CS180 Project 1: Colorizing Images of the Russian Empire

By Ethan Kuo

Introduction

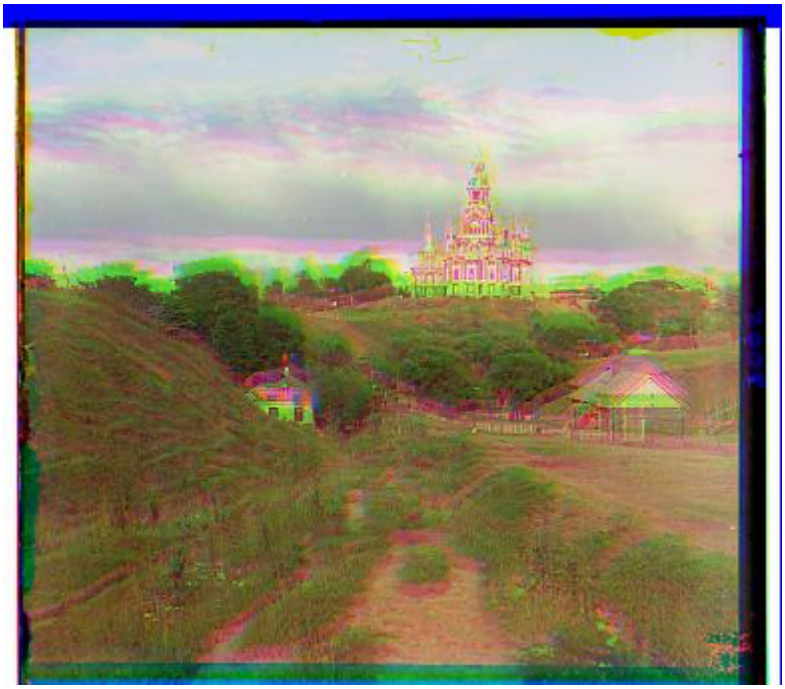
In the early 20th century, Sergei Mikhailovich Prokudin-Gorskii traveled throughout the Russian Empire and took thousands of pictures of everyday sights with three different filters: red, green and blue. Prokudin-Gorskii knew that, with these three colored exposures recorded onto glass plates, color pictures of these sights could eventually be brought to life. This project does just that.

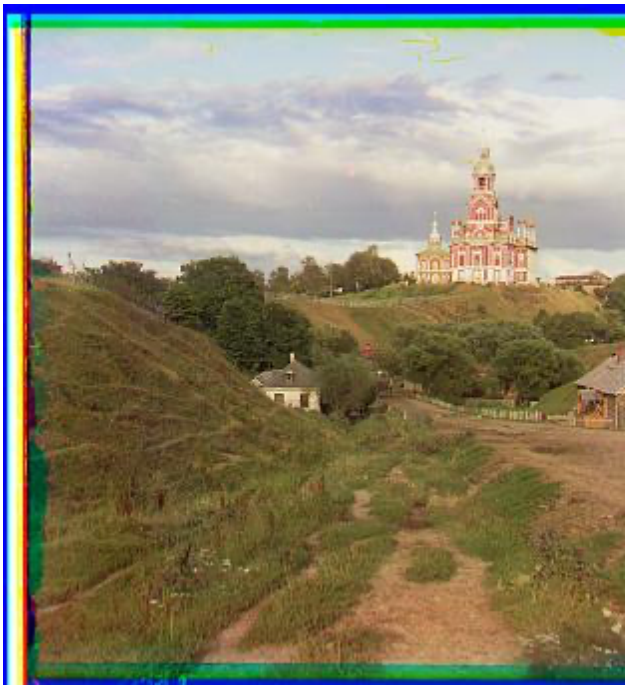
Methodology

My task was to take 3 grayscale images, each corresponding to the red, green, and blue channels for a colored image, and align them to construct that colored image. The difficulty here lies in that these filters must be aligned properly, and we need to try various alignments and assess their effectiveness using some similarity metric.

