TED University

CMPE 362 Digital Image Processing

Assignment 1

Images of the Russian Empire: Colorizing the Prokudin-Gorskii photo collection

Prepared By;

Name: Kaan Üstündağ

Student ID: 16171099958

Overview

So, by using the 3 different color filtered photographs (red, green and blue filters) taken by Sergey Prokudin-Gorskii, we can align these images trio as a separate color channel to recreate a full-color picture. (RGB)

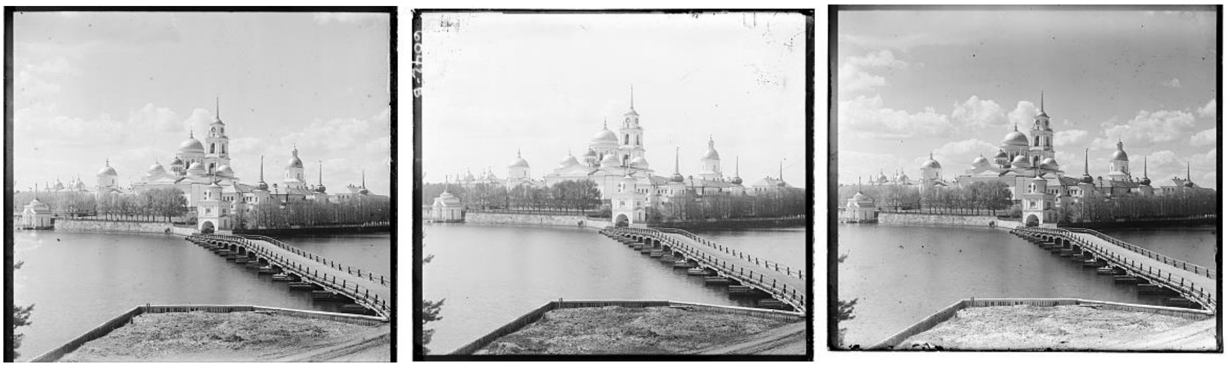
My approach to solve this homework with SSD, was to first crop 3 differently color filtered photos and then using sum of squared differences of two different color filtered photos. In my SSD solution, I aligned G and R channels, G and B channels together and after that I concatenated these two new channels and B channel to get the properly aligned photo. In my matlab code you may observe the functions that I used for these steps. (*align* and *SSD* functions). Finally, I produced full-colored version of the Prokudin-Gorskii photos.

Second approach to solve the problem was NCC (Normalized cross-correlation). In this approach, I again separated 3 channels from the input photograph by R, G and B. After that, I cropped image borders from the 3 separated channels, and then by using *normxcorr2* function and with the help of its documentation, I produced new 2 channel and aligned these two with the original blue channel that I gained at the first step. By the end, I produced full-colored version of the Prokudin-Gorskii photos.

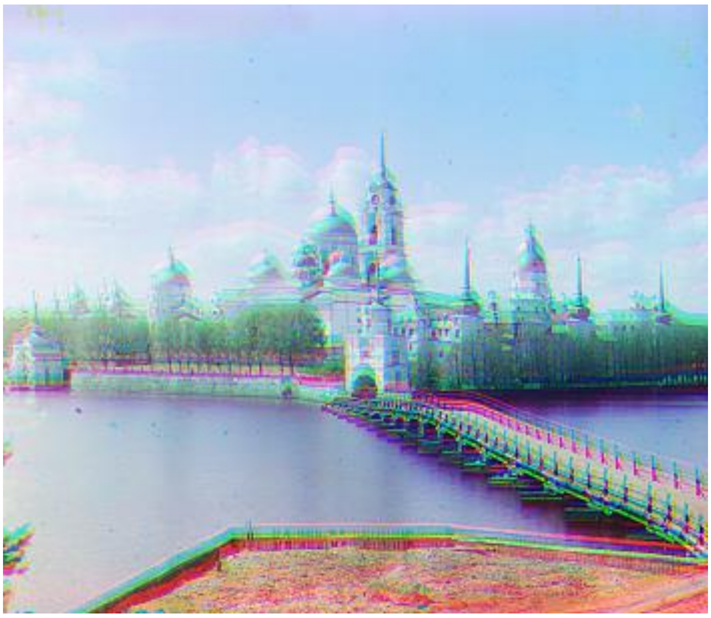
For both SSD and NCC approaches, I used gamma correction, histogram equalization and HSV color space transformations to improve the quality and contrast of the result images. (You may find the function that I created to improve the quality of the result images as *enhance* function in my matlab source code)

