**Matlab科学计算课程论文**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **考 察 项 目** | | | | |
| **界面的美观性** | **功能的完整性** | | **论文的规范性** | **代码的规范性** |
| **评语：** | | | | |
| **总 分** | |  | | |
| **阅 卷 人** | |  | | |
| **日 期** | |  | | |

**数字图像处理GUI程序设计**

**一、GUI的布局**

本程序中的GUI界面布局主要采取的是Box Layout的布局方式。界面的上侧依次是三个组件“轴”，分别用来显示原始图片，边缘检测后的图片以及图像模糊、恢复时的图片。界面下侧主要包括两个面板，分别是“图像边缘检测算法”面板，用于选择所使用的算子。以及“图像模糊与恢复”面板，包括对原始图形进行模糊的按钮以及恢复图像算法的选择菜单。界面的右下侧还有两个按钮，分别用来执行“退出”和“读取图片”的操作。具体的GUI布局界面如图1所示。



图1 GUI界面

**二、功能模块实现**

本程序中的功能模块主要分为三部分。第一部分是读取图片，并将原始图片显示在axes1组件中，实现代码如下：

1、读取图片的实现代码

function pushbutton1\_Callback(hObject, eventdata, handles)

%设置全局变量，可以在另外的函数中使用它们

global im;

%选择图片路径

[filename,pathname]=...

uigetfile({'\*.jpg';'\*.bmp';'\*.tif';'\*.png'},'图片选择');

%合成路径+文件名

str=[pathname,filename];

%没有选择任何文件，报错对话框

L=length(filename);

if L<5

errordlg('You do not choose any image','Notice');

return;

end;

test=filename(1,L-3:L);

if test=='.jpg'|test=='.bmp'|test=='.tif'|test=='.png'

%读取图片

im=imread(str);

%等待工具条

h=waitbar(0,'Please waiting, reading...');

%关键，指定axes1为当前显示区域

axes(handles.axes1);

%显示图片

imshow(im);

waitbar(1,h,'finish');

pause(2);

delete(h);

else

warndlg('format wrong','warn');

return;

end;

第二部分是图像边缘检测的处理。对读入进来的图片用Robert算子、Sobel算子、Prewitt算子、拉普拉斯算子、Canny算子等进行边缘检测。相应的模块界面设计如图2 ，代码如下：

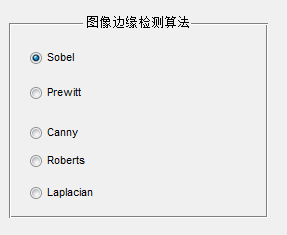


图2 边缘检测算法

2、边缘检测算法的实现代码

function uipanel1\_SelectionChangeFcn(hObject, eventdata, handles)

% hObject handle to uipanel1 (see GCBO)

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

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

global im

global BW

%拿到所选择按钮的名称

str=get(hObject,'string');

h=waitbar(0,'Please waiting, reading...');

%从此开始在axes2中作图

axes(handles.axes2);

if length(size(im))==3 %RGB图像，true color

switch str

case 'Sobel'

BW=edge(rgb2gray(im),'sobel');

imshow(BW);

case 'Prewitt'

BW=edge(rgb2gray(im),'prewitt');

imshow(BW);

case 'Canny'

BW=edge(rgb2gray(im),'canny');

imshow(BW);

case 'Roberts'

BW=edge(rgb2gray(im),'roberts');

imshow(BW);

case 'Laplacian'

BW=edge(rgb2gray(im),'log');

imshow(BW);

end;

else %gray image灰度图像

switch str

case 'Sobel'

BW=edge(im,'sobel');

imshow(BW);

case 'Prewitt'

BW=edge(im,'prewitt');

imshow(BW);

case 'Canny'

BW=edge(im,'canny');

imshow(BW);

case 'Roberts'

BW=edge(im,'roberts');

imshow(BW);

case 'Laplacian'

BW=edge(im,'log');

imshow(BW);

end;

end

waitbar(1,h,'finish');

pause(2);

delete(h);

第三部分是图像的模糊和恢复。对读取进来的图片首先进行模糊化，然后再用Wiener滤波器、regularized滤波器、Lucy-Richardson算法、盲去卷积算法进行恢复。相应的模块界面设计如图3 ，代码如下：

