CSCE-689 Computational Photography

Programming Assignment 1

Sriram Chakravarthy
UIN:629009746

February 19,2022

Objective

The goal of this assignment is to take the digitized Prokudin-Gorskii glass plate images and, using image processing techniques, automatically produce a color image with as few visual artifacts as possible. This should be done by extracting the three color channel images, placing them on top of each other, and aligning them, so that they form a single RGB color image.

Task 1: Single Scale Implementation

For this task, the blue channel image is taken as a reference and the red, green channel images are aligned by minimizing the SSD(Sum of Squared Distance). Since this is a small image, the single scale approach with a window of [-20,21] pixels is used. As a preprocessing step, the borders of the all the three channels are not considered for calculating the SSD.

The details of the steps are as follows:

1. Crop the original image into three separate channel images by using height / 3 as each channel's height.
2. Crop each channel by 20 pixels on all sides. This is done so that the meat of the image can be considered in the alignment, without the borders and other blemishes affecting any calculations.
3. Align the cropped red and green channels to the blue channel to get (x, y) displacements for both channels using the SSD as a metric.
4. Apply the displacements onto the uncropped versions of their corresponding channels.
5. Merge the three channels together into one final color image.

The resulting images along with the shifts for the green and blue channels are shown below:

Cathedral.jpg, shifts : R(12,3) , G(5,2)

Monastery.jpg, shifts : R(3,2) , G(-3,2)

nativity.jpg, shifts : R(7,1) , G(3,1)

settlers.jpg, shifts : R(14,-1) , G(7,0)

From the results, the SSD method seems to perform decently on these images. However, there is a need for use of better features like edge detection and a need for cropping the borders. This is done in the later sections.

Task 2: Multi Scale Implementation

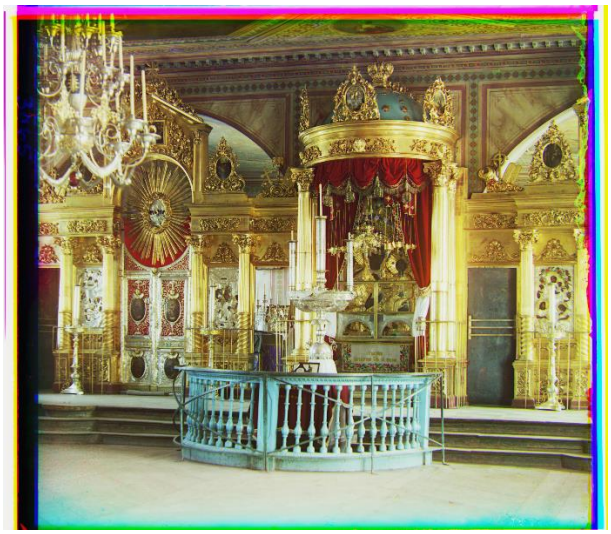
Using a single scale approach for large images is prohibitively time consuming. The solution to this is using a pyramid of scaled versions of the large image.
The steps are described below:

1. Separate the original image to its R, G, B channels as done in the previous task and discard 200 pixels which correspond to the borders in each channel.
2. Create an image pyramid for each channel.
3. Create a four level image pyramid with images of size 1x, (1/4)x , (1/16)x and (1/64)x of the original image.
4. Starting with the image of smallest size, search through each level of the red and green pyramids against the corresponding blue pyramid image using SSD as a metric to get displacement values.
5. Use the displacement values of one level in the next level after appropriately scaling it. This helps in getting an accurate displacement in the higher levels with a small search window.
6. After the displacement is found for the Apply the displacements onto the uncropped versions of their corresponding channels.
7. Merge the three channels together into one final color image.

The resulting images (with reduced resolution to fit here) along with the shifts for the Green and Blue channels are shown below:

emir.tif, shifts : R(88,44), G(49,24)

harvesters .tif , shifts : R(124,13), G(59,16)



icon.tif, *shifts :* R(89,23), G(40,17)

lady .tif *shifts :* R(113,11), G(47,8)

Better features

Using the SSD based on image intensity in the individual color channels to align them is a naïve method. By observing the R, G, B channel images of emir.tif image, I observed that there is no strong correlation between them. This made me explore other methods for image alignment.

