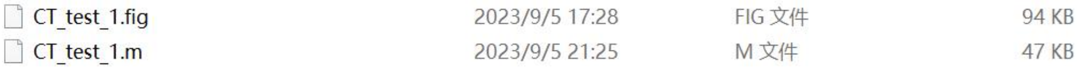
一、 系统功能概述

读取并显示所给 CT 或 PET 图像序列（横断面、矢状面和冠状面）；显示窗的选择；图像中感兴趣区域（ROI）的人工勾画，ROI 内 CT 值信息（均值、最大值、最小值、面积等）；CT，PET 图像的融合；融合图像浏览；权重选择。

二、 系统的使用说明（图文并茂）

1. 打开程序

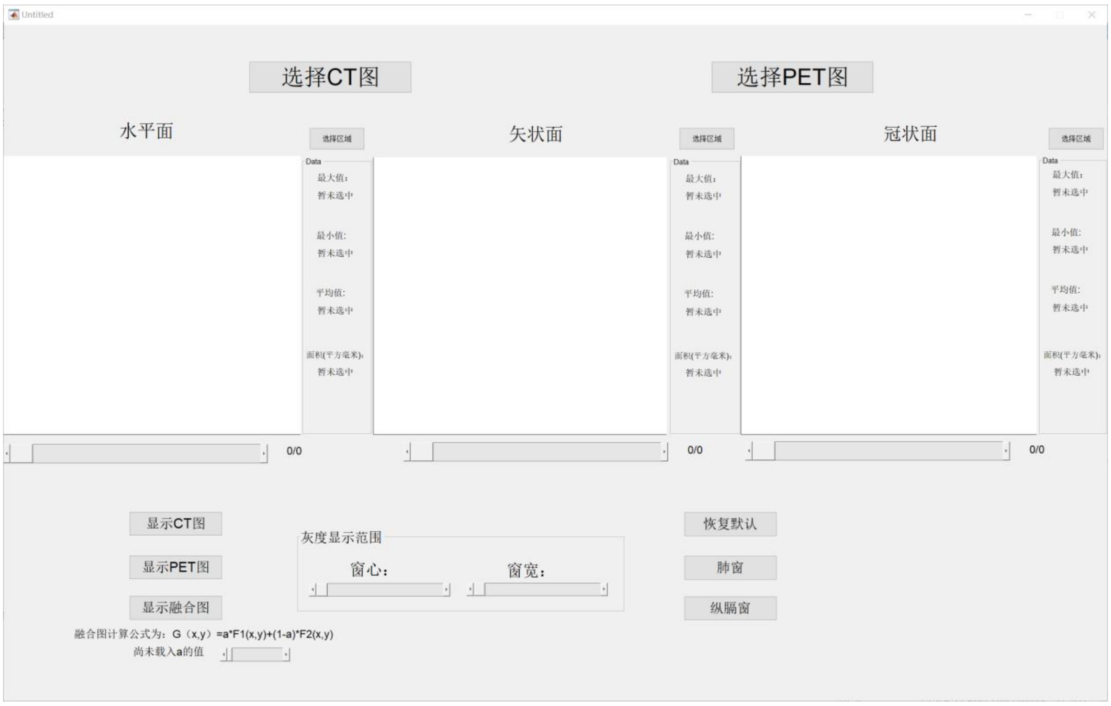
选择 CT_test_1.m 文件用 matlab2021 及以上版本启动，选择 CT_test_1.fig 文件用 matlab 内置的 GUIDE 启动。



