function [TotTim,TotDis,AvgSpeed,AvgAcc,AvgDeacc,Active,NoFrame,X,Y,Z,x\_max,y\_max,z\_max,V,A,Ang,AngNo,X1pix,Y1pix,Z1pix] = AvgCal(XYZ,SI1,TI1,Tank)

T = 1/8 - XYZ(:,1);

T=T.\*0+1/8;

TotTim=sum(T);

Mx1 = Tank.Length;%size(SI1,2);

My1 = Tank.Height;%size(SI1,1);

% R1 = size(SI1,2)/size(TI1,2);

x\_max=Mx1;

y\_max=Tank.Width;%size(TI1,1)\*R1;

z\_max=My1;

% X = Mx1-XYZ(:,2);

% Z = My1-XYZ(:,3);

% Y = XYZ(:,5)\*R1;

X = XYZ(:,2);

Z = z\_max-XYZ(:,3);

Y = XYZ(:,5);

X1pix=X\*(size(SI1,2)/Tank.Length);

Y1pix=Y\*(size(TI1,1)/Tank.Width);

Z1pix=(size(SI1,1))-Z\*(size(SI1,1)/Tank.Height);

D = sqrt((X(1:end-1)-X(2:end)).^2+(Y(1:end-1)-Y(2:end)).^2+(Z(1:end-1)-Z(2:end)).^2);

V = D./(T(2:end));

A = (V(1:end-1)-V(2:end))./(T(1:end-2)+T(2:end-1));

X1=X(1:end-2);

X2=X(2:end-1);

X3=X(3:end);

Y1=Y(1:end-2);

Y2=Y(2:end-1);

Y3=Y(3:end);

Z1=Z(1:end-2);

Z2=Z(2:end-1);

Z3=Z(3:end);

a=[X(1:end-2)-X(2:end-1),Y(1:end-2)-Y(2:end-1),Z(1:end-2)-Z(2:end-1)];

b=[X(2:end-1)-X(3:end-0),Y(2:end-1)-Y(3:end-0),Z(2:end-1)-Z(3:end-0)];

th=acosd((sum(a'.\*b'))./(sqrt(sum(a'.^2)).\*sqrt(sum(b'.^2))));

th(isnan(th))=0;

th(end+1:end+2)=th(end);

Ang =th';

V(end+1)=V(end);

A(end+1:end+2)=A(end);

% AB=sqrt((X1-X2).^2+(Y1-Y2).^2+(Z1-Z2).^2);

% BC=sqrt((X2-X3).^2+(Y2-Y3).^2+(Z2-Z3).^2);

% AC=sqrt((X3-X1).^2+(Y3-Y1).^2+(Z3-Z1).^2);

%

%

% Ang=acosd((BC.^2+AB.^2-AC.^2)./(2.\*BC.\*AB));

AngNo=100\*sum(Ang>90)/length(Ang);

TotDis=sum(D);

AvgSpeed=mean(V);

AvgAcc=mean(A(A>0));

AvgDeacc=-1\*mean(A(A<0));

Active=100\*sum(D>0.01)./length(D);

NoFrame=length(A);

s

function varargout = ProcessData(varargin)

% PROCESSDATA MATLAB code for ProcessData.fig

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

% singleton\*.

%

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

% the existing singleton\*.

%

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

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

%

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

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

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

% unrecognized property name or invalid value makes property application

% stop. All inputs are passed to ProcessData\_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 ProcessData

% Last Modified by GUIDE v2.5 07-Jun-2014 00:35:35

% Begin initialization code - DO NOT EDIT

gui\_Singleton = 1;

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

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

'gui\_OpeningFcn', @ProcessData\_OpeningFcn, ...

'gui\_OutputFcn', @ProcessData\_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 ProcessData is made visible.

function ProcessData\_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 ProcessData (see VARARGIN)

% Choose default command line output for ProcessData

handles.output = hObject;

% Update handles structure

guidata(hObject, handles);

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

% uiwait(handles.figure1);

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

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

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

% hObject handle to figure

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

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

% Get default command line output from handles structure

varargout{1} = handles.output;

function edit1\_Callback(hObject, eventdata, handles)

% hObject handle to edit1 (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 edit1 as text

% str2double(get(hObject,'String')) returns contents of edit1 as a double

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

function edit1\_CreateFcn(hObject, eventdata, handles)

% hObject handle to edit1 (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 pushbutton1.

function pushbutton1\_Callback(hObject, eventdata, handles)

% 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)

global FileName

global XYZ

global SI1

global TI1

global Tank

FileName=' ';

PathName=' ';

[FileName,PathName,FilterIndex] = uigetfile('\*.mat');

if(PathName~=0)

FileName=[PathName,FileName];

set(handles.edit1,'String',FileName);

load(FileName);

R1 = size(SI1,2)/size(TI1,2);

Tank.Height=str2double(get(handles.edit13,'String'));

Tank.Length=str2double(get(handles.edit14,'String'));

Tank.Width=str2double(get(handles.edit15,'String'));

Rlt=Tank.Length/size(TI1,2);

Rwt=Tank.Width/size(TI1,1);

Rls=Tank.Length/(size(SI1,2));

Rhs=Tank.Height/(size(SI1,1));

XYZ1=XYZ;

XYZ(:,2)=XYZ(:,2).\*Rls;

XYZ(:,3)=XYZ(:,3).\*Rhs;

XYZ(:,4)=XYZ(:,4).\*Rlt;

XYZ(:,5)=XYZ(:,5).\*Rwt;

if(strcmp(get(handles.axes1,'Visible'),'off')==1)

set(handles.axes1,'Visible','on');

end

pushbutton2\_Callback(hObject, eventdata, handles);

pushbutton7\_Callback(hObject, eventdata, handles);

else

set(handles.edit1,'String','Please Select \*.mat file');

end

% --- Executes on slider movement.

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

global X

global Y

global Z

global V

global x\_max

global y\_max

global z\_max

temp3=uint32(get(handles.slider1,'Value'))+1;

v1=get(handles.edit3,'String');

v2=get(handles.edit4,'String');

v3=get(handles.edit5,'String');

l1=get(handles.edit6,'String');

v1=str2num(v1);

v2=str2num(v2);

v3=str2num(v3);

l1=ceil(str2num(l1)/2);

VV=round(mean(V(min(temp3,length(V)):min(temp3+30,length(V)))));

if(VV<=v1)

c='g';

elseif(VV>v1 && VV<=v2)

c='y';

elseif(VV>v2 && VV<=v3)

c='m';

else

c='r';

end

cla(handles.axes1);

xlim(handles.axes1,[0 x\_max]);

ylim(handles.axes1,[0 y\_max]);

zlim(handles.axes1,[0 z\_max]);

hold(handles.axes1,'on');

plot3(X(max(1,temp3-l1):min(temp3+l1,length(X))),Y(max(1,temp3-l1):min(temp3+l1,length(Y))),Z(max(1,temp3-l1):min(temp3+l1,length(Z))),'-\*','color',c,'Parent',handles.axes1);

plot3(X(min(temp3+l1,length(X))),Y(min(temp3+l1,length(Y))),Z(min(temp3+l1,length(Z))),'Ob','Parent',handles.axes1);

title(handles.axes1,['Avg. V = ',num2str(VV)]);

view(handles.axes1,3);

axis(handles.axes1,'on');

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

function slider1\_CreateFcn(hObject, eventdata, handles)

% hObject handle to slider1 (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 checkbox1.

function checkbox1\_Callback(hObject, eventdata, handles)

% hObject handle to checkbox1 (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 checkbox1

% --- Executes on button press in pushbutton2.

function pushbutton2\_Callback(hObject, eventdata, handles)