I used sobel filter to detect edges in each channel and used that to align the three color channels. There are more similarities in the location of the edges detected across the three channels. Using SSD on these images containing the edges would make more sense. This gave a better alignment in the case of emir.tif(reduced resolution to fit here) which is shown below.

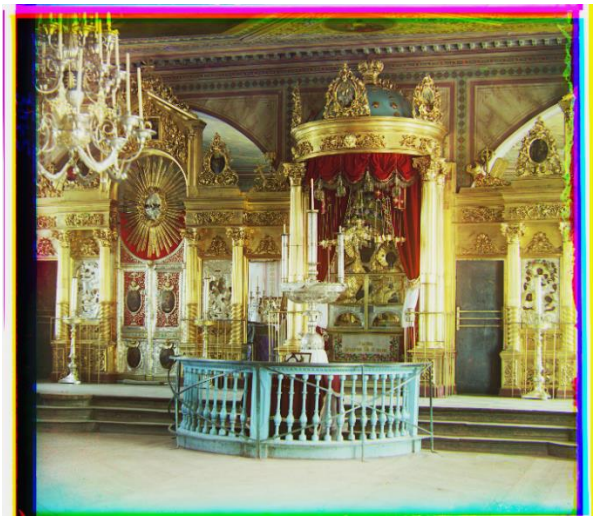
Emir.tif aligned using pixel intensity

emir.tif aligned using sobel edges

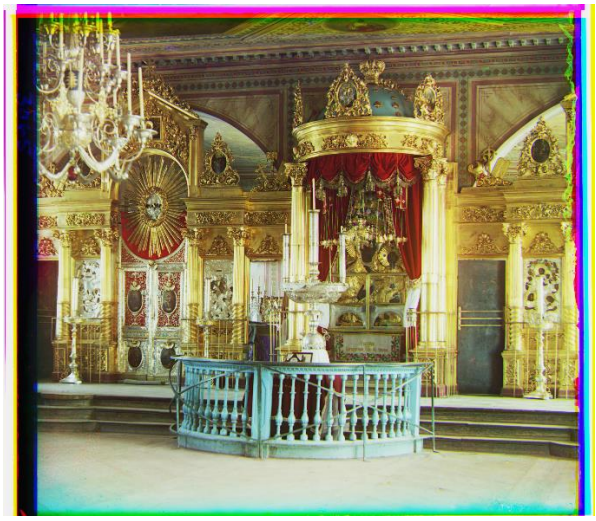
In other images, the improvements are not as significant as in the case of the previous image. Other results are shown below(scaled down to fit here).

harvesters.tif aligned using pixel intensity

harvesters.tif aligned using sobel edges



icon.tif aligned using pixel intensity



icon.tif aligned using sobel edges

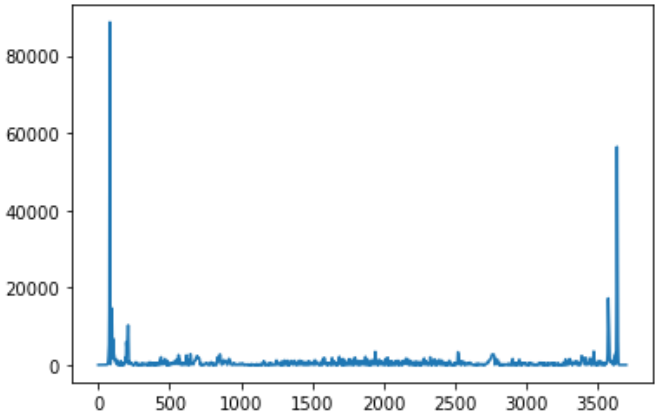
lady.tif aligned using pixel intensity

lady.tif aligned using pixel intensity

Rest of the images are included in the last sections.

Automatic Cropping

Instead of cropping the images arbitrarily, I found the edges along the x and y directions using sobel filter. For detecting edges along the x direction, I plotted the sum of intensities of all the columns in each channel.



I found that the absolute values are at a peak near the borders corresponding to a value of around 4000. I used this as a threshold and found the first column(nearest to the centre, within the search range of 400 pixels from the end points) that has the sum above threshold as the location of the border.

I repeated the above procedure in y direction to find all the four border start points for the three channels. After the images were aligned with the cropped borders, I cropped the individual channels and combined them to form the final cropped image.

The results of cropping are shown below.

