

---

```

function varargout = decorr3DGUI(varargin)
% DECORR3DGUI MATLAB code for decorr3DGUI.fig
%     DECORR3DGUI, by itself, creates a new DECORR3DGUI or raises the
%     existing
%     singleton*.
%
%     H = DECORR3DGUI returns the handle to a new DECORR3DGUI or the
%     handle to
%     the existing singleton*.
%
%     DECORR3DGUI('CALLBACK',hObject,eventData,handles,...) calls the
%     local
%     function named CALLBACK in DECORR3DGUI.M with the given input
%     arguments.
%
%     DECORR3DGUI('Property','Value',...) creates a new DECORR3DGUI
%     or raises the
%     existing singleton*. Starting from the left, property value
%     pairs are
%     applied to the GUI before decorr3DGUI_OpeningFcn gets called.
%     An
%     unrecognized property name or invalid value makes property
%     application
%     stop. All inputs are passed to decorr3DGUI_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 decorr3DGUI

% Last Modified by GUIDE v2.5 11-Apr-2019 20:50:12

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @decorr3DGUI_OpeningFcn, ...
                  'gui_OutputFcn',   @decorr3DGUI_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 decorr3DGUI is made visible.
function decorr3DGUI_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 decorr3DGUI (see VARARGIN)

% Choose default command line output for decorr3DGUI
handles.output = hObject;

% Update handles structure
setSerialPopUps(hObject, eventdata,handles);
handles.currComputationTime = 0;
handles.maxbscan = 1;
handles.maxdecorr = 1;
guidata(hObject, handles);
axes(handles.axes12);
disableButtonsForStart(hObject, eventdata, handles);

imshow('index.png');
% UIWAIT makes decorr3DGUI wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = decorr3DGUI_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;

% --- Executes on button press in dataSelect.
function dataSelect_Callback(hObject, eventdata, handles)
% hObject    handle to dataSelect (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
handles.activeExperiment = ExperimentClass();
handles.activeExperiment.initDataFolderGUI();
handles.myString = sprintf(handles.activeExperiment.activeFolder);
set(handles.dataSelectionString, 'String', handles.myString);
enableButtonsForStart(hObject, eventdata, handles)
updateSettingsButton_Callback(hObject, eventdata, handles)
guidata( hObject, handles);
% --- Executes on slider movement.

```

---

---

```

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

updateSliceImage_ROICorrected(hObject,eventdata,handles,'y');

% --- Executes during object creation, after setting all properties.
function ySlider_CreateFcn(hObject, eventdata, handles)
% hObject      handle to ySlider (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
               called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on button press in updateSettingsButton.
function updateSettingsButton_Callback(hObject, eventdata, handles)
% hObject      handle to updateSettingsButton (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
handles.sigmaLocal = (str2double(get(handles.windowSigma, 'String')));
handles.azimuthAngleLocal =
    2*pi*(str2double(get(handles.azimuthAngle, 'String')))/360;
handles.elevationAngleLocal =
    2*pi*(str2double(get(handles.rangeAngle, 'String')))/360;
handles.rmaxLocal = (str2double(get(handles.rMax, 'String')));
handles.rminLocal = 0 ;
handles.cartScalingFactorLocal =
    (str2double(get(handles.cartScalingFactor, 'String')));
handles.frameRate = (str2double(get(handles.framerate, 'String')));
handles.decorrthreshLocal =
    10^(str2double(get(handles.threshVal, 'String')));
handles.thetaminLocal = -handles.azimuthAngleLocal/2;
handles.thetamaxLocal = handles.azimuthAngleLocal/2;
handles.phiminLocal = -handles.elevationAngleLocal/2;
handles.phimaxLocal = handles.elevationAngleLocal/2;
handles.interFrameTimeLocal = 1/handles.frameRate;
handles.totalThreshLocal =
    (str2double(get(handles.totalThresh, 'String')));
handles.activeExperiment.ExperimentClassSetParams(handles.rminLocal,handles.rmaxLocal);
guidata( hObject, handles);
function rangeAngle_Callback(hObject, eventdata, handles)
% hObject      handle to rangeAngle (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 rangeAngle as text

```

---

---

```

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

% --- Executes during object creation, after setting all properties.
function rangeAngle_CreateFcn(hObject, eventdata, handles)
% hObject    handle to rangeAngle (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 slider movement.
function xSlider_Callback(hObject, eventdata, handles)
% hObject    handle to xSlider (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
updateSliceImage_ROICorrected(hObject,eventdata,handles,'x');

% --- Executes during object creation, after setting all properties.
function xSlider_CreateFcn(hObject, eventdata, handles)
% hObject    handle to xSlider (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on slider movement.
function zSlider_Callback(hObject, eventdata, handles)
% hObject    handle to zSlider (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

