PROJECT ONE

Overview

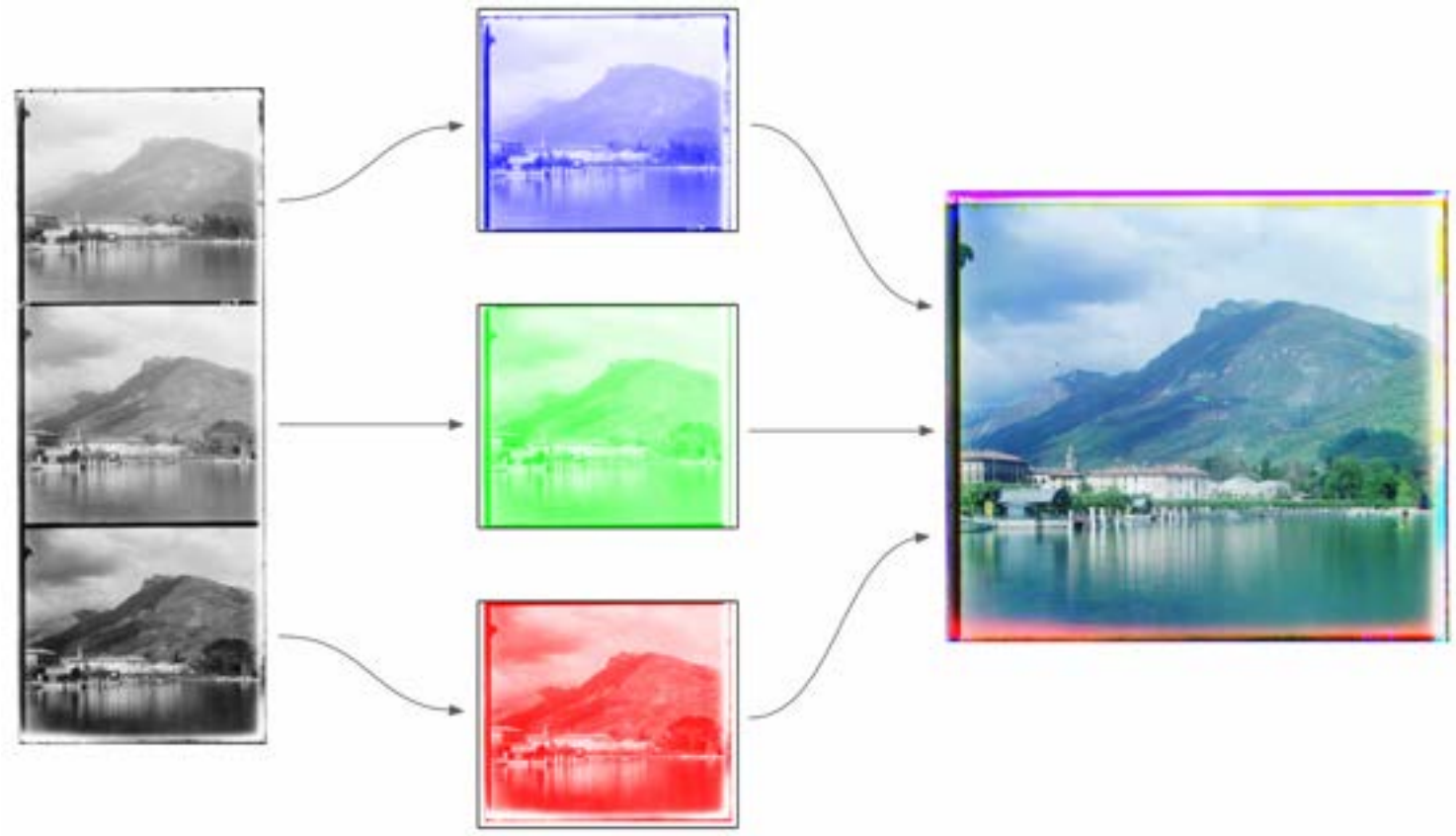
Sergei Prokudin-Gorskii, a pioneer of early color photography, captured thousands of scenes across the Russian Empire in the early 1900s by recording three separate exposures of each scene through red, green, and blue filters. Alas, he was limited by the technology of his time in displaying his photographs in color. Now, his glass plate negatives have been digitized by the Library of Congress, making them available for restoration. The goal of this project is to take those digitized glass plate scans, separate them into their three color channels, and then automatically align and combine them into a single RGB image.