Emir.tif cropped

harvesters.tif cropped

Icon.tif cropped

Icon.tif cropped

Results

In this section, the result images of all the tasks are included in their correct resolution.

1. **Task1**

Cathedral.jpg , Shifts: R(12,3), G(5,2)

monastery.jpg , Shifts: R(3,2), G(-3,2)

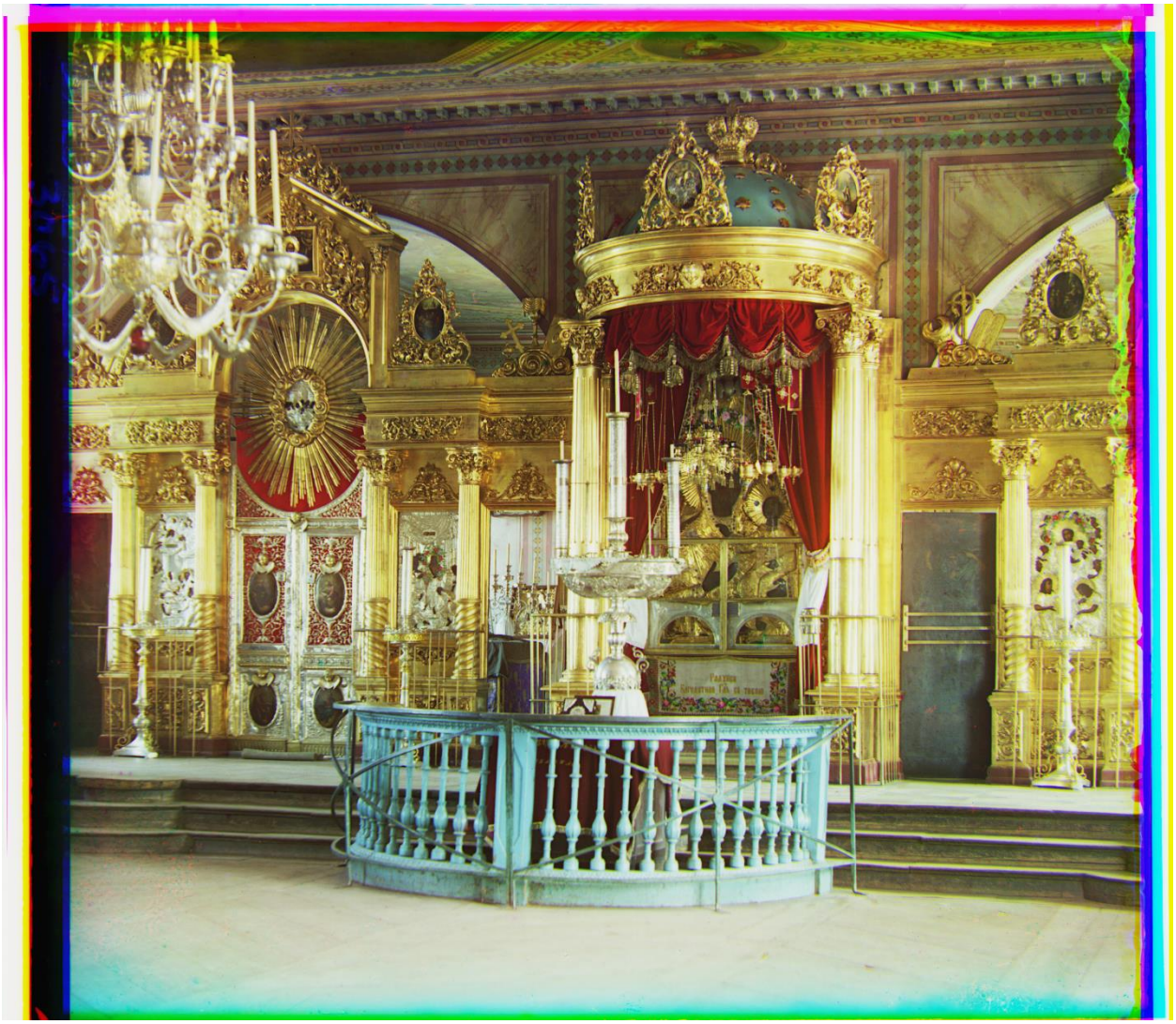
nativity.jpg , Shifts: R(7,1), G(3,1)

settlers.jpg , Shifts: R(14,-1), G(7,0)

