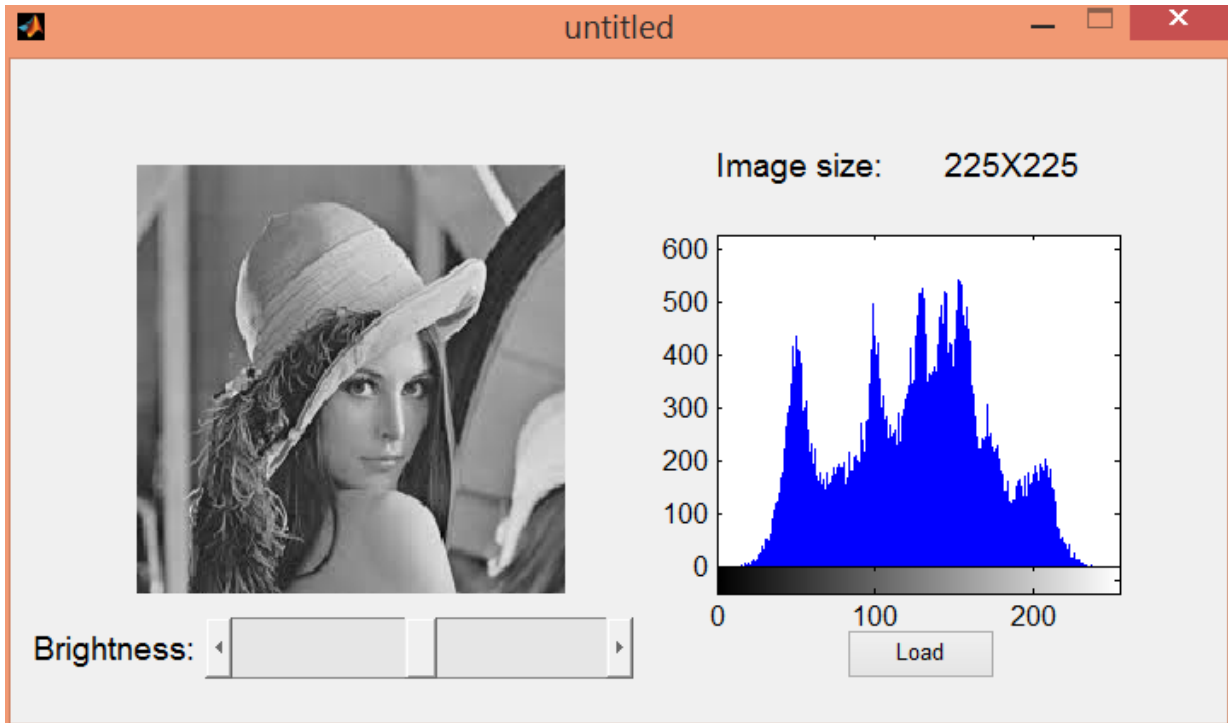


lab08_HW

1. A grayscale or gray-level image is an image in which the value associated with each pixel represents its intensity (or brightness) in the image. Design a GUI program such that:
 - (1) The user is able to load a gray-level image by using a Push Button
 - (2) The program then automatically displays the size of the image
 - (3) The program then automatically plots the histogram of the image
 - (4) The user can adjust the brightness of the image by using a Slider

Use "02Lena.png" from Lab 2 to test your program. The GUI program should look like this:



I slightly change the interface.