% 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)

global XYZ

global SI1

global TI1

global X

global Y

global Z

global x\_max

global y\_max

global z\_max

global V

global A

global Ang

global AngNo

global Tank

global X1pix

global Y1pix

global Z1pix

[TotTim,TotDis,AvgSpeed,AvgAcc,AvgDeacc,Active,NoFrame,X,Y,Z,x\_max,y\_max,z\_max,V,A,Ang,AngNo,X1pix,Y1pix,Z1pix] = AvgCal(XYZ,SI1,TI1,Tank);

MM=[0:0.1:x\_max];

xs=std(X);

ys=std(Y);

zs=std(Z);

rr=sqrt(xs.^2+ys.^2+zs.^2);

% vs=(4/3)\*pi\*rr.^3;

% vs=(2\*xs)\*(2\*ys)\*(2\*zs);

X1=12\*round(X/(12\*0.12));

Y1=5\*round(Y/(5\*0.16));

Z1=5\*round(Z/(5\*0.16));

DD=[X1,Y1,Z1];

[S1 S2 S3]=unique(round(DD),'rows');

vs=(length(S1).\*3\*pi\*0.5\*0.5);

va=x\_max.\*y\_max\*z\_max;

if(vs>va)

vs=0.97\*va;

end

%set(handles.text67,'String',[num2str(100\*3.14\*3\*3\*TotDis./(x\_max.\*y\_max\*z\_max)),'%']);

set(handles.text67,'String',[num2str(round(100\*100\*vs/va)/100),'%']);

set(handles.text24,'String',num2str(round(100\*TotTim)/100));

set(handles.text4,'String',num2str(round(100\*AvgSpeed)/100));

set(handles.text5,'String',num2str(round(100\*AvgAcc)/100));

set(handles.text26,'String',num2str(round(100\*AvgDeacc)/100));

set(handles.text6,'String',[num2str(round(100\*Active)/100),'%']);

set(handles.text8,'String',num2str(round(100\*TotDis)/100));

set(handles.text22,'String',[num2str(round(100\*AngNo)/100),'%']);

set(handles.slider1,'Max',NoFrame-1);

set(handles.slider2,'Max',size(SI1,1));

set(handles.slider3,'Max',size(SI1,1));

set(handles.slider4,'Max',size(SI1,1));

set(handles.slider5,'Max',size(SI1,1));

set(handles.slider6,'Max',size(SI1,1));

set(handles.slider7,'Max',size(SI1,1));

set(handles.slider2,'Value',0);

set(handles.slider3,'Value',0);

set(handles.slider4,'Value',0);

set(handles.slider5,'Value',0);

set(handles.slider6,'Value',0);

set(handles.slider7,'Value',0);

% --- Executes on button press in pushbutton3.

function pushbutton3\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton3 (see GCBO)

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

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

close

% --- Executes on button press in pushbutton4.

function pushbutton4\_Callback(hObject, eventdata, handles)

% 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)

global X

global Y

global Z

global timer1

global x\_max

global y\_max

global z\_max

global V

global A

temp1 = get(handles.checkbox1,'Value');

temp2 = str2num(get(handles.edit2,'String')); %plot framerate

if(temp2<0.02)

temp2=0.02;

end

if(temp1 == 1)

if(strcmp(get(handles.pushbutton4,'String'),'Stop')==0)

set(handles.pushbutton4,'String','Stop');

timer1.StartFcn = @(~,thisEvent)plotfun(handles,X,Y,Z,V,x\_max,y\_max,z\_max);

timer1.TimerFcn = @(~,thisEvent)plotfun(handles,X,Y,Z,V,x\_max,y\_max,z\_max);

timer1.Period=temp2;

timer1.ExecutionMode = 'fixedRate';

start(timer1);

else

stop(timer1);

set(handles.pushbutton4,'String','Plot');

end

else

%temp3=get(handles.slider1,'Value')+1;

slider1\_Callback(hObject, eventdata, handles)

end

function edit2\_Callback(hObject, eventdata, handles)

% hObject handle to edit2 (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 edit2 as text

% str2double(get(hObject,'String')) returns contents of edit2 as a double

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

function edit2\_CreateFcn(hObject, eventdata, handles)

% hObject handle to edit2 (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 edit3\_Callback(hObject, eventdata, handles)

% hObject handle to edit3 (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 edit3 as text

% str2double(get(hObject,'String')) returns contents of edit3 as a double

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

function edit3\_CreateFcn(hObject, eventdata, handles)

% hObject handle to edit3 (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 edit4\_Callback(hObject, eventdata, handles)

% hObject handle to edit4 (see GCBO)

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

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

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

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

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

function edit4\_CreateFcn(hObject, eventdata, handles)

% hObject handle to edit4 (see GCBO)

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

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

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

% See ISPC and COMPUTER.

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

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

end

function edit5\_Callback(hObject, eventdata, handles)

% hObject handle to edit5 (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 edit5 as text

% str2double(get(hObject,'String')) returns contents of edit5 as a double

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

function edit5\_CreateFcn(hObject, eventdata, handles)

% hObject handle to edit5 (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 edit6\_Callback(hObject, eventdata, handles)

% hObject handle to edit6 (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 edit6 as text

% str2double(get(hObject,'String')) returns contents of edit6 as a double

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

function edit6\_CreateFcn(hObject, eventdata, handles)

% hObject handle to edit6 (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 during object creation, after setting all properties.

function figure1\_CreateFcn(hObject, eventdata, handles)

% hObject handle to figure1 (see GCBO)

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

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

global timer1

global ROI1b

global ROI1t

global ROI2b

global ROI2t

global ROI3b

global ROI3t

ROI1b=1;

ROI1t=1;

ROI2b=1;

ROI2t=1;

ROI3b=1;

ROI3t=1;

timer1=timer;

function plotfun(handles,X,Y,Z,V,x\_max,y\_max,z\_max)

temp3=uint32(get(handles.slider1,'Value'))+1;

set( handles.slider1,'Value',temp3);

v1=get(handles.edit3,'String');

v2=get(handles.edit4,'String');

v3=get(handles.edit5,'String');

l1=get(handles.edit6,'String');

v1=str2num(v1);

v2=str2num(v2);

v3=str2num(v3);

l1=ceil(str2num(l1)/2);

VV=round(mean(V(temp3:min(temp3+30,length(X)))));

if(VV<=v1)

c='g';

elseif(VV>v1 && VV<=v2)

c='y';

elseif(VV>v2 && VV<=v3)

c='m';

else

c='r';

end

cla(handles.axes1);

xlim(handles.axes1,[0 x\_max]);

ylim(handles.axes1,[0 y\_max]);

zlim(handles.axes1,[0 z\_max]);

hold(handles.axes1,'on')

plot3(X(max(1,temp3-l1):min(temp3+l1,length(X))),Y(max(1,temp3-l1):min(temp3+l1,length(Y))),Z(max(1,temp3-l1):min(temp3+l1,length(Z))),'-\*','color',c,'Parent',handles.axes1);

plot3(X(min(temp3+l1,length(X))),Y(min(temp3+l1,length(Y))),Z(min(temp3+l1,length(Z))),'Ob','Parent',handles.axes1);

title(handles.axes1,['Avg. V = ',num2str(VV)]);

view(handles.axes1,3);

axis(handles.axes1,'on');

% --- Executes on button press in pushbutton5.

function pushbutton5\_Callback(hObject, eventdata, handles)

% 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)

global XYZ

global SI1

global TI1

global X

global Y

global Z

global x\_max

global y\_max

global z\_max

global V

global A

global X1pix

global Y1pix

global Z1pix

RGBm = repmat(SI1,[1,1,3]);

temp1=get(handles.checkbox2,'Value');

if(temp1==1)

circles=[X1pix Z1pix];

circles(:,3)=0.8;