Below is a sample of my RGB reconstructions. I aligned each image using a combination image pyramid and brute force displacement window, scoring with Euclidean Distance. I aligned each channel using the edges calculated with the `sci-kit` sobel edge filter.

You can view my full jupyter notebook here: `https://github.com/pauburk/CS180/tree/main/1/1.ipynb`. The original .tif files are omitted from the repository as they are too large. You can view them here instead: `https://drive.google.com/drive/folders/1b3CUGd8j3kIabL47yY8BZCVc-0Dff8pY?usp=sharing`.

Approach

After importing the data, I seperated each image into three color channels by dividing the height by three and taking each section.

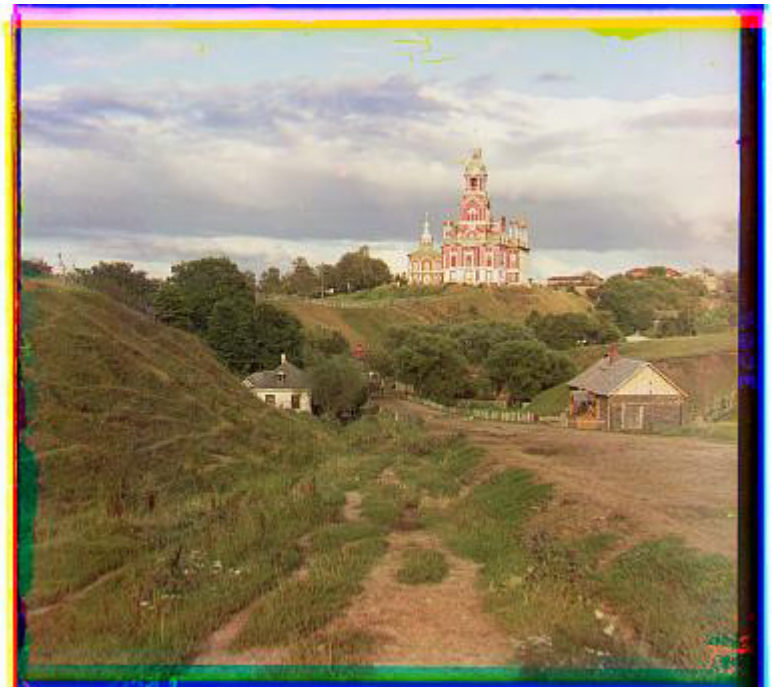


Then, I aligned the green and red channels to the blue channel, and stacked the images (red on top of blue on top of green). To align the images, I used a combination of brute force with an image pyramid. Brute force checked every displacement value within a range of `(-30, 30)` pixels in the `x` and `y` directions, and calculated the best displacement by scoring each shift using the Euclidean Distance formula: `sqrt(sum((img1-img2)^2))`. When calculating the best displacement, I first applied the `sci-kit` Sobel edge filter so that the alignment was guided by image edges rather than raw pixel intensities. The image pyramid recursively downscaled the images by `1/2` until the dimensions were `<400px`, upon when brute force would be executed on the small image. Then, the image was recursively upscaled by `1/2` and during each iteration the displacement was refined within a range of `(-10, 10)` pixels. When the images reached original scale they would be aligned.

I kept the borders in the final images to demonstrate the displacement.

