# Colorizing the Prokudin-Gorskii photo collection

## Project Overview

In this project, the goal was to align digitized versions of Prokudin-Gorskii glass plate images by splitting and aligning the three color channels (red, green, and blue) to create a composite color image. Multiple alignment techniques, including SSD (Sum of Squared Differences) and NCC (Normalized Cross Correlation), were explored for both TIF and JPG images. For JPG images, an extensive search method was used to find the optimal alignment. For TIF images, a pyramid approach was implemented to handle the larger size and complexity more efficiently. To further enhance the alignment results for `emir.tif`, image filtering techniques like Sobel were applied to minimize visual artifacts and improve accuracy.

## 1. Single-scale Implementation for JPG Images

Approach: To enhance the accuracy of image processing, I converted the images to a [0, 1] range using `sk.img_as_float`. I experimented with both SSD (Sum of Squared Differences) and NCC (Normalized Cross Correlation) for image alignment, and both yielded similar results. An extensive search method was used to find the best alignment due to the relatively smaller size of the JPG images.

## SSD (Sum of Squared Differences)

While using SSD for alignment, I used Euclidean Distance squared as the metric. I needed to negate the result, as larger values indicate worse alignment (i.e., larger differences between images), whereas smaller values suggest better alignment.

## NCC (Normalized Cross-Correlation)

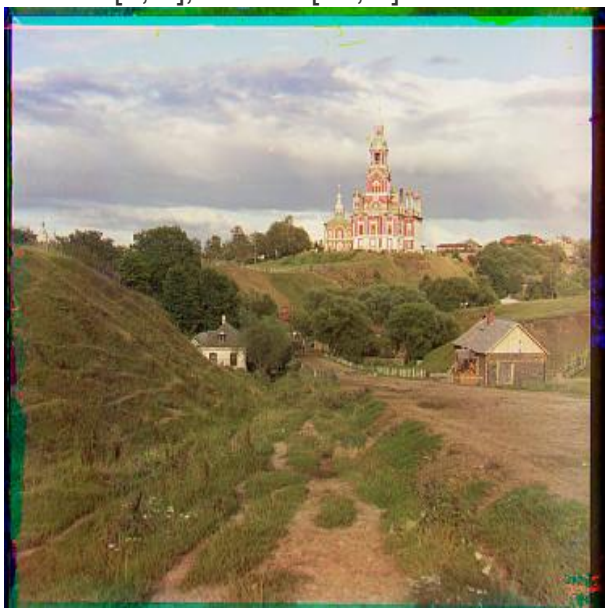
For NCC, I computed the similarity between images by normalizing them and calculating the dot product of the flattened, normalized images. This approach measures how well the images align by comparing their normalized pixel values, with higher values indicating better alignment.

### Cathedral - SSD

G shift: [5, 2], R shift: [12, 3]

## Cathedral - NCC

G shift: [5, 2], R shift: [12, 3]

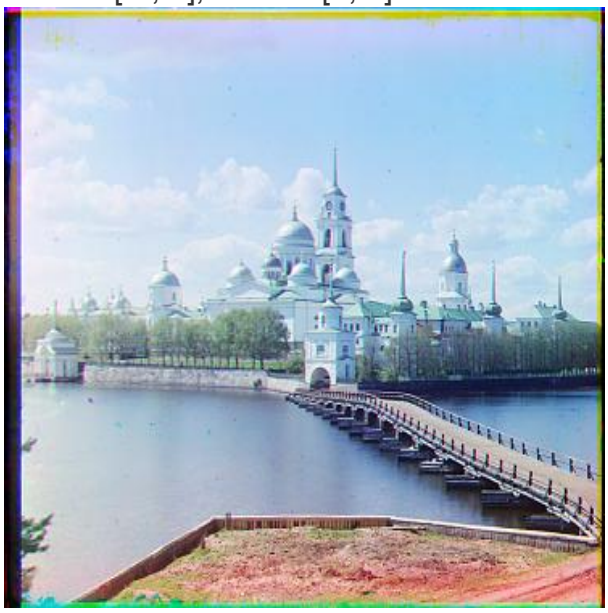


## Monastery - SSD

G shift: [-3, 2], R shift: [3, 2]

## Monastery - NCC

G shift: [-3, 2], R shift: [3, 2]



## Tobolsk - SSD

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

### Tobolsk - NCC

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

## 2. Multi-scale Pyramid Implementation for TIF Images

Approach: For TIF images, I implemented a multi-scale pyramid approach to efficiently align the images despite their larger size and complexity. The pyramid approach allows for progressively finer alignment by analyzing images at different scales. However, the initial results for `emir.tif` were not satisfactory. To address this, I incorporated Sobel edge detection to improve alignment accuracy.