RGBm=insertShape(RGBm,'FilledCircle',circles ,'Color','red','Opacity',0.1);

end

cla reset

cla(handles.axes1);

view(handles.axes1,[0 90]);

%imshow(SI1,'Parent',handles.axes1);

imshow(RGBm,'Parent',handles.axes1);

title(handles.axes1,'ROI');

set(handles.slider2,'Enable','on');

set(handles.slider3,'Enable','on');

set(handles.slider4,'Enable','on');

set(handles.slider5,'Enable','on');

set(handles.slider6,'Enable','on');

set(handles.slider7,'Enable','on');

set(handles.checkbox2,'Enable','on');

function edit9\_Callback(hObject, eventdata, handles)

% hObject handle to edit9 (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 edit9 as text

% str2double(get(hObject,'String')) returns contents of edit9 as a double

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

function edit9\_CreateFcn(hObject, eventdata, handles)

% hObject handle to edit9 (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 edit8\_Callback(hObject, eventdata, handles)

% hObject handle to edit8 (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 edit8 as text

% str2double(get(hObject,'String')) returns contents of edit8 as a double

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

function edit8\_CreateFcn(hObject, eventdata, handles)

% hObject handle to edit8 (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 edit7\_Callback(hObject, eventdata, handles)

% hObject handle to edit7 (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 edit7 as text

% str2double(get(hObject,'String')) returns contents of edit7 as a double

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

function edit7\_CreateFcn(hObject, eventdata, handles)

% hObject handle to edit7 (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 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

global XYZ

global SI1

global TI1

global X

global Y

global Z

global x\_max

global y\_max

global z\_max

global V

global A

global Ang

global ROI1b

global ROI1t

global ROI2b

global ROI2t

global ROI3b

global ROI3t

global X1pix

global Y1pix

global Z1pix

ROI1b=uint32(get(handles.slider2,'Value'));

rectangle1 = int32([0 ROI1t size(SI1,2) ROI1b]);

rectangle2 = int32([0 ROI2t size(SI1,2) ROI2b]);

rectangle3 = int32([0 ROI3t size(SI1,2) ROI3b]);

% set(handles.slider3,'Max',size(SI1,1)-ROI1t+1);

RGBm = repmat(SI1,[1,1,3]);

temp1=get(handles.checkbox2,'Value');

if(temp1==1)

circles=[X1pix Z1pix];

circles(:,3)=0.8;

RGBm=insertShape(RGBm,'FilledCircle',circles ,'Color','red','Opacity',0.1);

end

RGBm=insertShape(RGBm,'FilledRectangle',rectangle1,'Color','yellow','Opacity',0.4);

RGBm=insertShape(RGBm,'FilledRectangle',rectangle2,'Color','green','Opacity',0.4);

RGBm=insertShape(RGBm,'FilledRectangle',rectangle3,'Color','blue','Opacity',0.4);

cla(handles.axes1);

view(handles.axes1,[0 90]);

imshow(RGBm,'Parent',handles.axes1);

title(handles.axes1,'ROI');

% z\_to=double(size(SI1,1))-double(ROI1t);

% z\_from=double(size(SI1,1))-double(ROI1t)-double(ROI1b);

z\_from=double(ROI1t);

z\_to=double(ROI1t)+double(ROI1b);

AA=A;

AA(A<0)=0;

Ai=A;

Ai(A<=0)=0;

Ai(A>0)=1;

D = sqrt((X(1:end-1)-X(2:end)).^2+(Y(1:end-1)-Y(2:end)).^2+(Z(1:end-1)-Z(2:end)).^2);

D(end+1)=0;

AngV=Ang./V;

AngV(isnan(AngV))=0;

set(handles.text18,'String',[num2str(round(10000\*sum((Z1pix>=z\_from) & (Z1pix<=z\_to))/length(Z1pix))/100),'%']);

set(handles.text35,'String',[num2str(round(max(V(Z1pix>=z\_from & Z1pix<=z\_to))))]);

set(handles.text36,'String',[num2str(round(mean(V(Z1pix>=z\_from & Z1pix<=z\_to))))]);

set(handles.text37,'String',[num2str(round(max(AA(Z1pix>=z\_from & Z1pix<=z\_to))))]);

set(handles.text38,'String',[num2str(round(sum(AA(Z1pix>=z\_from & Z1pix<=z\_to))./(sum(Ai(Z1pix>=z\_from & Z1pix<=z\_to)))))]);

set(handles.text39,'String',[num2str(round(sum(D(Z1pix>=z\_from & Z1pix<=z\_to))))]);

set(handles.text40,'String',[num2str(round(mean(AngV(Z1pix>=z\_from & Z1pix<=z\_to))))]);

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

function slider2\_CreateFcn(hObject, eventdata, handles)

% hObject handle to slider2 (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 slider3\_Callback(hObject, eventdata, handles)

% 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

global XYZ

global SI1

global TI1

global X

global Y

global Z

global x\_max

global y\_max

global z\_max

global V

global A

global Ang

global ROI1b

global ROI1t

global ROI2b

global ROI2t

global ROI3b

global ROI3t

global X1pix

global Y1pix

global Z1pix

ROI1t=uint32(get(handles.slider3,'Value'));

rectangle1 = int32([0 ROI1t size(SI1,2) ROI1b]);

rectangle2 = int32([0 ROI2t size(SI1,2) ROI2b]);

rectangle3 = int32([0 ROI3t size(SI1,2) ROI3b]);

%set(handles.slider2,'Min',ROI1t+1);

RGBm = repmat(SI1,[1,1,3]);

temp1=get(handles.checkbox2,'Value');

if(temp1==1)

circles=[X1pix Z1pix];

circles(:,3)=0.8;

RGBm=insertShape(RGBm,'FilledCircle',circles ,'Color','red','Opacity',0.1);

end

RGBm=insertShape(RGBm,'FilledRectangle',rectangle1,'Color','yellow','Opacity',0.4);

RGBm=insertShape(RGBm,'FilledRectangle',rectangle2,'Color','green','Opacity',0.4);

RGBm=insertShape(RGBm,'FilledRectangle',rectangle3,'Color','blue','Opacity',0.4);

cla(handles.axes1);

view(handles.axes1,[0 90]);

imshow(RGBm,'Parent',handles.axes1);

title(handles.axes1,'ROI');

z\_from=double(ROI1t);

z\_to=double(ROI1t)+double(ROI1b);

%plot(X,size(SI1,1)-Z,'\*b')

AA=A;

AA(A<0)=0;

Ai=A;

Ai(A<=0)=0;

Ai(A>0)=1;

D = sqrt((X(1:end-1)-X(2:end)).^2+(Y(1:end-1)-Y(2:end)).^2+(Z(1:end-1)-Z(2:end)).^2);

D(end+1)=0;

AngV=Ang./V;

AngV(isnan(AngV))=0;

set(handles.text18,'String',[num2str(round(10000\*sum((Z1pix>=z\_from) & (Z1pix<=z\_to))/length(Z1pix))/100),'%']);

set(handles.text35,'String',[num2str(round(max(V(Z1pix>=z\_from & Z1pix<=z\_to))))]);

set(handles.text36,'String',[num2str(round(mean(V(Z1pix>=z\_from & Z1pix<=z\_to))))]);

set(handles.text37,'String',[num2str(round(max(AA(Z1pix>=z\_from & Z1pix<=z\_to))))]);

set(handles.text38,'String',[num2str(round(sum(AA(Z1pix>=z\_from & Z1pix<=z\_to))./(sum(Ai(Z1pix>=z\_from & Z1pix<=z\_to)))))]);

set(handles.text39,'String',[num2str(round(sum(D(Z1pix>=z\_from & Z1pix<=z\_to))))]);

set(handles.text40,'String',[num2str(round(mean(AngV(Z1pix>=z\_from & Z1pix<=z\_to))))]);

set(handles.text18,'String',[num2str(round(10000\*sum((Z1pix>=z\_from) & (Z1pix<=z\_to))/length(Z1pix))/100),'%']);