2. Task 2

Emir.tif, shifts: R(88,44), G(49,24)

harvesters.tif, shifts: R(124,13), G(59,16)



icon.tif , shifts: R(89,23), G(40,17)

lady.tif , shifts: R(113,11), G(47,8)

Self_portrait.tif , shifts: R(176,36), G(78,28)

three_generations.tif , shifts: R(111,10), G(53,14)

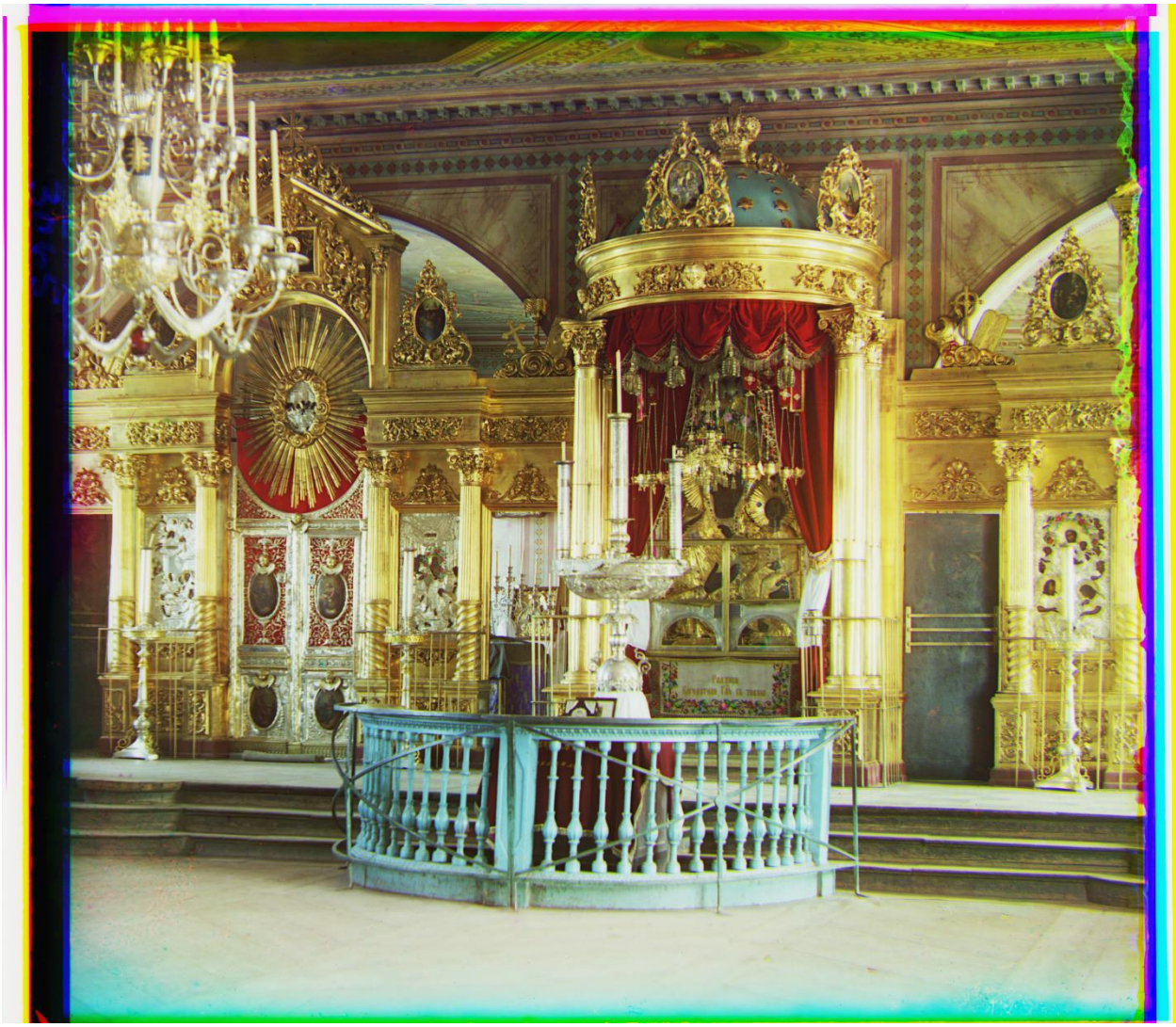
train.tif , shifts: R(87,31), G(42,5)

turkmen.tif , shifts: R(115,26), G(55,18)

village.tif , shifts: R(137,22), G(65,12)