The similarity metric I used was **normalized cross correlation** (NCC). Intuitively, NCC yields a high similarity score for an alignment of 2 channels when high values are matched with high values, and low values are matched with low values.

Crucially, when aligning color channels, the "similarity" we are looking for is not in the RGB values themselves but rather the structural similarity of the images. For example, for an image with a green hill, the RGB values for that hill would be close to (255, 0, 0), but NCC would punish the correct alignment of the red and green filters since their values are so different. Thus, I applied **Canny edge detection** on each channel, an algorithm that identifies the edges or boundaries of objects within an image. Then, I applied NCC on the "edge versions" of the channels so that they would be aligned based on structural similarity.

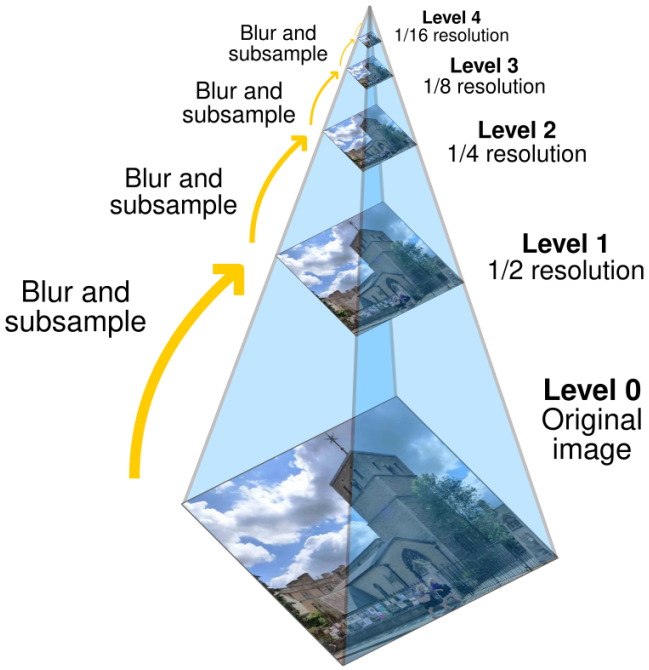




ıages NCC on direct RGB values NCC on Canny edge detection image

For small .jpg images, I searched a range of displacements, such as [-15, 15] pixels, for the red or green filters relative to

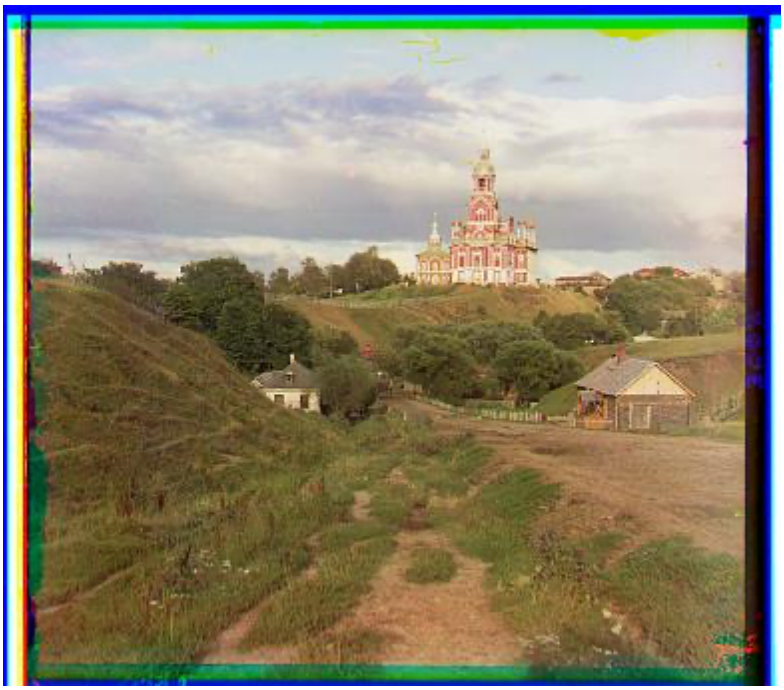
the blue filter. For larger .tif images, I used a **recursive image pyramid** search. Image pyramid search works by repeatedly downscaling an image to create a pyramid of progressively smaller images, then using the best displacement vector for the previous resolution level as a starting point for the search for the next level. This significantly reduces the search space of displacement vectors by progressively honing in on high potential vectors.