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

function slider3\_CreateFcn(hObject, eventdata, handles)

% hObject handle to slider3 (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 checkbox2.

function checkbox2\_Callback(hObject, eventdata, handles)

% hObject handle to checkbox2 (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 checkbox2

global ROI1b

global ROI1t

global ROI2b

global ROI2t

global ROI3b

global ROI3t

global X1pix

global Z1pix

global SI1

rectangle1 = int32([0 ROI1t size(SI1,2) ROI1b]);

rectangle2 = int32([0 ROI2t size(SI1,2) ROI2b]);

rectangle3 = int32([0 ROI3t size(SI1,2) ROI3b]);

% set(handles.slider3,'Max',size(SI1,1)-ROI1t+1);

RGBm = repmat(SI1,[1,1,3]);

temp1=get(handles.checkbox2,'Value');

if(temp1==1)

circles=[X1pix Z1pix];

circles(:,3)=0.8;

RGBm=insertShape(RGBm,'FilledCircle',circles ,'Color','red','Opacity',0.1);

end

RGBm=insertShape(RGBm,'FilledRectangle',rectangle1,'Color','yellow','Opacity',0.4);

RGBm=insertShape(RGBm,'FilledRectangle',rectangle2,'Color','green','Opacity',0.4);

RGBm=insertShape(RGBm,'FilledRectangle',rectangle3,'Color','blue','Opacity',0.4);

cla(handles.axes1);

view(handles.axes1,[0 90]);

imshow(RGBm,'Parent',handles.axes1);

title(handles.axes1,'ROI');

% --- Executes on slider movement.

function slider4\_Callback(hObject, eventdata, handles)

% hObject handle to slider4 (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

global XYZ

global SI1

global TI1

global X

global Y

global Z

global x\_max

global y\_max

global z\_max

global V

global A

global Ang

global ROI1b

global ROI1t

global ROI2b

global ROI2t

global ROI3b

global ROI3t

global X1pix

global Y1pix

global Z1pix

ROI2b=uint32(get(handles.slider4,'Value'));

rectangle1 = int32([0 ROI1t size(SI1,2) ROI1b]);

rectangle2 = int32([0 ROI2t size(SI1,2) ROI2b]);

rectangle3 = int32([0 ROI3t size(SI1,2) ROI3b]);

% set(handles.slider3,'Max',size(SI1,1)-ROI1t+1);

RGBm = repmat(SI1,[1,1,3]);

temp1=get(handles.checkbox2,'Value');

if(temp1==1)

circles=[X1pix Z1pix];

circles(:,3)=0.8;

RGBm=insertShape(RGBm,'FilledCircle',circles ,'Color','red','Opacity',0.1);

end

RGBm=insertShape(RGBm,'FilledRectangle',rectangle1,'Color','yellow','Opacity',0.4);

RGBm=insertShape(RGBm,'FilledRectangle',rectangle2,'Color','green','Opacity',0.4);

RGBm=insertShape(RGBm,'FilledRectangle',rectangle3,'Color','blue','Opacity',0.4);

cla(handles.axes1);

view(handles.axes1,[0 90]);

imshow(RGBm,'Parent',handles.axes1);

title(handles.axes1,'ROI');

z\_from=double(ROI2t);

z\_to=double(ROI2t)+double(ROI2b);

%plot(X,size(SI1,1)-Z,'\*b')

AA=A;

AA(A<0)=0;

Ai=A;

Ai(A<=0)=0;

Ai(A>0)=1;

D = sqrt((X(1:end-1)-X(2:end)).^2+(Y(1:end-1)-Y(2:end)).^2+(Z(1:end-1)-Z(2:end)).^2);

D(end+1)=0;

AngV=Ang./V;

AngV(isnan(AngV))=0;

set(handles.text19,'String',[num2str(round(10000\*sum((Z1pix>=z\_from) & (Z1pix<=z\_to))/length(Z1pix))/100),'%']);

set(handles.text47,'String',[num2str(round(max(V(Z1pix>=z\_from & Z1pix<=z\_to))))]);

set(handles.text48,'String',[num2str(round(mean(V(Z1pix>=z\_from & Z1pix<=z\_to))))]);

set(handles.text49,'String',[num2str(round(max(AA(Z1pix>=z\_from & Z1pix<=z\_to))))]);

set(handles.text50,'String',[num2str(round(sum(AA(Z1pix>=z\_from & Z1pix<=z\_to))./(sum(Ai(Z1pix>=z\_from & Z1pix<=z\_to)))))]);

set(handles.text51,'String',[num2str(round(sum(D(Z1pix>=z\_from & Z1pix<=z\_to))))]);

set(handles.text52,'String',[num2str(round(mean(AngV(Z1pix>=z\_from & Z1pix<=z\_to))))]);

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

function slider4\_CreateFcn(hObject, eventdata, handles)

% hObject handle to slider4 (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 slider5\_Callback(hObject, eventdata, handles)

% hObject handle to slider5 (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

global XYZ

global SI1

global TI1

global X

global Y

global Z

global x\_max

global y\_max

global z\_max

global V

global A

global Ang

global ROI1b

global ROI1t

global ROI2b

global ROI2t

global ROI3b

global ROI3t

global X1pix

global Y1pix

global Z1pix

ROI2t=uint32(get(handles.slider5,'Value'));

rectangle1 = int32([0 ROI1t size(SI1,2) ROI1b]);

rectangle2 = int32([0 ROI2t size(SI1,2) ROI2b]);

rectangle3 = int32([0 ROI3t size(SI1,2) ROI3b]);

%set(handles.slider2,'Min',ROI1t+1);

RGBm = repmat(SI1,[1,1,3]);

temp1=get(handles.checkbox2,'Value');

if(temp1==1)

circles=[X1pix Z1pix];

circles(:,3)=0.8;

RGBm=insertShape(RGBm,'FilledCircle',circles ,'Color','red','Opacity',0.1);

end

RGBm=insertShape(RGBm,'FilledRectangle',rectangle1,'Color','yellow','Opacity',0.4);

RGBm=insertShape(RGBm,'FilledRectangle',rectangle2,'Color','green','Opacity',0.4);

RGBm=insertShape(RGBm,'FilledRectangle',rectangle3,'Color','blue','Opacity',0.4);

cla(handles.axes1);

view(handles.axes1,[0 90]);

imshow(RGBm,'Parent',handles.axes1);

title(handles.axes1,'ROI');

z\_from=double(ROI2t);

z\_to=double(ROI2t)+double(ROI2b);

%plot(X,size(SI1,1)-Z,'\*b')

AA=A;

AA(A<0)=0;

Ai=A;

Ai(A<=0)=0;

Ai(A>0)=1;

D = sqrt((X(1:end-1)-X(2:end)).^2+(Y(1:end-1)-Y(2:end)).^2+(Z(1:end-1)-Z(2:end)).^2);

D(end+1)=0;

AngV=Ang./V;

AngV(isnan(AngV))=0;

set(handles.text19,'String',[num2str(round(10000\*sum((Z1pix>=z\_from) & (Z1pix<=z\_to))/length(Z1pix))/100),'%']);

set(handles.text47,'String',[num2str(round(max(V(Z1pix>=z\_from & Z1pix<=z\_to))))]);

set(handles.text48,'String',[num2str(round(mean(V(Z1pix>=z\_from & Z1pix<=z\_to))))]);

set(handles.text49,'String',[num2str(round(max(AA(Z1pix>=z\_from & Z1pix<=z\_to))))]);

set(handles.text50,'String',[num2str(round(sum(AA(Z1pix>=z\_from & Z1pix<=z\_to))./(sum(Ai(Z1pix>=z\_from & Z1pix<=z\_to)))))]);

set(handles.text51,'String',[num2str(round(sum(D(Z1pix>=z\_from & Z1pix<=z\_to))))]);

set(handles.text52,'String',[num2str(round(mean(AngV(Z1pix>=z\_from & Z1pix<=z\_to))))]);