```

---

---

```

updateSliceImage_ROICorrected(hObject,eventdata,handles,'z');

% --- Executes during object creation, after setting all properties.
function zSlider_CreateFcn(hObject, eventdata, handles)
% hObject    handle to zSlider (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
              called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on button press in beginButton.
function beginButton_Callback(hObject, eventdata, handles)
% hObject    handle to beginButton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.beginButton, 'String', "Processing...");
drawnow;
handles.activeN = 1;
if(handles.activeExperiment.checkFolder())
    handles.activeExperiment.nextDataSetInFolder();
    updateSliceImage_ROICorrected(hObject,eventdata,handles,'x');
    updateSliceImage_ROICorrected(hObject,eventdata,handles,'y');
    updateSliceImage_ROICorrected(hObject,eventdata,handles,'z');
    set(handles.beginButton, 'String', "Ready");
    updateFrameDropDown(hObject,eventdata,handles)
    set(handles.beginButton,'Enable','off')
    enableButtonsControlPanel(hObject, eventdata, handles);
    setROIRange_sliders(hObject, eventdata, handles);
    handles.activeExperiment.initExperiment();
else
    set(handles.beginButton, 'String', "The folder is empty");
end
guidata( hObject, handles);
function azimuthAngle_Callback(hObject, eventdata, handles)
% hObject    handle to azimuthAngle (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 azimuthAngle as
        text
%        str2double(get(hObject,'String')) returns contents of
        azimuthAngle as a double

% --- Executes during object creation, after setting all properties.
function azimuthAngle_CreateFcn(hObject, eventdata, handles)
% hObject    handle to azimuthAngle (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 windowSigma_Callback(hObject, eventdata, handles)
% hObject      handle to windowSigma (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 windowSigma as text
%         str2double(get(hObject,'String')) returns contents of
%         windowSigma as a double

% --- Executes during object creation, after setting all properties.
function windowSigma_CreateFcn(hObject, eventdata, handles)
% hObject      handle to windowSigma (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 rMax_Callback(hObject, eventdata, handles)
% hObject      handle to rMax (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 rMax as text
%         str2double(get(hObject,'String')) returns contents of rMax as
%         a double

% --- Executes during object creation, after setting all properties.
function rMax_CreateFcn(hObject, eventdata, handles)
% hObject      handle to rMax (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 cartScalingFactor_Callback(hObject, eventdata, handles)
% hObject    handle to cartScalingFactor (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 cartScalingFactor
%         as text
%         str2double(get(hObject,'String')) returns contents of
%         cartScalingFactor as a double

% --- Executes during object creation, after setting all properties.
function cartScalingFactor_CreateFcn(hObject, eventdata, handles)
% hObject    handle to cartScalingFactor (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 framerate_Callback(hObject, eventdata, handles)
% hObject    handle to framerate (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 framerate as text
%         str2double(get(hObject,'String')) returns contents of
%         framerate as a double

% --- Executes during object creation, after setting all properties.
function framerate_CreateFcn(hObject, eventdata, handles)
% hObject    handle to framerate (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 slider movement.
function frameSlider_Callback(hObject, eventdata, handles)
% hObject      handle to frameSlider (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

% --- Executes during object creation, after setting all properties.
function frameSlider_CreateFcn(hObject, eventdata, handles)
% hObject      handle to frameSlider (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
%              called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

function computeNextVol_Callback(hObject, eventdata, handles)
% hObject      handle to computeNextVol (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 computeNextVol as
%       text
%       str2double(get(hObject,'String')) returns contents of
%       computeNextVol as a double
%       tic
set(handles.computeNextVol, 'String', "Processing...");
drawnow;
if(handles.activeExperiment.checkFolder())
    handles.activeExperiment.nextDataSetInFolder();
    handles.activeN = handles.activeExperiment.folderIndex-1;
    set(handles.computeNextVol, 'String', "Compute Next Volume");
else
    set(handles.computeNextVol, 'String', "No more volumes remain");
    drawnow;
end
updateSliceImage_ROICorrected(hObject,eventdata,handles,'x');

```

---



---

```

updateSliceImage_ROICorrected(hObject,eventdata,handles,'y');
updateSliceImage_ROICorrected(hObject,eventdata,handles,'z');
updateDecorrPlot(hObject,eventdata,handles)
updateFrameDropDown(hObject,eventdata,handles)
drawnow;
guidata( hObject, handles);

% --- Executes during object creation, after setting all properties.
function computeNextVol_CreateFcn(hObject, eventdata, handles)
% hObject    handle to computeNextVol (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 cycleDisplayVol_Callback(hObject, eventdata, handles)
% hObject    handle to cycleDisplayVol (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 cycleDisplayVol as
         text
%       str2double(get(hObject,'String')) returns contents of
         cycleDisplayVol as a double

% --- Executes during object creation, after setting all properties.
function cycleDisplayVol_CreateFcn(hObject, eventdata, handles)
% hObject    handle to cycleDisplayVol (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 runVols_Callback(hObject, eventdata, handles)
% hObject    handle to runVols (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 runVols as text
%         str2double(get(hObject,'String')) returns contents of runVols
%         as a double
set(handles.computeNextVol, 'String', "Processing...");
drawnow;
while(handles.activeExperiment.checkFolder())
    handles.activeExperiment.nextDataSetInFolder();
    handles.activeN = handles.activeExperiment.folderIndex-1;
    set(handles.computeNextVol, 'String', "Compute Next Volume");
    updateSliceImage_ROICorrected(hObject,eventdata,handles,'x');
    updateSliceImage_ROICorrected(hObject,eventdata,handles,'y');
    updateSliceImage_ROICorrected(hObject,eventdata,handles,'z');
    updateFrameDropDown(hObject,eventdata,handles)
    updateDecorrPlot(hObject,eventdata,handles)
    drawnow;
end
set(handles.computeNextVol, 'String', "No more volumes remain");
drawnow;

guidata( hObject, handles);

% --- Executes during object creation, after setting all properties.
function runVols_CreateFcn(hObject, eventdata, handles)
% hObject    handle to runVols (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 backVol.
function backVol_Callback(hObject, eventdata, handles)
% hObject    handle to backVol (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles.activeN = handles.activeN - 1;
if(handles.activeN <= 1)
    handles.activeN = 1;
end
set(handles.selectVolPopup, 'Value',handles.activeN);
updateSliceImage_ROICorrected(hObject,eventdata,handles,'x');
updateSliceImage_ROICorrected(hObject,eventdata,handles,'y');
updateSliceImage_ROICorrected(hObject,eventdata,handles,'z');
updateDecorrPlot(hObject,eventdata,handles);

