NOTE: I have emailed this to the instructor before the deadline

Assignment 0

CS 543 - Computer Vision

Colorizing Prokudin-Gorskii images of the Russian Empire

Naman Shukla (UIN : 656868559, netID : namans2)

---

1. Introduction:

In this assignment, I have attempted to extract the three color channel images by Sergei Mikhailovich Prokudin-Gorskii (1863-1944), place them on top of each other, and align them so that they form a single RGB color image. I have implemented the channel detection with simply stacking three layers with different combination of RGB, aligning with sum of squared difference (SSD) method and with more sophisticated method of normalized cross-correlation (NCC). Additionally, I have implemented image pyramid (BONUS) with binary image conversion and tested the implementation with TIF images

2. Preprocessing :

The given image is in the form of a matrix with dimensions (1024 X 393) implicitly containing three images of dimensions (341 X 393). Hence, we need to divide the image into three parts and supply it as an input for RGB channel.

3. Implementation:

3.1 Checking RGB channels:

I have stacked the three layers of R, G and B with all possible combinations to figure out which is the best overall combination. The following collage represent the best is RGB (standard) channel is it represents the natural colors. Image is still blur. Hence, we have to align the template to remove the blur.

Figure : From left to right is in the following order- BGR, BRG, GBR, GRB, RBG, RGB

3.2 Sum of squared difference (SSD) method:

This method is brute force method to align based on finding the best (minimum) differences in the pixel values with respect to different layers on set base layer. Hence, I have set each of the three template (R, G and B) as base one-by-one. Then, the best out of the three base is selected as the output of SSD.

Additional Implementation :
Borders are the region of highest darkness. Therefore, its influence on the finding the difference in the pixel value with respect to layers on set base is the highest. In order to properly align the image, I have cropped the borders. Hence, I have used only 84% (because 8% of the width and length are covered by black pixels i.e. borders on each sides) of the image from the center to align the layer over the base. This eventually resulted in far better alignment and comparable to other sophisticated method like NCC.

Here is one of the example of the output :