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

function slider5\_CreateFcn(hObject, eventdata, handles)

% hObject handle to slider5 (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 slider6\_Callback(hObject, eventdata, handles)

% hObject handle to slider6 (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

global XYZ

global SI1

global TI1

global X

global Y

global Z

global x\_max

global y\_max

global z\_max

global V

global A

global Ang

global ROI1b

global ROI1t

global ROI2b

global ROI2t

global ROI3b

global ROI3t

global X1pix

global Y1pix

global Z1pix

ROI3b=uint32(get(handles.slider6,'Value'));

rectangle1 = int32([0 ROI1t size(SI1,2) ROI1b]);

rectangle2 = int32([0 ROI2t size(SI1,2) ROI2b]);

rectangle3 = int32([0 ROI3t size(SI1,2) ROI3b]);

% set(handles.slider3,'Max',size(SI1,1)-ROI1t+1);

RGBm = repmat(SI1,[1,1,3]);

temp1=get(handles.checkbox2,'Value');

if(temp1==1)

circles=[X1pix Z1pix];

circles(:,3)=0.8;

RGBm=insertShape(RGBm,'FilledCircle',circles ,'Color','red','Opacity',0.1);

end

RGBm=insertShape(RGBm,'FilledRectangle',rectangle1,'Color','yellow','Opacity',0.4);

RGBm=insertShape(RGBm,'FilledRectangle',rectangle2,'Color','green','Opacity',0.4);

RGBm=insertShape(RGBm,'FilledRectangle',rectangle3,'Color','blue','Opacity',0.4);

cla(handles.axes1);

view(handles.axes1,[0 90]);

imshow(RGBm,'Parent',handles.axes1);

title(handles.axes1,'ROI');

z\_from=double(ROI3t);

z\_to=double(ROI3t)+double(ROI3b);

AA=A;

AA(A<0)=0;

Ai=A;

Ai(A<=0)=0;

Ai(A>0)=1;

D = sqrt((X(1:end-1)-X(2:end)).^2+(Y(1:end-1)-Y(2:end)).^2+(Z(1:end-1)-Z(2:end)).^2);

D(end+1)=0;

AngV=Ang./V;

AngV(isnan(AngV))=0;

set(handles.text20,'String',[num2str(round(10000\*sum((Z1pix>=z\_from) & (Z1pix<=z\_to))/length(Z1pix))/100),'%']);

set(handles.text60,'String',[num2str(round(max(V(Z1pix>=z\_from & Z1pix<=z\_to))))]);

set(handles.text61,'String',[num2str(round(mean(V(Z1pix>=z\_from & Z1pix<=z\_to))))]);

set(handles.text62,'String',[num2str(round(max(AA(Z1pix>=z\_from & Z1pix<=z\_to))))]);

set(handles.text63,'String',[num2str(round(sum(AA(Z1pix>=z\_from & Z1pix<=z\_to))./(sum(Ai(Z1pix>=z\_from & Z1pix<=z\_to)))))]);

set(handles.text64,'String',[num2str(round(sum(D(Z1pix>=z\_from & Z1pix<=z\_to))))]);

set(handles.text65,'String',[num2str(round(mean(AngV(Z1pix>=z\_from & Z1pix<=z\_to))))]);

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

function slider6\_CreateFcn(hObject, eventdata, handles)

% hObject handle to slider6 (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 slider7\_Callback(hObject, eventdata, handles)

% hObject handle to slider7 (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

global XYZ

global SI1

global TI1

global X

global Y

global Z

global x\_max

global y\_max

global z\_max

global V

global A

global Ang

global ROI1b

global ROI1t

global ROI2b

global ROI2t

global ROI3b

global ROI3t

global X1pix

global Y1pix

global Z1pix

ROI3t=uint32(get(handles.slider7,'Value'));

rectangle1 = int32([0 ROI1t size(SI1,2) ROI1b]);

rectangle2 = int32([0 ROI2t size(SI1,2) ROI2b]);

rectangle3 = int32([0 ROI3t size(SI1,2) ROI3b]);

%set(handles.slider2,'Min',ROI1t+1);

RGBm = repmat(SI1,[1,1,3]);

temp1=get(handles.checkbox2,'Value');

if(temp1==1)

circles=[X1pix Z1pix];

circles(:,3)=0.8;

RGBm=insertShape(RGBm,'FilledCircle',circles ,'Color','red','Opacity',0.1);

end

RGBm=insertShape(RGBm,'FilledRectangle',rectangle1,'Color','yellow','Opacity',0.4);

RGBm=insertShape(RGBm,'FilledRectangle',rectangle2,'Color','green','Opacity',0.4);

RGBm=insertShape(RGBm,'FilledRectangle',rectangle3,'Color','blue','Opacity',0.4);

cla(handles.axes1);

view(handles.axes1,[0 90]);

imshow(RGBm,'Parent',handles.axes1);

title(handles.axes1,'ROI');

z\_from=double(ROI3t);

z\_to=double(ROI3t)+double(ROI3b);

%plot(X,size(SI1,1)-Z,'\*b')

AA=A;

AA(A<0)=0;

Ai=A;

Ai(A<=0)=0;

Ai(A>0)=1;

D = sqrt((X(1:end-1)-X(2:end)).^2+(Y(1:end-1)-Y(2:end)).^2+(Z(1:end-1)-Z(2:end)).^2);

D(end+1)=0;

AngV=Ang./V;

AngV(isnan(AngV))=0;

set(handles.text20,'String',[num2str(round(10000\*sum((Z1pix>=z\_from) & (Z1pix<=z\_to))/length(Z1pix))/100),'%']);

set(handles.text60,'String',[num2str(round(max(V(Z1pix>=z\_from & Z1pix<=z\_to))))]);

set(handles.text61,'String',[num2str(round(mean(V(Z1pix>=z\_from & Z1pix<=z\_to))))]);

set(handles.text62,'String',[num2str(round(max(AA(Z1pix>=z\_from & Z1pix<=z\_to))))]);

set(handles.text63,'String',[num2str(round(sum(AA(Z1pix>=z\_from & Z1pix<=z\_to))./(sum(Ai(Z1pix>=z\_from & Z1pix<=z\_to)))))]);

set(handles.text64,'String',[num2str(round(sum(D(Z1pix>=z\_from & Z1pix<=z\_to))))]);

set(handles.text65,'String',[num2str(round(mean(AngV(Z1pix>=z\_from & Z1pix<=z\_to))))]);

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

function slider7\_CreateFcn(hObject, eventdata, handles)

% hObject handle to slider7 (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 pushbutton7.

function pushbutton7\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton7 (see GCBO)

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

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

global X

global Y

global Z

global V

global x\_max

global y\_max

global z\_max

% cla reset

set(handles.pushbutton7,'Enable','off');

pause(0.01);

v1=get(handles.edit3,'String');

v2=get(handles.edit4,'String');

v3=get(handles.edit5,'String');

l1=length(X);

l2=get(handles.edit6,'String');

v1=str2num(v1);

v2=str2num(v2);

v3=str2num(v3);

% l1=ceil(str2num(l1)/2);

l2=ceil(str2num(l2)/2);

% VV=V;

% CC(VV<=v1)='g';

% CC(VV>v1 & VV<=v2)='y';

% CC(VV>v2 & VV<=v3)='m';

% CC(VV>v3)='r';

