Coloring Grayscale Images with Color Images

1. Motivation and impact

Converting grayscale images to proper color images is an interesting topic in computational photography. There are many practical applications for it, such as restoring old black and white photographs and coloring scientific images (e.g., coloring microscopic images of cells). Thus, this project focuses on how to colorize grayscale photos using color photos, which is based on Welsh's work.

2. Approach

In a grayscale image, the image grayscale level can reflect the differences between the segments on the image. The classical color transfer technique is implemented by finding the best match for the target pixel based on the similarity of the pixel luminance and its neighborhood statistical features. That is, the color of the source color image is colored to the target grayscale source image based on the luminance. Welsh's algorithm focuses on color transfer by finding the best luminance matching pixel of the target pixel in the source image.

In this section, two approaches will be introduced, global image matching and swatches matching, where swatches matching is based on the idea of global image matching.

2.1 Global Image Matching

First of all, in order to obtain the pixel luminance, the BGR color space of the reference image and grayscale image is transformed into LAB space. The luminance of the reference image is remapped according to the luminance and the standard deviation of the grayscale image. The luminance remapping is performed by this equation:

$$L = \frac{\sigma t}{\sigma s}(l - \mu s) + \mu t$$

where l is the luminance value of the source image, L is the remapping luminance value, μs and μt are the mean values of the luminance of the source and grayscale images, and σs and σt are their standard variance of the luminance.

Then an appropriate number of color sample points are selected using randomly jittered grid, and in the project, the project select 225 sample points. The weighted average of the luminance of the pixel points and the standard deviation of the luminance in the neighborhood range is used as the weight value. The neighborhood size of 5*5 is chosen in the project. For each pixel point in the grayscale image, the best match that has the minimum weight value is found among the sample points of the reference image, and the chromatic value (A, B) of the matching point is assigned to the corresponding pixel point of the grayscale image. Finally, the target image needs to be converted from LAB space to RGB space.

2.2 Swatches

In this section, the color space of the images is converted to LAB as previous method. The user then selects the corresponding swatches between source image and target image. Luminance remapping, jittered sampling, and color transfer is only performed on the source swatches and target swatches. The method uses L2 error to find the best match point between swatches and 3*3 neighborhood of target pixels. After the final colorization, the result image would be converted to RGB space.

3. Implementation details

The approach is accomplished via Python3 in Jupyter Notebook using several Python packages: cv2, numpy, random, matplotlib, and scipy.ndimage. Cv2 is used for processing image channels, display and interaction. Numpy handles matrix calculations such as luminance remapping and weight value of color transfer. Random is used for selecting color points on a randomly jittered grid. Scipy.ndim age is for multidimensional image processing in calculation of neighborhood standard deviation. Matplotlib is for visualization of image. The source images and grayscale images come from Welsh's paper and photos I took.

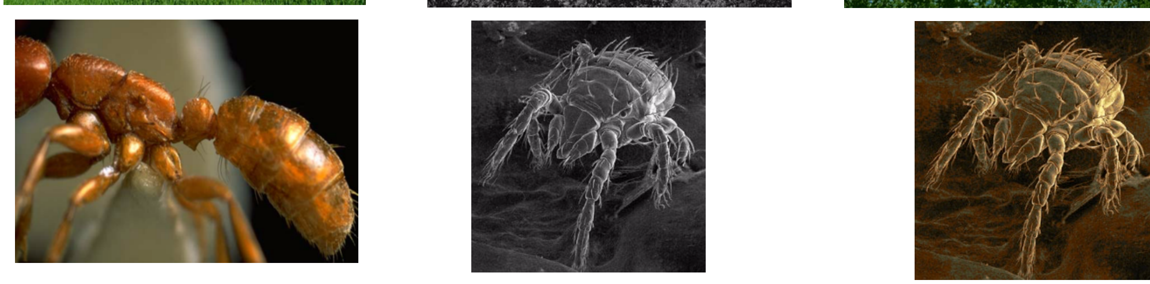
4. Results

4.1 Global Image Matching

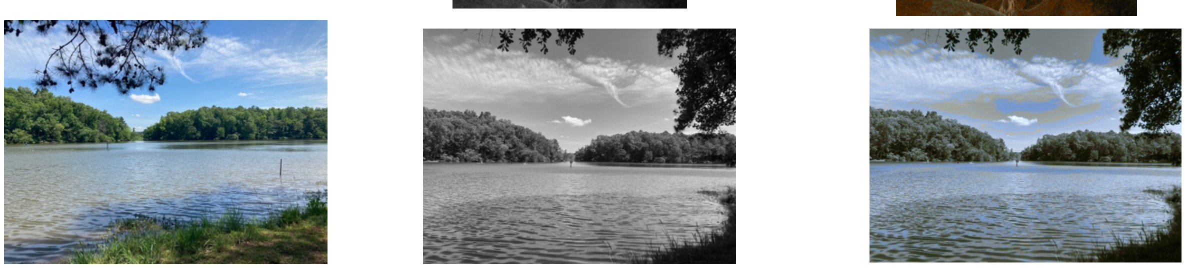
The following four sets of figures are the results of running the global image matching method. The first column is the source image, the second column is the grayscale image, and the third column is the result of coloring the grayscale graph. The input images of Figure (a)-(c) are from Welsh's paper and get similar results, which indicates the approach was successfully implemented. The input images of Figure (d) come from the photos taken by myself. The effect is not as good as the previous, and the coloring is not uniform in some places.

(a)

(b)



(c)

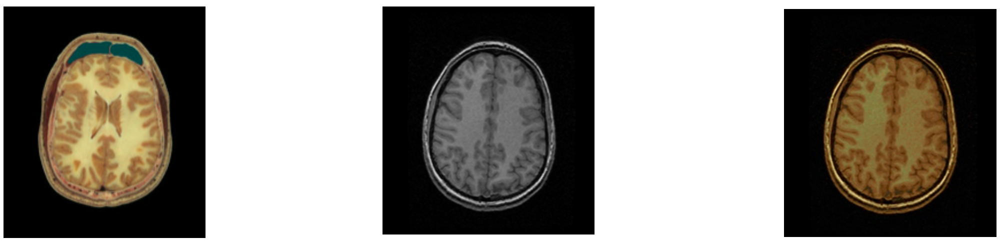


(d)

4.2 **Swatches**

The following four sets of figures are the results of swatches method. Figure (e)-(g) use the source and grayscale images from Welsh's paper and produce similar results, so this method was also successfully implemented. However, the same problem is that when I run it with photos I took (see Figure (h)), the result is strange.

(e)



(f)

(g)

(h)

5. Challenge / innovation

I think this project is challenging in two ways. First, this paper uses calculation methods from other papers, but in this paper only the ideas are mentioned and no explicit formulas are given, so it requires reading other papers such as luminance remapping. Secondly, neither method gave good results when tested using my own photos. I think this is because my photos do not have a high consistency of brightness corresponding to color. That is, different colors will have similar luminance.

I expect to get at least 10 points in this section. I regret not figuring out how to improve this approach effectively, but I think I did a good job of implementing the method of the paper.

Reference

Welsh, Tomihisa, Michael Ashikhmin, and Klaus Mueller. "Transferring color to greyscale images." *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*. 2002.

Hertzmann, Aaron, et al. "Image analogies." *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. 2001