Aligned Images

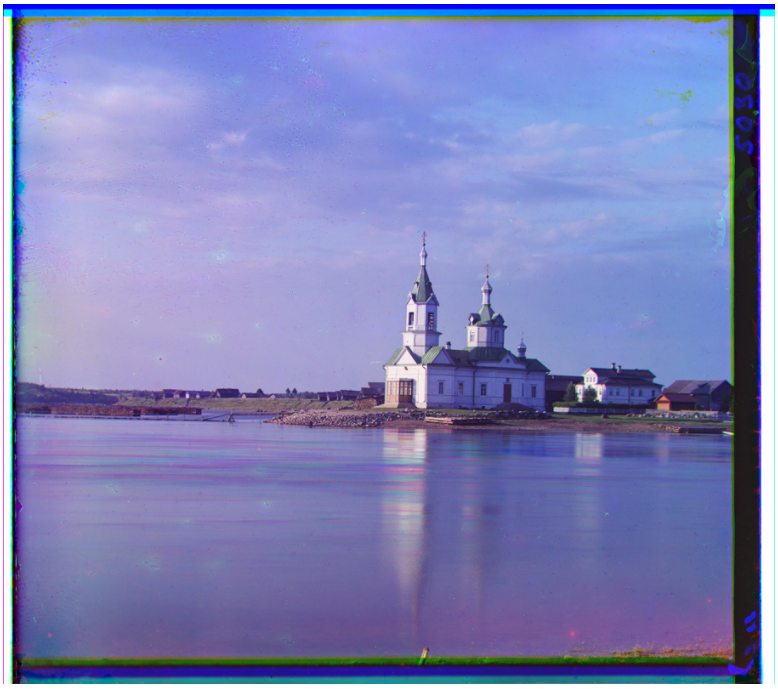
Like the previous example, each colorized image is the result of aligning the Canny edge versions of the color channels using the NCC similarity metric. This scheme was successful on both the small and large images.

Cathedral



Original images G: [2, 5], R: [3, 12]

Church



Original images G: [3, 25], R: [-4, 58]

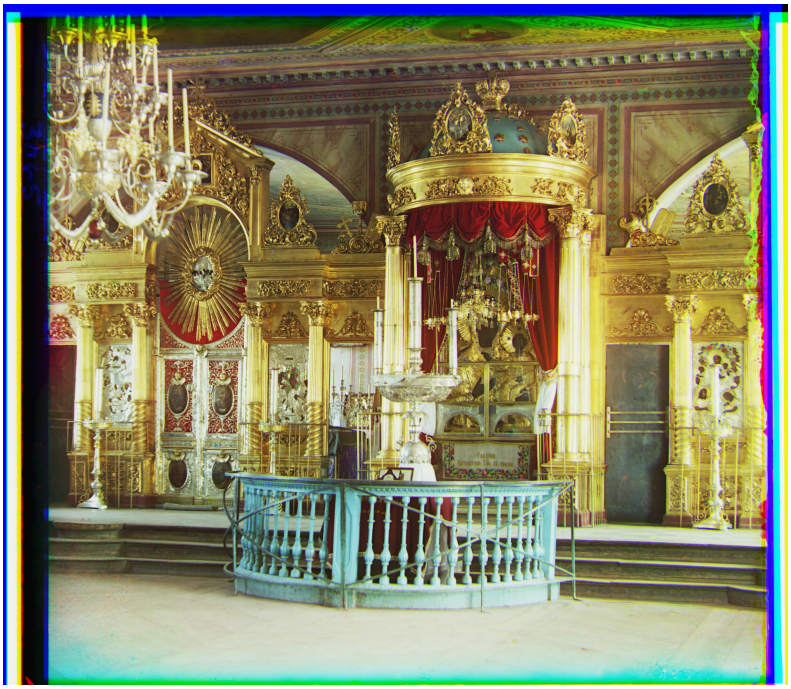
Emir

Original images G: [23, 49], R: [40, 107]