```

---

---

```

guidata( hObject, handles);
% --- Executes on button press in forwardVol.
function forwardVol_Callback(hObject, eventdata, handles)
% hObject      handle to forwardVol (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
handles.activeN = handles.activeN + 1;
if(handles.activeN > handles.activeExperiment.folderIndex-1)
    handles.activeN = handles.activeExperiment.folderIndex-1;
end
set(handles.selectVolPopup, 'Value',handles.activeN);
updateSliceImage_ROICorrected(hObject,eventdata,handles,'x');
updateSliceImage_ROICorrected(hObject,eventdata,handles,'y');
updateSliceImage_ROICorrected(hObject,eventdata,handles,'z');
updateDecorrPlot(hObject,eventdata,handles);
guidata( hObject, handles);

% --- Executes on selection change in selectVolPopup.
function selectVolPopup_Callback(hObject, eventdata, handles)
% hObject      handle to selectVolPopup (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
%         selectVolPopup contents as cell array
%         contents{get(hObject,'Value')} returns selected item from
%         selectVolPopup
%handles.sigma = (str2double(get(handles.selectVolPopup, 'Value')));
% --- Executes during object creation, after setting all properties.
handles.activeN = get(handles.selectVolPopup, 'Value');
updateSliceImage_ROICorrected(hObject,eventdata,handles,'x');
updateSliceImage_ROICorrected(hObject,eventdata,handles,'y');
updateSliceImage_ROICorrected(hObject,eventdata,handles,'z');
updateDecorrPlot(hObject,eventdata,handles)
guidata( hObject, handles);
function selectVolPopup_CreateFcn(hObject, eventdata, handles)
% hObject      handle to selectVolPopup (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
%              called

% Hint: popupmenu 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 realTimeButton.
function realTimeButton_Callback(hObject, eventdata, handles)
% hObject      handle to realTimeButton (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

```

---

---

```

% Hint: get(hObject,'Value') returns toggle state of realTimeButton

% --- Executes on button press in prerecordButton.
function prerecordButton_Callback(hObject, eventdata, handles)
% hObject    handle to prerecordButton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of prerecordButton

% --- Executes during object creation, after setting all properties.
function threshVal_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function threshVal_Callback(hObject, eventdata, handles)

% --- Executes on button press in pauseExperiment.
function pauseExperiment_Callback(hObject, eventdata, handles)
% hObject    handle to pauseExperiment (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 EndButton.
function EndButton_Callback(hObject, eventdata, handles)
% hObject    handle to EndButton (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 continueButton.
function continueButton_Callback(hObject, eventdata, handles)
% hObject    handle to continueButton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.continueButton,'value',0);
set(handles.continueButton,'enable','off');
set(handles.pauseButton,'enable','on');
handles.activeExperiment.totalThresh =
    str2double(get(handles.totalThresh,'String'));
updateDecorrPlot(hObject,eventdata,handles)
set(handles.currentStatusString,'String','Waiting for volumes');
drawnow;
if (handles.activeExperiment.decorrExceedsThresh())
    set(handles.currentStatusString,'String','Threshold Reached');
    set(handles.continueButton,'enable','on');
    set(handles.pauseButton,'enable','off');
else
    handles.activeExperiment.sendSerialData();