Steps of Producing Full-colored Image (SSD)

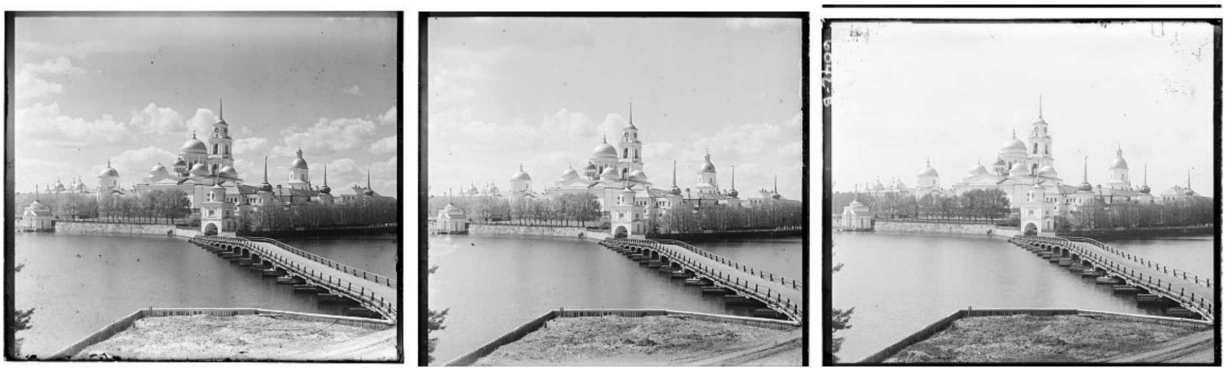
1. R, G and B channels separated from input image



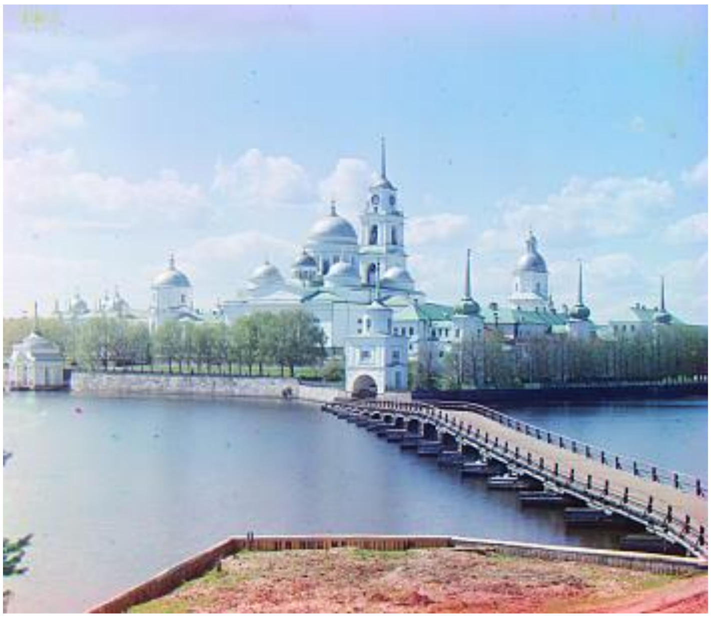
If I concatenated these three original channels directly, the result was like the below image.



2. Aligning the second and the third parts (G and R channels) to the first one (B channel).



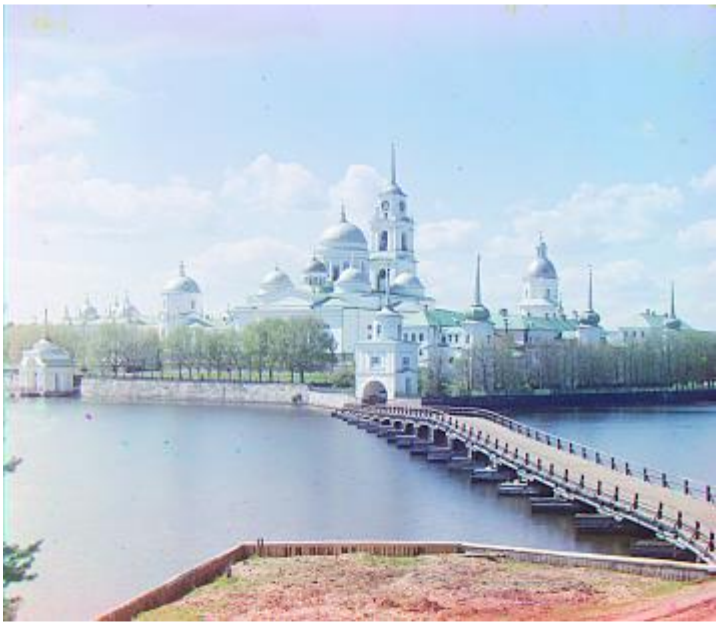
Using the above three channels (two of them are recreated and one of them is old G channel) first cropping these images and then by using alignment process with SSD rule, by the end of this step, after I concatenated these three channels and produced the below image.