%

k1=find(V<=v1);

k2=find(V>v1 & V<=v2);

k3=find(V>v2 & V<=v3);

k4=find(V>v3);

% cla reset

cla(handles.axes1);

xlim(handles.axes1,[0 x\_max]);

ylim(handles.axes1,[0 y\_max]);

zlim(handles.axes1,[0 z\_max]);

hold(handles.axes1,'on');

plot3(X,Y,Z,'-','color',[0.90 0.90 0.90]);

% i=length(X)-1;

% while(i>0)

% plot3(X(i),Y(i),Z(i),'color',CC(i),'Parent',handles.axes1);

% hold(handles.axes1,'on');

% end

plot3(X(k1),Y(k1),Z(k1),'.g');

plot3(X(k2),Y(k2),Z(k2),'.y');

plot3(X(k3),Y(k3),Z(k3),'.m');

plot3(X(k4),Y(k4),Z(k4),'.r');

% for i=1:l2:length(X)-l2

%

% VV=round(mean(V(i:i+l2)));

%

% if(VV<=v1)

% c='g';

% elseif(VV>v1 && VV<=v2)

% c='y';

% elseif(VV>v2 && VV<=v3)

% c='m';

% else

% c='r';

% end

% %cla(handles.axes1);

%

% xlim(handles.axes1,[0 x\_max]);

% ylim(handles.axes1,[0 y\_max]);

% zlim(handles.axes1,[0 z\_max]);

% hold(handles.axes1,'on');

%

% plot3(X(i:i+l2),Y(i:i+l2),Z(i:i+l2),'.','color',c,'Parent',handles.axes1);

% %plot3(X,Y,Z,'\*');

% %plot3(X(min(temp3+l1,length(X))),Y(min(temp3+l1,length(Y))),Z(min(temp3+l1,length(Z))),'Ob','Parent',handles.axes1);

%

% end

title(handles.axes1,['All dots']);

view(handles.axes1,3);

axis(handles.axes1,'on');

set(handles.pushbutton7,'Enable','on');

% --- Executes on button press in pushbutton8.

function pushbutton8\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton8 (see GCBO)

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

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

Summary='';

Fn=get(handles.edit1,'String');

Summary=[Summary, 'Duration(sec): ',get(handles.text24,'String'),'\n'];

Summary=[Summary, '\nAverage Speed (pix): ',get(handles.text4,'String'),'\n'];

Summary=[Summary, '\nAverage Acceleration (pix^2/sec): ',get(handles.text5,'String'),'\n'];

Summary=[Summary, '\nAverage Deacceleration (pix^2/sec): ',get(handles.text26,'String'),'\n'];

Summary=[Summary, '\nActive (%%): ',get(handles.text6,'String'),'%\n'];

Summary=[Summary, '\nTotal Distance (pix): ',get(handles.text8,'String'),'\n'];

Summary=[Summary, '\nAverage Turning (%%): ',get(handles.text22,'String'),'%\n'];

Summary=[Summary, '\nOcc. Volume (%%): ',get(handles.text67,'String'),'%\n'];

Fn=[Fn(1:end-3),'txt'];

fid=fopen(Fn,'w');

fprintf(fid,Summary);

fclose(fid);

% --- Executes on button press in pushbutton9.

function pushbutton9\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton9 (see GCBO)

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

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

global XYZ

global SI1

global TI1

global X

global Y

global Z

global x\_max

global y\_max

global z\_max

global V

global A

global Ang

global AngNo

Xaxis=get(handles.popupmenu1,'Value');

Yaxis=get(handles.popupmenu2,'Value');

[TotTim,TotDis,AvgSpeed,AvgAcc,AvgDeacc,Active,NoFrame,X,Y,Z,x\_max,y\_max,z\_max,V,A,Ang,AngNo] = AvgCal(XYZ,SI1,TI1);

if(Xaxis==1)

X=V;

xlabel(handles.axes1,'Speed');

elseif(Xaxis==2)

X=A;

xlabel(handles.axes1,'Acceleration');

elseif(Xaxis==3)

X=Ang;

xlabel(handles.axes1,'Angle');

else

X=[];

end

if(Yaxis==1)

Y=V;

ylabel(handles.axes1,'Speed');

elseif(Yaxis==2)

Y=A;

ylabel(handles.axes1,'Acceleration');

elseif(Yaxis==3)

Y=Ang;

ylabel(handles.axes1,'Angle');

else

Y=[];

end

cla(handles.axes1);

xlim(handles.axes1,[min(X) max(X)]);

ylim(handles.axes1,[min(Y) max(Y)]);

%zlim(handles.axes1,[0 0]);

hold(handles.axes1,'on');

plot(X,Y,'.b');%,'color',[0.90 0.90 0.90]);

line([min(X) max(X)],[mean(Y) mean(Y)],'color','r');

line([mean(X) mean(X)],[min(Y) max(Y)],'color','r');

title('');

view(2);

% --- Executes on selection change in popupmenu1.

function popupmenu1\_Callback(hObject, eventdata, handles)

% hObject handle to popupmenu1 (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 popupmenu1 contents as cell array

% contents{get(hObject,'Value')} returns selected item from popupmenu1

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

function popupmenu1\_CreateFcn(hObject, eventdata, handles)

% hObject handle to popupmenu1 (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 popupmenu2.

function popupmenu2\_Callback(hObject, eventdata, handles)

% hObject handle to popupmenu2 (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 popupmenu2 contents as cell array

% contents{get(hObject,'Value')} returns selected item from popupmenu2

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

function popupmenu2\_CreateFcn(hObject, eventdata, handles)

% hObject handle to popupmenu2 (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 popupmenu3.

function popupmenu3\_Callback(hObject, eventdata, handles)

% hObject handle to popupmenu3 (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 popupmenu3 contents as cell array

% contents{get(hObject,'Value')} returns selected item from popupmenu3

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

function popupmenu3\_CreateFcn(hObject, eventdata, handles)

% hObject handle to popupmenu3 (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 popupmenu4.

function popupmenu4\_Callback(hObject, eventdata, handles)

% hObject handle to popupmenu4 (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 popupmenu4 contents as cell array

% contents{get(hObject,'Value')} returns selected item from popupmenu4

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

function popupmenu4\_CreateFcn(hObject, eventdata, handles)

% hObject handle to popupmenu4 (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 pushbutton10.

function pushbutton10\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton10 (see GCBO)

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

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

global XYZ

global SI1

global TI1

global X

global Y

global Z

global x\_max

global y\_max

global z\_max

global V

global A

global Ang

global AngNo

global Tank

Xaxis=get(handles.popupmenu3,'Value');

Yaxis=get(handles.popupmenu4,'Value');

[TotTim,TotDis,AvgSpeed,AvgAcc,AvgDeacc,Active,NoFrame,X,Y,Z,x\_max,y\_max,z\_max,V,A,Ang,AngNo,X1pix,Y1pix,Z1pix] = AvgCal(XYZ,SI1,TI1,Tank);

flg1=get(handles.radiobutton3,'Value');

cla reset

if(Xaxis==1)

Xaxis1=V;

% xlabel(handles.axes1,'Speed');

elseif(Xaxis==2)

Xaxis1=A;

% xlabel(handles.axes1,'Acceleration');

elseif(Xaxis==3)

Xaxis1=Ang;

% xlabel(handles.axes1,'Angle');

else

Xaxis1=[];

end

if(Yaxis==1)

Yaxis1=V;

% ylabel(handles.axes1,'Speed');

elseif(Yaxis==2)

Yaxis1=A;

% ylabel(handles.axes1,'Acceleration');

elseif(Yaxis==3)

Yaxis1=Ang;