CT_test_1.fig	2023/9/5 17:28	FIG 文件	94 KB
CT_test_1.m	2023/9/5 21:25	M 文件	47 KB

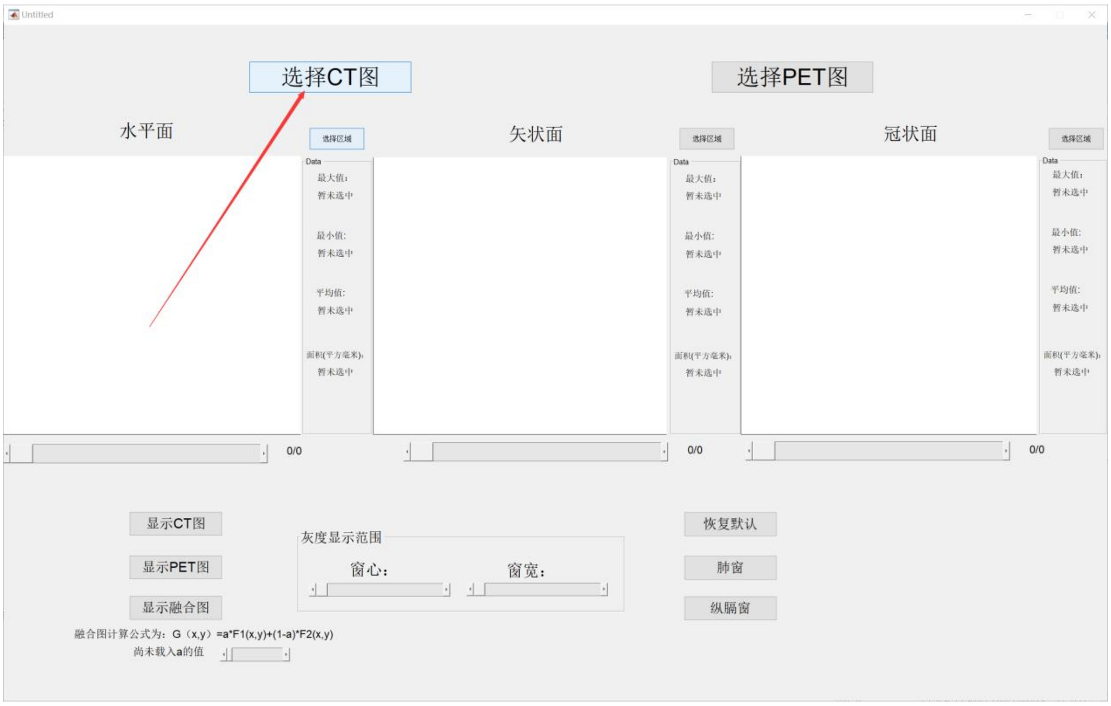
2. 程序初始界面

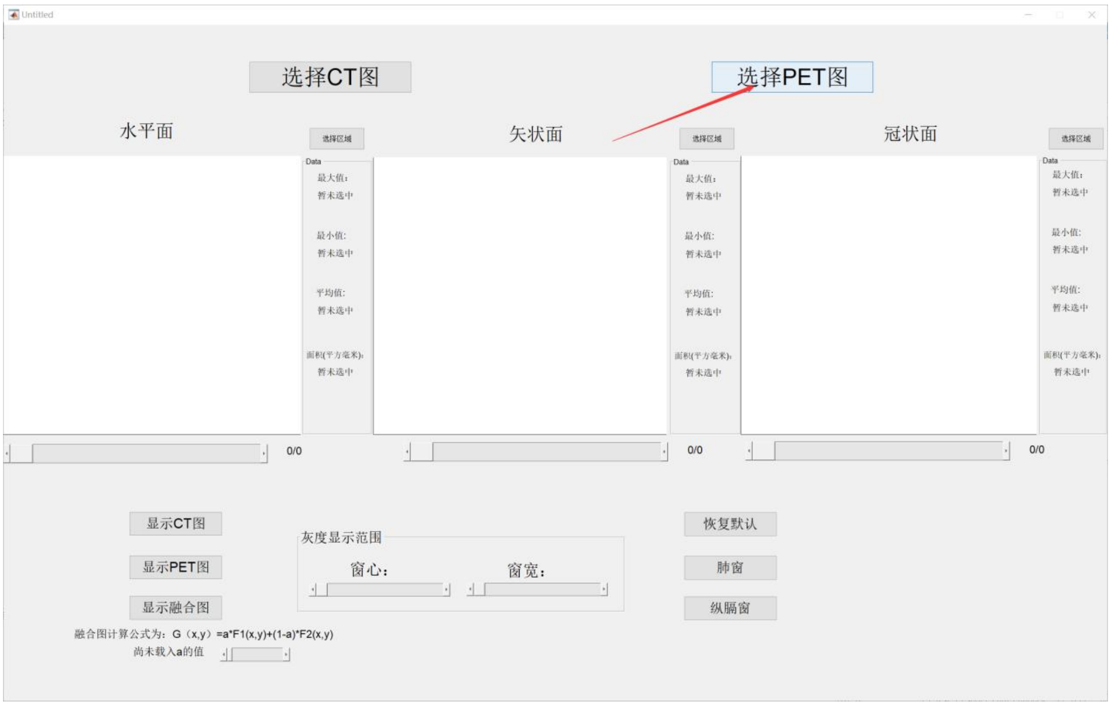
初始界面如下



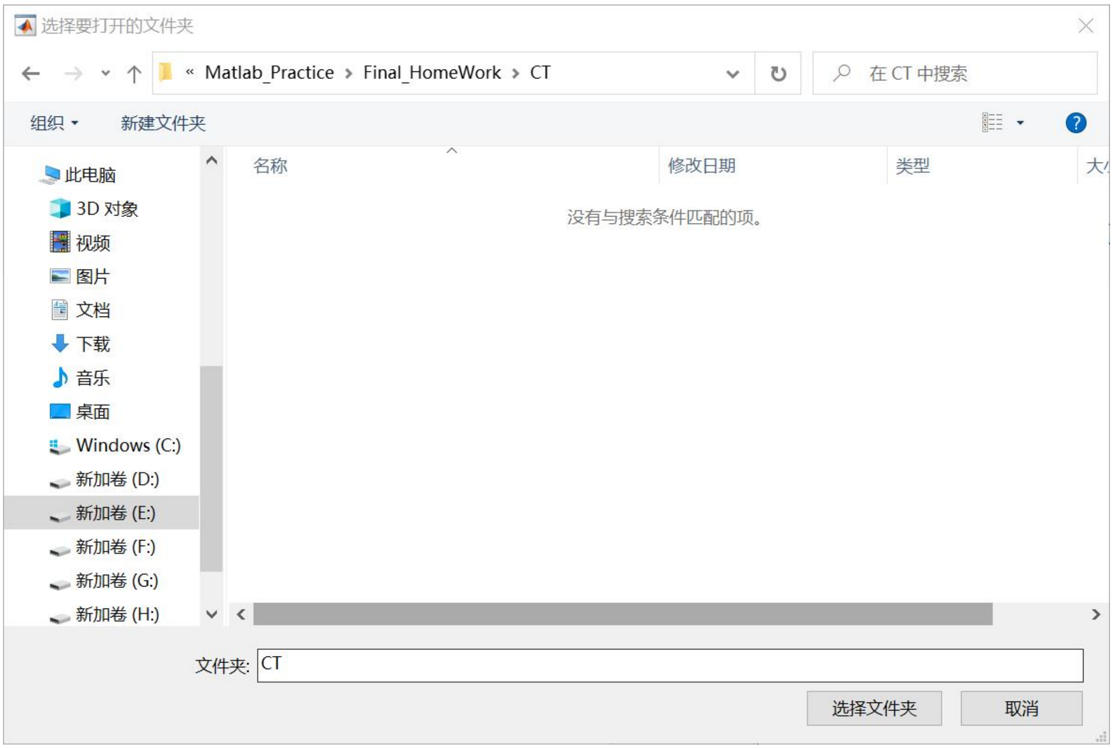
3. 选择 CT 图或 PET 图

点击“选择 CT 图”来选择 CT 目标文件夹，或者点击“选择 PET 图”来选择 PET 目标文件夹。





点击按钮之后会弹出选择文件夹界面，选择目标文件夹



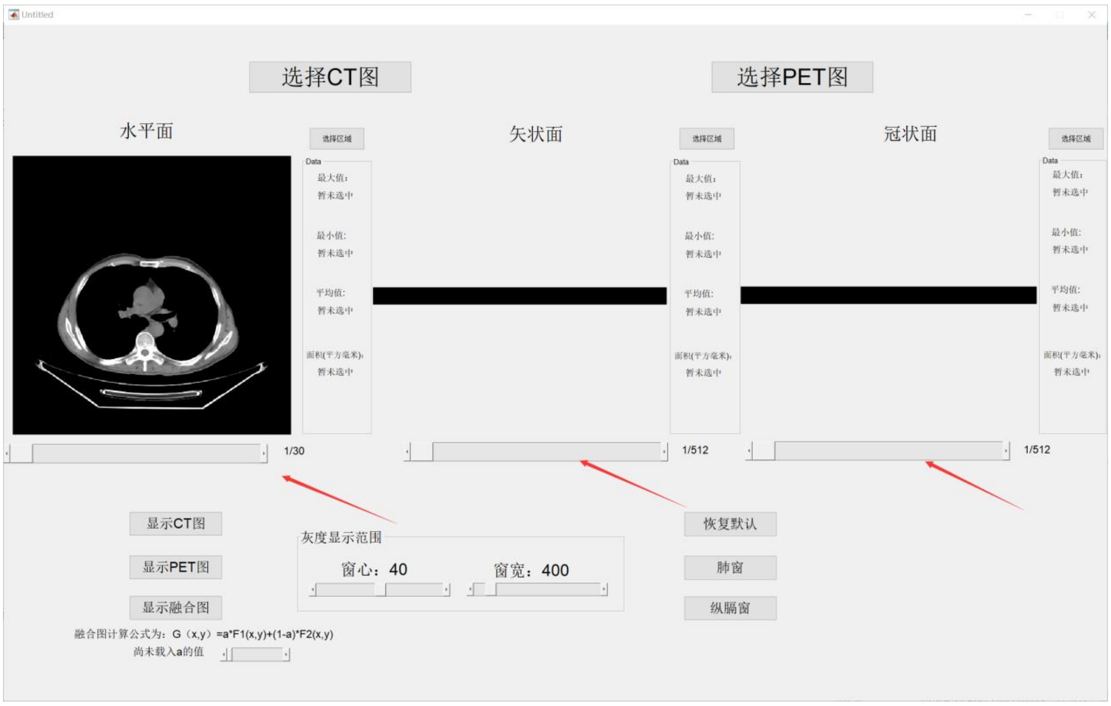
没有与搜索条件匹配的项。

文件夹: CT

4. 连续图像查看

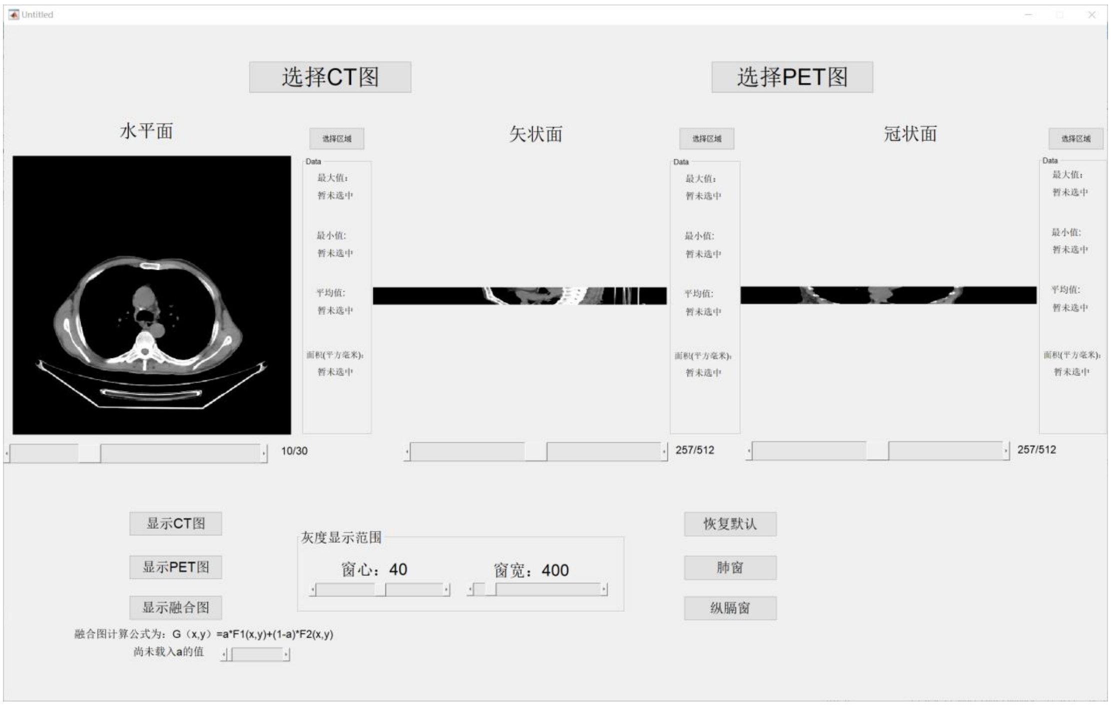
以 CT 图为例，选择完毕后会在下面三个窗口依次显示“水平面”“矢状面”“冠状面”。

每个图下面都有一个滑块，和显示当前为第几张图，滑动滑块可切换要查看第几张图像



融合图计算公式为：G（x,y）=a*F1(x,y)+(1-a)*F2(x,y)

尚未载入a的值

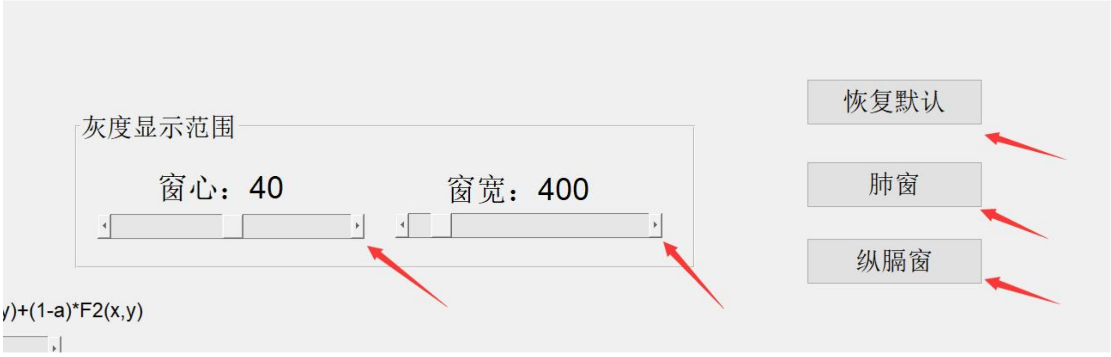


融合图计算公式为：G（x,y）=a*F1(x,y)+(1-a)*F2(x,y)

尚未载入a的值

5. 显示窗的切换与选择

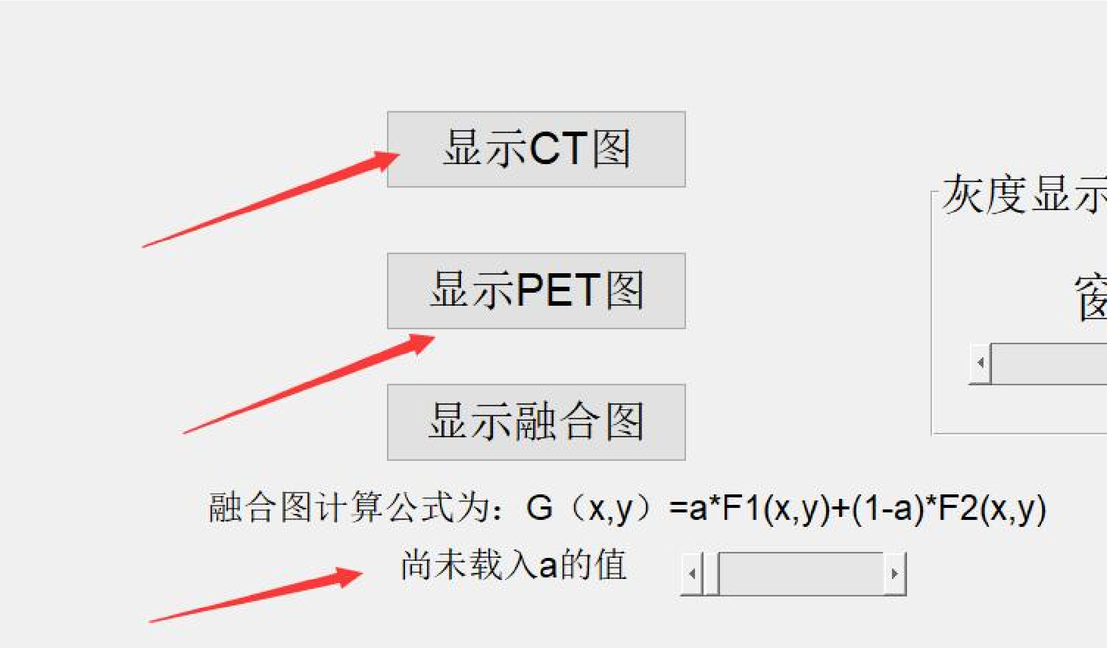
如图所示区域，显示了当前显示窗的窗心与窗宽，可点击右侧“恢复默认”“肺窗”“纵膈窗”按钮，分别一键切换到默认窗、肺窗、纵膈窗。或者滑动下方滑块，分别改变窗心与窗宽。



