

14:440:127– Introduction to Computers for Engineers

PROJ03 - 7%

Rutgers University, Fall 2010

Instructor - Brenda V. Cortez

1 Deliverables:

For this project you are required to create a GUI and submit:

- the non-crashable **M-file** for you GUI named *rec00_netid_PROJ03.m* and the automatically generated **.fig** file

2 Grading Rubric:

2.1 Full 7% grade if:

- Your GUI meets all requirements below under "Project Description" - **4 points**
- Your file is named correctly (*rec00_netid_PROJ03.m*) - **0.5 points**
- You check for bad input name files (i.e. if image file queried is not in the current directory, display "error" message) - **1.5 points**
- You reset the axes every time a new image is queried - **0.5 points**
- You made comment/s where it was required - **0.5 points**

2.2 Between 1% & 7% of the grade:

- Grades will be assigned as listed above.

2.3 0% of the grade if:

- Your code crashes. It's your responsibility to make sure your file is **DEBUGGED**.

3 Project Description:

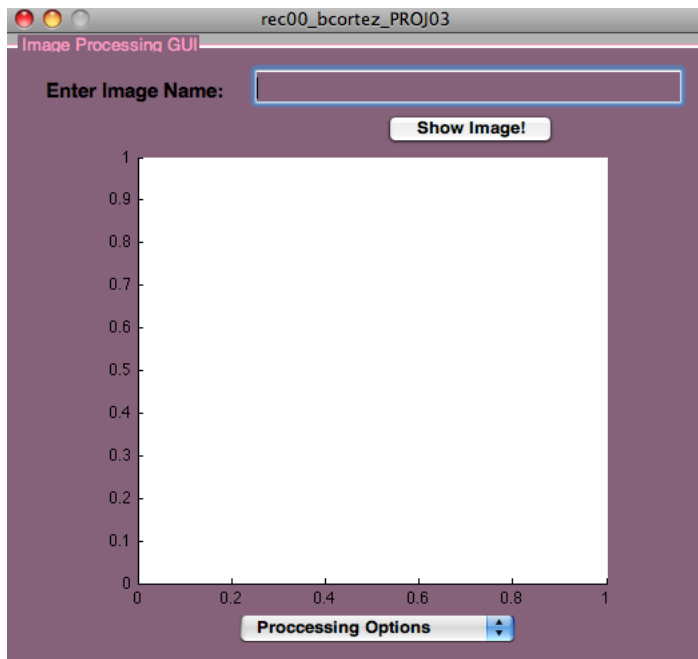
For this project you are required to create a GUI that:

- allows the *user* to type in the name of an image (found in the current directory), in an “*editable text component*”
- displays image within the GUI, on an “*axes component*” by clicking on a “*push button component*”
- allows the user to apply any of FIVE image processing algorithms to the image, which can be chosen from a “*pop-up menu component*”
- and renders the processed image immediately, *on the axes component*

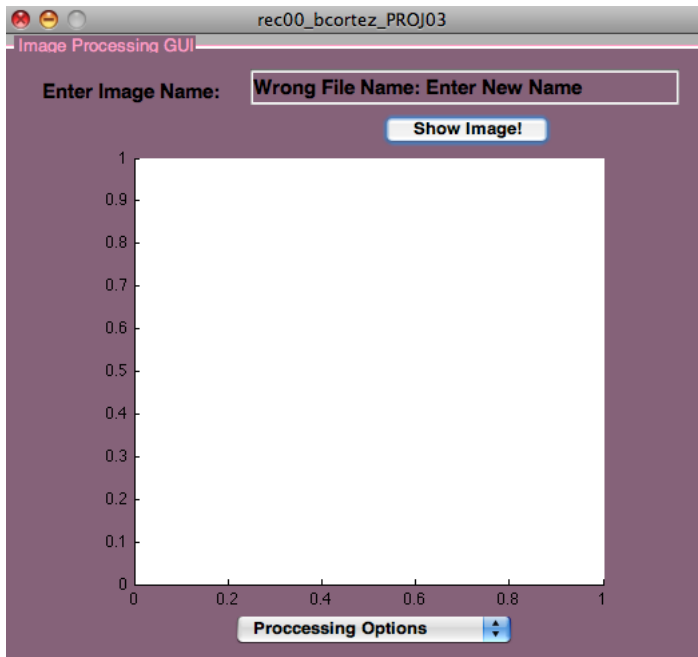
The first thing you need to do is download the images your GUI will have to process, to the directory where you will save your GUI file/s. When your project gets graded, only the “*building.jpg*”, “*sears.jpg*”, and “*empire.jpg*” will be used to test your GUI.

