

CS 101  
Fall 2014  
Program Assignment # 6  
Algorithm Due : Sunday, Nov. 6th, 2014  
Program Due : Sunday, Nov. 13th 2014

## Tiling Multiple Images

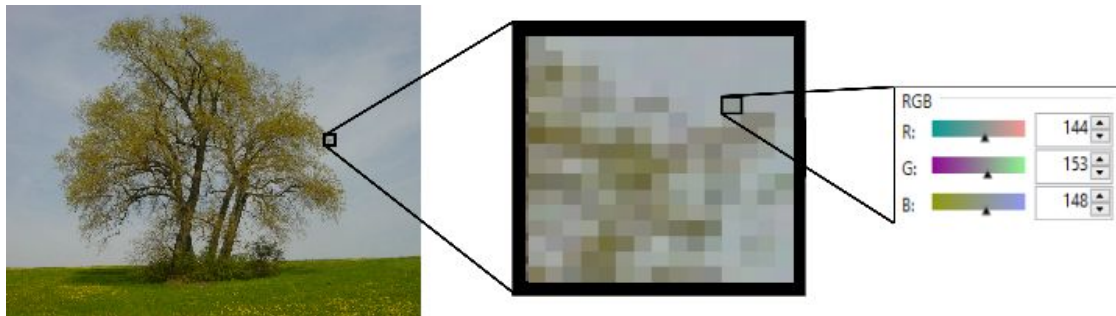
Multiple images can be taken from the same vantage point and tiled as a single image to show the progression over time. Below you can see multiple images that have taken and horizontal tiles or strips have been used from each one to make one image. We will be using vertical strips to show progression of time from left to right. In this example each image is of a different season. Spring, summer, fall and winter.



As you can tell, we take a slice from each image and form a new image. It gives the illusion of a time lapse in a single image.

Every image is a grid of colors. Each pixel contains a red, green and blue value. These values

form the colors you see on your screen.



You'll be given directories that have 2 file formats. You are required to implement a solution that reads the text based PPM files. Since they are text files you can open with them with the python open function just like any text file. The directories will also have jpg files. Using the jpg files will be extra credit. The program you write will ask the user for a directory, if the files in that directory are valid image files then you will sort them by their names and create a new image with slices of the other images. The number of images in each directory will not be constant. If there are 10 source images, then your output image will take 1/10 of a horizontal slice from the input images.

The first thing we need to do is understand how the text based ppm files are organized.

## PPM Files

Portable PixMap format or PPM is a simple image file format. There are a few variants of the PPM format, but we will be concentrating on one particular specification of the text file versions. Since it's just a normal text file, we can open, read and write the image files just like they were any normal text document. This makes it very convenient since we can use all the Python file handling techniques we've learned this semester. We have included ***ppmReader.html*** to view the ppm files that have been supplied. You may also want to use an image editor like gimp to view the images ( <http://www.gimp.org>). You will also want to examine the files in a text editor.

PPM Format ( IMG\_4833.ppm )

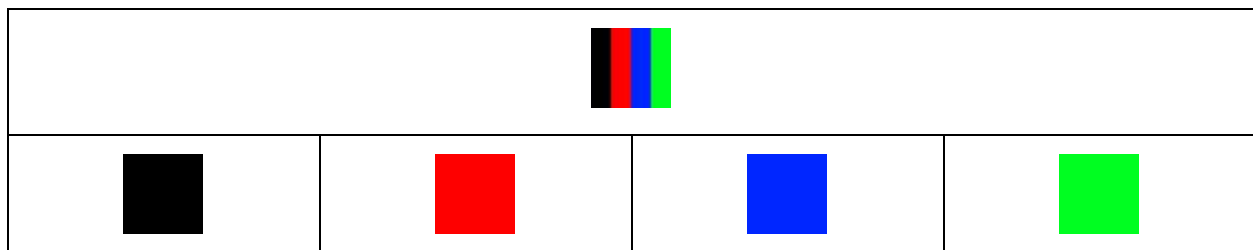
Line #	Example Data	Description
1	P3	PPM Header information. P followed by a number. We'll be using the P3 format.
2	800 600	Size of the image in pixels. This 800 wide and 600 pixels tall.
3	255	Color Depth of the Image. Each Red, Green, Blue component will be 0-255 in value. This is always 255 in our examples.

4	169	Red Component of the first pixel.
5	157	Green component of the first pixel
6	135	Blue Component of the first pixel
7	170	Red component of the second pixel
...	...	...
144001	87	Red component of last pixel
144002	121	Green component of last pixel
144003	71	Blue Component for the last pixel.

## Requirements

- You must prompt the user for a valid source directory. If the directory does not exist, then display an error and ask them to enter the source directory again.
- Ask the user for the name of the output file. If the user enters an empty string, or the file cannot be opened for write access, then warn them and ask for the input again. You may want to add the appropriate extension yourself, so the user doesn't name the ppm file saved as .doc.
- The source directory may contain other files types. You'll notice our Sample directory contains jpg and ppm extensions. You should only operate on the type of file you are looking for.
- You should validate that all the files are the same image size, and the header to each ppm file is valid. To validate the header, the first line must be P3, and the bitdepth must be 255..
- Create a valid image output with of the slices from the original image

In order to help debug your program you've been given a directory with small images. This will make it easier to debug and work with these files. There is a directory called extra\_small\_trial and small\_trial. You can see the result would have a slice of the black image, a slice of the red image and a slice of the green image.