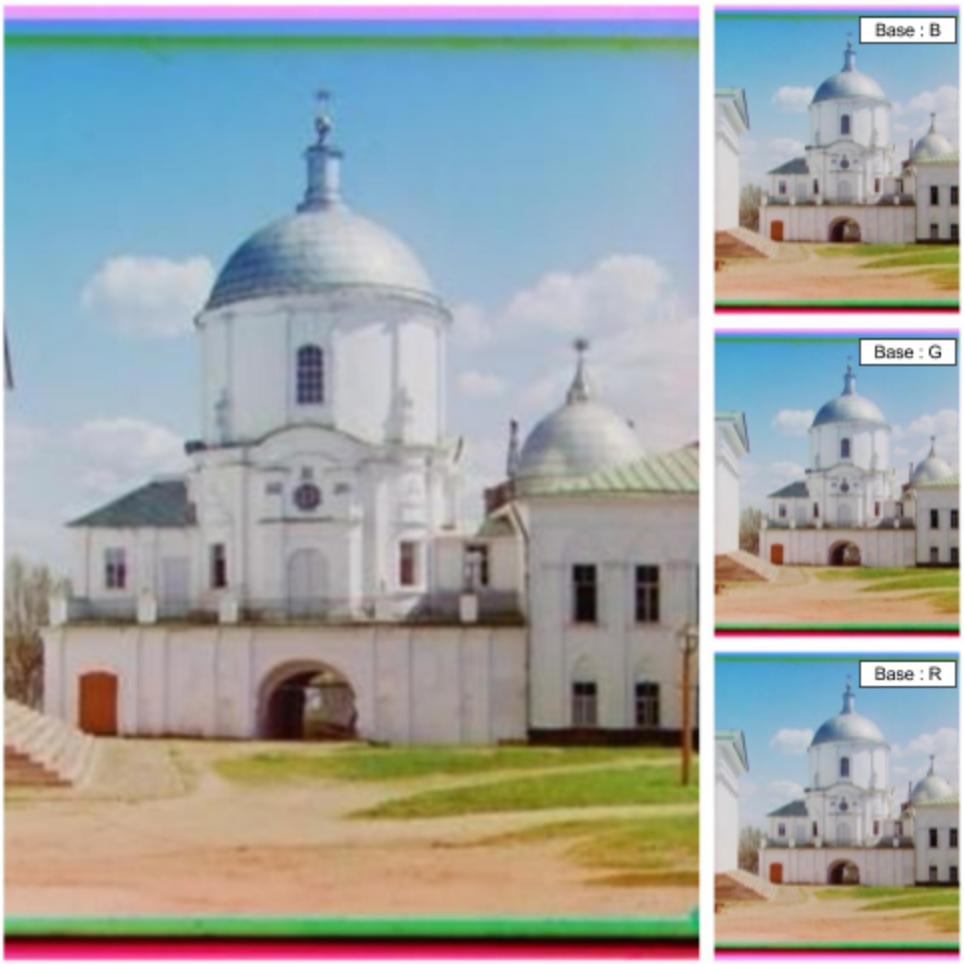

Figure : Image with the best base (G) with all

Here are all of the images with the corresponding best bases.

