|  |  |
| --- | --- |
| [Mini Project 1: Colorizing ProkudinGorskii images of the Russian Empir] | Abstract  The goal of this assignment is to take RGB individual images and with some image processing techniques, to automatically produce a color image. In order to do this, we will need to extract the three-color channel images, place them on top of each other, and align them so that they form a single RGB color image.  Author:  Abdallah El-Sawy |

# **Contents:**

1. Background ……………………………………………………………………………….……………….2
2. Overview……………………………………………………………………………………………………. 2
3. Steps ……………………………………………………………………………….…………….…………...2
4. Bonus points…………………………………………………………………….………………………… 3
5. Instruction’s Implementation……………………………………………….………………………3

5.1 Extract R, G and B channels …………………………………………………………………….….………... 3

5.2 Mutual alignment between channels ……………………………………………………….…………….7

5.3.1 The sum of squared differences (SSD) …………………………………………….………………….10

5.3.2 Improved SSD ………………………………………….…………………………………….…………………….14

5.4 the normalized cross-correlation (NCC) ………………………………………..…………………………14

5.5 [BONUS POINT] pyramid approach ……………………………………………………….…………………18

6. conclusion…………………………………………………………………………………………….……….18

7. References…………………………………………………………………………………………..…….…. 24

**Mini Project 1: Colorizing ProkudinGorskii images of the Russian Empire**

1. **Background:**

[Sergei Mikhailovich Prokudin-Gorskii](http://en.wikipedia.org/wiki/Prokudin-Gorskii) (1863-1944) was a photographer who, between the years 1909-1915, traveled the Russian empire and took thousands of photos of everything he saw. He used an early color technology that involved recording three exposures of every scene onto a glass plate using a red, green, and blue filter. Back then, there was no way to print such photos, and they had to be displayed using a special projector. Prokudin-Gorskii left Russia in 1918. His glass plate negatives survived and were purchased by the Library of Congress in 1948. Today, a digitized version of the Prokudin-Gorskii collection is [available online](http://www.loc.gov/exhibits/empire/gorskii.html).

1. **Overview:**

The goal of this mini project is to learn to work with images in MATLAB by taking the digitized Prokudin-Gorskii glass plate images and automatically producing a color image with as few visual artifacts as possible. In order to do this, you will need to extract the three-color channel images, place them on top of each other, and align them so that they form a single RGB color image.

1. **Steps:**
2. Divide the image into three equal parts (channels).
3. Aligning two of the channels to the third (try different orders). display the colorized output and report the (x,y) displacement vector that was used to align the channels.
4. SSD: Score each one using some image matching metric, and take the displacement with the best score. Use the sum of squared differences (SSD) to score how well the images match.
5. Improved SSD method.
6. NCC: Using the normalized cross-correlation (NCC) to over come the problem of brightness.
7. **Bonus Points:**
8. Multiscale alignment: use an image pyramid to implement a faster search procedure to search over all possible displacements will become prohibitively expensive.
9. Additional ideas may improve the quality of the colorized images such dealing with the borders of the photograph.
10. **Instruction’s implementation:**
11. First, we should extract each of R, G, B channels for every given photo like the following:





