

Image processing in Python: assignment

This assignment is the first part of the assessed assignments that you need to complete. In this assignment we will be writing some software to do manual image registration using the keyboard interface. Everything you need should be in the repository.

The code we write today will form the basis of the code we will write tomorrow, where we automate the registration process, so make sure it is good and well commented!

Part 1

- I. Write some useful information like a title and your names in some comments at the top of the code
- II. Import the modules you will need, these include `matplotlib.pyplot`, `numpy` and `imread` from `skimage.io`
- III. Load the file “lungs.jpg” and display it
- IV. Plot a histogram of the image
- V. Write a function that displays the CT image with different window levels. The input variables should be the window and the level.

```
def yourfunction(window, level):  
    #your code here
```

- VI. Identify which parts of the histogram relate to which parts of the image. Record this information in some comments in your code.

Part 2

- VII. Start a new script
- VIII. Write some useful information like a title and your names in some comments at the top of the code.
- IX. Import the modules that you will need. You'll need `matplotlib.pyplot` and `numpy`, but also `imread` from `skimage.io`, and `interpolation` and `rotate` from `scipy.ndimage`
- X. Load the images “lungs.jpg” and “lungs2.jpg”
- XI. Display the images
- XII. Make a plot where you can see both images on the same axes using transparency.

We will keep one of the images fixed (lungs.jpg) and make one of them able to move (lungs2.jpg). So that you can reference this moving image, specify a name for the axis when you plot it, e.g.

```
floating = ax.imshow(...)
```

Later you can use

```
floating.set_data(...)
```

To update the image in these axes to something else (like the image after we have manipulated it)

- XIII. Write a function that shifts your second image given an input argument called `shifts` which is a list of shifts horizontally and vertically. When your function is called, the command `shiftImage([10,20])` should shift the image down 10 pixels and to the right 20 pixels. Here is a skeleton implementation:

```
def shiftImage(shifts):  
    global image2  
    # your code here...  
  
    floating.set_data(image2)  
    fig.canvas.draw()
```

By defining `image2` as global here, changes we make to it inside the function have an effect outside the function too

If you called the axes showing the second image “floating”, then the figure will be updates with the second to last line.

The last line will update the figure if it is already displayed. This will make more sense later when our figure becomes interactive.

- XIV. Evaluate your function by calling it. What does `shiftImage([10,20])` do?
- XV. What are the shifts needed to align the images? Make a note of them in some comments.
- XVI. Modify your code to include rotations. Load “lungs3.jpg” and use that as your floating image instead. What are the shifts/rotations required to align the images now?
- XVII. Lets make it so we can shift the floating image using keyboard presses.

To do that, we will need to connect our matplotlib figure to a `key_press_event`.

We will do it with the code below:

```
fig.canvas.mpl_connect('key_press_event', eventHandler)
```

This connects the figure “fig” to key presses. When a key is pressed the specified function (in this case `eventHandler`) is called. The first argument to that function is the event. We haven’t defined this function yet though. You will need to write it; below is an example to get you started:

```
def eventHandler(event):  
    """  
    This function handles deciphering what the user wants us  
    to do, the event knows which key has been pressed.  
    """  
    up = 0  
    whichKey = event.key  
    if whichKey == "up":  
        up = 1  
    print(up)
```

After you have written the function, you can connect the event handler to the plot like so:

```
fig.canvas.mpl_connect('key_press_event', eventHandler)
plt.show()
```

What happens when you press the up key? What happens when you press any other key.

Try printing whichKey to show what each key is called.

Your eventHandler should call the function shiftImage that you wrote before and send it the required arguments.

- XVIII. Finish off the implementation of the keyboard interface, including rotations. Use it to register lungs2 to lungs1, and then lungs3 to lungs1. Do the shifts and rotations match what you found by trial and error?