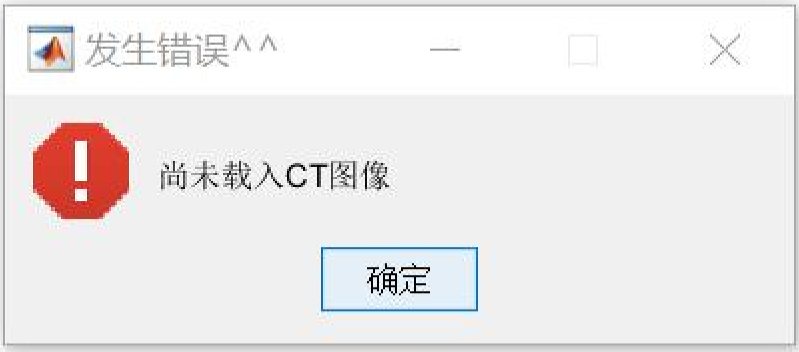
6. 不同图像间的切换

本程序可单独载入 CT 图或 PET 图，也可同时载入 CT 图和 PET 图并将它们融合显示。

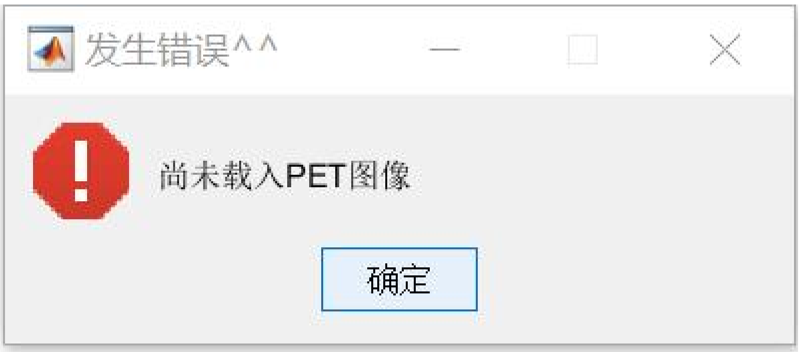
图示区域可实现图像切换功能



分别点击显示 CT 图、显示 PET 图、显示融合图，即可切换到 CT 图、PET 图、融合图；若当前未载入 CT 图或 PET 图，会进行相应的报错。

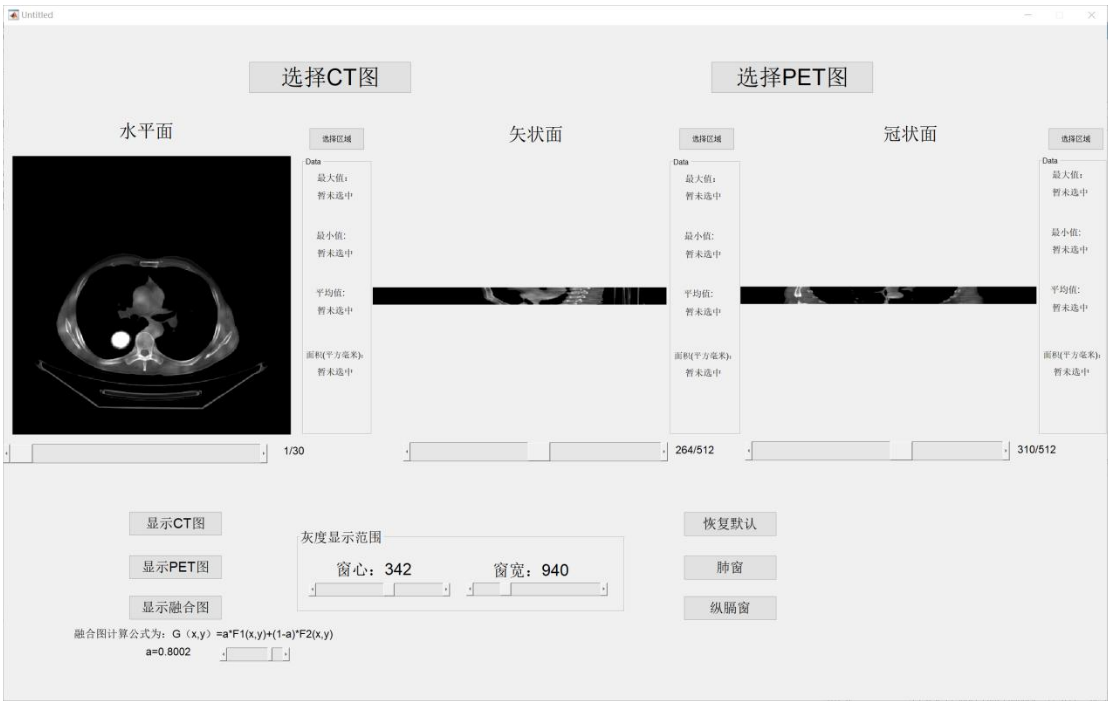


尚未载入CT图像



尚未载入PET图像

同时，融合图像的计算公式也在程序界面展示，用户可自行改变 a 的值。



窗心：342 窗宽：940

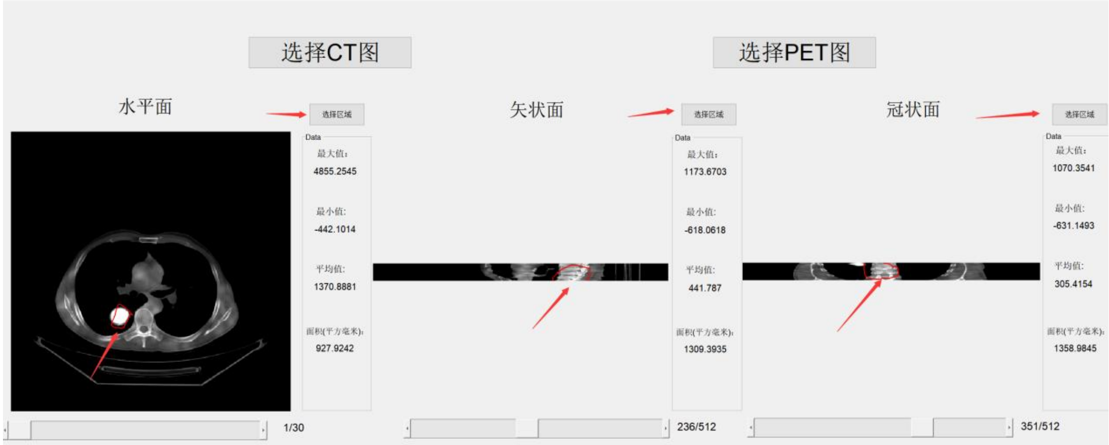
融合图计算公式为：G（x,y）=a*F1(x,y)+(1-a)*F2(x,y)

a=0.8002

7. 感兴趣区域的选择

点击每个显示区域右侧的“选择区域”按钮，即可在相应区域

进行圈画，圈画完毕后双击圈画区域，程序会在右侧按钮下方显示出所圈画区域的最大值、最小值、平均值和面积，此功能可用在任意一张图上，但需要注意的是，每个区域各使用一次此功能后，需要重启程序以便再次使用。



Data
最大值：
4855.2545
最小值：
-442.1014
平均值：
1370.8881
面积(平方毫米)：
927.9242

1/30

Data
最大值：
1173.6703
最小值：
-618.0618
平均值：
441.787
面积(平方毫米)：
1309.3935

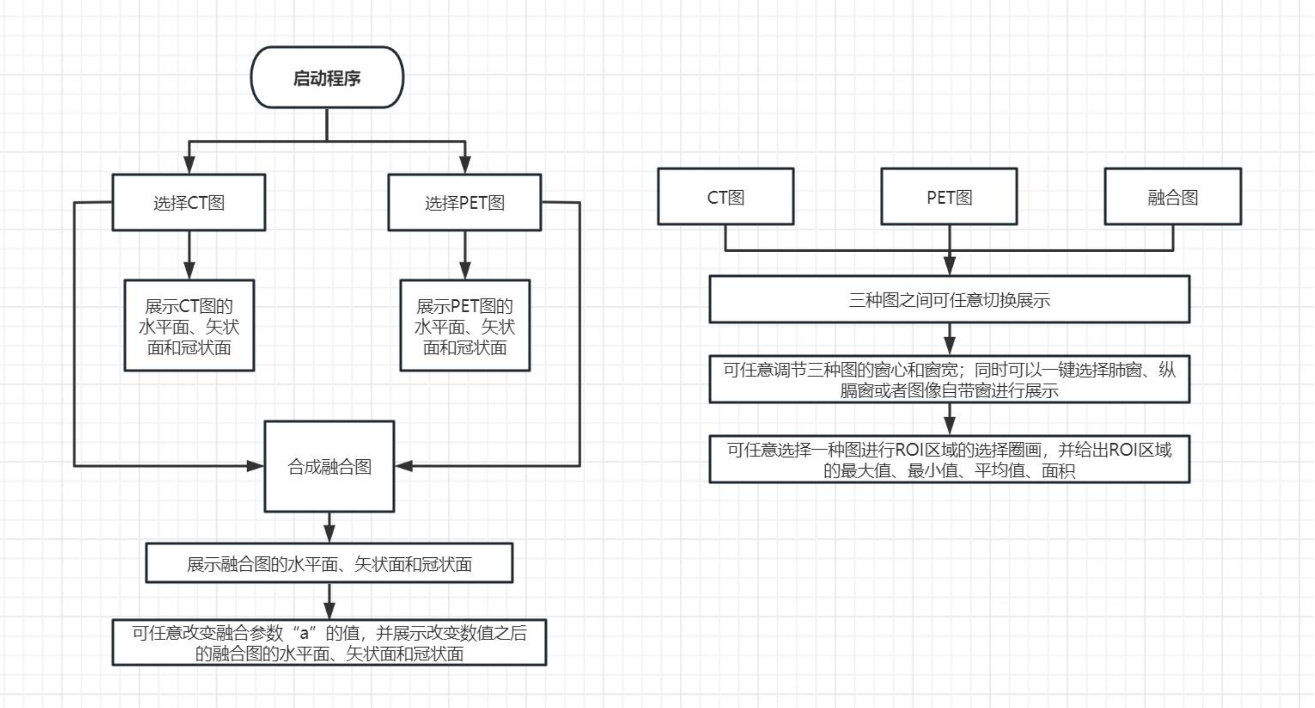
236/512

Data
最大值：
1070.3541
最小值：
-631.1493
平均值：
305.4154
面积(平方毫米)：
1358.9845

351/512

三、 设计流程及相关程序说明

1. GUI 设计流程框图


启动程序
选择CT图
选择PET图
展示CT图的水平面、矢状面和冠状面
展示PET图的水平面、矢状面和冠状面
合成融合图
展示融合图的水平面、矢状面和冠状面
可任意改变融合参数“a”的值，并展示改变数值之后的融合图的水平面、矢状面和冠状面
CT图
PET图
融合图
三种图之间可任意切换展示
可任意调节三种图的窗心和窗宽；同时可以一键选择肺窗、纵膈窗或者图像自带窗进行展示
可任意选择一种图进行ROI区域的选择圈画，并给出ROI区域的最大值、最小值、平均值、面积

2. 每个控件程序及相关说明（复制代码和相关注释，请不要截图）

2.1.程序启动时，预先加入 CT_Point 和 PET_Point 来记录是否已载入 CT 和

PET 文件

```
function CT_test_1_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 CT_test_1 (see VARARGIN)

% Choose default command line output for CT_test_1
handles.output = hObject;


handles.CT_Point=0;                                                %%
标记是否载入 CT

handles.PET_Point=0;                                               %%
标记是否载入 PET

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes CT_test_1 wait for user response (see UIRESUME)
% uiwait(handles.figure1);
```