```

---

---

```

        while(~get(handles.pauseButton,'value') &&
~handles.activeExperiment.decorrExceedsThresh())
            if(handles.activeExperiment.checkFolder())
                set(handles.currentStatusString,'String','Processing
Volume');
                drawnow;
                tic;
                handles.activeExperiment.nextDataSetInFolder();
                handles.currComputationTime = toc;
                handles.activeN = handles.activeExperiment.folderIndex-1;

updateSliceImage_ROICorrected(hObject,eventdata,handles,'x');

updateSliceImage_ROICorrected(hObject,eventdata,handles,'y');

updateSliceImage_ROICorrected(hObject,eventdata,handles,'z');
                updateDecorrrPlot(hObject,eventdata,handles)
                updateFrameDropDown(hObject,eventdata,handles)
                computationTimeString =
strcat(num2str(handles.currComputationTime),{' ','s'});
                set(handles.compTimeString,'String',computationTimeString)
                drawnow
                %pause(.01);
            else
                set(handles.currentStatusString,'String','Waiting for
volumes');
                drawnow;
            end

            pause(.01);
        end
    end
end
if(get(handles.pauseButton,'value'))
    set(handles.currentStatusString,'String','Paused');
    handles.activeExperiment.sendSerialData();
end
set(handles.pauseButton,'value',0)
if (handles.activeExperiment.decorrExceedsThresh())
    set(handles.currentStatusString,'String','Threshold Reached');
    set(handles.continueButton,'enable','on');
    set(handles.pauseButton,'enable','off');
    handles.activeExperiment.sendSerialData();
end

drawnow;
guidata( hObject, handles);

% --- Executes on button press in resetButton.
function resetButton_Callback(hObject, eventdata, handles)
% hObject    handle to resetButton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
disableButtonsForStart(hObject, eventdata, handles)
cla(handles.xCartVol);

```

---

---

```

cla(handles.yCartVol);
cla(handles.zCartVol);
cla(handles.xDecorrVol);
cla(handles.yDecorrVol);
cla(handles.zDecorrVol);
cla(handles.decorPlot);
set(handles.selectVolPopup,'String',{' '});
set(handles.beginButton, 'String', 'Initialize');
removeSerialConnection();
guidata(hObject,handles);

function totalThresh_Callback(hObject, eventdata, handles)
% hObject      handle to totalThresh (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 totalThresh as text
%         str2double(get(hObject,'String')) returns contents of
%         totalThresh as a double

% --- Executes during object creation, after setting all properties.
function totalThresh_CreateFcn(hObject, eventdata, handles)
% hObject      handle to totalThresh (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 updateSliceImage(hObject,eventdata,handles,direction)
tempSize =
    size(handles.activeExperiment.ultrasoundDataSeries(handles.activeN).rawData_cart)
decorSliderValue
    = .1*(.0001+get(handles.dynRangeDecorrSlider,'Value'));
bscanSliderValue =
    10*(.0001+get(handles.dynRangeBScanSlider,'Value'));
switch direction
    case 'x'
        SliderValue =
            floor((tempSize(1)-1)*get(handles.xSlider,'Value')+1);

        % x axis
        axes(handles.xCartVol);

        imagesc(abs(log10(handles.activeExperiment.getDataSlice_cart(direction,handles.ac
[0 bscanSliderValue]));
        colormap(handles.xCartVol,gray);

```

---

---

```

        edgeVals =
(squeeze(abs(handles.activeExperiment.getDataSlice_decorrMask(direction,handles.a
    alphamask(edgeVals,[256 0 0],.3);
    % decorr plots
    % x axis
    axes(handles.xDecorrVol);

    imagesc(squeeze(abs(handles.activeExperiment.getDataSlice_cumulativeDecorr(direct
[0, decorrSliderValue]));
        set(handles.xDecorrVol,'XTicklabel',[])
        set(handles.xDecorrVol,'YTicklabel',[])
        colormap(handles.xDecorrVol,hot(20));
    case 'y'
        SliderValue =
floor((tempSize(2)-1)*get(handles.ySlider,'Value'))+1;
        % x axis
        axes(handles.yCartVol);

    imagesc(abs(log10(handles.activeExperiment.getDataSlice_cart(direction,handles.ac
[0 bscanSliderValue]));
        colormap(handles.yCartVol,gray);
        edgeVals =
(squeeze(abs(handles.activeExperiment.getDataSlice_decorrMask(direction,handles.a
    alphamask(edgeVals,[256 0 0],.3);
    % decorr plots
    % x axis
    axes(handles.yDecorrVol);

    imagesc(squeeze(abs(handles.activeExperiment.getDataSlice_cumulativeDecorr(direct
[0, decorrSliderValue]));
        set(handles.yDecorrVol,'XTicklabel',[])
        set(handles.yDecorrVol,'YTicklabel',[])
        colormap(handles.yDecorrVol,hot(20));
    case 'z'
        SliderValue =
floor((tempSize(3)-1)*get(handles.zSlider,'Value'))+1;
        % x axis
        axes(handles.zCartVol);

    imagesc(abs(log10(handles.activeExperiment.getDataSlice_cart(direction,handles.ac
[0 bscanSliderValue]));
        colormap(handles.zCartVol,gray);
        edgeVals =
(squeeze(abs(handles.activeExperiment.getDataSlice_decorrMask(direction,handles.a
    alphamask(edgeVals,[256 0 0],.3);
    % decorr plots
    axes(handles.zDecorrVol);

    imagesc(squeeze(abs(handles.activeExperiment.getDataSlice_cumulativeDecorr(direct
[0, decorrSliderValue]));
        set(handles.zDecorrVol,'XTicklabel',[])