3. Better Features

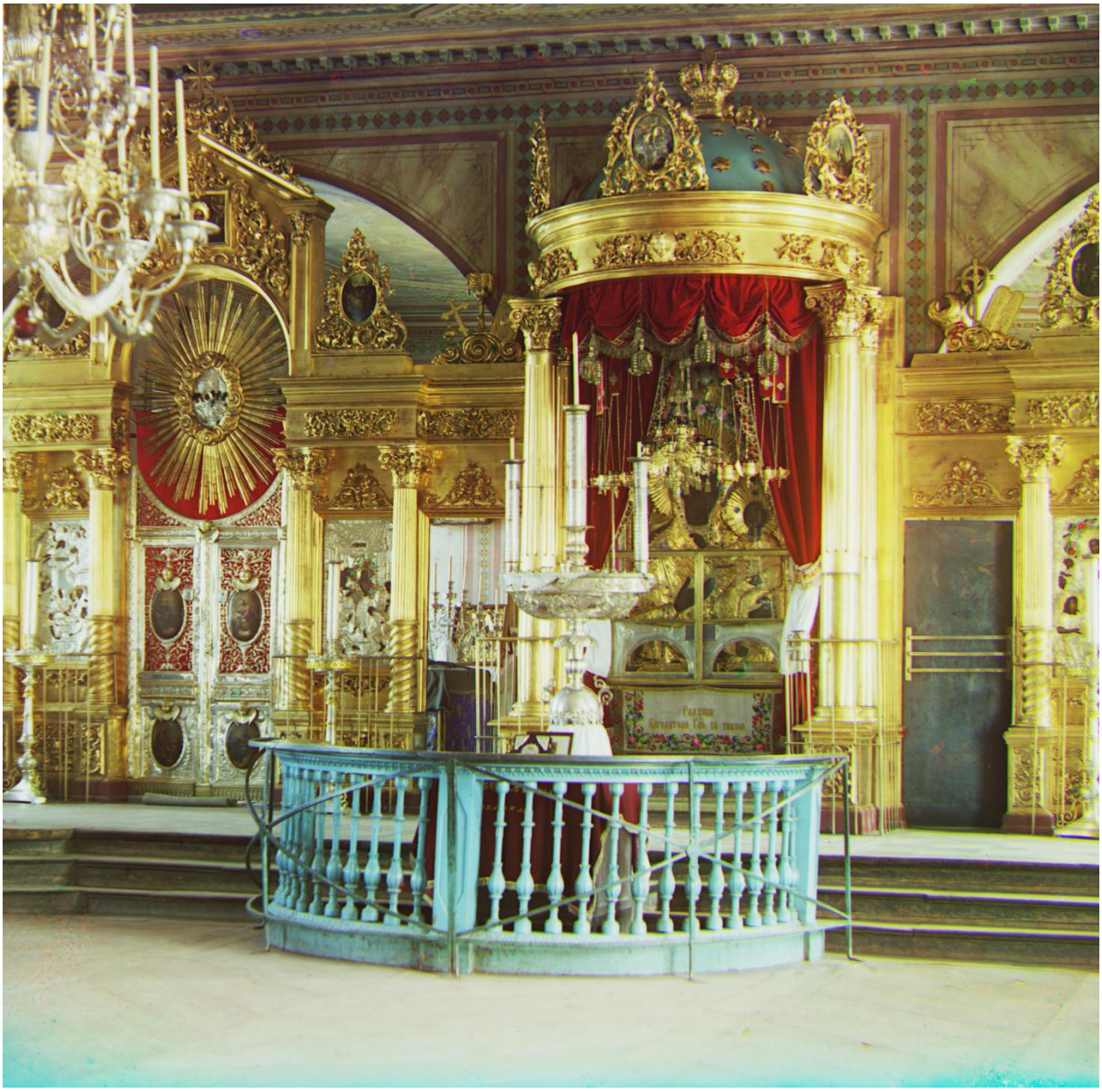
S




7293


Аь 132

4. **Auto-Cropping**




Аб 132

SS