4 Sample Test

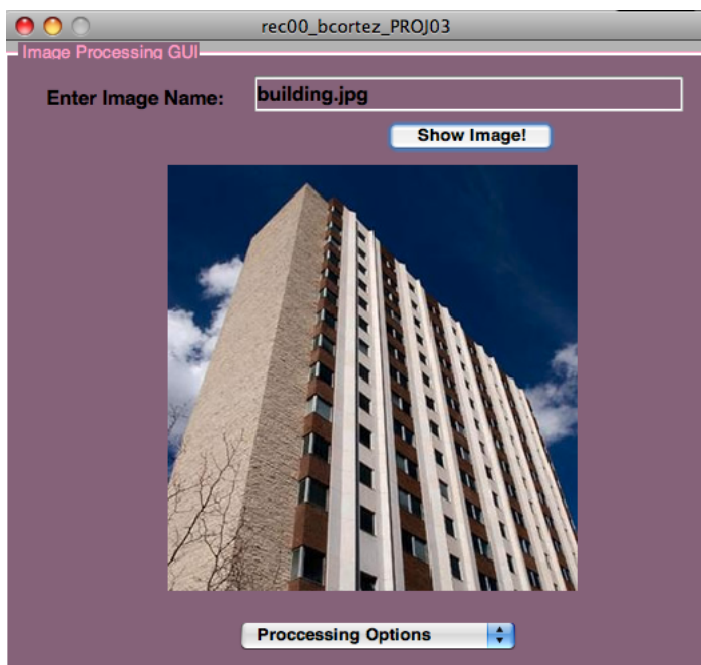
Initially, you should have an GUI that looks as follows (you can personalize it by using your own color squeme and/or layout, but functionality should be the same):



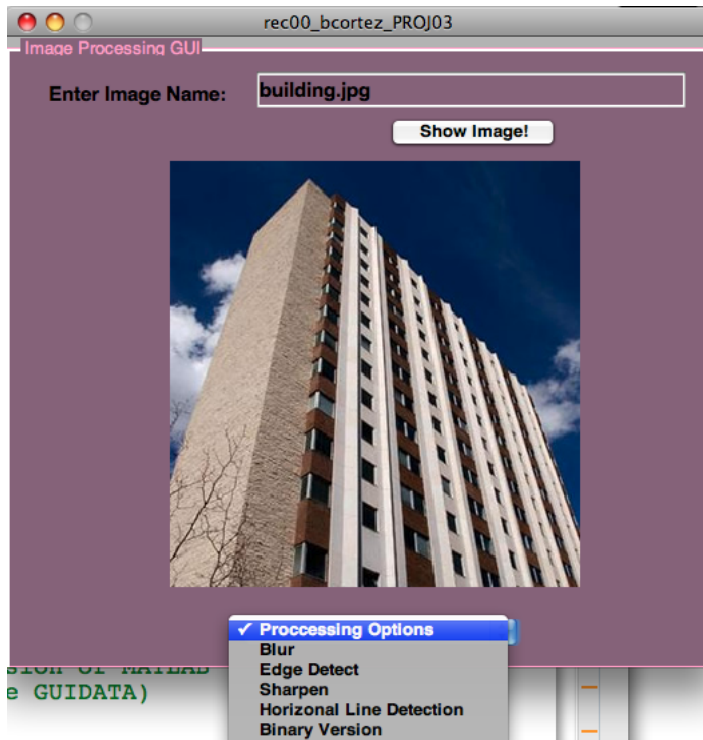
If the user types in the name of a file that is NOT found in the current directory, and the “*push button component*” is pressed, you should populate the “*editable text component*” with a message saying: “**Wrong File Name: Enter New Name**”. As you can see in the following image, the “*axes component*” and the “*pop-up menu component*” should remain unchanged.



The *user* can then simply re-type the a correct name of an image file (i.e. *“building.jpg”*, *“sears.jpg”*, and *“empire.jpg”*) and by pressing the *“push button component”* should see it rendered on the axes as follows:



Now that a valid image has been displayed, the *user* should be able to select from the pop-up menu component, which should have as its first option **“Processing Options”** and the other options should be: Blur, Edge Detect, Sharpen, Horizontal Line Detection & Binary Version.



- If the *user* selects “**Blur**” as the processing option, you should use the **fspecial** command to create a ”rotationally symmetric Gaussian lowpass filter of” size 7x7 with standard deviation sigma=5, and apply it to the image.
- If the *user* selects “**Edge Detect**” as the processing option, you should convert the image to gray before being able to used the intensity command **edge** to apply the ”Sobel horizontal edge-emphasizing filter” method to the image.
- If the *user* selects “**Sharpen**” as the processing option, you should use the **fspecial** command to create an ”unsharp contrast enhancement filter”, and apply it to the image.
- If the *user* selects “**Horizontal Line Detection**” as the processing option, you should apply to the image a horizontal Kernel represented by the array $[-1,-1,-1; 2,2,2; -1,-1,-1]$.
- If the *user* selects “**Binary Version**” as the processing option, you should convert the image to binary using **im2bw** using threshold=0.5
- If, **after** processing the image the *user* selects the first option in the menu (i.e. Processing Options), you should display the original image.