```

---

---

```

        set(handles.zDecorrVol, 'YTicklabel', [])
        colormap(handles.zDecorrVol, hot(20));
    otherwise
end

function resetAll(hObject, eventdata, handles, direction)
    set(handles.beginButton, 'Enable', 'on')
    set(handles.beginButton, 'String', 'Initialize')

function updateFrameDropDown(hObject, eventdata, handles)
    numFrameCell = {};
    for currCell = 1:handles.activeExperiment.folderIndex-1
        numFrameCell{currCell} = currCell;
    end
    set(handles.selectVolPopup, 'String', numFrameCell);

    set(handles.selectVolPopup, 'Value', handles.activeExperiment.folderIndex-1);
function updateDecorrPlot(hObject, eventdata, handles)
    threshValue = str2double(get(handles.totalThresh, 'String'));
    currLength = length(handles.activeExperiment.decorSumSeries);
    threshSeries = threshValue*ones(currLength,1);
    axes(handles.decorPlot);
    set(handles.decorPlot, 'YScale', 'log')

    plot(1:currLength, handles.activeExperiment.averageDecorr, 1:currLength, threshSeries);
    hold on;
    plot(handles.activeN,
handles.activeExperiment.averageDecorr(handles.activeN), 'r*')
    plot(handles.activeN,
log10(handles.activeExperiment.decorSumSeriesROI(handles.activeN)), 'r*')
    hold off;
    drawnow;
    legend('Cumulative Decorrelation Sum', 'target')

% --- Executes on button press in recomputeDecorr.
function recomputeDecorr_Callback(hObject, eventdata, handles)
% hObject      handle to recomputeDecorr (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
set(handles.recomputeDecorr, 'String', 'Processing...');
drawnow;
handles.activeExperiment.recomputeDecorr();
updateSliceImage_ROICorrected(hObject, eventdata, handles, 'x');
updateSliceImage_ROICorrected(hObject, eventdata, handles, 'y');
updateSliceImage_ROICorrected(hObject, eventdata, handles, 'z');
updateDecorrPlot(hObject, eventdata, handles);
set(handles.recomputeDecorr, 'String', 'Recompute Decorr');
drawnow;
guidata(hObject, handles);
function sendSerialData(hObject, eventdata, handles)

% --- Executes on selection change in inSerialPopUp.

```

---



---

```

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

% Hints: contents = cellstr(get(hObject,'String')) returns
%         inSerialPopUp contents as cell array
%         contents{get(hObject,'Value')} returns selected item from
%         inSerialPopUp

% --- Executes during object creation, after setting all properties.
function inSerialPopUp_CreateFcn(hObject, eventdata, handles)
% hObject      handle to inSerialPopUp (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
%              called

% Hint: popupmenu 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 selection change in outSerialPopUp.
function outSerialPopUp_Callback(hObject, eventdata, handles)
% hObject      handle to outSerialPopUp (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
%         outSerialPopUp contents as cell array
%         contents{get(hObject,'Value')} returns selected item from
%         outSerialPopUp

% --- Executes during object creation, after setting all properties.
function outSerialPopUp_CreateFcn(hObject, eventdata, handles)
% hObject      handle to outSerialPopUp (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
%              called

% Hint: popupmenu 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 setSerialPopUps(hObject, eventdata,handles)
tempDir = seriallist;

```

---

---

```

tempArr = {};
for n = 1:length(tempDir)
    tempArr{n} = tempDir(n);
end
set(handles.inSerialPopUp, 'String',tempArr);
set(handles.outSerialPopUp, 'String',tempArr);

% --- Executes on slider movement.
function dynRangeDecorrSlider_Callback(hObject, eventdata, handles)
% hObject      handle to dynRangeDecorrSlider (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.dynRangeMaxDecorr = handles.dynRangeMinVal
+4*(handles.dynRangeMinVal)*((get(handles.dynRangeDecorrSlider,'Value')));
updateSliceImage_ROICorrected(hObject,eventdata,handles,'x');
updateSliceImage_ROICorrected(hObject,eventdata,handles,'y');
updateSliceImage_ROICorrected(hObject,eventdata,handles,'z');
drawnow;
guidata(hObject, handles);
% --- Executes during object creation, after setting all properties.
function dynRangeDecorrSlider_CreateFcn(hObject, eventdata, handles)
% hObject      handle to dynRangeDecorrSlider (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
%              called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on slider movement.
function dynRangeBScanSlider_Callback(hObject, eventdata, handles)
% hObject      handle to dynRangeBScanSlider (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
updateSliceImage_ROICorrected(hObject,eventdata,handles,'x');
updateSliceImage_ROICorrected(hObject,eventdata,handles,'y');
updateSliceImage_ROICorrected(hObject,eventdata,handles,'z');
drawnow;
guidata(hObject, handles);

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