2.2.“选择 CT 图”按钮，点击该按钮选择要读取的 CT 图文件夹路径，并显示 CT 图及其相关数据。

```
function pushbutton1_Callback(hObject, eventdata,
handles)                    %%读取 CT 文件函数
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

AIM=uigetdir();                                                   %%
选择目标文件夹

AIMPOT=dir(AIM);                                                  %%
取目标文件夹路径
   MARK=[];
   for
```

```
i=1:length(AIMPOT)-2                                                      %%循环
读取每个文件路径

CT_PATH=[AIMPOT(i+2).folder,'\'];                                           %%循
环读取每个文件路径
        CT_PATH=[CT_PATH,AIMPOT(i+2).name];
        CT_Info =
dicominfo(CT_PATH);                                               %%其他数据信息
        CT_Image =
dicomread(CT_PATH);                                              %%灰度矩阵
        CT_Image_Total(:,:,CT_Info.InstanceNumber) =
double(CT_Image*CT_Info.RescaleSlope+ CT_Info.RescaleIntercept);
                                                                             %%
将灰度矩阵转换为 CT 值矩阵储存在新的三维数组中

MARK=[MARK,CT_Info.InstanceNumber];                                          %%
记录三维数组中的有效维度
    end

    handles.CT_Info =
CT_Info;                                                    %%将其他数据信息存入
handles 全局结构体, 便于其他函数调用


CT_MAXM_Origen=(CT_Info.WindowWidth+CT_Info.WindowCenter*2)/2;
      %%CT 图原始最大值

CT_MINM_Origen=CT_MAXM_Origen-CT_Info.WindowWidth;
   %%CT 图原始最小值


Slice_Number1_Now=1;                                                         %%
水平面当前页数

Slice_Number2_Now=1;                                                         %%
矢状面当前页数

Slice_Number3_Now=1;                                                         %%
冠状面当前页数



SliceNumberTotal_2=512;                                                       %%
存入矢状面总页数
```

```
SliceNumberTotal_3=512;                                                      %%
存入冠状面总页数


handles.CT_Image_Total=CT_Image_Total(:,:,MARK);
   %%将有效的三维数组存入全局三维数组中
    CT_Image_Total=CT_Image_Total(:,:,MARK);


handles.Slice_Number1_Now=Slice_Number1_Now;
  %%将当前页数存入 handles 全局结构体中
    handles.Slice_Number2_Now=Slice_Number2_Now;
    handles.Slice_Number3_Now=Slice_Number3_Now;



handles.SliceNumberTotal_1=length(AIMPOT)-2;
  %%将每张图的总页数存入全局结构体中
    handles.SliceNumberTotal_2=SliceNumberTotal_2;
    handles.SliceNumberTotal_3=SliceNumberTotal_3;


handles.CT_MAXM=CT_MAXM_Origen;
%%将 CT 图原始最大最小值存入全局结构体中
    handles.CT_MINM=CT_MINM_Origen;
    handles.CT_MAXM_Origen=CT_MAXM_Origen;
    handles.CT_MINM_Origen=CT_MINM_Origen;


handles.CT_Figure=handles.CT_Image_Total;
  %%将原始 CT 图及相关信息存入全局结构体
    handles.CT_Point=handles.CT_Point+1;
    handles.CT_MAX_Origin=handles.CT_MAXM;
    handles.CT_MIN_Origin=handles.CT_MINM;
    handles.CT_Info_Origin=CT_Info;


axes(handles.axes1);                                                        %%
在 axes1 处显示 CT 图的水平面
    imshow(CT_Image_Total(:,:,Slice_Number1_Now),[CT_MINM_Origen
CT_MAXM_Origen]);

axes(handles.axes2);                                                        %%
```

```
在 axes2 处显示 CT 图的矢状面

imshow(reshape(CT_Image_Total(:,Slice_Number1_Now,:),[SliceNumber
Total_3 length(AIMPOT)-2])',[CT_MINM_Origen CT_MAXM_Origen]);

axes(handles.axes3);                                                %%
在 axes3 处显示 CT 图的冠状面

imshow(reshape(CT_Image_Total(Slice_Number1_Now,:,:),[SliceNumber
Total_2 length(AIMPOT)-2])',[CT_MINM_Origen CT_MAXM_Origen]);


set(handles.slider1,'Max',length(AIMPOT)-2);
  %%设置水平面滑块的信息
    set(handles.slider1,'Min',1);
    set(handles.slider1,'Value',1);

set(handles.text1,'String',[num2str(Slice_Number2_Now),'/',num2st
r(length(AIMPOT)-2)]);
                                                                    %%
设置水平面的页码数据


set(handles.slider2,'Max',SliceNumberTotal_3);
   %%同上设置矢状面的信息
    set(handles.slider2,'Min',1);
    set(handles.slider2,'Value',1);

set(handles.text2,'String',[num2str(Slice_Number1_Now),'/',num2st
r(SliceNumberTotal_2)]);


set(handles.slider3,'Max',SliceNumberTotal_2);
   %%同上设置冠状面的信息
    set(handles.slider3,'Min',1);
    set(handles.slider3,'Value',1);

set(handles.text3,'String',[num2str(Slice_Number1_Now),'/',num2st
r(SliceNumberTotal_2)]);


handles.windowcenter=(CT_MAXM_Origen+CT_MINM_Origen)/2;
    %%将当前窗心存入全局结构体
```

```
handles.windowwidth=CT_MAXM_Origen-CT_MINM_Origen;
   %%将当前窗宽存入全局结构体

                                                                            %%
设置显示窗心和窗宽数据
    set(handles.text_windowcenter,'String',['窗心:
',num2str(handles.windowcenter)]);
    set(handles.text_windowwidth,'String',['窗宽:
',num2str(handles.windowwidth)]);
                                                                            %%
设置窗心和窗宽滑块的相关信息
    set(handles.slider_windowcenter,'Max',2048);
    set(handles.slider_windowcenter,'Min',-2048);
    set(handles.slider_windowcenter,'Value',handles.windowcenter);

    set(handles.slider_windowwidth,'Max',4096);
    set(handles.slider_windowwidth,'Min',0);
    set(handles.slider_windowwidth,'Value',handles.windowwidth);



    guidata(hObject,handles);
  %%记录对 handles 结构体中的数据变更
```

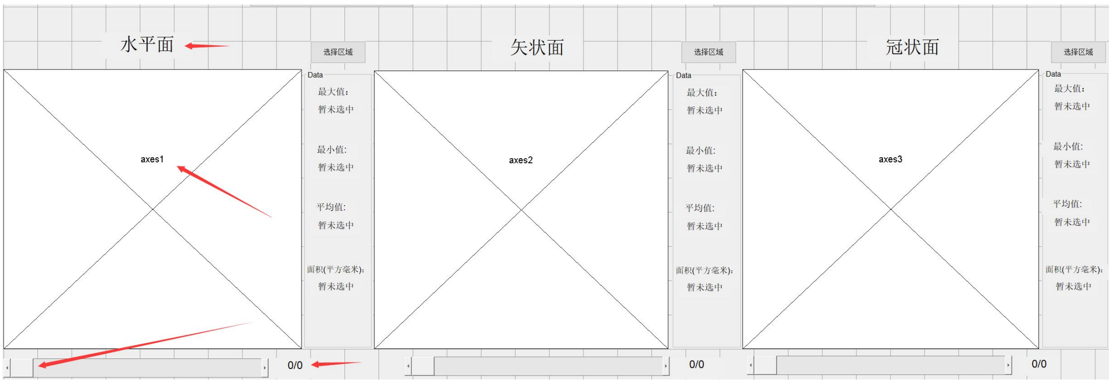
2.3. “选择 PET 图”按钮，点击该按钮选择要读取的 PET 图文件夹路径，并显示 CT 图及其相关数据。

```
function pushbutton2_Callback(hObject, eventdata,
handles)                       %%读取 PET 文件函数
% 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)

                                                                            %%
同 CT 图的读取和存储，便不再过多注释
    AIM=uigetdir();
    AIMPOT=dir(AIM);%%取文件地址
    MARK=[];
    for i=1:length(AIMPOT)-2
        CT_PATH=[AIMPOT(i+2).folder,'\'];
        CT_PATH=[CT_PATH,AIMPOT(i+2).name];
        CT_Info = dicominfo(CT_PATH);%%取 CT
        CT_Image = dicomread(CT_PATH);
        CT_Image_Total(:,:,CT_Info.InstanceNumber) =
CT_Image;            %%PET 图不需要转化为 CT 值
        MARK=[MARK,CT_Info.InstanceNumber];
    end

    handles.CT_Info = CT_Info;

    Slice_Number1_Now=1;
    Slice_Number2_Now=1;
    Slice_Number3_Now=1;

%
SliceNumberTotal_2=size(CT_Image_Total(:,Slice_Number1_Now,Slice_
Number1_Now));
%
SliceNumberTotal_3=size(CT_Image_Total(Slice_Number1_Now,:,Slice_
Number1_Now));

    SliceNumberTotal_2=512;
    SliceNumberTotal_3=512;

    handles.CT_Image_Total=CT_Image_Total(:,:,MARK);
    CT_Image_Total=CT_Image_Total(:,:,MARK);


    CT_MAXM_Origen=max(max(CT_Image));%%CT 图范围
    CT_MINM_Origen=min(min(CT_Image));

    handles.Slice_Number1_Now=Slice_Number1_Now;
    handles.Slice_Number2_Now=Slice_Number2_Now;
    handles.Slice_Number3_Now=Slice_Number3_Now;


    handles.SliceNumberTotal_1=length(AIMPOT)-2;
    handles.SliceNumberTotal_2=SliceNumberTotal_2;
    handles.SliceNumberTotal_3=SliceNumberTotal_3;

    handles.CT_MAXM=CT_MAXM_Origen;
    handles.CT_MINM=CT_MINM_Origen;
    handles.CT_MAXM_Origen=CT_MAXM_Origen;
    handles.CT_MINM_Origen=CT_MINM_Origen;
    handles.PET_Figure=handles.CT_Image_Total;
    handles.PET_Point=handles.PET_Point+1;
    handles.PET_MAX_Origin=handles.CT_MAXM;
    handles.PET_MIN_Origin=handles.CT_MINM;
    handles.PET_Info_Origin=CT_Info;

    axes(handles.axes1);
    imshow(CT_Image_Total(:,:,Slice_Number1_Now),[CT_MINM_Origen
CT_MAXM_Origen]);
    axes(handles.axes2);

imshow(reshape(CT_Image_Total(:,Slice_Number1_Now,:),[SliceNumber
Total_3 length(AIMPOT)-2])',[CT_MINM_Origen CT_MAXM_Origen]);
    axes(handles.axes3);

imshow(reshape(CT_Image_Total(Slice_Number1_Now,:,:),[SliceNumber
Total_2 length(AIMPOT)-2])',[CT_MINM_Origen CT_MAXM_Origen]);

    set(handles.slider1,'Max',length(AIMPOT)-2);
    set(handles.slider1,'Min',1);
    set(handles.slider1,'Value',1);

set(handles.text1,'String',[num2str(Slice_Number2_Now),'/',num2st
r(length(AIMPOT)-2)]);

    set(handles.slider2,'Max',SliceNumberTotal_3);
    set(handles.slider2,'Min',1);
    set(handles.slider2,'Value',1);

set(handles.text2,'String',[num2str(Slice_Number1_Now),'/',num2st
r(SliceNumberTotal_2)]);

    set(handles.slider3,'Max',SliceNumberTotal_2);
    set(handles.slider3,'Min',1);
    set(handles.slider3,'Value',1);

set(handles.text3,'String',[num2str(Slice_Number1_Now),'/',num2st
r(SliceNumberTotal_2)]);

    handles.windowcenter=(CT_MAXM_Origen+CT_MINM_Origen)/2;
    handles.windowwidth=CT_MAXM_Origen-CT_MINM_Origen;

    set(handles.text_windowcenter,'String',['窗心:
',num2str(handles.windowcenter)]);
```

```
    set(handles.text_windowwidth,'String',['窗宽:
',num2str(handles.windowwidth)]);

    set(handles.slider_windowcenter,'Max',2048);
    set(handles.slider_windowcenter,'Min',-2048);
    set(handles.slider_windowcenter,'Value',handles.windowcenter);

    set(handles.slider_windowwidth,'Max',4096);
    set(handles.slider_windowwidth,'Min',0);
    set(handles.slider_windowwidth,'Value',handles.windowwidth);


    guidata(hObject,handles);

% --- Executes on slider movement.
```

2.4.axes1 区域以及其滑块，axes1 区域上方的静态文本框显示此区域为“水平面”，axes1 区域显示图像，下方的滑块来改变图像页数，右下角的静态文本框显示当前页数/总页数。

axes2 与 axes3 区域同理，便不再过多赘述。



axes1 区域滑块控件代码展示：

```
function slider1_Callback(hObject, eventdata, handles)
% hObject    handle to slider1 (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,'Value') returns position of slider
%        get(hObject,'Min') and get(hObject,'Max') to determine range
of slider
    handles.Slice_Number1_Now=
round(get(handles.slider1,'Value'));          %%将当前滑块所代表的图的页
数存入全局结构体
```

```
axes(handles.axes1);                                              %%
在 axes1 处显示当前页数的水平面

imshow(handles.CT_Image_Total(:,:,handles.Slice_Number1_Now),[han
dles.CT_MINM handles.CT_MAXM]);

set(handles.text1,'String',[num2str(handles.Slice_Number1_Now),'/
',num2str(handles.SliceNumberTotal_1)]);

   guidata(hObject,handles);

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

axes2 区域滑块控件代码展示：

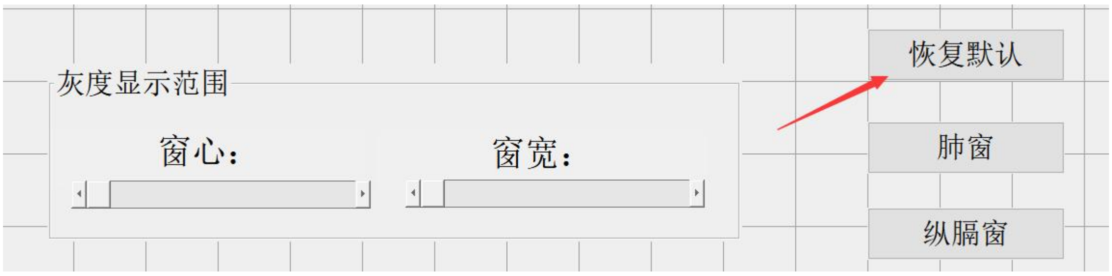
```
function slider2_Callback(hObject, eventdata, handles)
% hObject    handle to slider2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%        get(hObject,'Min') and get(hObject,'Max') to determine range
of slider
   handles.Slice_Number2_Now=
round(get(handles.slider2,'Value'));        %%同水平面滑块 slider1 的
处理，不再过多注释
   axes(handles.axes2);

imshow(reshape(handles.CT_Image_Total(:,handles.Slice_Number2_Now,
:),[handles.SliceNumberTotal_3
handles.SliceNumberTotal_1])',[handles.CT_MINM handles.CT_MAXM]);

set(handles.text2,'String',[num2str(handles.Slice_Number2_Now),'/
',num2str(handles.SliceNumberTotal_2)]);

   guidata(hObject,handles);