Given Examples

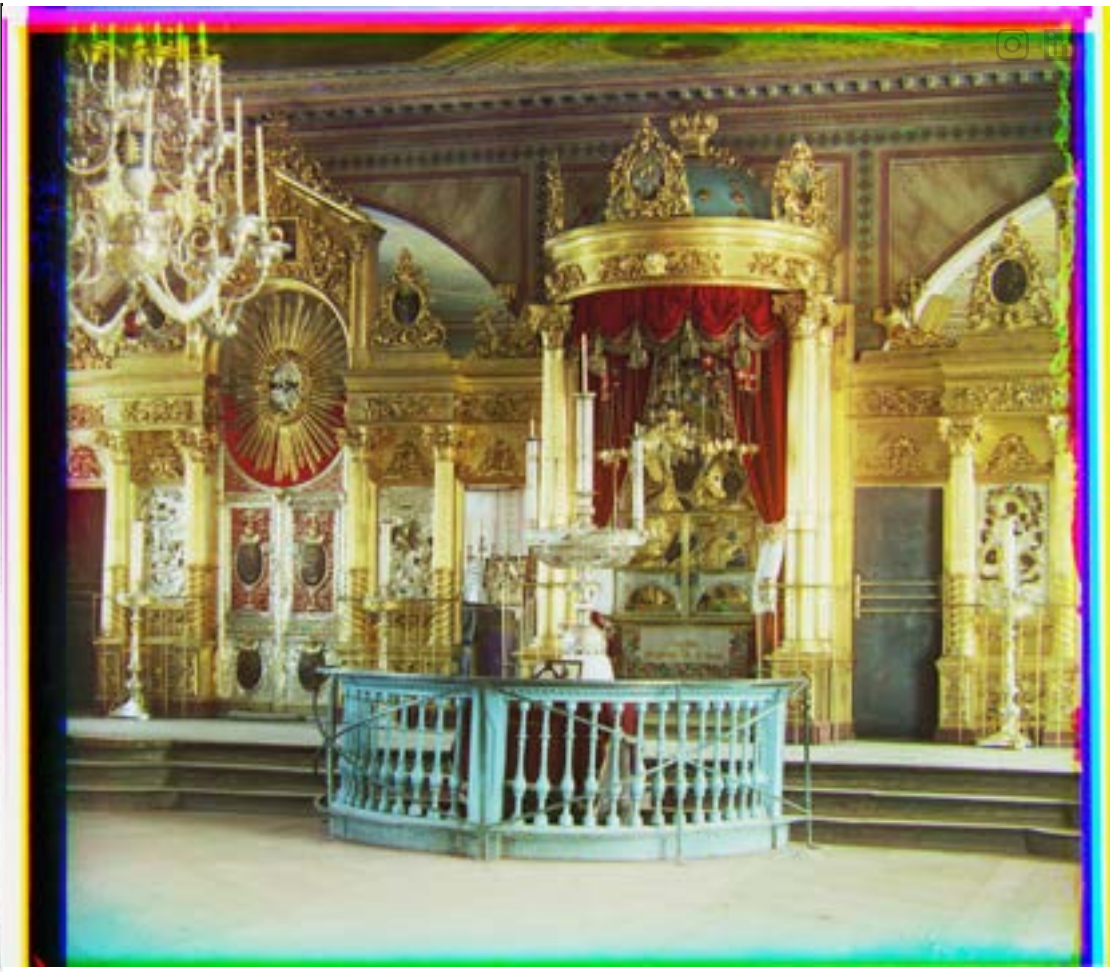
emir – G shift (dy, dx): (49, 24) – R shift (dy, dx): (107, 40)



cathedral – G shift (dy, dx): (5, 2) – R shift (dy, dx): (12, 3)

church – G shift (dy, dx): (25, 4) – R shift (dy, dx): (58, -4)

harvesters – G shift (dy, dx): (60, 18) – R shift (dy, dx): (123, 14)



icon – G shift (dy, dx): (41, 17) – R shift (dy, dx): (90, 23)

italil – G shift (dy, dx): (38, 22) – R shift (dy, dx): (77, 36)

lastochikino – G shift (dy, dx): (−3, −1) – R shift (dy, dx): (76, −8)