```

---

---

```

function dynRangeBScanSlider_CreateFcn(hObject, eventdata, handles)
% hObject    handle to dynRangeBScanSlider (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
              called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

function disableButtonsForStart(hObject, eventdata, handles)
set(handles.resetButton, 'Enable', 'off');
set(handles.recomputeDecorr, 'Enable', 'off');
set(handles.beginButton, 'Enable', 'off');
set(handles.continueButton, 'Enable', 'off');
set(handles.computeNextVol, 'Enable', 'off');
set(handles.runVols, 'Enable', 'off');
set(handles.selectVolPopup, 'Enable', 'off');
set(handles.backVol, 'Enable', 'off');
set(handles.forwardVol, 'Enable', 'off');
set(handles.xSlider, 'Enable', 'off');
set(handles.ySlider, 'Enable', 'off');
set(handles.zSlider, 'Enable', 'off');
set(handles.dynRangeBScanSlider, 'Enable', 'off');
set(handles.dynRangeDecorrSlider, 'Enable', 'off');
set(handles.updateSettingsButton, 'Enable', 'off');
set(handles.pauseButton, 'enable', 'off');

function enableButtonsForStart(hObject, eventdata, handles)
set(handles.resetButton, 'Enable', 'on');
set(handles.recomputeDecorr, 'Enable', 'on');
set(handles.beginButton, 'Enable', 'on');
set(handles.updateSettingsButton, 'Enable', 'on');

function enableButtonsControlPanel(hObject, eventdata, handles)
set(handles.continueButton, 'Enable', 'on');
set(handles.computeNextVol, 'Enable', 'on');
set(handles.runVols, 'Enable', 'on');
set(handles.selectVolPopup, 'Enable', 'on');
set(handles.backVol, 'Enable', 'on');
set(handles.forwardVol, 'Enable', 'on');
set(handles.xSlider, 'Enable', 'on');
set(handles.ySlider, 'Enable', 'on');
set(handles.zSlider, 'Enable', 'on');
set(handles.dynRangeBScanSlider, 'Enable', 'on');
set(handles.dynRangeDecorrSlider, 'Enable', 'on');

% --- Executes on button press in pauseButton.
function pauseButton_Callback(hObject, eventdata, handles)
% hObject    handle to pauseButton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

```

---

---

```

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

% Hint: get(hObject,'Value') returns toggle state of pauseButton
set(handles.continueButton,'enable','on');
set(handles.pauseButton,'enable','off');

function dynRangeMin_Callback(hObject, eventdata, handles)
% hObject      handle to dynRangeMin (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 dynRangeMin as text
%          str2double(get(hObject,'String')) returns contents of
%          dynRangeMin as a double
tempStrinArr = get(handles.outSerialPopUp,'String');
handles.activeExperiment.outSerialString =
    tempStrinArr{get(handles.outSerialPopUp, 'Value')};

handles.dynRangeMinVal = get(handles.dynRangeMin,'String');
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function dynRangeMin_CreateFcn(hObject, eventdata, handles)
% hObject      handle to dynRangeMin (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 sendButton.
function sendButton_Callback(hObject, eventdata, handles)
% hObject      handle to sendButton (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
%handles.activeExperiment.setUpSerialOutConnection();
handles.activeExperiment.sendSerialData();

% --- Executes on button press in serialSetupButton.
function serialSetupButton_Callback(hObject, eventdata, handles)
% hObject      handle to serialSetupButton (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
tempStrinArr = get(handles.outSerialPopUp,'String');

```

---

---

```

handles.activeExperiment.outSerialString =
    tempStrinArr{get(handles.outSerialPopUp, 'Value')}
display(handles.activeExperiment.outSerialString);
handles.activeExperiment.setUpSerialOutConnection();

% --- Executes on button press in roiSelectButton.
function roiSelectButton_Callback(hObject, eventdata, handles)
% hObject    handle to roiSelectButton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
axes(handles.xCartVol);
[BW Xi_xslice, Yi_xSlice] = roipoly;
axes(handles.yCartVol);
[BW Xi_yslice, Yi_ySlice] = roipoly;
axes(handles.zCartVol);
[BW Xi_zslice, Yi_zSlice] = roipoly;

% --- Executes on slider movement.
function xroiSlider_1_Callback(hObject, eventdata, handles)
% hObject    handle to xroiSlider_1 (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
updateSliceImage_ROICorrected(hObject,eventdata,handles,'x');
updateSliceImage_ROICorrected(hObject,eventdata,handles,'y');
updateSliceImage_ROICorrected(hObject,eventdata,handles,'z');
handles.activeExperiment.updateROIDataSet();
drawnow;
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function xroiSlider_1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to xroiSlider_1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
%            called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on slider movement.
function xroiSlider_2_Callback(hObject, eventdata, handles)
% hObject    handle to xroiSlider_2 (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
updateSliceImage_ROICorrected(hObject,eventdata,handles,'x');
updateSliceImage_ROICorrected(hObject,eventdata,handles,'y');
updateSliceImage_ROICorrected(hObject,eventdata,handles,'z');
handles.activeExperiment.updateROIDataSet();
drawnow;
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function xroiSlider_2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to xroiSlider_2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
%            called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on slider movement.
function yroiSlider_1_Callback(hObject, eventdata, handles)
% hObject    handle to yroiSlider_1 (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
updateSliceImage_ROICorrected(hObject,eventdata,handles,'x');
updateSliceImage_ROICorrected(hObject,eventdata,handles,'y');
updateSliceImage_ROICorrected(hObject,eventdata,handles,'z');
handles.activeExperiment.updateROIDataSet();
drawnow;
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function yroiSlider_1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to yroiSlider_1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
%            called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