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

axes3 区域滑块控件代码展示：

```
function slider3_Callback(hObject, eventdata,
handles)                        %%同水平面滑块 slider1 的处理，不再过多注释
```

```
% hObject    handle to slider3 (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,'Value') returns position of slider
%        get(hObject,'Min') and get(hObject,'Max') to determine range
of slider
    handles.Slice_Number3_Now= round(get(handles.slider3,'Value'));
    axes(handles.axes3);

imshow(reshape(handles.CT_Image_Total(handles.Slice_Number3_Now,:,
:),[handles.SliceNumberTotal_2
handles.SliceNumberTotal_1])',[handles.CT_MINM handles.CT_MAXM]);

set(handles.text3,'String',[num2str(handles.Slice_Number3_Now),'/
',num2str(handles.SliceNumberTotal_3)]);

    guidata(hObject,handles);

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

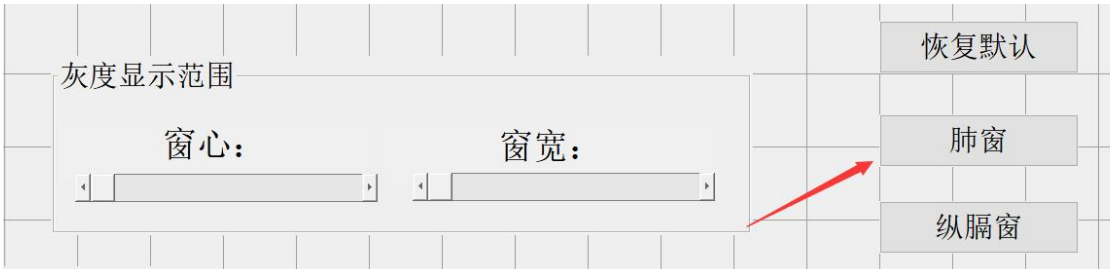
2.5. 点击“恢复默认”按钮，会将窗心和窗宽变为自带窗并显示三张图。



```
function pushbutton_reset_Callback(hObject, eventdata,
handles)            %%重置窗心窗口显示
% hObject    handle to pushbutton_reset (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


handles.CT_MAXM=handles.CT_MAXM_Origen;
 %%重置最大最小值，于读取时的操作同理，不再过多注释
    handles.CT_MINM=handles.CT_MINM_Origen;


handles.windowcenter=(handles.CT_MAXM_Origen+handles.CT_MINM_Orig
en)/2;

handles.windowwidth=handles.CT_MAXM_Origen-handles.CT_MINM_Origen
```

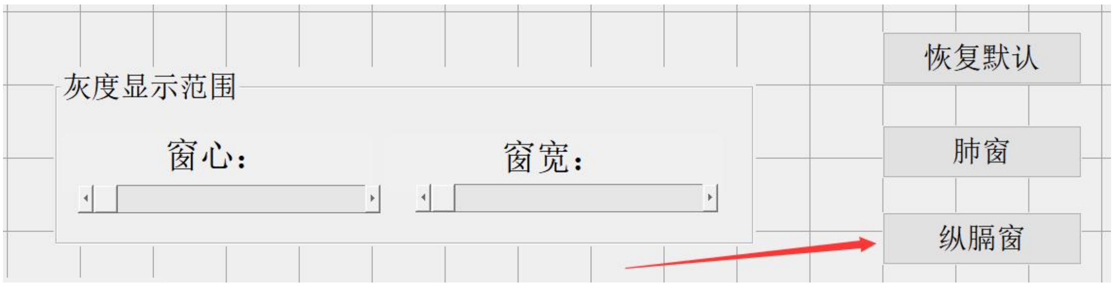
```
;

    set(handles.text_windowcenter,'String',['窗心:
',num2str(handles.windowcenter)]);
    set(handles.text_windowwidth,'String',['窗宽:
',num2str(handles.windowwidth)]);

    set(handles.slider_windowcenter,'Max',2048);
    set(handles.slider_windowcenter,'Min',-2048);
    set(handles.slider_windowcenter,'Value',handles.windowcenter);

    set(handles.slider_windowwidth,'Max',4096);
    set(handles.slider_windowwidth,'Min',0);
    set(handles.slider_windowwidth,'Value',handles.windowwidth);

    axes(handles.axes1);

imshow(handles.CT_Image_Total(:,:,handles.Slice_Number1_Now),[han
dles.CT_MINM handles.CT_MAXM]);
    axes(handles.axes2);

imshow(reshape(handles.CT_Image_Total(:,handles.Slice_Number2_Now,
:),[handles.SliceNumberTotal_3
handles.SliceNumberTotal_1])',[handles.CT_MINM handles.CT_MAXM]);
    axes(handles.axes3);

imshow(reshape(handles.CT_Image_Total(handles.Slice_Number3_Now,:,
:),[handles.SliceNumberTotal_2
handles.SliceNumberTotal_1])',[handles.CT_MINM handles.CT_MAXM]);

    guidata(hObject,handles);
% --- Executes on button press in pushbutton4.
```

2.6. 点击“肺窗”按钮，会将窗心和窗宽变为自带窗并显示三张图。



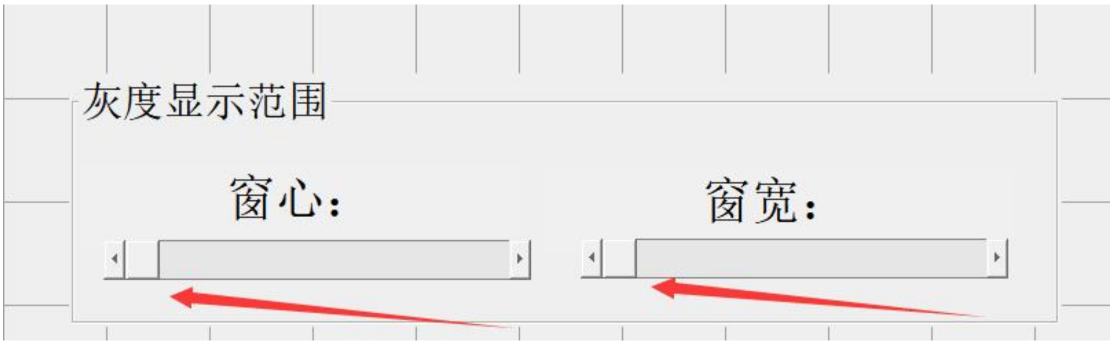
```
function pushbutton4_Callback(hObject, eventdata,
handles)                    %%将窗心和窗宽改为肺窗的函数，与重置按钮同理，不再过
多注释
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
    handles.windowcenter=-500;
    handles.windowwidth=700;


handles.CT_MAXM=(handles.windowwidth+handles.windowcenter*2)/2;
    handles.CT_MINM=handles.CT_MAXM-handles.windowwidth;

    set(handles.text_windowcenter,'String',['窗心:
',num2str(handles.windowcenter)]);
    set(handles.text_windowwidth,'String',['窗宽:
',num2str(handles.windowwidth)]);

    set(handles.slider_windowcenter,'Value',handles.windowcenter);
    set(handles.slider_windowwidth,'Value',handles.windowwidth);

    axes(handles.axes1);

imshow(handles.CT_Image_Total(:,:,handles.Slice_Number1_Now),[han
dles.CT_MINM handles.CT_MAXM]);
    axes(handles.axes2);

imshow(reshape(handles.CT_Image_Total(:,handles.Slice_Number2_Now,
:),[handles.SliceNumberTotal_3
handles.SliceNumberTotal_1])',[handles.CT_MINM handles.CT_MAXM]);
    axes(handles.axes3);

imshow(reshape(handles.CT_Image_Total(handles.Slice_Number3_Now,:,
:),[handles.SliceNumberTotal_2
handles.SliceNumberTotal_1])',[handles.CT_MINM handles.CT_MAXM]);

    guidata(hObject,handles);
% --- Executes on button press in pushbutton5.
```

2.7.点击“纵膈窗”按钮，会将窗心和窗宽变为自带窗并显示三张图。



```
function pushbutton5_Callback(hObject, eventdata,
handles)                %%将窗心和窗宽改为纵膈窗的函数，与重置按钮同理，不再
过多注释
% hObject    handle to pushbutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

    handles.windowcenter=50;
    handles.windowwidth=400;


handles.CT_MAXM=(handles.windowwidth+handles.windowcenter*2)/2;
    handles.CT_MINM=handles.CT_MAXM-handles.windowwidth;

    set(handles.text_windowcenter,'String',['窗心:
',num2str(handles.windowcenter)]);
    set(handles.text_windowwidth,'String',['窗宽:
',num2str(handles.windowwidth)]);

    set(handles.slider_windowcenter,'Value',handles.windowcenter);
    set(handles.slider_windowwidth,'Value',handles.windowwidth);

    axes(handles.axes1);

imshow(handles.CT_Image_Total(:,:,handles.Slice_Number1_Now),[han
dles.CT_MINM handles.CT_MAXM]);
    axes(handles.axes2);

imshow(reshape(handles.CT_Image_Total(:,handles.Slice_Number2_Now,
:),[handles.SliceNumberTotal_3
handles.SliceNumberTotal_1])',[handles.CT_MINM handles.CT_MAXM]);
    axes(handles.axes3);

imshow(reshape(handles.CT_Image_Total(handles.Slice_Number3_Now,:,
:),[handles.SliceNumberTotal_2
handles.SliceNumberTotal_1])',[handles.CT_MINM handles.CT_MAXM]);

    guidata(hObject,handles);
% --- Executes on slider movement.
```

2.8. “窗心”滑块和“窗宽”滑块，拖动对应的滑块，可以改变窗心和窗宽的具体数值，并用改变后的窗显示出三张图。



窗心滑块的代码展示：