% ylabel(handles.axes1,'Angle');

else

Yaxis1=[];

end

if(flg1==1)

if(Xaxis==1)

xlabel(handles.axes1,'Speed');

elseif(Xaxis==2)

xlabel(handles.axes1,'Acceleration');

elseif(Xaxis==3)

xlabel(handles.axes1,'Angle');

else

xlabel(handles.axes1,'');

end

if(Yaxis==1)

ylabel(handles.axes1,'Speed');

elseif(Yaxis==2)

ylabel(handles.axes1,'Acceleration');

elseif(Yaxis==3)

ylabel(handles.axes1,'Angle');

else

ylabel(handles.axes1,'');

end

cla(handles.axes1);

xlim(handles.axes1,[min(Xaxis1) max(Xaxis1)]);

ylim(handles.axes1,[min(Yaxis1) max(Yaxis1)]);

%zlim(handles.axes1,[0 0]);

hold(handles.axes1,'on');

plot(Xaxis1,Yaxis1,'.b');%,'color',[0.90 0.90 0.90]);

line([min(Xaxis1) max(Xaxis1)],[mean(Yaxis1) mean(Yaxis1)],'color','r');

line([mean(Xaxis1) mean(Xaxis1)],[min(Yaxis1) max(Yaxis1)],'color','r');

title('');

view(2);

hold(handles.axes1,'off');

else

colormap(jet);

xlabel(handles.axes1,'');

ylabel(handles.axes1,'');

cla(handles.axes1);

hold(handles.axes1,'on');

[a1 b1]=hist(Xaxis1);

xlim([min(b1) max(b1)]);

ylim([min(a1) max(a1)]);

hist(Xaxis1);

title('');

view(2);

end

% --- Executes on button press in radiobutton1.

function radiobutton1\_Callback(hObject, eventdata, handles)

% hObject handle to radiobutton1 (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 radiobutton1

% --- Executes on button press in radiobutton2.

function radiobutton2\_Callback(hObject, eventdata, handles)

% hObject handle to radiobutton2 (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 radiobutton2

% --- If Enable == 'on', executes on mouse press in 5 pixel border.

% --- Otherwise, executes on mouse press in 5 pixel border or over radiobutton4.

function radiobutton4\_ButtonDownFcn(hObject, eventdata, handles)

% hObject handle to radiobutton4 (see GCBO)

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

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

% aa=get(hObject.Value);

% % if(aa==1)

% beep;

% % end

function edit11\_Callback(hObject, eventdata, handles)

% hObject handle to edit11 (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 edit11 as text

% str2double(get(hObject,'String')) returns contents of edit11 as a double

pushbutton13\_Callback(hObject, eventdata, handles);

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

function edit11\_CreateFcn(hObject, eventdata, handles)

% hObject handle to edit11 (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 edit12\_Callback(hObject, eventdata, handles)

% hObject handle to edit12 (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 edit12 as text

% str2double(get(hObject,'String')) returns contents of edit12 as a double

pushbutton13\_Callback(hObject, eventdata, handles);

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

function edit12\_CreateFcn(hObject, eventdata, handles)

% hObject handle to edit12 (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 pushbutton13.

function pushbutton13\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton13 (see GCBO)

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

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

global XYZ

global SI1

global TI1

global X

global Y

global Z

global x\_max

global y\_max

global z\_max

global V

global A

global Ang

global AngNo

Vset=str2double(get(handles.edit11,'String'));

Aset=str2double(get(handles.edit12,'String'));

P = (V>=Vset)&(Ang>=Aset);

Pr=round(mean(P)\*10000)/100;

set(handles.text68,'String',[num2str(Pr),'%']);

% --- Executes on slider movement.

function slider10\_Callback(hObject, eventdata, handles)

% hObject handle to slider10 (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

global V

p = get(hObject,'Value');

set(handles.text73,'String',[num2str(p),'%']);

Vx=round(max(V)\*(p/100));

set(handles.edit11,'String',num2str(Vx));

pushbutton13\_Callback(hObject, eventdata, handles);

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

function slider10\_CreateFcn(hObject, eventdata, handles)

% hObject handle to slider10 (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 slider11\_Callback(hObject, eventdata, handles)

% hObject handle to slider11 (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

global Ang

p = get(hObject,'Value');

set(handles.text74,'String',[num2str(p),'%']);

Angx=round(max(Ang)\*(p/100));

set(handles.edit12,'String',num2str(Angx));

pushbutton13\_Callback(hObject, eventdata, handles);

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

function slider11\_CreateFcn(hObject, eventdata, handles)

% hObject handle to slider11 (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 edit13\_Callback(hObject, eventdata, handles)

% hObject handle to edit13 (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 edit13 as text

% str2double(get(hObject,'String')) returns contents of edit13 as a double

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

function edit13\_CreateFcn(hObject, eventdata, handles)

% hObject handle to edit13 (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 edit14\_Callback(hObject, eventdata, handles)

% hObject handle to edit14 (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 edit14 as text

% str2double(get(hObject,'String')) returns contents of edit14 as a double

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

function edit14\_CreateFcn(hObject, eventdata, handles)

% hObject handle to edit14 (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 edit15\_Callback(hObject, eventdata, handles)

% hObject handle to edit15 (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 edit15 as text

% str2double(get(hObject,'String')) returns contents of edit15 as a double

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

function edit15\_CreateFcn(hObject, eventdata, handles)

% hObject handle to edit15 (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 during object creation, after setting all properties.

function axes2\_CreateFcn(hObject, eventdata, handles)

% hObject handle to axes2 (see GCBO)

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

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

% Hint: place code in OpeningFcn to populate axes2

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

function axes3\_CreateFcn(hObject, eventdata, handles)

% hObject handle to axes3 (see GCBO)

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

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

% Hint: place code in OpeningFcn to populate axes3

% --- Executes on button press in pushbutton15.

function pushbutton15\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton15 (see GCBO)

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

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

global SI1

global TI1

Xms=size(SI1,2)/2;

Yms=size(SI1,1)/2;

Xmt=size(TI1,2)/2;

Ymt=size(TI1,1)/2;

RecXs=200;

RecYs=100;

RecXt=100;

RecYt=100;

cla reset

cla(handles.axes1);

view(handles.axes1,[0 90]);

imshow(SI1,'Parent',handles.axes1);

hold on

rectangle('Position',[Xms-RecXs/2,Yms-RecYs/2,RecXs,RecYs],'EdgeColor','r','Parent',handles.axes1);

% --- Executes on slider movement.

function slider12\_Callback(hObject, eventdata, handles)

% hObject handle to slider12 (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

global HV

global VV

global SI1

global TI1

global X

global Y

global Z

global Tank

VV=get(hObject,'Value');

set(handles.text85,'String',num2str(HV));

set(handles.text86,'String',num2str(VV));

Xms=size(SI1,2)/2;

Yms=size(SI1,1)/2;

Xmt=size(TI1,2)/2;

Ymt=size(TI1,1)/2;

RecXs=(get(handles.slider13,'Value')/100)\*size(SI1,2);

RecYs=(get(handles.slider12,'Value')/100)\*size(SI1,1);

RecXt=(get(handles.slider13,'Value')/100)\*size(TI1,2);

RecYt=(get(handles.slider12,'Value')/100)\*size(TI1,1);

cla reset

cla(handles.axes1);

view(handles.axes1,[0 90]);

if(get(handles.radiobutton5,'Value')==1)

imshow(TI1,'parent',handles.axes1);

hold on

rectangle('Position',[Xmt-RecXt/2,Ymt-RecYt/2,RecXt,RecYt],'EdgeColor','r','linewidth',3,'Parent',handles.axes1);

XminC=Tank.Length\*(Xmt-RecXt/2)/size(TI1,2);

XmaxC=Tank.Length\*(Xmt+RecXt/2)/size(TI1,2);

YminC=Tank.Width\*(Ymt-RecYt/2)/size(TI1,1);