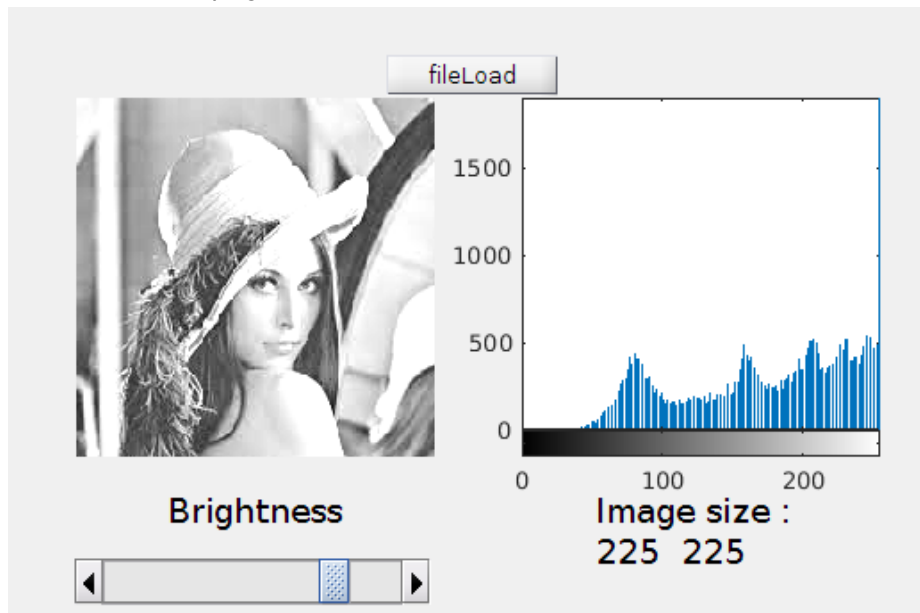
If you want pop window of selcecting file, write like this :

```
[filename, pathname] = uigetfile({'*.jpg;*.tif;*.png;*.gif', ...  
    'All Image Files'; ...  
    '*.*','All Files' }, 'Select Image');
```

see lab08_1.m

see lab08_1.fig

see lab08_1.png



2. Design a GUI program to blend 2 images. The program needs to contain 3 Axes . Two of the Axes show two original images. The other Axes shows the blended image. The program need to contain a slider with which the user can adjust the weights of the two original images to the blended image. Use "rice.png" and "cameraman.tif" to test your program. The GUI program should look like this:



Be care for that two picture size may be different
so resize it when It load

If you want to get demo image.

```
>> fullfile(matlabroot, 'toolbox/images/imdata')
```

The default path is

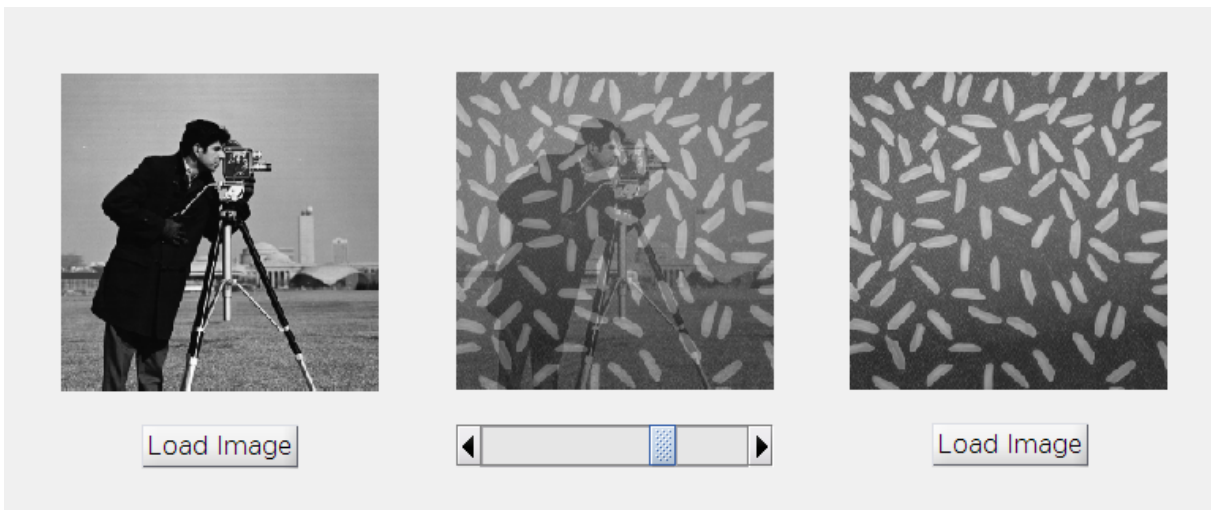
```
/usr/local/MATLAB/R2017a/toolbox/images/imdata
```

We can use `imlincomb` to blended image

see lab08_2.m

see lab08_2.fig

see lab08_2.png



3. Design a GUI program to rotate an image. The program needs to allow the user to load an image, to display the loaded image, to provide a rotation angle, and to display the rotated image. The user should be able to store the rotated image too. Do NOT use the built-in

```
function imrotate .
```

if you are lazy, you can use still using builtin function (without imrotate)

```
>> th = 10;
>> tform = affine2d([cosd(th) -sind(th) 0; ...
                    sind(th) cosd(th) 0; ...
                    0 0 1]);
>> imshow(imwarp(imread('02Lena.png'), tform)) ;
```