```
function slider_windowcenter_Callback(hObject, eventdata,
handles)          %%改变窗心数值的滑块
% hObject    handle to slider_windowcenter (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)



handles.windowcenter=round(get(handles.slider_windowcenter,'Value
'));   %%读取此时窗心的值

handles.CT_MAXM=(handles.windowwidth+handles.windowcenter*2)/2;
      %%计算此时的最大最小值
    handles.CT_MINM=handles.CT_MAXM-handles.windowwidth;

    set(handles.text_windowcenter,'String',['窗心:
',num2str(handles.windowcenter)]);
                                                                       %%
将改变之后的窗心值显示在 text

    set(handles.slider_windowcenter,'Value',handles.windowcenter);


axes(handles.axes1);                                                   %%
显示此时的三张图

imshow(handles.CT_Image_Total(:,:,handles.Slice_Number1_Now),[han
dles.CT_MINM handles.CT_MAXM]);
    axes(handles.axes2);

imshow(reshape(handles.CT_Image_Total(:,handles.Slice_Number2_Now,
:),[handles.SliceNumberTotal_3
handles.SliceNumberTotal_1])',[handles.CT_MINM handles.CT_MAXM]);
    axes(handles.axes3);
```

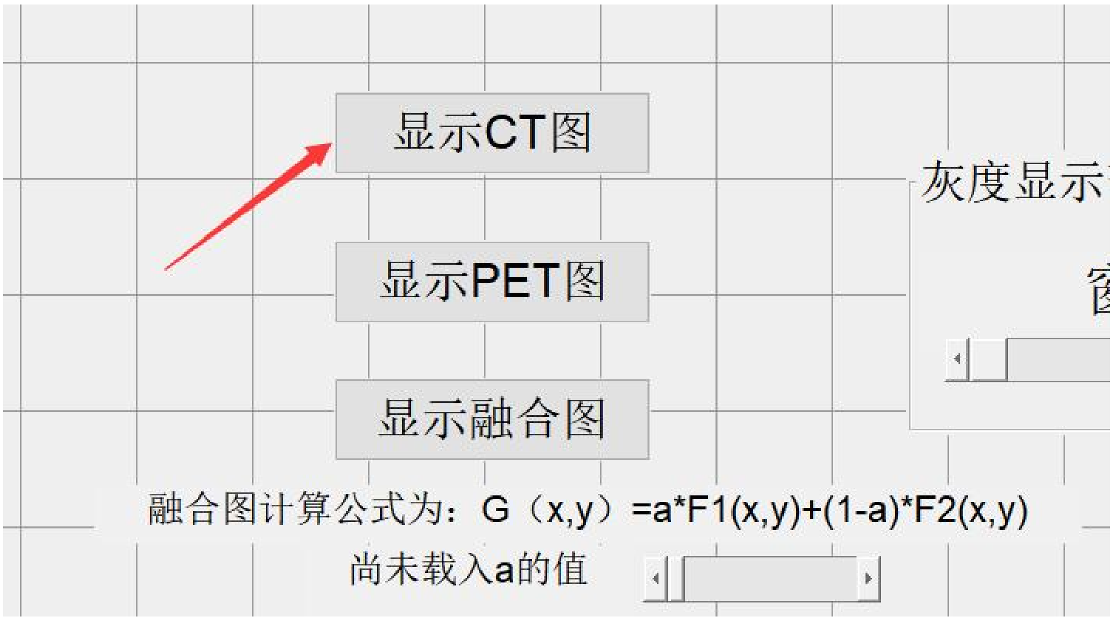
```
imshow(reshape(handles.CT_Image_Total(handles.Slice_Number3_Now,:,
:),[handles.SliceNumberTotal_2
handles.SliceNumberTotal_1])',[handles.CT_MINM handles.CT_MAXM]);

    guidata(hObject,handles);


% Hints: get(hObject,'Value') returns position of slider
%        get(hObject,'Min') and get(hObject,'Max') to determine range
of slider


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

窗宽滑块的代码展示：

```
function slider_windowwidth_Callback(hObject, eventdata,
handles)          %%改变窗宽数值的滑块，与窗心同理，不再过多注释
% hObject    handle to slider_windowwidth (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,'Value') returns position of slider
%        get(hObject,'Min') and get(hObject,'Max') to determine range
of slider


handles.windowwidth=round(get(handles.slider_windowwidth,'Value'))
;


handles.CT_MAXM=(handles.windowwidth+handles.windowcenter*2)/2;
    handles.CT_MINM=handles.CT_MAXM-handles.windowwidth;

    set(handles.text_windowwidth,'String',['窗宽:
',num2str(handles.windowwidth)]);

    set(handles.slider_windowwidth,'Value',handles.windowwidth);

    axes(handles.axes1);

imshow(handles.CT_Image_Total(:,:,handles.Slice_Number1_Now),[han
dles.CT_MINM handles.CT_MAXM]);
    axes(handles.axes2);
```

```
imshow(reshape(handles.CT_Image_Total(:,handles.Slice_Number2_Now,
:),[handles.SliceNumberTotal_3
handles.SliceNumberTotal_1])',[handles.CT_MINM handles.CT_MAXM]);
    axes(handles.axes3);

imshow(reshape(handles.CT_Image_Total(handles.Slice_Number3_Now,:,
:),[handles.SliceNumberTotal_2
handles.SliceNumberTotal_1])',[handles.CT_MINM handles.CT_MAXM]);

    guidata(hObject,handles);

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

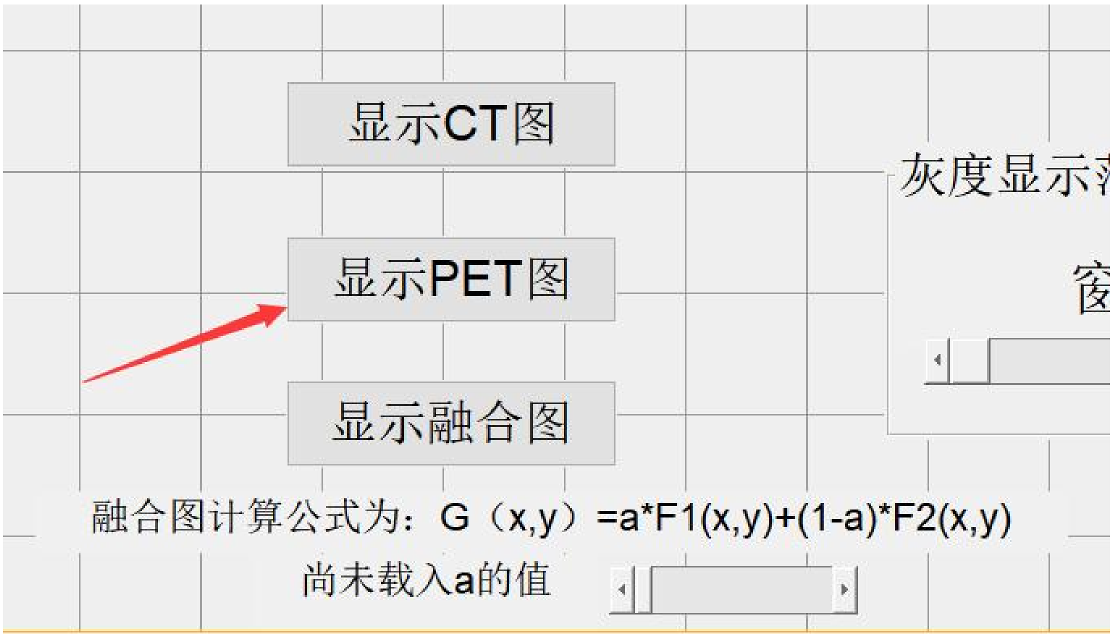
2.9. “显示 CT 图”按钮，将要展示的图改为 CT 图，并能判断用户是否已成功载入 CT 图。并显示出三张 CT 图。



```
function pushbutton_showCT_Callback(hObject, eventdata,
handles)            %%选择 CT 图像进行显示
% hObject    handle to pushbutton_showCT (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

    if
handles.CT_Point==0                                           %%判
断是否已读取 CT 文件，若未读取则提示报错
        msgbox('尚未载入 CT 图像','发生错误^^','error');
    else

handles.CT_Image_Total=handles.CT_Figure;                        %%
```

```
将当前显示的图改为 CT 图

handles.CT_MAXM_Origen=handles.CT_MAX_Origin;                           %
%将当前所存取有关图的数据全部换为 CT 图的数据
       handles.CT_MINM_Origen=handles.CT_MIN_Origin;
       handles.CT_MAXM=handles.CT_MAX_Origin;
       handles.CT_MINM=handles.CT_MIN_Origin;
       handles.CT_Info=handles.CT_Info_Origin;




handles.windowcenter=(handles.CT_MAX_Origin+handles.CT_MIN_Origin)
/2;

handles.windowwidth=handles.CT_MAX_Origin-handles.CT_MIN_Origin;

       handles.CT_MAXM=handles.CT_MAX_Origin;
       handles.CT_MINM=handles.CT_MIN_Origin;

       set(handles.text_windowcenter,'String',['窗心:
',num2str(handles.windowcenter)]);
       set(handles.text_windowwidth,'String',['窗宽:
',num2str(handles.windowwidth)]);

       set(handles.slider_windowcenter,'Max',2048);
       set(handles.slider_windowcenter,'Min',-2048);

set(handles.slider_windowcenter,'Value',handles.windowcenter);

       set(handles.slider_windowwidth,'Max',4096);
       set(handles.slider_windowwidth,'Min',0);

set(handles.slider_windowwidth,'Value',handles.windowwidth);

       axes(handles.axes1);

imshow(handles.CT_Image_Total(:,:,handles.Slice_Number1_Now),[han
dles.CT_MIN_Origin handles.CT_MAX_Origin]);
       axes(handles.axes2);

imshow(reshape(handles.CT_Image_Total(:,handles.Slice_Number2_Now,
:),[handles.SliceNumberTotal_3
handles.SliceNumberTotal_1])',[handles.CT_MIN_Origin
handles.CT_MAX_Origin]);
        axes(handles.axes3);

imshow(reshape(handles.CT_Image_Total(handles.Slice_Number3_Now,:,
:),[handles.SliceNumberTotal_2
handles.SliceNumberTotal_1])',[handles.CT_MIN_Origin
handles.CT_MAX_Origin]);

        guidata(hObject,handles);
    end
```

2.10. “显示 PET 图”按钮，将要展示的图改为 PET 图，并能判断用户是否已成功载入 PET 图。并显示出三张 PET 图。



融合图计算公式为：G（x,y）=a*F1(x,y)+(1-a)*F2(x,y)

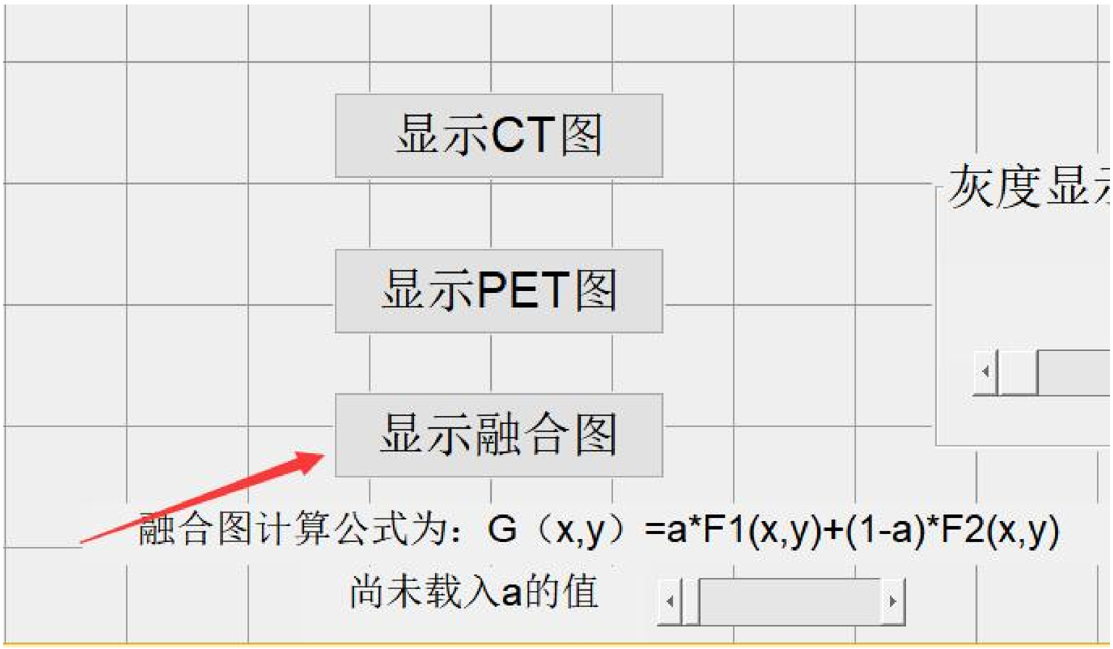
尚未载入a的值