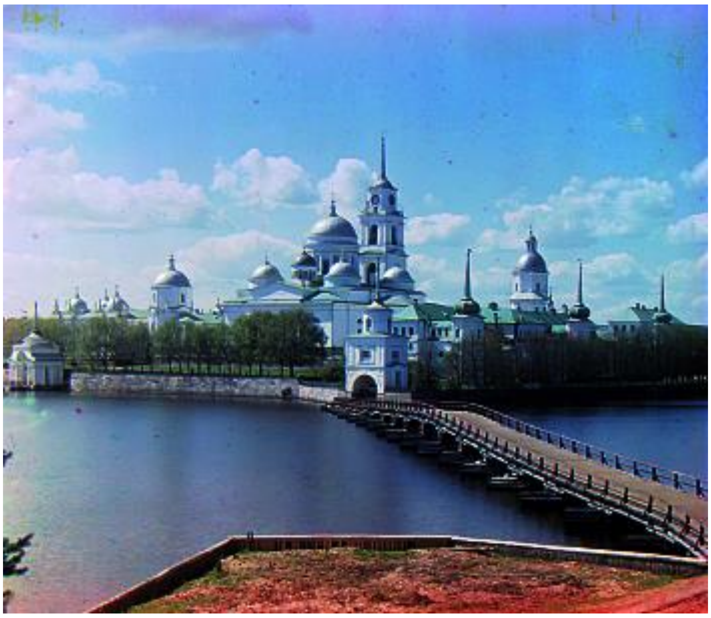
3. After producing a proper fully colored image with SSD, I started to improve the images' quality and the contrast. Firstly, I used gamma correction rule which is,

$$s = round(255 * (r/255)^{\gamma})$$

By applying gamma correction to the result image, I produced image below;



After this correction, I used histogram equalization to improve the quality of the image by applying *histeq()* function, and for the last, I used HSV color space to end my enhance step, I converted the RGB image to HSV format by using *rgb2hsv,* after applying a factor value to the vChannel I concatenated three channels again and gained a result like below, which seems much more vivid and satisfying image for me.;



- Best result for SSD (the most satisfactory result for me), NCC failed on this image;

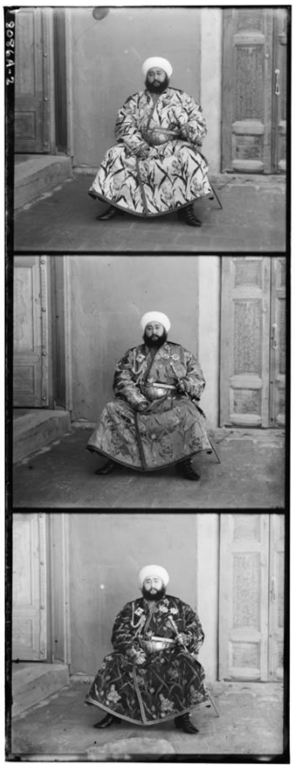


Figure 1: Input

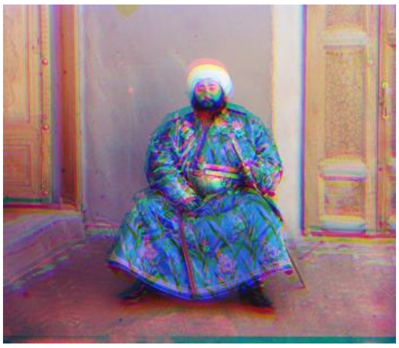


Figure 2: Before alignment

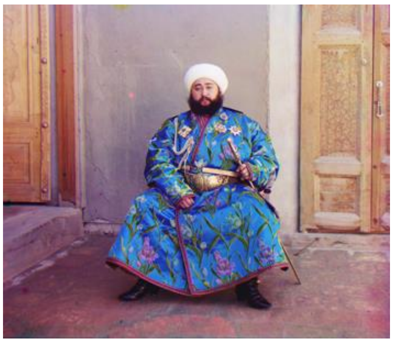


Figure 3: After alignment

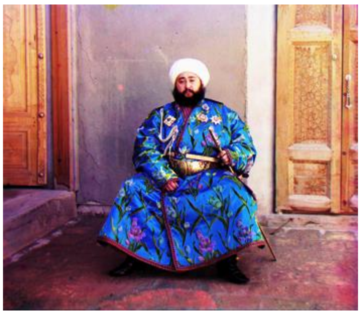


Figure 4: After Enhancing Quality

- Failed result for SSD (The worst output for me);