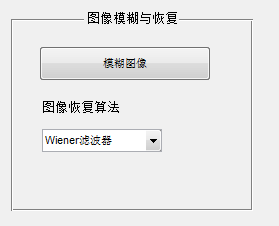


图3 图像的模糊和恢复

3、图像模糊的实现代码

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 im

global H

global B

LEN=35;

THETA=13;

H=fspecial('motion',LEN,THETA);

B=imfilter(im,H,'circular','conv');

axes(handles.axes3);

imshow(B)

4、图像恢复的实现代码

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)

%图像恢复

global H

global B

global im

h=waitbar(0,'Please waiting, reading...');

axes(handles.axes3);

val = get(hObject ,'value');

switch(val)

case 1

C=deconvwnr(B,H); % 用Wiener滤波器进行恢复

imshow(C);

case 2

INITPSF=ones(size(H));

[D,P]=deconvblind(B,INITPSF,30); %用盲去卷积算法进行恢复

imshow(D);

case 3

V=0.02;

NP=V\*prod(size(im));

[E, LAGRA]=deconvreg(B,H,NP); %用regularized滤波器进行恢复图像，指定PSF和噪声幂次NP

imshow(E);

case 4

F=deconvlucy(B,H,5);%用Lucy-Richardson算法进行恢复

end

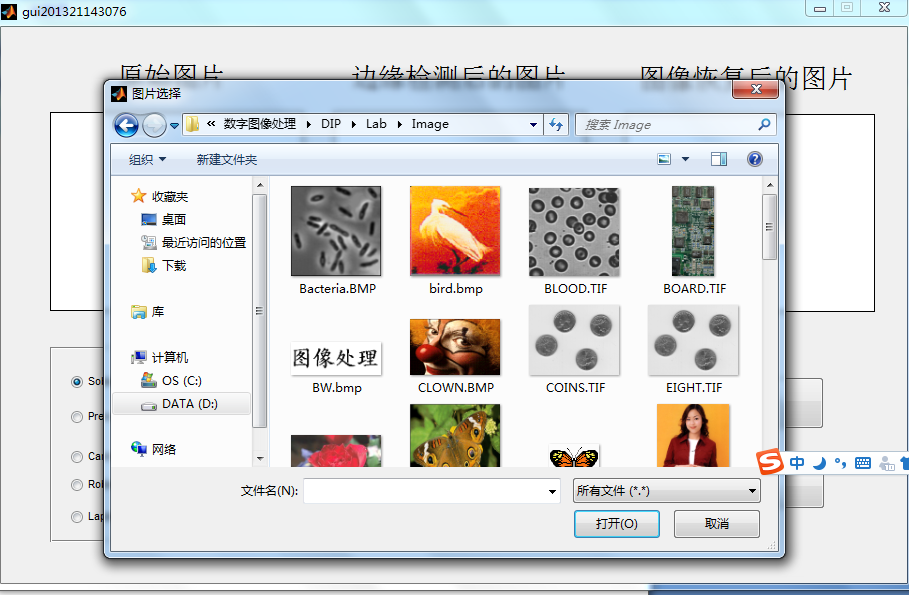
waitbar(1,h,'finish');

pause(2);

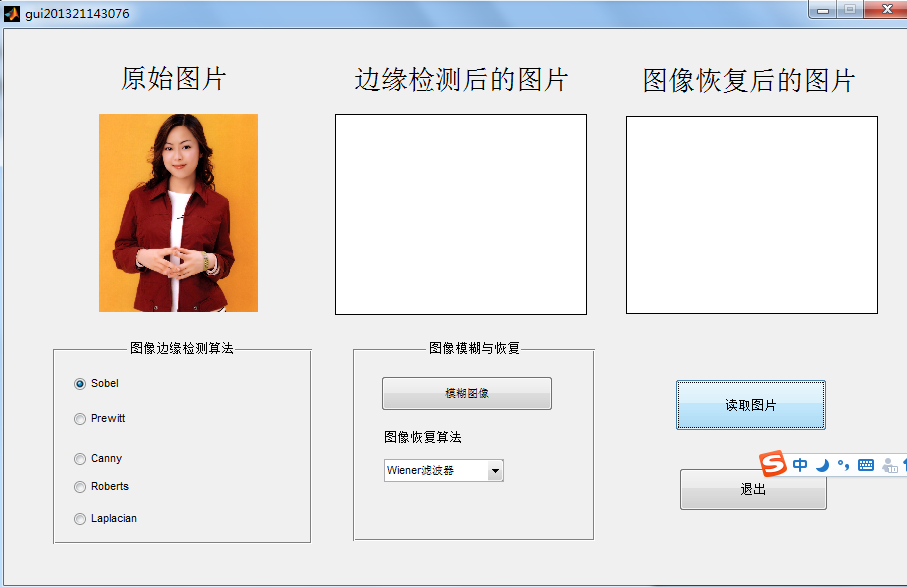
delete(h);