```

---

---

```

% --- Executes on slider movement.
function yroiSlider_2_Callback(hObject, eventdata, handles)
% hObject      handle to yroiSlider_2 (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
updateSliceImage_ROICorrected(hObject,eventdata,handles,'x');
updateSliceImage_ROICorrected(hObject,eventdata,handles,'y');
updateSliceImage_ROICorrected(hObject,eventdata,handles,'z');
handles.activeExperiment.updateROIDataSet();
drawnow;
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function yroiSlider_2_CreateFcn(hObject, eventdata, handles)
% hObject      handle to yroiSlider_2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
%              called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on slider movement.
function zroiSlider_1_Callback(hObject, eventdata, handles)
% hObject      handle to zroiSlider_1 (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
updateSliceImage_ROICorrected(hObject,eventdata,handles,'x');
updateSliceImage_ROICorrected(hObject,eventdata,handles,'y');
updateSliceImage_ROICorrected(hObject,eventdata,handles,'z');
drawnow;
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function zroiSlider_1_CreateFcn(hObject, eventdata, handles)
% hObject      handle to zroiSlider_1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
%              called

% Hint: slider controls usually have a light gray background.

```

---

---

```

if isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on slider movement.
function slider19_Callback(hObject, eventdata, handles)
% hObject      handle to slider19 (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
updateSliceImage_ROICorrected(hObject,eventdata,handles,'z');
drawnow;
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function slider19_CreateFcn(hObject, eventdata, handles)
% hObject      handle to slider19 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
%              called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

function
    updateSliceImage_ROICorrected(hObject,eventdata,handles,direction)
tempSize =
    size(handles.activeExperiment.ultrasoundDataSeries(handles.activeN).rawData_cart)
decorrSliderValue
    = .1*(.0001+get(handles.dynRangeDecorrSlider,'Value'));
bscanSliderValue =
    10*(.0001+get(handles.dynRangeBScanSlider,'Value'));
xMax = tempSize(3);
yMax = tempSize(2);
zMax = tempSize(1);
xMid = floor(xMax/2);
yMid = floor(yMax/2);
zMid = floor(zMax/2);

handles.x_roi_left =
    floor((xMid-1)*get(handles.xroiSlider_1,'Value'))+1; %1:94
handles.x_roi_right = xMid
+floor((xMid)*get(handles.xroiSlider_2,'Value')); % 94:188
handles.y_roi_left =
    floor((yMid-1)*get(handles.xroiSlider_3,'Value'))+1; %1:94

```

---



---

```

handles.y_roi_right =
    floor((yMid)+floor((yMid)*get(handles.xroiSlider_4, 'Value')));
handles.z_roi_left =
    floor((zMid)*get(handles.yroiSlider_1, 'Value'))+1; % 1:74
handles.z_roi_right =
    floor(zMid-1)+floor((zMid)*get(handles.yroiSlider_2, 'Value'));
handles.activeExperiment.ROI_xRange =
    [handles.x_roi_left, handles.x_roi_right]
handles.activeExperiment.ROI_yRange =
    [handles.y_roi_left, handles.y_roi_right]
handles.activeExperiment.ROI_zRange =
    [handles.z_roi_left, handles.z_roi_right]

switch direction
    case 'x'
        SliderValue =
            floor((tempSize(1)-1)*get(handles.xSlider, 'Value'))+1;

            % x axis
            axes(handles.xCartVol);
            tempPic =
handles.activeExperiment.getDataSlice_cart(direction, handles.activeN, 1, SliderValue);

            imagesc(abs(log10(tempPic)), [0 bscanSliderValue]);
            colormap(handles.xCartVol, gray);

            edgeVals =
(squeeze(abs(handles.activeExperiment.getDataSlice_decorrMask(direction, handles.a
            alphamask(edgeVals, [256 0 0], .3);
            hold on;

rectangle('position', drawRect(handles.x_roi_left, handles.y_roi_left, handles.x_roi
            hold off;
            % decorr plots
            % x axis
            axes(handles.xDecorrVol);

            imagesc(squeeze(abs(handles.activeExperiment.getDataSlice_cumulativeDecorr(direct
[0, decorrSliderValue]));
            set(handles.xDecorrVol, 'XTicklabel', [])
            set(handles.xDecorrVol, 'YTicklabel', [])
            colormap(handles.xDecorrVol, hot(20));
        case 'y'
            SliderValue =
                floor((tempSize(2)-1)*get(handles.ySlider, 'Value'))+1;
                % y axis
                axes(handles.yCartVol);
                % y axis
                tempPic =
handles.activeExperiment.getDataSlice_cart(direction, handles.activeN, 1, SliderValue);

%         tempPic(:, 1:handles.x_roi_left) = 0;
%         tempPic(:, handles.x_roi_right:end) = 0;
%         tempPic(1:handles.z_roi_left, :) = 0;