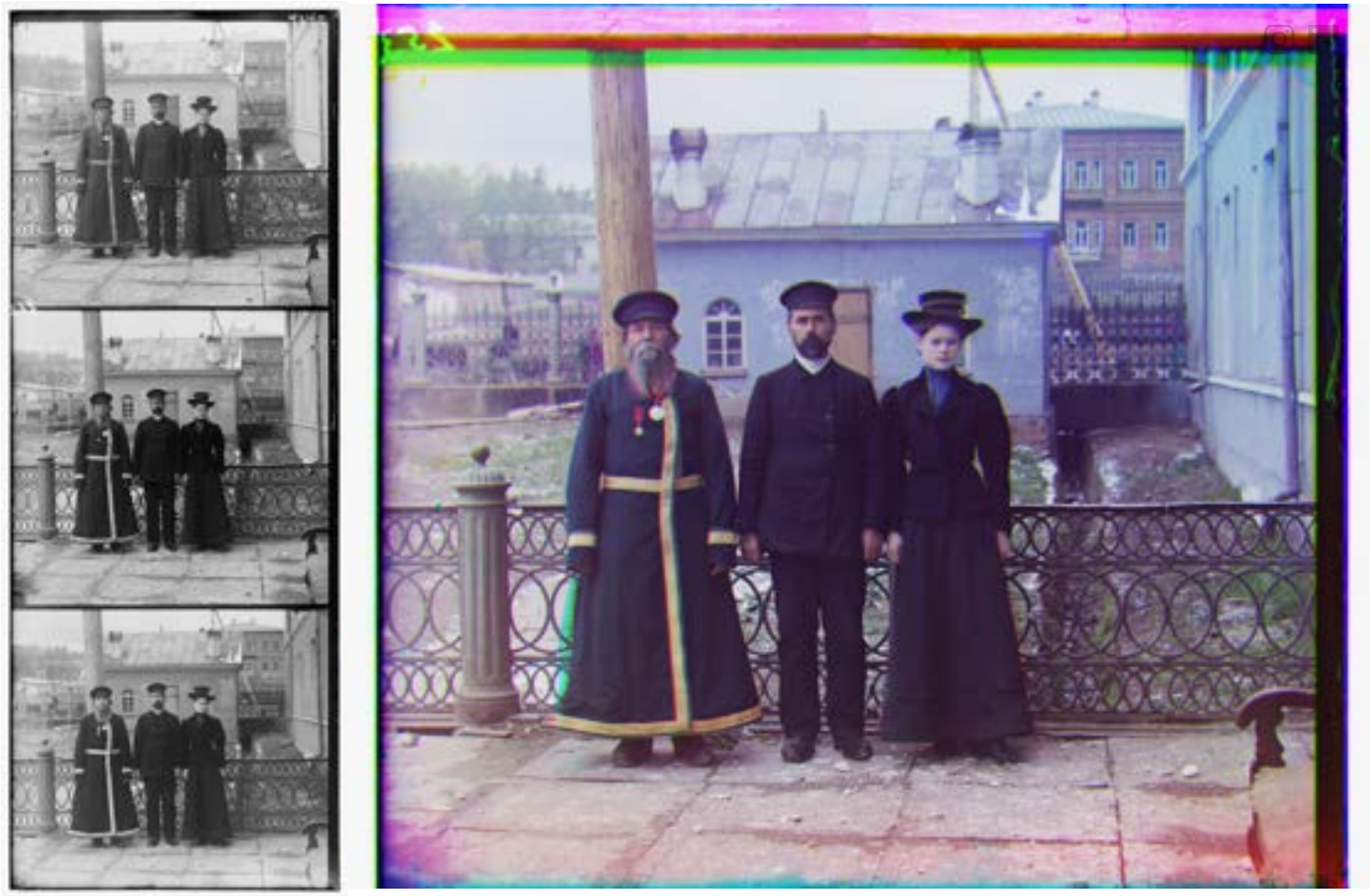
lugano – G shift (dy, dx): (41, -17) – R shift (dy, dx): (92, -29)

melons – G shift (dy, dx): (80, 10) – R shift (dy, dx): (177, 14)

monastery – G shift (dy, dx): (-3, 2) – R shift (dy, dx): (3, 2)

self_portrait - G shift (dy, dx): (78, 29) - R shift (dy, dx): (176, 37)

siren – G shift (dy, dx): (49, –6) – R shift (dy, dx): (96, –24)