YmaxC=Tank.Width\*(Ymt+RecYt/2)/size(TI1,1);

PP=(X>=XminC)&(X<=XmaxC)&(Y>=YminC)&(Y<=YmaxC);

PPP=round(mean(PP)\*100\*100)/100;

set(handles.text87,'String',[num2str(PPP),'%']);

set(handles.text87,'ForegroundColor','r');

elseif(get(handles.radiobutton6,'Value')==1)

imshow(SI1,'parent',handles.axes1);

hold on

rectangle('Position',[Xms-RecXs/2,Yms-RecYs/2,RecXs,RecYs],'EdgeColor','b','linewidth',3,'Parent',handles.axes1);

XminC=Tank.Length\*(Xms-RecXs/2)/size(SI1,2);

XmaxC=Tank.Length\*(Xms+RecXs/2)/size(SI1,2);

YminC=Tank.Height\*(Yms-RecYs/2)/size(SI1,1);

YmaxC=Tank.Height\*(Yms+RecYs/2)/size(SI1,1);

PP=(X>=XminC)&(X<=XmaxC)&(Z>=YminC)&(Z<=YmaxC);

PPP=round(mean(PP)\*100\*100)/100;

set(handles.text87,'String',[num2str(PPP),'%']);

set(handles.text87,'ForegroundColor','b');

end

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

function slider12\_CreateFcn(hObject, eventdata, handles)

% hObject handle to slider12 (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 slider13\_Callback(hObject, eventdata, handles)

% hObject handle to slider13 (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

global HV

global VV

global SI1

global TI1

global X

global Y

global Z

global Tank

HV=get(hObject,'Value');

set(handles.text85,'String',[num2str(HV),'%']);

set(handles.text86,'String',[num2str(VV),'%']);

Xms=size(SI1,2)/2;

Yms=size(SI1,1)/2;

Xmt=size(TI1,2)/2;

Ymt=size(TI1,1)/2;

RecXs=(get(handles.slider13,'Value')/100)\*size(SI1,2);

RecYs=(get(handles.slider12,'Value')/100)\*size(SI1,1);

RecXt=(get(handles.slider13,'Value')/100)\*size(TI1,2);

RecYt=(get(handles.slider12,'Value')/100)\*size(TI1,1);

cla reset

cla(handles.axes1);

view(handles.axes1,[0 90]);

if(get(handles.radiobutton5,'Value')==1)

imshow(TI1,'parent',handles.axes1);

hold on

rectangle('Position',[Xmt-RecXt/2,Ymt-RecYt/2,RecXt,RecYt],'EdgeColor','r','linewidth',3,'Parent',handles.axes1);

XminC=Tank.Length\*(Xmt-RecXt/2)/size(TI1,2);

XmaxC=Tank.Length\*(Xmt+RecXt/2)/size(TI1,2);

YminC=Tank.Width\*(Ymt-RecYt/2)/size(TI1,1);

YmaxC=Tank.Width\*(Ymt+RecYt/2)/size(TI1,1);

PP=(X>=XminC)&(X<=XmaxC)&(Y>=YminC)&(Y<=YmaxC);

PPP=round(mean(PP)\*100\*100)/100;

set(handles.text87,'String',[num2str(PPP),'%']);

set(handles.text87,'ForegroundColor','r');

elseif(get(handles.radiobutton6,'Value')==1)

imshow(SI1,'parent',handles.axes1);

hold on

rectangle('Position',[Xms-RecXs/2,Yms-RecYs/2,RecXs,RecYs],'EdgeColor','b','linewidth',3,'Parent',handles.axes1);

XminC=Tank.Length\*(Xms-RecXs/2)/size(SI1,2);

XmaxC=Tank.Length\*(Xms+RecXs/2)/size(SI1,2);

YminC=Tank.Height\*(Yms-RecYs/2)/size(SI1,1);

YmaxC=Tank.Height\*(Yms+RecYs/2)/size(SI1,1);

PP=(X>=XminC)&(X<=XmaxC)&(Z>=YminC)&(Z<=YmaxC);

PPP=round(mean(PP)\*100\*100)/100;

set(handles.text87,'String',[num2str(PPP),'%']);

set(handles.text87,'ForegroundColor','b');

end

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

function slider13\_CreateFcn(hObject, eventdata, handles)

% hObject handle to slider13 (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 radiobutton5.

function radiobutton5\_Callback(hObject, eventdata, handles)

% hObject handle to radiobutton5 (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 radiobutton5

% --- Executes on button press in radiobutton6.

function radiobutton6\_Callback(hObject, eventdata, handles)

% hObject handle to radiobutton6 (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 radiobutton6

% --- Executes when selected object is changed in uipanel12.

function uipanel12\_SelectionChangeFcn(hObject, eventdata, handles)

% hObject handle to the selected object in uipanel12

% eventdata structure with the following fields (see UIBUTTONGROUP)

% EventName: string 'SelectionChanged' (read only)

% OldValue: handle of the previously selected object or empty if none was selected

% NewValue: handle of the currently selected object

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

global SI1

global TI1

global X

global Y

global Z

global Tank

Xms=size(SI1,2)/2;

Yms=size(SI1,1)/2;

Xmt=size(TI1,2)/2;

Ymt=size(TI1,1)/2;

RecXs=(get(handles.slider13,'Value')/100)\*size(SI1,2);

RecYs=(get(handles.slider12,'Value')/100)\*size(SI1,1);

RecXt=(get(handles.slider13,'Value')/100)\*size(TI1,2);

RecYt=(get(handles.slider12,'Value')/100)\*size(TI1,1);

cla reset

cla(handles.axes1);

view(handles.axes1,[0 90]);

if(get(handles.radiobutton5,'Value')==1)

imshow(TI1,'parent',handles.axes1);

hold on

rectangle('Position',[Xmt-RecXt/2,Ymt-RecYt/2,RecXt,RecYt],'EdgeColor','r','linewidth',3,'Parent',handles.axes1);

XminC=Tank.Length\*(Xmt-RecXt/2)/size(TI1,2);

XmaxC=Tank.Length\*(Xmt+RecXt/2)/size(TI1,2);

YminC=Tank.Width\*(Ymt-RecYt/2)/size(TI1,1);

YmaxC=Tank.Width\*(Ymt+RecYt/2)/size(TI1,1);

PP=(X>=XminC)&(X<=XmaxC)&(Y>=YminC)&(Y<=YmaxC);

PPP=round(mean(PP)\*100\*100)/100;

set(handles.text87,'String',[num2str(PPP),'%']);

set(handles.text87,'ForegroundColor','r');

elseif(get(handles.radiobutton6,'Value')==1)

imshow(SI1,'parent',handles.axes1);

hold on

rectangle('Position',[Xms-RecXs/2,Yms-RecYs/2,RecXs,RecYs],'EdgeColor','b','linewidth',3,'Parent',handles.axes1);

XminC=Tank.Length\*(Xms-RecXs/2)/size(SI1,2);

XmaxC=Tank.Length\*(Xms+RecXs/2)/size(SI1,2);

YminC=Tank.Height\*(Yms-RecYs/2)/size(SI1,1);

YmaxC=Tank.Height\*(Yms+RecYs/2)/size(SI1,1);

PP=(X>=XminC)&(X<=XmaxC)&(Z>=YminC)&(Z<=YmaxC);

PPP=round(mean(PP)\*100\*100)/100;

set(handles.text87,'String',[num2str(PPP),'%']);

set(handles.text87,'ForegroundColor','b');

end

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

function axes1\_CreateFcn(hObject, eventdata, handles)

% hObject handle to axes1 (see GCBO)

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

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

% Hint: place code in OpeningFcn to populate axes1

aaaa=imread('screen.jpg');

imshow(aaaa,'parent',hObject);