### Church

G shift: [25 4], R shift: [58 -4]

## Emir

G shift: [49, 24], R shift: [72, 43]

## Harvesters

G shift: [60, 16], R shift: [124, 14]

## Icon

G shift: [41, 17], R shift: [90, 23]

## Lady

G shift: [52, 9], R shift: [112, 12]

## Melons

G shift: [82 10], R shift: [178, 13]

## Onion Church

G shift: [52, 26], R shift: [108, 36]

## Sculpture

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

## Self Portrait

G shift: [176, 29], R shift: [176, 37]

## Three Generations

G shift: [53, 14], R shift: [122, 11]

## Train

G shift: [42, 6], R shift: [87, 32]


# 3. Bells & Whistles (Extra Credit)

Although most of the images aligned really well, emir.tif was not aligned perfectly. To address this, I implemented the Sobel filter. The Sobel filter is an edge detection algorithm that highlights edges in an image by calculating the gradient of pixel intensities. This helps in refining the alignment by focusing on the prominent edges, leading to a more accurate composite image.

## Emir

G shift: [49, 24], R shift: [72, 43]

### Emir with Sobel

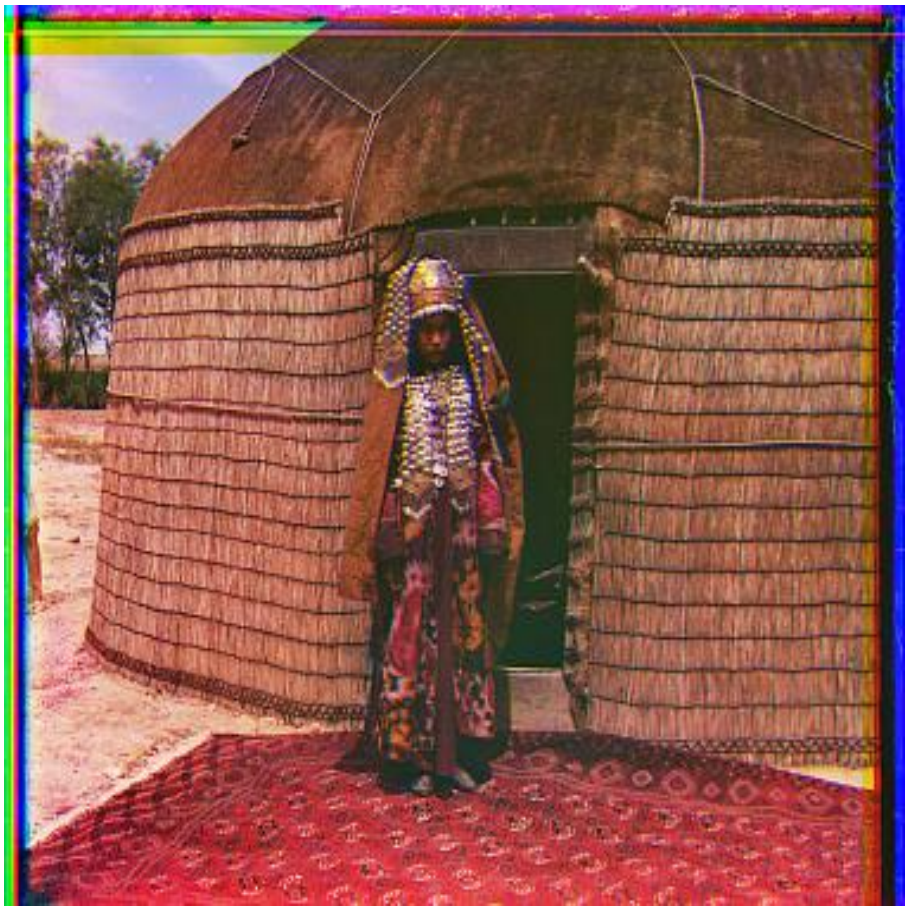
G shift: [49, 24], R shift: [107, 40]

## 4. Additional image

A few examples of my own choosing from (https://www.loc.gov/collections/prokudin-gorskii/?st=grid)

### Women in traditional dress - extensive search SSD

G shift: [3, 1], R shift: [8, 4]



## Tree - extensive search SSD

G shift: [5, 4], R shift: [11, 6]

# Embroidered Cloth - pyramid + sobel

G shift: [75, 9], R shift: [158, 16]