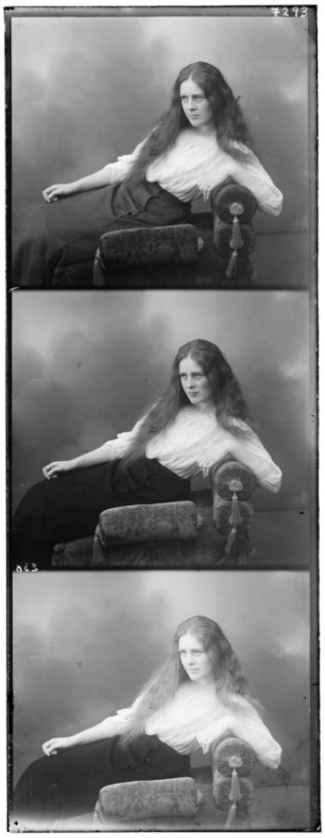


Figure 1: Input

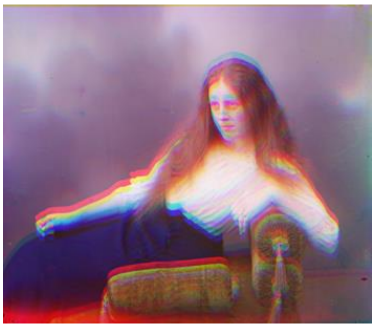


Figure 2: Before Alignment

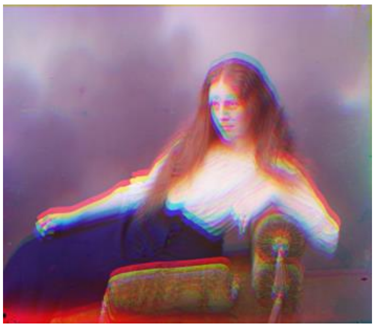


Figure 3: After Alignment

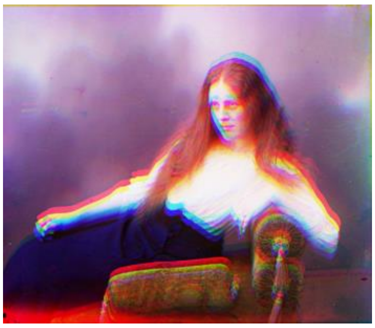
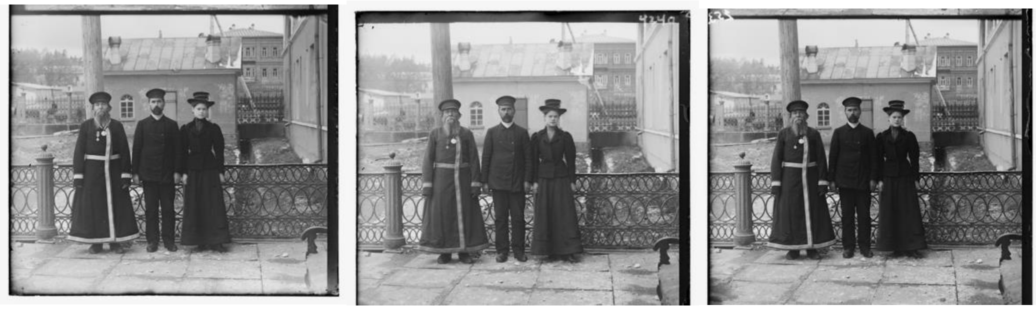