```
function pushbutton_showPET_Callback(hObject, eventdata,
handles)             %%选择 PET 图像进行显示
% hObject    handle to pushbutton_showPET (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
    if
handles.PET_Point==0                                           %%判
断是否已读取 PET 文件，若未读取则提示报错
        msgbox('尚未载入 PET 图像','发生错误^^','error');
    else

handles.CT_Image_Total=handles.PET_Figure;                        %%
将当前显示的图改为 PET 图

handles.CT_MAXM_Origen=handles.PET_MAX_Origin;                     %
```

```
%将当前所存取有关图的数据全部换为 PET 图的数据
       handles.CT_MINM_Origen=handles.PET_MIN_Origin;
       handles.CT_MAXM=handles.PET_MAX_Origin;
       handles.CT_MINM=handles.PET_MIN_Origin;
       handles.CT_Info=handles.PET_Info_Origin;



handles.windowcenter=(handles.PET_MAX_Origin+handles.PET_MIN_Orig
in)/2;

handles.windowwidth=handles.PET_MAX_Origin-handles.PET_MIN_Origin
;

       handles.CT_MAXM=handles.PET_MAX_Origin;
       handles.CT_MINM=handles.PET_MIN_Origin;

       set(handles.text_windowcenter,'String',['窗心:
',num2str(handles.windowcenter)]);
       set(handles.text_windowwidth,'String',['窗宽:
',num2str(handles.windowwidth)]);

       set(handles.slider_windowcenter,'Max',2048);
       set(handles.slider_windowcenter,'Min',-2048);

set(handles.slider_windowcenter,'Value',handles.windowcenter);

       set(handles.slider_windowwidth,'Max',4096);
       set(handles.slider_windowwidth,'Min',0);

set(handles.slider_windowwidth,'Value',handles.windowwidth);

       axes(handles.axes1);

imshow(handles.CT_Image_Total(:,:,handles.Slice_Number1_Now),[han
dles.PET_MIN_Origin handles.PET_MAX_Origin]);
       axes(handles.axes2);

imshow(reshape(handles.CT_Image_Total(:,handles.Slice_Number2_Now,
:),[handles.SliceNumberTotal_3
handles.SliceNumberTotal_1])',[handles.PET_MIN_Origin
handles.PET_MAX_Origin]);
       axes(handles.axes3);
```

```
imshow(reshape(handles.CT_Image_Total(handles.Slice_Number3_Now,:,
:),[handles.SliceNumberTotal_2
handles.SliceNumberTotal_1])',[handles.PET_MIN_Origin
handles.PET_MAX_Origin]);

        guidata(hObject,handles);
    end
```

2.11.“显示融合图”按钮，融合 CT 和 PET 图像，将要展示的图改为融合图，并能判断用户是否已成功载入 CT 图或 PET 图。并显示出三张融合图。



融合图计算公式为：G（x,y）=a*F1(x,y)+(1-a)*F2(x,y)

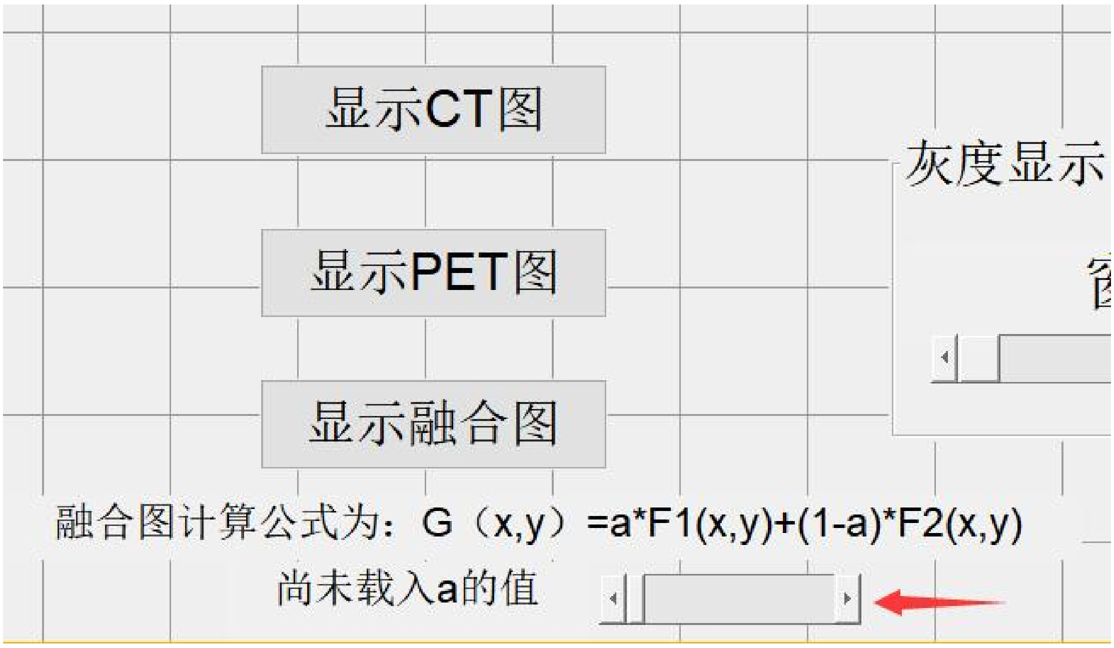
尚未载入a的值

```
function pushbutton_showMIX_Callback(hObject, eventdata,
handles)            %%选择 PET 图像进行显示
% hObject    handle to pushbutton_showMIX (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

    if
handles.PET_Point==0                                          %%判
断是否已读取 CT 或 PET 文件, 若未读取则提示报错
        msgbox('尚未载入 PET 图像','发生错误^^','error');
    end
    if handles.CT_Point==0
        msgbox('尚未载入 CT 图像','发生错误^^','error');
    end
    if handles.PET_Point>0 && handles.CT_Point>0


Para=0.9;                                                 %%参数 a
的值
```

```
handles.Para=Para;                                                  %%将参
数 a 存入 handles 全局结构体

handles.CT_Image_Total=double(Para*handles.CT_Figure)+double((1-P
ara)*handles.PET_Figure);
                                                                     %%
将 CT 图与 PET 图进行融合，将当前显示的图改为融合图

handles.CT_MAXM_Origen=Para*handles.CT_MAX_Origin+(1-Para)*handle
s.PET_MAX_Origin;
                                                                     %%
将当前所存取有关图的数据全部换为融合图的数据

handles.CT_MINM_Origen=Para*handles.CT_MIN_Origin+(1-Para)*handle
s.PET_MIN_Origin;

handles.CT_MAXM=Para*handles.CT_MAX_Origin+(1-Para)*handles.PET_M
AX_Origin;

handles.CT_MINM=Para*handles.CT_MIN_Origin+(1-Para)*handles.PET_M
IN_Origin;
       handles.CT_Info=handles.CT_Info_Origin;


handles.windowcenter=(Para*handles.CT_MAX_Origin+(1-Para)*handles.
PET_MAX_Origin+Para*handles.CT_MIN_Origin+(1-Para)*handles.PET_MI
N_Origin)/2;

handles.windowwidth=Para*handles.CT_MAX_Origin+(1-Para)*handles.P
ET_MAX_Origin-Para*handles.CT_MIN_Origin+(1-Para)*handles.PET_MIN
_Origin;


handles.CT_MAXM=Para*handles.CT_MAX_Origin+(1-Para)*handles.PET_M
AX_Origin;

handles.CT_MINM=Para*handles.CT_MIN_Origin+(1-Para)*handles.PET_M
IN_Origin;

       set(handles.text_windowcenter,'String',['窗心:
```

```
',num2str(handles.windowcenter)]);
       set(handles.text_windowwidth,'String',['窗宽:
',num2str(handles.windowwidth)]);

       set(handles.slider_windowcenter,'Max',2048);
       set(handles.slider_windowcenter,'Min',-2048);

set(handles.slider_windowcenter,'Value',handles.windowcenter);

       set(handles.slider_windowwidth,'Max',4096);
       set(handles.slider_windowwidth,'Min',0);

set(handles.slider_windowwidth,'Value',handles.windowwidth);

       set(handles.slider_Para,'Max',0.999);
       set(handles.slider_Para,'Min',0.001);
       set(handles.text_Para,'String',['a=',num2str(Para)]);
       set(handles.slider_Para,'Value',Para);

       axes(handles.axes1);

imshow(handles.CT_Image_Total(:,:,handles.Slice_Number1_Now),[Par
a*handles.CT_MIN_Origin+(1-Para)*handles.PET_MIN_Origin
Para*handles.CT_MAX_Origin+(1-Para)*handles.PET_MAX_Origin]);
       axes(handles.axes2);

imshow(reshape(handles.CT_Image_Total(:,handles.Slice_Number2_Now,
:),[handles.SliceNumberTotal_3
handles.SliceNumberTotal_1])',[Para*handles.CT_MIN_Origin+(1-Para)
*handles.PET_MIN_Origin
Para*handles.CT_MAX_Origin+(1-Para)*handles.PET_MAX_Origin]);
       axes(handles.axes3);

imshow(reshape(handles.CT_Image_Total(handles.Slice_Number3_Now,:,
:),[handles.SliceNumberTotal_2
handles.SliceNumberTotal_1])',[Para*handles.CT_MIN_Origin+(1-Para)
*handles.PET_MIN_Origin
Para*handles.CT_MAX_Origin+(1-Para)*handles.PET_MAX_Origin]);

       guidata(hObject,handles);
   end
```

2.12. “a 的数值改变”滑块，能判断用户是否已成功载入 CT 图或 PET 图。可以通过滑动改变特征参数‘a’的数值，并显示出改变数值之后的三张融合图。



融合图计算公式为：G（x,y）=a*F1(x,y)+(1-a)*F2(x,y)

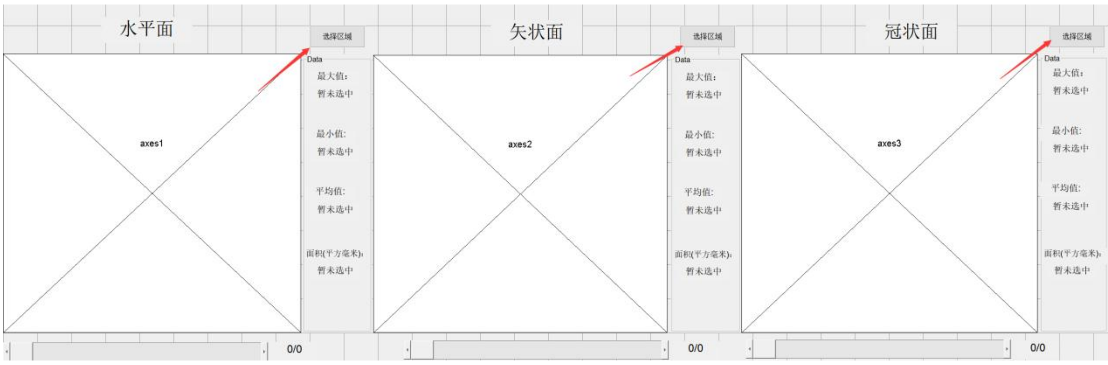
尚未载入a的值

```
function slider_Para_Callback(hObject, eventdata,
handles)                    %%融合图像特征参数 a 的选择
% hObject    handle to slider_Para (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

   if handles.CT_Point==0 ||
handles.PET_Point==0                           %%判断是否已载入融合图像，若
未载入则报错
       msgbox('尚未载入完整融合图像','发生错误^^','error');
   else



handles.Para=get(handles.slider_Para,'Value');                   %
%读取此时滑块的值给 Para