1. REQUIRED COMMENT: Explain how each process is affecting the image. For each process, create a block comment and answer the question in full sentences to get full credit. You should type in the *help* command along with the different commands and read the documentation. If necessary, look up the mathematical definitions using wikipedia.

Following is the code that will render the processed image to the GUI; insert it after your code for processing the image.

```
hAxes = gca;  
imshow(processedImage, 'Parent', hAxes);
```

2. REQUIRED COMMENT: How does the above code work and what is it doing? Create a block comment and answer these questions in full sentences to get full credit.

5 Hints

5.1 Checking if the image file name is correct

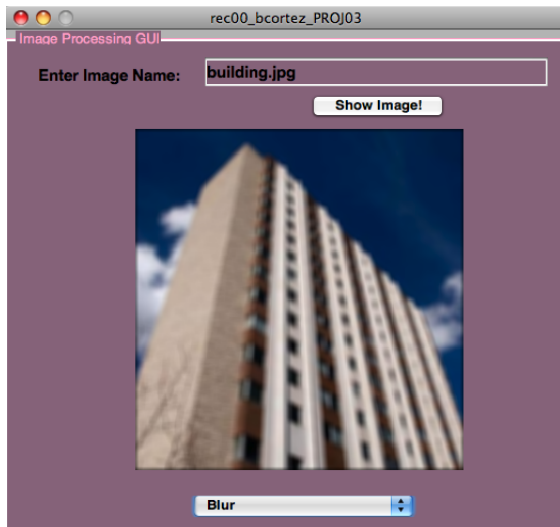
First of all, it is important that you understand that in order for your functions to operate on the SAME image, you must define a global variable for the image and perhaps another variable (to flag when the image file is found, i.e. 0 if not found, 1 if found), so that you can access it throughout. However, MatLab requires that you define it everywhere you are going to use it for safety precautions, so don't forget to do this within the function where you will be using the image.

Second, in order to check if the image file is found in the current directory (i.e. where your GUI's **M-file** and **.fig** file are saved), you need to know the current directory, which in MatLab can be determine by using the **pwd** command. Once you know the directory path, all you need is the image file name, which you can get using the **get** function; just remember what *tag* you assigned to the "*editable text field*". Then to locate the image file, you will need to **concatenate** the path name, a backslash and the file name, and use the **exist** function to determine if in fact the file is in the current directory (see quiz answer to remember what *concatenate* means). If the file is not found, you should *set* the "*editable text field*" to display the error message given previously and set your flag variable to "**0**", since this will mean that the image file was not found. But if the image file is found, then you should read it in to your global variable and set your flag variable to "**1**" now that it has been found.

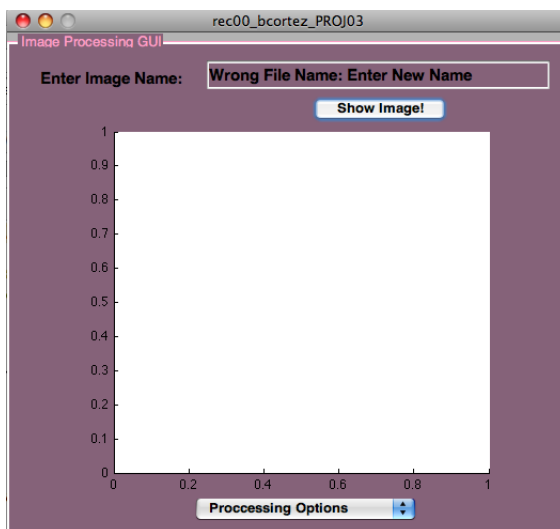
5.2 Displaying Image

The code is similar to that given in above in the "Sample Test" section, but use your global flag variable to either render the image that was read in OR to simply reset the axes. You will need to read documentation for command **cla**, so that you can learn how to reset the axes, if an invalid image name is entered after another image which has already been processed. Following is an example:

Here is the image "building.jpg" rendered to the GUI axes after it was processed using the *Horizontal Line Detection* process.



And here is what the user should see if a new file name is entered, and this file is not found in the current directory.



The “*pop-up menu*” was restored to the first option and the “*axes component*” was reset.

5.3 Useful Documentation

If you open up the help documentation and type in “*Modifying Properties*”, you will find pretty good information about the properties that you can modify and some explanation about them as well.

You will also find that the Lecture Presentations that covered **Images** and **Graphics Handles** will be helpful, and well as the **tutorials** for HW04.

NO collaborators for PROJs.
FOLLOW INSTRUCTIONS!!
DUE DATE - TUESDAY, NOVEMBER 23, 2010 @ 4:30PM SHARP!!