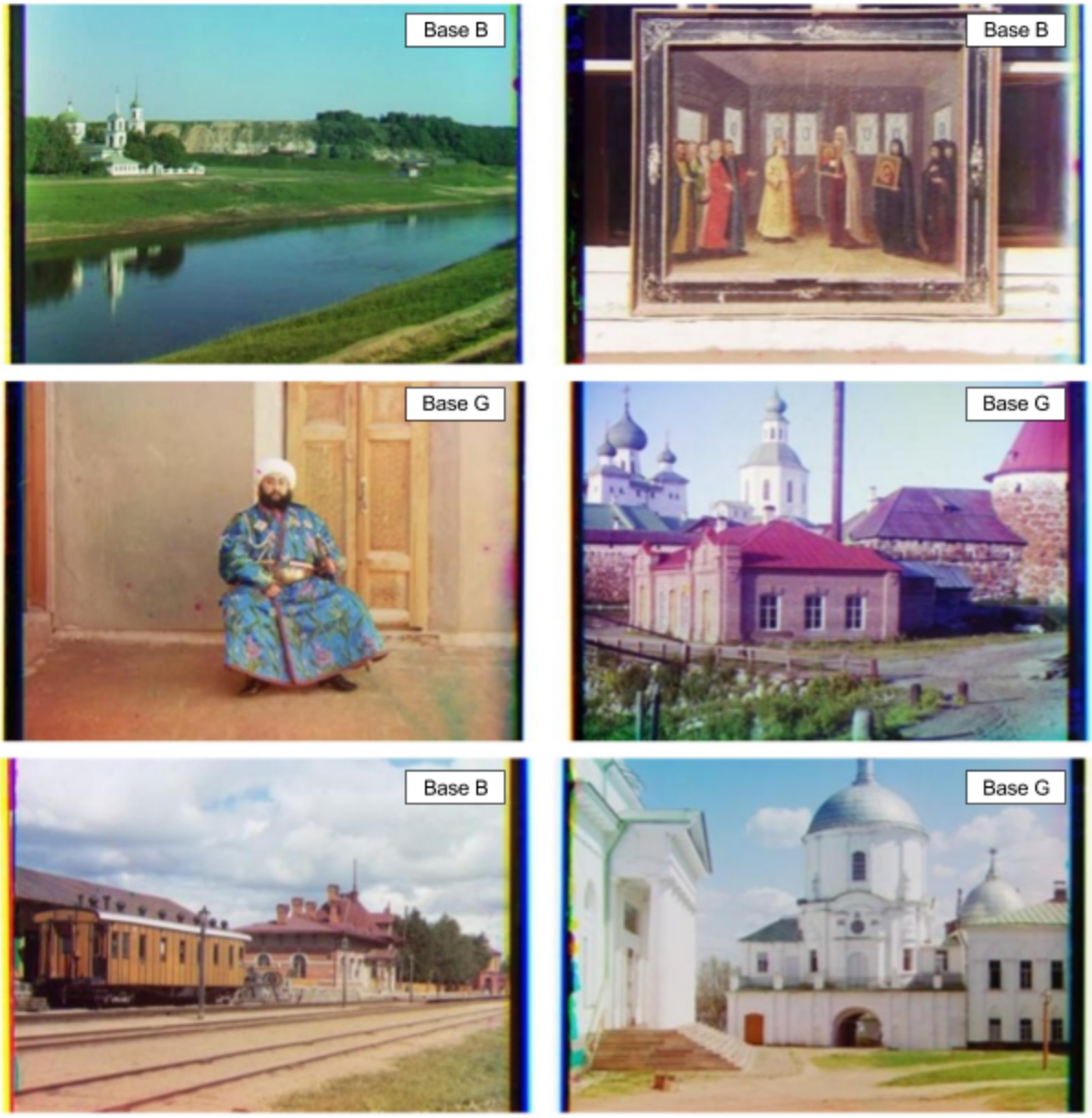

Figure : Collage of all of the colored images obtained from SSD with corresponding best bases.

3.3 Normalized Cross Correlation (NCC) method :

In this method we have more sophisticated way instead of brute force to find the perfect alignment of one image (referred as template) with respect to other (referred as base). It is a four step process :

- First vectorize and normalize the Images.
- Move the template over base with the given shifting window (here I've used -15 to 15).

- Find correlation values for each shift by taking the dot product of normalized vectors.
- Find the maximum correlation value and the corresponding shift.

Note :
In this method, I have not used any cropping criterion. Instead, supplied the image as it is. Even then, NCC performs comparable to SSD with the additional boundary adjustment.