## Extra Credit ( 5 Points )

- Ask the user for which file type they would like to use to create a new image. PPM or JPG. This obviously requires you to implement both solutions.

## Tips and Suggestions

- You want to practice simple operations with the ppm files
  - Write a program to increase the color of a ppm file.
  - Write a program to invert the colors in a ppm file. Each color should be 255-color.
  -
- Play around with storing the images in different structures.
  - You may try using a list of lists, where each sublist is a row. You can then store a list as the red, green and blue value. So accessing the green color at y position 50 and x position 100 might be accessed with `img[50][100][1]`.

```
img = [  
    [ [ 10, 20, 40], [ 5, 20, 30], ... ]  
    [ [ 128, 90, 240], [240, 150, 90], ... ]  
    ...  
    [[ 5, 100, 20], [100, 122, 240], ... ]  
]
```
  - You could also keep each pixel in a dictionary. There are quite a few ways to do this. The key could be a tuple of the y, x, and color, and the value would be the color value at that position. `img[(50, 100, 1)]`
  - You can also use a list of tuples for the red, green blue. Then if you were given a row, and column you'd want to find the index that it would be a one-dimensional list. `index=row*total_columns + col`. If you had a list of 100 items, and it was representing a 10x10 grid. If you were wanting the value at row 3, and column 8, the corresponding index is `3*10+8`.
  - There are many more options for how to represent the information, so play around with ideas before you implement.
- If you load all the files in a directory at once, you won't want to use a different variable for each since each subdirectory of images contain a different number of images. You probably would want to store each open file handle in a list. You alternatively, may not want to open all the files at once. If the image was 100x100 and there were 4 images, then you might want to open the first input file and get all the columns from 0-24 for all rows.
- Practice portions of the program. You can't do it all at once. Try a program that can list all the files in a directory of a certain type. Jpg, ppm etc.
- You'll want to sort the names of the files so that you process each image in the order it

was taken in.

## JPG Files using Pillow

Python has an easy to use image module called Pillow. Pillow is pre-installed on the lab computers. You might find that using Pillow to manipulate the jpg files is indeed easier than working with PPM. However, you will have to do some work in reading and understanding the Pillow documentation. You should examine and play with different properties and methods in Pillow to understand and find the parts you need to make this assignment work.

## Example Output Shown with Extra Credit

```
>>> ===== RESTART =====
>>>
    Image Slicer

Welcome to the image slicer. You can provide a directory that has several images
of the same size, and this program will take equal vertical slices out of each
and construct a new image.

1. Slice a PPM file.
2. Slice a Jpg Image
3. Quit

==> e
You must enter one of the following 1,2,3
    Image Slicer

Welcome to the image slicer. You can provide a directory that has several images
of the same size, and this program will take equal vertical slices out of each
and construct a new image.

1. Slice a PPM file.
2. Slice a Jpg Image
3. Quit

==> 1

Enter a directory that contains valid image files ==> invalid_dir
invalid_dir is not a valid directory. Please enter a valid directory

Enter a directory that contains valid image files ==> bad_images
image3.ppm does not have correct number of pixels, 39x40 != 1599 pixels
image2.ppm does not have 255 as its bit depth
image4.ppm does not have correct number of pixels, 40x39 != 1599 pixels
image1.ppm does not have PPM as its first line
image2.ppm does not have the same width as image3.ppm
image4.ppm does not have the same width as image3.ppm
image4.ppm does not have the same height as image3.ppm
image1.ppm does not have the same width as image3.ppm
```

```
Enter a directory that contains valid image files ==> small_trial

Enter the file to save the ppm to. ==> output*.ppm
Could not open the file output*.ppm for write access. Please try another

Enter the file to save the ppm to. ==> out_trial.ppm

Image out_trial.ppm has been saved..
```

### Image Slicer

Welcome to the image slicer. You can provide a directory that has several images of the same size, and this program will take equal vertical slices out of each and construct a new image.

1. Slice a PPM file.
2. Slice a Jpg Image
3. Quit

```
==> 3
>>>
```

## Useful Modules and functions

- os.path.isdir ( path )** - Given a path as a string returns True if the path is a valid directory.
- os.listdir( path )** - Returns a list of strings of filenames in the path given. If no Path is given the current working directory is used.
- os.path.splitext ( file )** - Given a filename it splits the extension off. You get a tuple of 2 strings. The first part being the portion before the extension, and the second is the file extension.
- os.path.join( value, [value]... )** - Takes any number of strings and returns a concatenated string separated by the default file system separator. ( / for Unix and mac, \ for Windows )

## Definitions

**Pixel** - any one of the very small dots that together form the picture on a television screen, computer monitor, etc. <sup>1</sup>

## Resources

1. "Pixel." *Merriam-Webster*. Merriam-Webster, n.d. Web. 15 July 2014.  
<<http://www.merriam-webster.com/dictionary/pixel>>.
2. Pillow documentation <https://pillow.readthedocs.org/en/latest/>

