

Blind to Change

More on Graphical User Interfaces



CS112 Scientific Computation

Department of Computer Science
Wellesley College

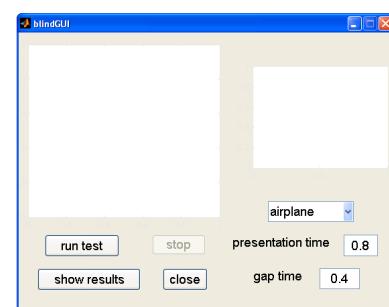
Exploring “change blindness”

The human visual system can be “blind” to changes in a scene,
unless we draw our attention to the area undergoing change

This GUI program displays a pair of alternating images on the left, until the user notices the change and presses the stop button

The user can select which pair to view, amount of time each image is displayed, and time between image frames

Results on the total time needed to detect the change for each pair is displayed on the right



Functions defined in blindGUI.m

blindGUI: top-level function at the start of the file, called from the Command Window. This function initializes the GUI program and opens the GUI window. *This function is not modified*

blindGUI_OpeningFcn: executed just before GUI window is made visible. Code is added to this function to set up an initial pair of images for testing, and to create a vector to store the experimental results

blindGUI_OutputFcn: returns outputs to the Command Window. *This function is not modified*

3

Changes to blindGUI_OpeningFcn

```
% --- Executes just before blindGUI is made visible.  
function blindGUI_OpeningFcn (hObject, eventdata, handles, varargin)
```

```
% variables to store image pair, initialized to airplane images  
handles.im1 = imread('plane1.bmp');  
handles.im2 = imread('plane2.bmp');  
% results of tests: number of repeats needed to detect change,  
% for each pair of images (currently four image pairs)  
handles.results = zeros(1,4);
```

```
% Choose default command line output for blindGUI  
handles.output = hObject;  
% Update handles structure  
guidata(hObject, handles);
```

4

imagesMenu_Callback

```
function imagesMenu_Callback(hObject, eventdata, handles)
% loads selected image pair
imageChoice = get(handles.imagesMenu, 'Value');
if (imageChoice == 1)
    handles.im1 = imread('plane1.bmp');
    handles.im2 = imread('plane2.bmp');
elseif (imageChoice == 2)
    handles.im1 = imread('dinner1.bmp');
    handles.im2 = imread('dinner2.bmp');
elseif (imageChoice == 3)
    handles.im1 = imread('farm1.bmp');
    handles.im2 = imread('farm2.bmp');
else
    handles.im1 = imread('market1.bmp');
    handles.im2 = imread('market2.bmp');
end
% update handles structure
guidata(hObject, handles);
```



Don't forget to update
global handles structure!

5

demoButton_Callback (part 1)

```
function demoButton_Callback(hObject, eventdata, handles)
% create blank frame of the same size as the image pair
[rows cols rgb] = size(handles.im1);
blank = 0.8*ones(rows, cols, rgb);
% get presentation and gap times from the GUI
presentTime = str2double(get(handles.presentTextbox, 'String'));
gapTime = str2double(get(handles.gapTextbox, 'String'));
% switch to testAxes display area and enable the stop button
axes(handles.testAxes);
set(handles.stopButton, 'Enable', 'on');
% initialize number of repetitions needed to detect change
nreps = 0;
```

...

6

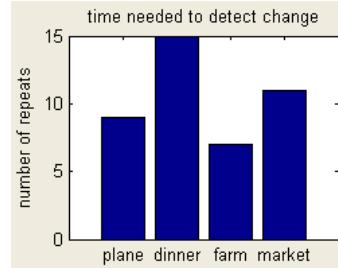
demoButton_Callback (part 2)

```
% alternate two images until the user clicks the stop button
while (get(handles.stopButton, 'Value') == 0)
    imshow(handles.im1);
    pause(presentTime);
    imshow(blank);
    pause(gapTime);
    imshow(handles.im2);
    pause(presentTime);
    imshow(blank);
    pause(gapTime);
    nreps = nreps + 1;
end
% record results (number of repeats needed to detect change)
handles.results(get(handles.imagesMenu, 'Value')) = nreps;
% reset and disable the stop button
set(handles.stopButton, 'Value', 0);
set(handles.stopButton, 'Enable', 'off');
% update handles structure
guidata(hObject, handles);
```

7

resultsButton_Callback

```
function resultsButton_Callback(hObject, eventdata, handles)
% display results of tests as a bar graph of the number of
% repeats needed to detect change for each image pair
axes(handles.resultsAxes);
bar(handles.results);
axis([0 5 0 max(handles.results)])
set(gca, 'XTickLabel', {'plane' 'dinner' 'farm' 'market'});
ylabel('number of repeats');
title('time needed to detect change');
```



8

Bells and whistles

Other things you may want to explore in a GUI for your final project:

- adding animation
- other GUI components
 - Slider
 - Listbox
 - Radio Buttons
- dialog boxes
- pull-down menus
- mouse control
- multi-line text boxes