**三、结果演示和说明**

1、读取图片



2、显示原始图片



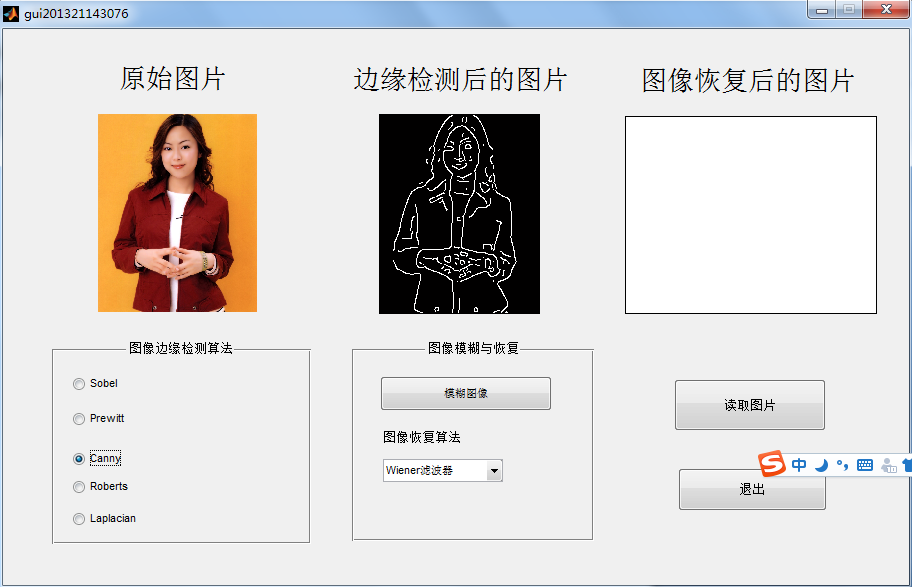
3、使用Prewiit算子进行边缘检测



4、使用Roberts算子进行边缘检测



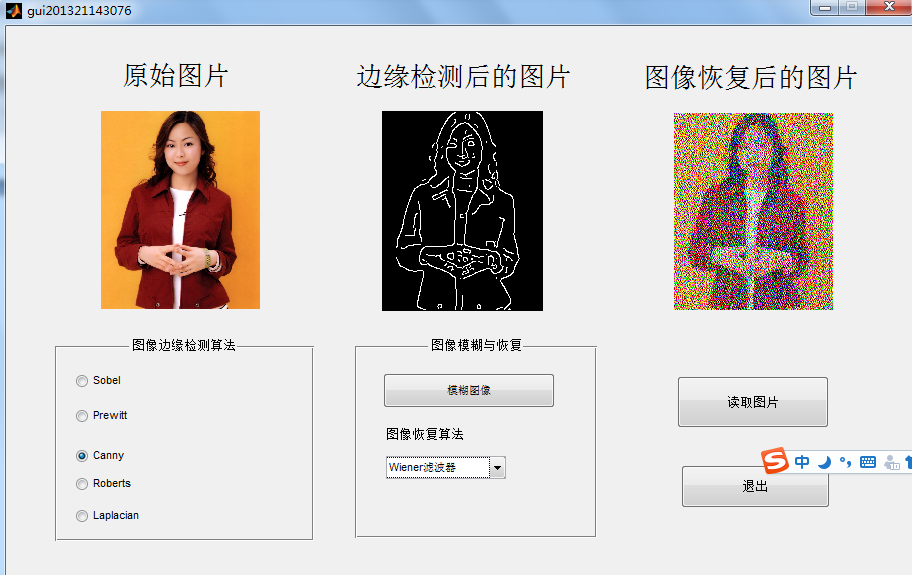
5、使用Canny算子进行边缘检测



6、点击“模糊图像”按钮，对图像进行模糊处理



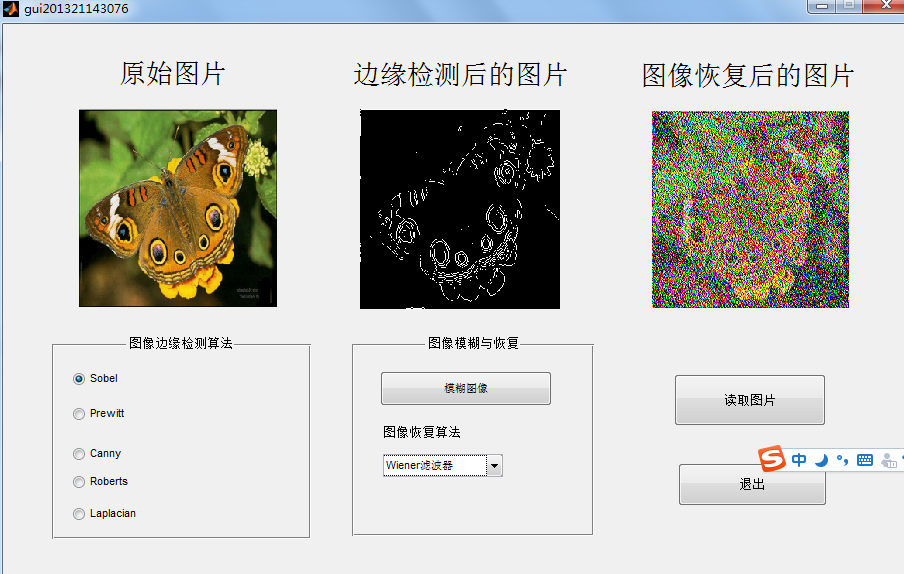
7、使用Wiener滤波器进行图像恢复

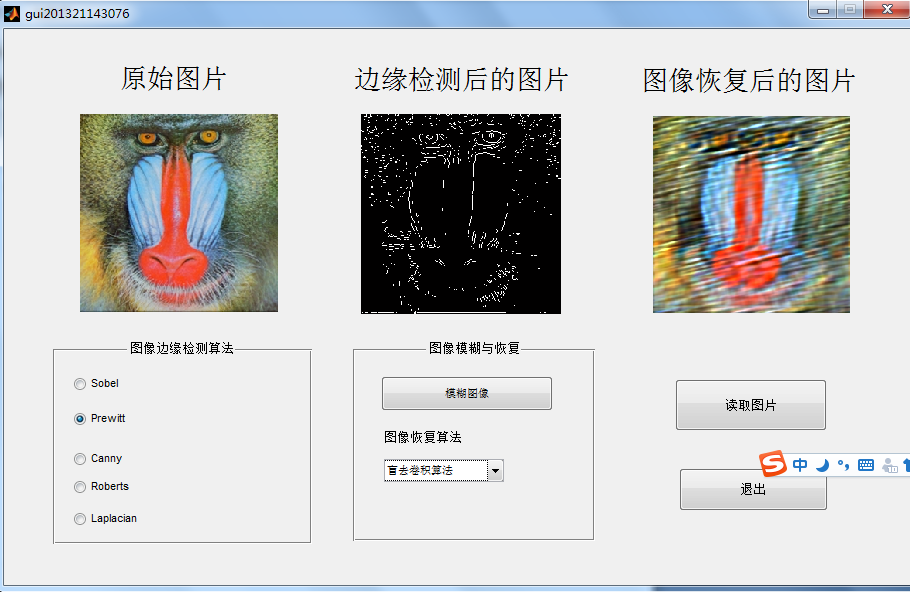


8、使用盲去卷积算法进行图像恢复



9、使用其他图片的综合测试





**附录（可执行的完整代码）**

function varargout = gui201321143076(varargin)

% GUI201321143076 M-file for gui201321143076.fig

% GUI201321143076, by itself, creates a new GUI201321143076 or raises the existing

% singleton\*.

%

% H = GUI201321143076 returns the handle to a new GUI201321143076 or the handle to

% the existing singleton\*.

%

% GUI201321143076('CALLBACK',hObject,eventData,handles,...) calls the local

% function named CALLBACK in GUI201321143076.M with the given input arguments.

%

% GUI201321143076('Property','Value',...) creates a new GUI201321143076 or raises the

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

% applied to the GUI before color\_image\_edge\_OpeningFunction gets called. An

% unrecognized property name or invalid value makes property application

% stop. All inputs are passed to gui201321143076\_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 gui201321143076

% Last Modified by GUIDE v2.5 28-Jun-2016 17:02:02

% Begin initialization code - DO NOT EDIT

gui\_Singleton = 1;

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

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

'gui\_OpeningFcn', @gui201321143076\_OpeningFcn, ...