Figure 4: After Enhancing Quality

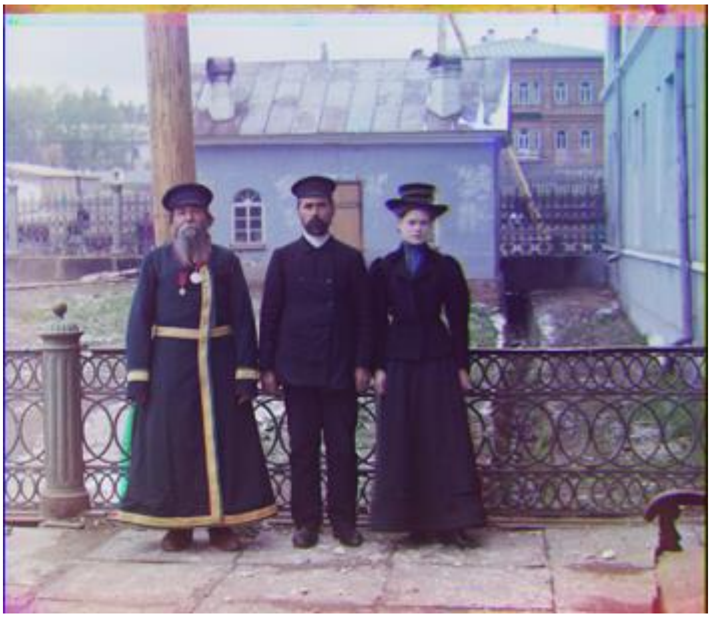
Steps of Producing Full-colored Image (NCC)

- Firstly, I cropped the input image into 3 separated images as R, G and B channels.



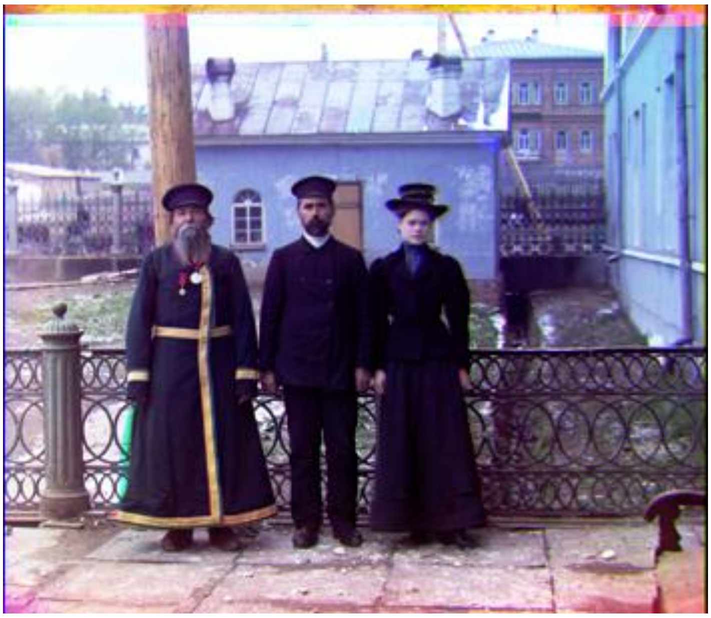
- Secondly, I applied *normxcorr2* function to G and B, R and B channels. After that I used *circshift* the result of the previous two outcomes to shift arrays circularly. I also cropped the borders from the images before alignment. So, I had 3 channels like below;

- After aligning these 3 channels, I had an image like below;



By the way, I choose this image to give details of my NCC solution, since this images' result was failed at SSD part and worked perfectly for NCC.

- I again enhanced the result image by applying the same steps as I mentioned in the SSD part below, and the outcome is shown below;



- Best result for NCC (the most satisfying result image for me), SSD failed on this image;