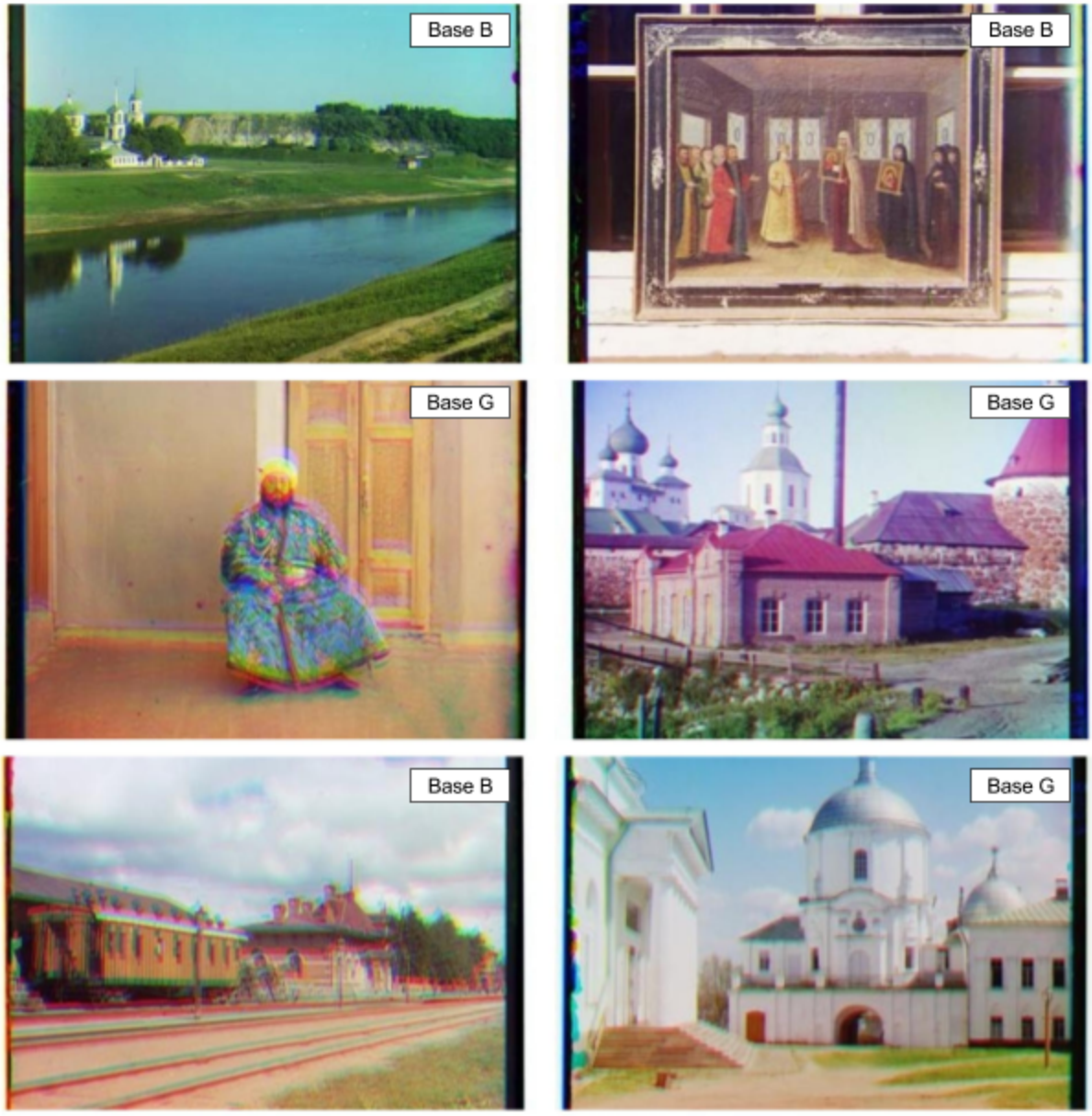

Figure : Collage of all of the colored images obtained from NCC with corresponding best bases.

Comment:

Images like 00125v, 00149v and 00351v looks quite same as the images from SSD. But rest of the images (00153v, 00398v and 01112v) are blur and not exactly aligned. This observation is explained in the following example,

This image is obtained by SSD method (with additional boundary removal) with base G.

This image is obtained by NCC method with base G. The layers are not properly aligned. There exist an observable shadow which results in blur effect.

Hence, SSD works better with some images but with the trade off of boundary removal. Whereas, NCC is more analytical method that could render information about the boundary and thereby could perform equally well as SSD (with few exceptions).

3.4 Image Pyramid (BONUS)

In this method, the image is iteratively reduced multiple times. I have used 2x as multiplication factor i.e. the image would be reduced by ½ of its dimension. Then starting from the top of the pyramid (smallest picture), we should find the best alignment of an image with respect to the base image. I have used 4 layer pyramid. After the above step, the shifting coordinates should be multiplied with the multiplication factor and we should again find the alignment of an image with respect to the base image for 2nd layer of the pyramid from the top. Repeat the above steps till the end of the pyramid.

Additional Implementation :
In addition to the boundary removal, I have used Binary Images for the alignment.

This implementation resulted in a far better output than regular image with SSD in a relatively lesser computation time.

The following output is the selected best images obtained from ‘Image Pyramid’ method.