'gui\_OutputFcn', @gui201321143076\_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 gui201321143076 is made visible.

function gui201321143076\_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 gui201321143076 (see VARARGIN)

% Choose default command line output for gui201321143076

handles.output = hObject;

% Update handles structure

guidata(hObject, handles);

% UIWAIT makes gui201321143076 wait for user response (see UIRESUME)

% uiwait(handles.figure1);

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

function varargout = gui201321143076\_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 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)

%设置全局变量，可以在另外的函数中使用它们

global im;

%选择图片路径

[filename,pathname]=...

uigetfile({'\*.jpg';'\*.bmp';'\*.tif';'\*.png'},'图片选择');

% [filename, pathname] = uigetfile( ...

% {'\*.m;\*.fig;\*.mat;\*.mdl', 'All MATLAB Files (\*.m, \*.fig, \*.mat, \*.mdl)';

% '\*.m', 'M-files (\*.m)'; ...

% '\*.fig','Figures (\*.fig)'; ...

% '\*.mat','MAT-files (\*.mat)'; ...

% '\*.mdl','Models (\*.mdl)'; ...

% '\*.\*', 'All Files (\*.\*)'}, ...

% 'Pick a file');

%合成路径+文件名

str=[pathname,filename];

%没有选择任何文件，报错对话框

L=length(filename);

if L<5

errordlg('You do not choose any image','Notice');

return;

end;

test=filename(1,L-3:L);

if test=='.jpg'|test=='.bmp'|test=='.tif'|test=='.png'

%读取图片

im=imread(str);

%等待工具条

h=waitbar(0,'Please waiting, reading...');

%关键，指定axes1为当前显示区域

axes(handles.axes1);

%显示图片

imshow(im);

waitbar(1,h,'finish');

pause(2);

delete(h);

else

warndlg('format wrong','warn');

return;

end;

% --- 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)

close(gcf);

% --------------------------------------------------------------------

function uipanel1\_SelectionChangeFcn(hObject, eventdata, handles)

% hObject handle to uipanel1 (see GCBO)

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

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

global im

global BW

%拿到所选择按钮的名称

str=get(hObject,'string');

h=waitbar(0,'Please waiting, reading...');

%从此开始在axes2中作图

axes(handles.axes2);

if length(size(im))==3 %RGB图像，true color

switch str

case 'Sobel'

BW=edge(rgb2gray(im),'sobel');

imshow(BW);

case 'Prewitt'

BW=edge(rgb2gray(im),'prewitt');

imshow(BW);

case 'Canny'

BW=edge(rgb2gray(im),'canny');

imshow(BW);

case 'Roberts'

BW=edge(rgb2gray(im),'roberts');

imshow(BW);

case 'Laplacian'

BW=edge(rgb2gray(im),'log');

imshow(BW);

end;

else %gray image灰度图像

switch str

case 'Sobel'

BW=edge(im,'sobel');

imshow(BW);

case 'Prewitt'

BW=edge(im,'prewitt');

imshow(BW);

case 'Canny'

BW=edge(im,'canny');

imshow(BW);

case 'Roberts'

BW=edge(im,'roberts');

imshow(BW);

case 'Laplacian'

BW=edge(im,'log');

imshow(BW);

end;

end

waitbar(1,h,'finish');

pause(2);

delete(h);

% --- 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 im

global H

global B

LEN=35;

THETA=13;

H=fspecial('motion',LEN,THETA);

B=imfilter(im,H,'circular','conv');

axes(handles.axes3);

imshow(B)

% --- 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)

%图像恢复

global H

global B

global im

h=waitbar(0,'Please waiting, reading...');

axes(handles.axes3);

val = get(hObject ,'value');

switch(val)

case 1

C=deconvwnr(B,H); % 用Wiener滤波器进行恢复

imshow(C);

case 2

INITPSF=ones(size(H));

[D,P]=deconvblind(B,INITPSF,30); %用盲去卷积算法进行恢复

imshow(D);

case 3

V=0.02;

NP=V\*prod(size(im));

[E, LAGRA]=deconvreg(B,H,NP); %用regularized滤波器进行恢复图像，指定PSF和噪声幂次NP

imshow(E);

case 4

F=deconvlucy(B,H,5);%用Lucy-Richardson算法进行恢复

end

waitbar(1,h,'finish');

pause(2);

delete(h);

% 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