However I write my own function by myself

see `imgRot.m`

see `imgRot.m#testRot`

First, write transtrom function

```
t = [cosd(th) sind(th); -sind(th) cosd(th)];
```

Seoncd, I rotate from the center of picture, use `meshgrid`

```
[y, x] = meshgrid(-c/2:c/2, -r/2:r/2);
newxy = round(reshape(newsize,2,1) ./ 2 + 0.5 + ...
              t' * [reshape(x, 1, numel(x)); reshape(y, 1, numel(y))]);
```

Third, create new blank size

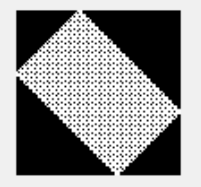
and need to care the boundary, and Matlab is start from 1

be care of size and newsize

```
blank = zeros(round(newsize + 0.5), 'uint8');
newsize = [r c] * abs(t);
```

Test it,

```
blank(sub2ind(size(blank), newxy(1,:), newxy(2,:))) = 255;
```



Notice that it is not one-by-one (float precision error), so there are some black point on it.

To solve this, need the reverse this transform methods to make N-to-1, to 1-to-N

see `imgRot.m#testRotinv`

Third, reversed the transform methods and clip out of oversize xy,

```
filter_cond = newxy(1,:)>=1 & newxy(2,:)>=1 & newxy(1,:)<=r & newxy(2,:)
newxy = newxy(:, filter_cond);
blank(reshape(filter_cond, size(blank))) = 255;
```

Test it



Now it is great!

The last, assign image to new image

```
blank(reshape(filter_cond, size(blank))) = base(sub2ind(size(base),
newxy(1,:), newxy(2,:)));
```

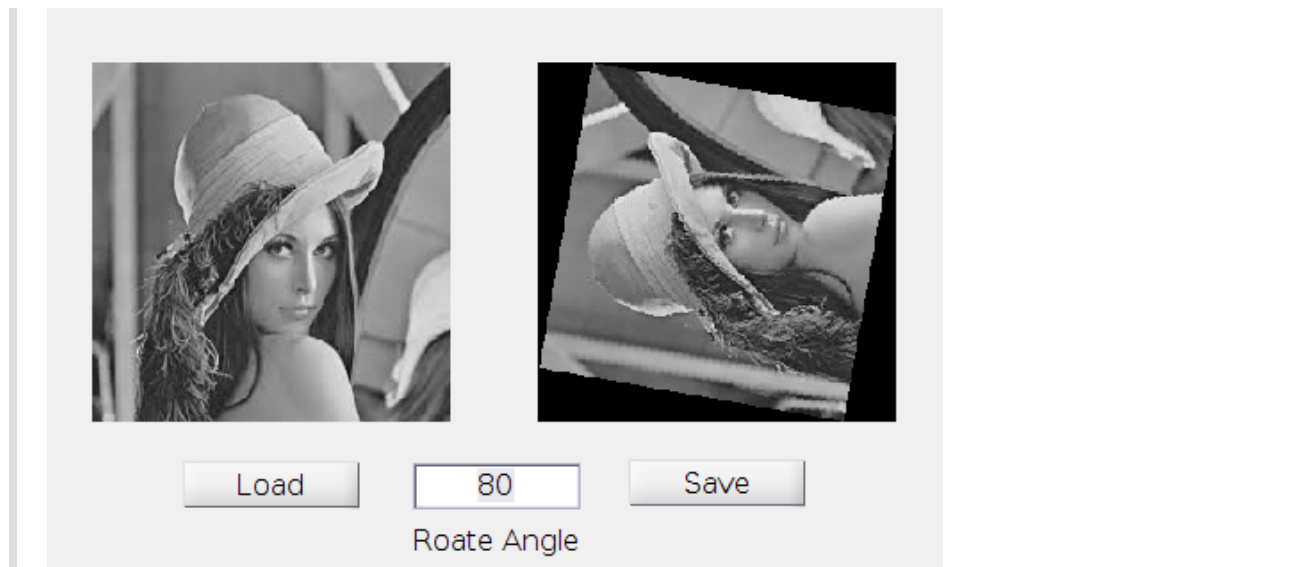


Then write GUI and use this function

see lab08_3.m

see lab08_3.fig

see lab08_3.png



Test on ubuntu16.04 + Matlab R0217a academic use