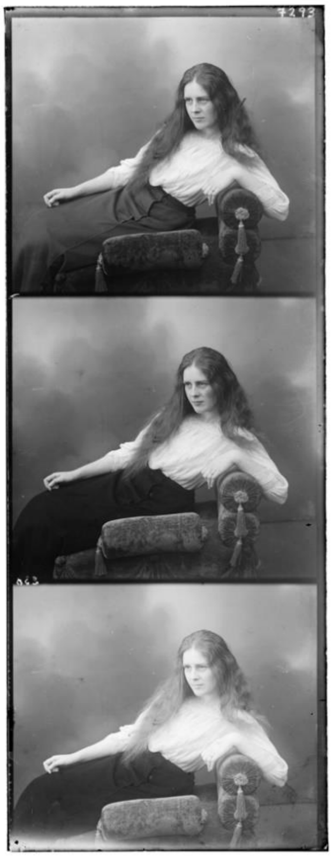


Figure 1: Input

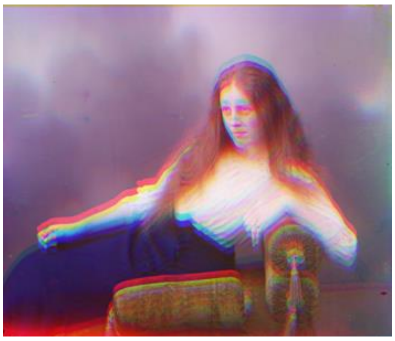


Figure 4: Before Alignment

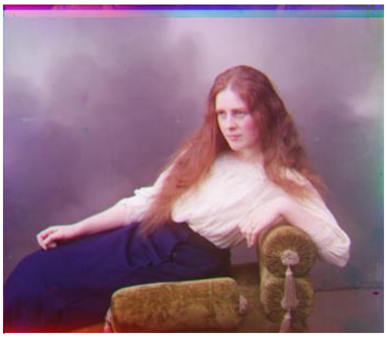


Figure 3: Result of Alignment

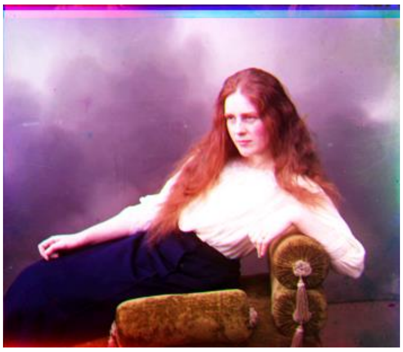


Figure 2: Result of Enhancements

- Failed result for NCC (The worst result image for me);

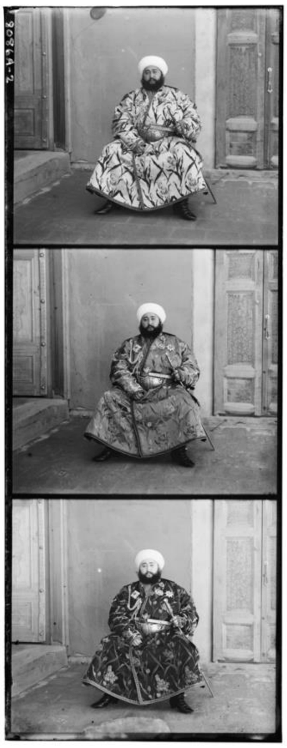


Figure 8: Input Image

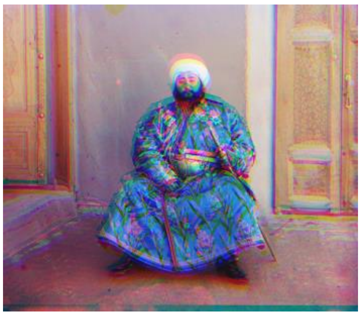


Figure 6: Before Alignment

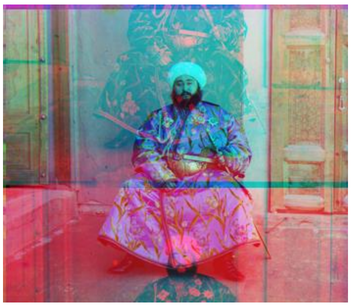


Figure 7: Result of the alignment

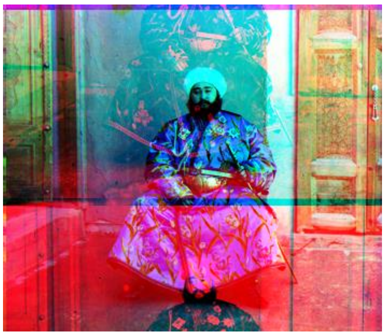


Figure 5: Result of enhancements

Conclusion

I used the Prokudin-Gorskii photo collection to reproduce them to have full-colored versions with using SSD and NCC approaches. For SSD solution, I had more satisfactory results and observed faster image process, however for two image inputs my SSD solution failed with the result (*lady.png* and *three_generations.png* you may see all the results within the zipped file*).* For NCC solution, I had a slower image process, but it only failed on one input image (*emir.png).* So, both algorithms generally worked fine, SSD had 2 failed results and NCC had 1 failed result, over 13 input images.

Important Note: In the zipped file, you may find all the results for 13 input images for both SSD and NCC ways. The name of the result images are in the format of;

- "….Before.png" -> Unaligned result
- "….ResultNormSSD.png" -> Aligned result for SSD
- "…EnhanceResultSSD.png" -> Aligned and enhanced result for SSD
- "…ResultNCC.png" -> Aligned result for NCC
- "…EnhanceResultNCC.png" ->Aligned and enhanced result for NCC