      Para=get(handles.slider_Para,'Value');

                                                                %%
计算新的 Para，并显示新的融合图像及其数据，与前同理，不再过多注释

handles.CT_Image_Total=double(Para*handles.CT_Figure)+double((1-P
ara)*handles.PET_Figure);

handles.CT_MAXM_Origen=Para*handles.CT_MAX_Origin+(1-Para)*handle
s.PET_MAX_Origin;
```

```
handles.CT_MINM_Origen=Para*handles.CT_MIN_Origin+(1-Para)*handle
s.PET_MIN_Origin;

handles.CT_MAXM=Para*handles.CT_MAX_Origin+(1-Para)*handles.PET_M
AX_Origin;

handles.CT_MINM=Para*handles.CT_MIN_Origin+(1-Para)*handles.PET_M
IN_Origin;
       handles.CT_Info=handles.CT_Info_Origin;



handles.windowcenter=(Para*handles.CT_MAX_Origin+(1-Para)*handles.
PET_MAX_Origin+Para*handles.CT_MIN_Origin+(1-Para)*handles.PET_MI
N_Origin)/2;

handles.windowwidth=Para*handles.CT_MAX_Origin+(1-Para)*handles.P
ET_MAX_Origin-Para*handles.CT_MIN_Origin+(1-Para)*handles.PET_MIN
_Origin;


handles.CT_MAXM=Para*handles.CT_MAX_Origin+(1-Para)*handles.PET_M
AX_Origin;

handles.CT_MINM=Para*handles.CT_MIN_Origin+(1-Para)*handles.PET_M
IN_Origin;

       set(handles.text_windowcenter,'String',['窗心：
',num2str(handles.windowcenter)]);
       set(handles.text_windowwidth,'String',['窗宽：
',num2str(handles.windowwidth)]);

       set(handles.slider_windowcenter,'Max',2048);
       set(handles.slider_windowcenter,'Min',-2048);

set(handles.slider_windowcenter,'Value',handles.windowcenter);

       set(handles.slider_windowwidth,'Max',4096);
       set(handles.slider_windowwidth,'Min',0);

set(handles.slider_windowwidth,'Value',handles.windowwidth);

       set(handles.slider_Para,'Max',0.999);
```

```
        set(handles.slider_Para,'Min',0.001);
        set(handles.text_Para,'String',['a=',num2str(Para)]);
        set(handles.slider_Para,'Value',Para);

        axes(handles.axes1);

imshow(handles.CT_Image_Total(:,:,handles.Slice_Number1_Now),[Para*handles.CT_MIN_Origin+(1-Para)*handles.PET_MIN_Origin
Para*handles.CT_MAX_Origin+(1-Para)*handles.PET_MAX_Origin]);
        axes(handles.axes2);

imshow(reshape(handles.CT_Image_Total(:,handles.Slice_Number2_Now,:),[handles.SliceNumberTotal_3
handles.SliceNumberTotal_1])',[Para*handles.CT_MIN_Origin+(1-Para)*handles.PET_MIN_Origin
Para*handles.CT_MAX_Origin+(1-Para)*handles.PET_MAX_Origin]);
        axes(handles.axes3);

imshow(reshape(handles.CT_Image_Total(handles.Slice_Number3_Now,:,:),[handles.SliceNumberTotal_2
handles.SliceNumberTotal_1])',[Para*handles.CT_MIN_Origin+(1-Para)*handles.PET_MIN_Origin
Para*handles.CT_MAX_Origin+(1-Para)*handles.PET_MAX_Origin]);

        guidata(hObject,handles);
    end
```

2.13.“选择区域”按钮，每个 axes 区域的右边都有一个“选择区域”按钮，点击可在相应的 axes 区域圈画用户感兴趣的区域，圈画完毕后双击圈画区域，可在相应的按钮下方显示出该区域的最大值、最小值、平均值和面积四类数据。



水平面区域选择按钮代码展示：

```
function pushbutton_draw1_Callback(hObject, eventdata,
handles)                  %%水平面的 ROI 选择
% hObject    handle to pushbutton_draw1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
    axes(handles.axes1);

imshow(handles.CT_Image_Total(:,:,handles.Slice_Number1_Now),[han
dles.CT_MINM handles.CT_MAXM]);
                                                                %%
选择 axes1 处为 ROI 目标
    h=imfreehand();
    wait(h);
    maskFreeHand=createMask(h);
    axes(handles.axes1);

imshow(handles.CT_Image_Total(:,:,handles.Slice_Number1_Now),[han
dles.CT_MINM handles.CT_MAXM]);
    hold on;
    contour(maskFreeHand,'r');

handles.Figure1_MAXM=-99999;                                     %
%所划区域的最大值

handles.Figure1_MINM=99999;                                      %
%所划区域的最小值

handles.Figure1_AVR=0.0;                                         %
%所划区域的平均值

handles.Figure1_S=0;                                            %%
所划区域的面积
    for
i=1:handles.SliceNumberTotal_2                                %%通
过循环寻找以上四种数据
        for k=1:handles.SliceNumberTotal_3
            if
(maskFreeHand(i,k)==1)                                  %%判断是否为
所划区域

handles.Figure1_S=handles.Figure1_S+1;                   %%S 先存储
像素点的个数

handles.Figure1_AVR=double(handles.Figure1_AVR+handles.CT_Image_T
otal(i,k,handles.Slice_Number1_Now));
                                                                %%
AVR 先存取所划矩阵的数值和
```

```
                if
handles.Figure1_MAXM<=handles.CT_Image_Total(i,k,handles.Slice_Nu
mber1_Now)

handles.Figure1_MAXM=double(handles.CT_Image_Total(i,k,handles.Sl
ice_Number1_Now));
                end
                if
handles.Figure1_MINM>=handles.CT_Image_Total(i,k,handles.Slice_Nu
mber1_Now)

handles.Figure1_MINM=double(handles.CT_Image_Total(i,k,handles.Sl
ice_Number1_Now));
                end
            end
        end
    end

handles.Figure1_AVR=double(handles.Figure1_AVR/handles.Figure1_S)
;       %%用总和/像素点个数得到平均值

handles.Figure1_S=handles.Figure1_S*handles.CT_Info.PixelSpacing(
1)*handles.CT_Info.PixelSpacing(2);
                                                                  %%
用像素个数*每个像素点的面积，得到总面积，单位为“/平方毫米”

set(handles.text1_MAXM,'String',num2str(handles.Figure1_MAXM));
      %%显示上述四种数据

set(handles.text1_MINM,'String',num2str(handles.Figure1_MINM));
    set(handles.text1_AVR,'String',num2str(handles.Figure1_AVR));
    set(handles.text1_S,'String',num2str(handles.Figure1_S));

    guidata(hObject,handles);
```

矢状面区域选择按钮代码展示：

```
function pushbutton_draw2_Callback(hObject, eventdata,
handles)                %%矢状面的 ROI 选择，与水平面同理，不再过多注释
% hObject    handle to pushbutton_draw2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
    axes(handles.axes2);

imshow(reshape(handles.CT_Image_Total(:,handles.Slice_Number2_Now,
:),[handles.SliceNumberTotal_3
handles.SliceNumberTotal_1])',[handles.CT_MINM handles.CT_MAXM]);
    h=imfreehand();
    wait(h);
    maskFreeHand=createMask(h);
    axes(handles.axes2);

imshow(reshape(handles.CT_Image_Total(:,handles.Slice_Number2_Now,
:),[handles.SliceNumberTotal_3
handles.SliceNumberTotal_1])',[handles.CT_MINM handles.CT_MAXM]);
    hold on;
    contour(maskFreeHand,'r');
    handles.Figure2_MAXM=-99999;
    handles.Figure2_MINM=99999;
    handles.Figure2_AVR=0.0;
    handles.Figure2_S=0;
    for i=1:handles.SliceNumberTotal_3
        for k=1:handles.SliceNumberTotal_1
            if (maskFreeHand(k,i)==1)
                handles.Figure2_S=handles.Figure2_S+1;

handles.Figure2_AVR=double(handles.Figure2_AVR+handles.CT_Image_T
otal(i,handles.Slice_Number2_Now,k));
                if
handles.Figure2_MAXM<=handles.CT_Image_Total(i,handles.Slice_Numb
er2_Now,k)

handles.Figure2_MAXM=double(handles.CT_Image_Total(i,handles.Slic
e_Number2_Now,k));
                end
                if
handles.Figure2_MINM>=handles.CT_Image_Total(i,handles.Slice_Numb
er2_Now,k)

handles.Figure2_MINM=double(handles.CT_Image_Total(i,handles.Slic
e_Number2_Now,k));
                end
            end
        end
    end

handles.Figure2_AVR=double(handles.Figure2_AVR/handles.Figure2_S)
;
```

```
handles.Figure2_S=handles.Figure2_S*handles.CT_Info.PixelSpacing(
1)*handles.CT_Info.PixelSpacing(2);


set(handles.text2_MAXM,'String',num2str(handles.Figure2_MAXM));

set(handles.text2_MINM,'String',num2str(handles.Figure2_MINM));
    set(handles.text2_AVR,'String',num2str(handles.Figure2_AVR));
    set(handles.text2_S,'String',num2str(handles.Figure2_S));

    guidata(hObject,handles);
```

冠状面区域选择按钮代码展示：

```
function pushbutton_draw3_Callback(hObject, eventdata,
handles)              %%冠状面的 ROI 选择，与水平面同理，不再过多注释
% hObject    handle to pushbutton_draw3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
    axes(handles.axes3);

imshow(reshape(handles.CT_Image_Total(handles.Slice_Number3_Now,:,
:),[handles.SliceNumberTotal_2
handles.SliceNumberTotal_1])',[handles.CT_MINM handles.CT_MAXM]);
    h=imfreehand();
    wait(h);
    maskFreeHand=createMask(h);
    axes(handles.axes3);

imshow(reshape(handles.CT_Image_Total(handles.Slice_Number3_Now,:,
:),[handles.SliceNumberTotal_2
handles.SliceNumberTotal_1])',[handles.CT_MINM handles.CT_MAXM]);
    hold on;
    contour(maskFreeHand,'r');
    handles.Figure3_MAXM=-99999;
    handles.Figure3_MINM=99999;
    handles.Figure3_AVR=0.0;
    handles.Figure3_S=0;
    for i=1:handles.SliceNumberTotal_2
        for k=1:handles.SliceNumberTotal_1
            if (maskFreeHand(k,i)==1)
                handles.Figure3_S=handles.Figure3_S+1;

handles.Figure3_AVR=double(handles.Figure3_AVR+handles.CT_Image_T
otal(handles.Slice_Number3_Now,i,k));
```

```
                if
handles.Figure3_MAXM<=handles.CT_Image_Total(handles.Slice_Number
3_Now,i,k)

handles.Figure3_MAXM=double(handles.CT_Image_Total(handles.Slice_
Number3_Now,i,k));
                end
                if
handles.Figure3_MINM>=handles.CT_Image_Total(handles.Slice_Number
3_Now,i,k)

handles.Figure3_MINM=double(handles.CT_Image_Total(handles.Slice_
Number3_Now,i,k));
                end
            end
        end
    end

handles.Figure3_AVR=double(handles.Figure3_AVR/handles.Figure3_S)
;

handles.Figure3_S=handles.Figure3_S*handles.CT_Info.PixelSpacing(
1)*handles.CT_Info.PixelSpacing(2);


set(handles.text3_MAXM,'String',num2str(handles.Figure3_MAXM));

set(handles.text3_MINM,'String',num2str(handles.Figure3_MINM));
    set(handles.text3_AVR,'String',num2str(handles.Figure3_AVR));
    set(handles.text3_S,'String',num2str(handles.Figure3_S));

    guidata(hObject,handles);
```