```

---

---

```

%         tempPic(handles.z_roi_right:end,:) = 0;

        imagesc(abs(log10(tempPic)),[0 bscanSliderValue]);
        colormap(handles.yCartVol,gray);
        edgeVals =
(squeeze(abs(handles.activeExperiment.getDataSlice_decorrMask(direction,handles.a
        alphamask(edgeVals,[256 0 0],.3);
        hold on;

rectangle('position',drawRect(handles.x_roi_left,handles.z_roi_left,handles.x_roi
        hold off;
        % decorr plots
        % x axis
        axes(handles.yDecorrVol);

        imagesc(squeeze(abs(handles.activeExperiment.getDataSlice_cumulativeDecorr(direct
[0, decorrSliderValue]));
        set(handles.yDecorrVol,'XTicklabel',[])
        set(handles.yDecorrVol,'YTicklabel',[])
        colormap(handles.yDecorrVol,hot(20));
        case 'z'
            SliderValue =
            floor((tempSize(1)-1)*get(handles.zSlider,'Value')+1;
            % x axis
            axes(handles.zCartVol);
            % x axis
            tempPic =
            handles.activeExperiment.getDataSlice_cart(direction,handles.activeN,1,SliderValu
%         tempPic(:,1:handles.x_roi_left) = 0;
%         tempPic(:,handles.x_roi_right:end) = 0;
%         tempPic(1:handles.y_roi_left,:) = 0;
%         tempPic(handles.y_roi_right:end,:) = 0;

            imagesc(abs(log10(tempPic)),[0 bscanSliderValue]);
            colormap(handles.zCartVol,gray);
            edgeVals =
(squeeze(abs(handles.activeExperiment.getDataSlice_decorrMask(direction,handles.a
            alphamask(edgeVals,[256 0 0],.3);
            hold on;

rectangle('position',drawRect(handles.y_roi_left,handles.z_roi_left,handles.y_roi
            hold off;
            % decorr plots
            axes(handles.zDecorrVol);

            imagesc(squeeze(abs(handles.activeExperiment.getDataSlice_cumulativeDecorr(direct
[0, decorrSliderValue]));
            set(handles.zDecorrVol,'XTicklabel',[])
            set(handles.zDecorrVol,'YTicklabel',[])
            colormap(handles.zDecorrVol,hot(20));
        otherwise
end

```

---

---

```

% --- Executes on slider movement.
function xroiSlider_3_Callback(hObject, eventdata, handles)
% hObject      handle to xroiSlider_3 (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
updateSliceImage_ROICorrected(hObject,eventdata,handles,'x');
updateSliceImage_ROICorrected(hObject,eventdata,handles,'y');
updateSliceImage_ROICorrected(hObject,eventdata,handles,'z');
drawnow;
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function xroiSlider_3_CreateFcn(hObject, eventdata, handles)
% hObject      handle to xroiSlider_3 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
%              called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on slider movement.
function xroiSlider_4_Callback(hObject, eventdata, handles)
% hObject      handle to xroiSlider_4 (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
updateSliceImage_ROICorrected(hObject,eventdata,handles,'x');
updateSliceImage_ROICorrected(hObject,eventdata,handles,'y');
updateSliceImage_ROICorrected(hObject,eventdata,handles,'z');
drawnow;
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function xroiSlider_4_CreateFcn(hObject, eventdata, handles)
% hObject      handle to xroiSlider_4 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
%              called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))

```

---

---

```

        set(hObject,'BackgroundColor',[.9 .9 .9]);
    end

    function setROIRange_sliders(hObject, eventdata, handles)
    tempSize =
        size(handles.activeExperiment.ultrasoundDataSeries(handles.activeN).rawData_cart);
    xMax = tempSize(3);
    yMax = tempSize(2);
    zMax = tempSize(1);
    xMid = floor(xMax/2);
    yMid = floor(yMax/2);
    zMid = floor(zMax/2);

    handles.x_roi_left =
        floor((xMid-1)*get(handles.xroiSlider_1,'Value'))+1; %1:94
    handles.x_roi_right = xMid
    +floor((xMid)*get(handles.xroiSlider_2,'Value')); % 94:188
    handles.y_roi_left =
        floor((yMid-1)*get(handles.xroiSlider_3,'Value'))+1; %1:94
    handles.y_roi_right =
        floor((yMid)+floor((yMid)*get(handles.xroiSlider_4,'Value')));
    handles.z_roi_left =
        floor((zMid)*get(handles.yroiSlider_1,'Value'))+1; % 1:74
    handles.z_roi_right =
        floor(zMid-1)+floor((zMid)*get(handles.yroiSlider_2,'Value'));
    handles.activeExperiment.ROI_xRange =
        [handles.x_roi_left,handles.x_roi_right]
    handles.activeExperiment.ROI_yRange =
        [handles.y_roi_left,handles.y_roi_right]
    handles.activeExperiment.ROI_zRange =
        [handles.z_roi_left,handles.z_roi_right]
    guidata(hObject, handles);

    function rectInput = drawRect(p1,p2,p3,p4)
        pWidth = p3-p1;
        pHeight = p4-p2;
        rectInput = [p1,p2,pWidth,pHeight];

```

*Error using dbstatus*

*Error: File: /Users/peter/Documents/MATLAB/3DEchoDecorrelationScripts/  
DataScripts/Matlab/GUI/decorr3DGUI.m Line: 209 Column: 36  
The input character is not valid in MATLAB statements or expressions.*

*Published with MATLAB® R2016a*