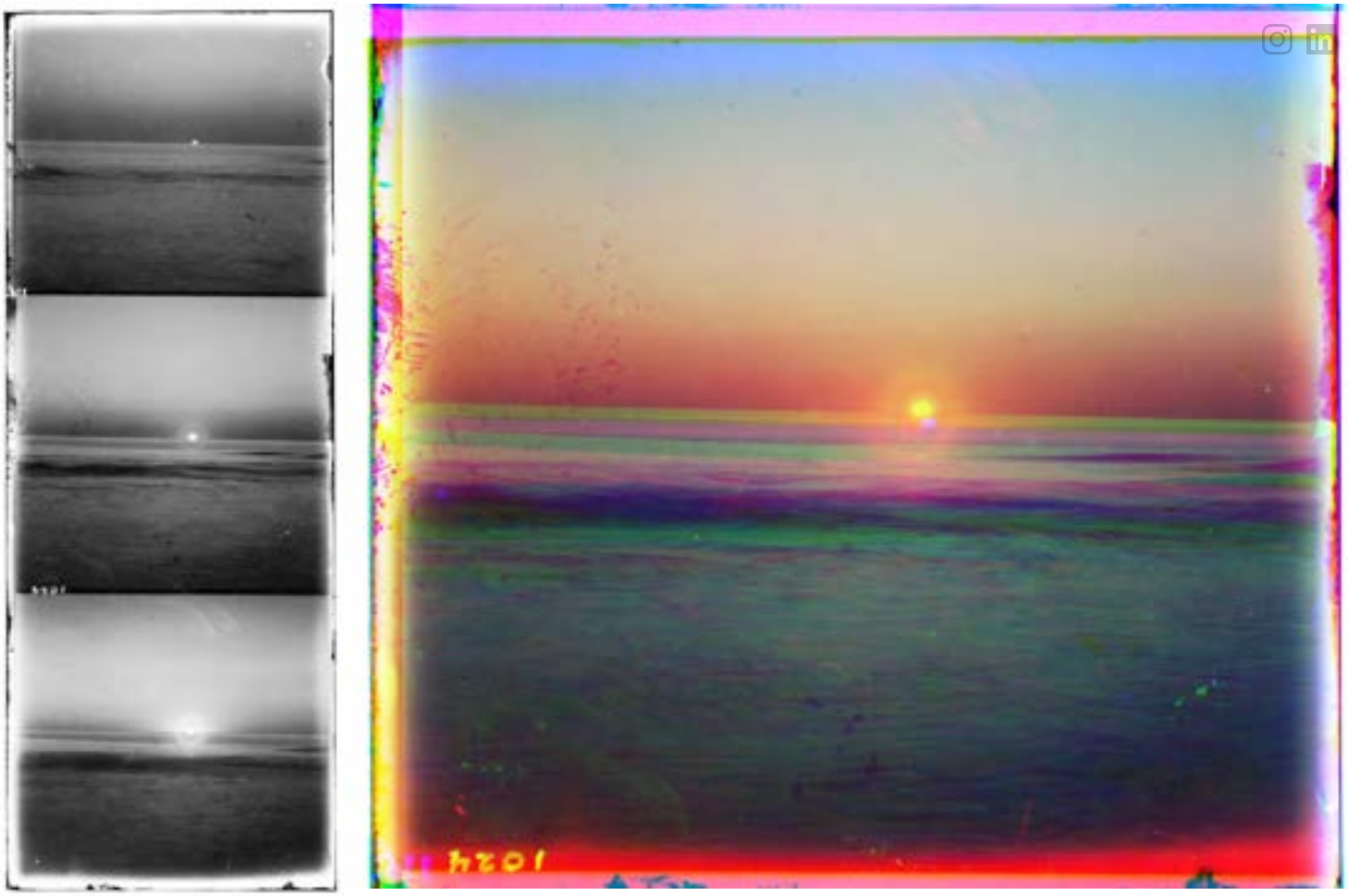


three_generations – G shift (dy, dx): (53, 13) – R shift (dy, dx): (111, 9)

tobolsk – G shift (dy, dx): (3, 3) – R shift (dy, dx): (7, 3)

Custom Examples

Below are two images I reconstructed from the `Prokudin-Gorskii collection`.



zakat na morie – G shift (dy, dx): (64, 368) – R shift (dy, dx): (120, -12)

ekaterinburg – G shift (dy, dx): (49, 16) – R shift (dy, dx): (115, 20)