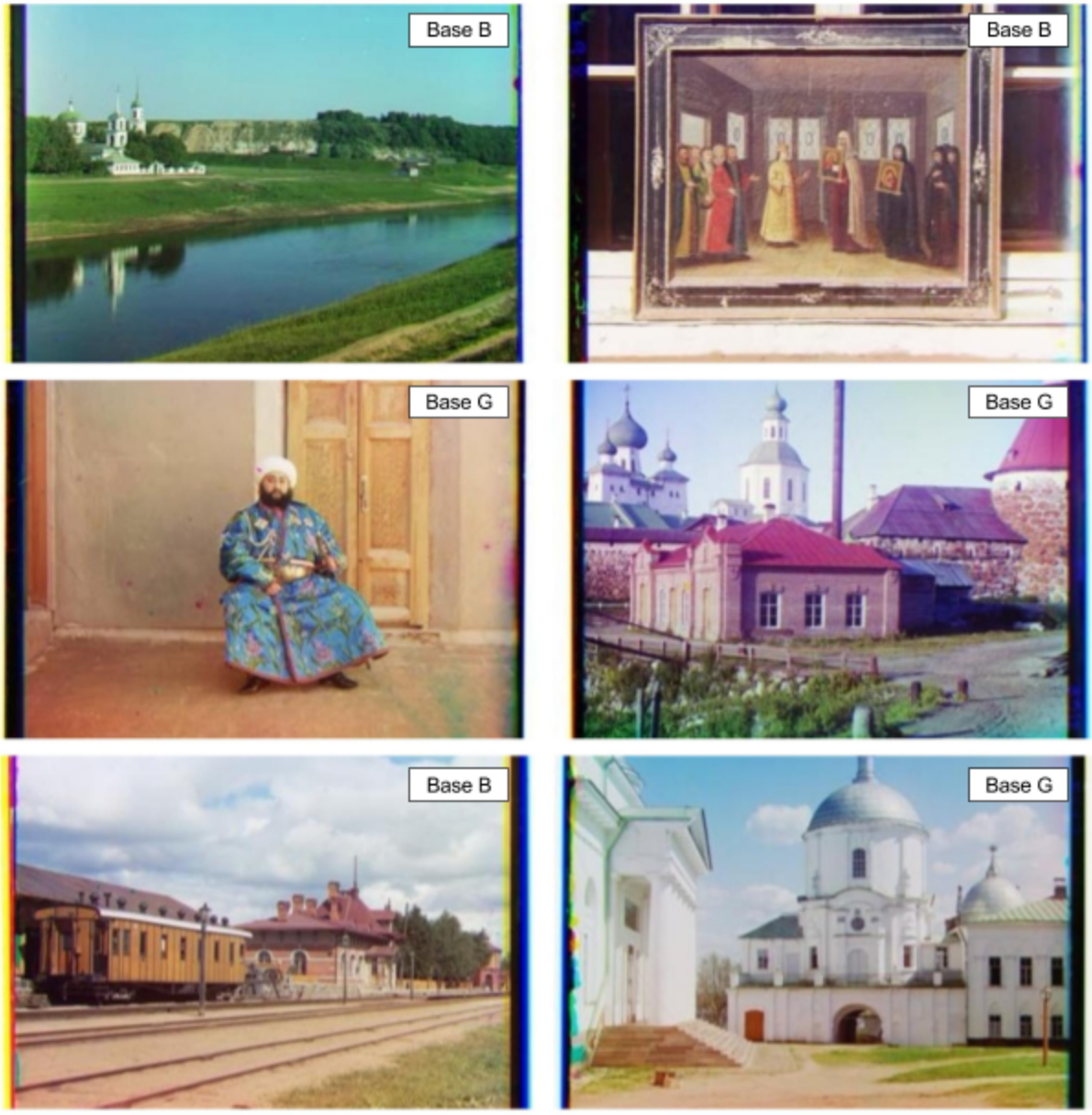

Figure : Collage of all of the colored images obtained from Image Pyramid with corresponding best bases.

The results of the additional Images in TIF format are given below:

Figure : Image 0104u colored after Image Pyramid implementation with base B

Figure : Image 01657u colored after Image Pyramid implementation with base B

Figure : Image 01861a colored after Image Pyramid implementation with base G

Comment :

The computation time for running image pyramid algorithm is around 24 seconds (depending on machine) for a single image with 4 layers of pyramid and around 16 seconds for a single image with 5 layers of pyramid. It decreases with the increases in the height of the pyramid but the trade off exists with the accuracy of alignment.

Note: all images are attached in Images directory in the zip file.