Harvesters

Original images G: [17, 60], R: [13, 123]

Icon

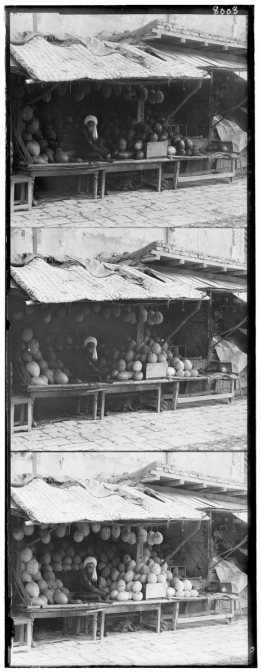


Original images G: [16, 39], R: [22, 88]

Lady

Original images G: [10, 56], R: [13, 120]

Melons



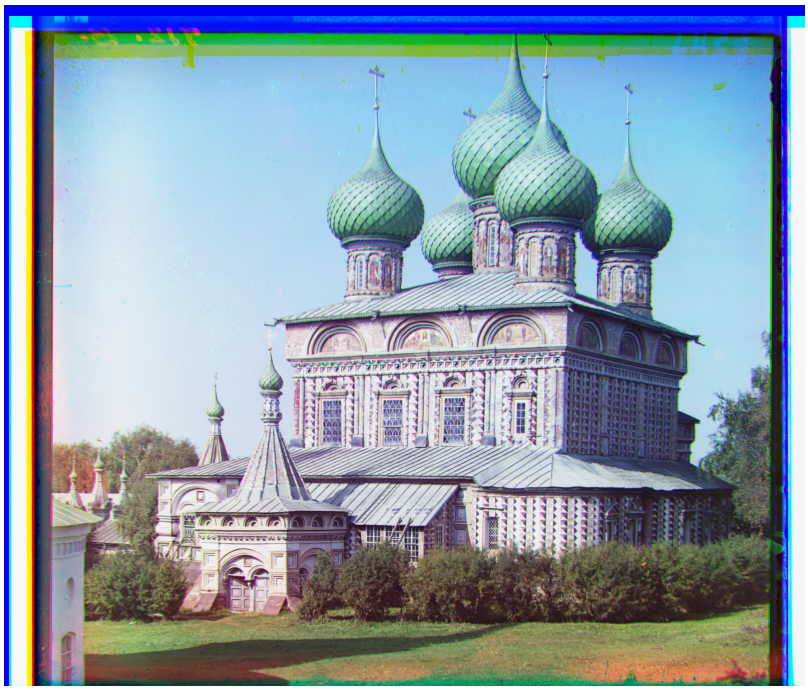
Original images G: [10, 80], R: [14, 177]

Monastery

Original images G: [2, -3], R: [2, 3]

Onion Church

Original images



G: [24, 52], R: [35, 107]

Sculpture

Original images

G: [-11, 33], R: [-27, 140]

Self Portrait

Original images G: [29, 77], R: [37, 175]

Three Generations

Original images G: [12, 56], R: [10, 114]

Tobolsk

Original images

G: [3, 3], R: [3, 6]

Train

Original images

G: [0, 47], R: [29, 85]

Extra images

Adobe

Original images G: [3, 33], R: [7, 79]

Siren

Original images G: [-8, 48], R: [-24, 96]

Boy

Original images

G: [-13, 45], R: [-10, 103]

Railroad

Original images

G: [0, 3], R: [0, 10]