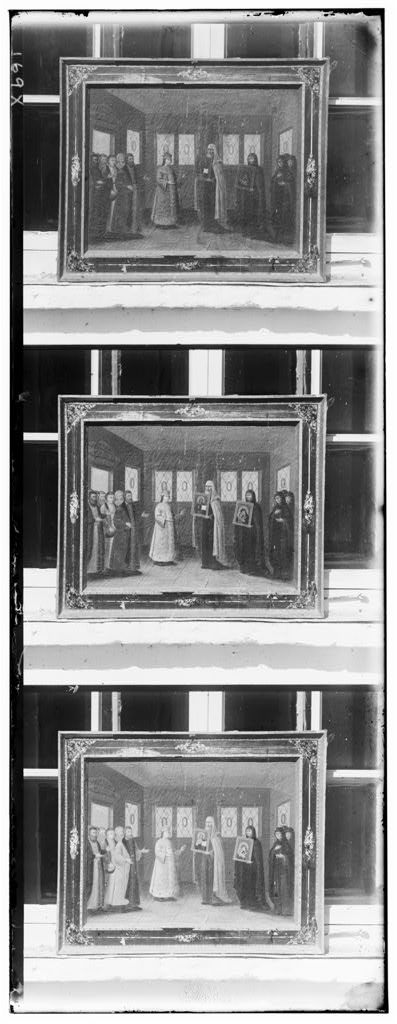
**R channel**

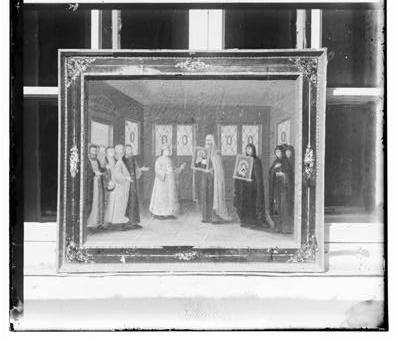
**B channel**

**G channel**





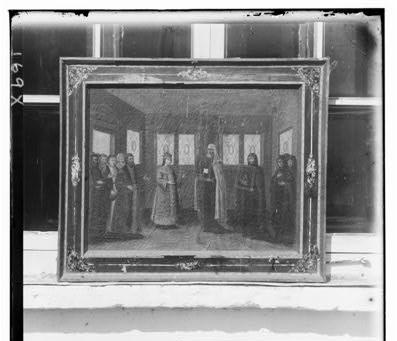


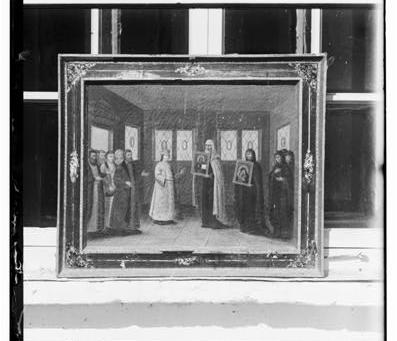


**R channel**

**B channel**

**G channel**









**R channel**

**B channel**

**G channel**









**R channel**

**B channel**

**G channel**









**R channel**

**B channel**

**G channel**









**R channel**

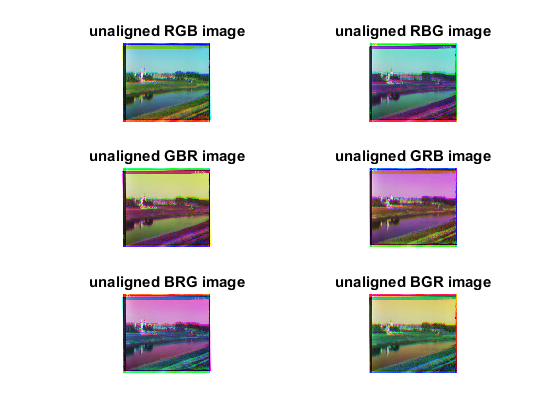


**B channel**

**G channel**



1. aligning two of the channels to the third channel which will be the reference. And produce the following combinations:













1. SSD: Using the sum of squared differences (SSD) to score how well the images match. we will take every time one of the channels for the background and one of other channels to be the foreground and applying the function 'align\_Img’ on every combination of them to find the best aligned images.

* 'align\_Img’ function definition:

Input Arguments:

* foreground channel: which is one of the three RGB channels.
* background channel: is another one of the three RGB channels.
* r: is the rotation window size used to search over certain channel.
* p: is the cropping border Percentage.
* using the function 'SSD=sum(sum(image1-image2). ^2)' after cropping the background channel then trying to shift the foreground channel from [-r:r -r;r] and after every shift we crop this channel then checking the SSD function and take the best of them.
* It will out best align to two of the three channels to the remain channel as base like the following:

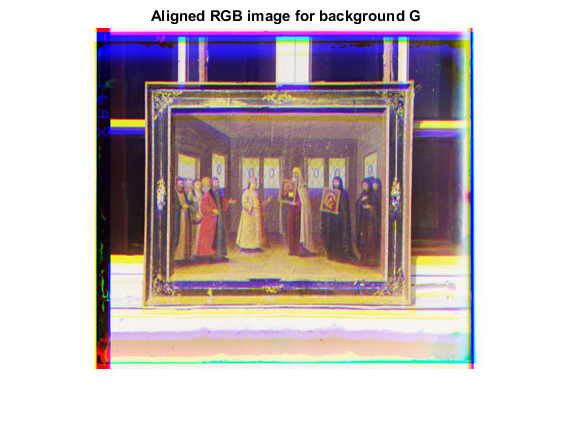
r = 4

p = 0.09

*For 00125v.jpg the best alignment is which have the background channel R*



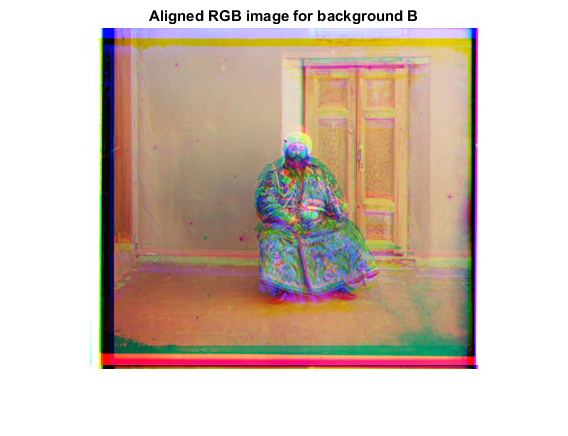
*For 00149v.jpg it seems that all channels in background have the same quality.*



*For 00351v.jpg the best alignment is which have the background channel B*



*For 00153v.jpg the best alignment is which have the background channel B*



*For 00398v.jpg the best alignment is which have the background channel B*

*For 01112v.jpg the best alignment is which have the background channel R*





1. Improved SSD method: when we see the previous results of the SSD method, we find there is unneeded border which effect on our results so we should crop this channel with some ratio and redo the SSD method on the new channels.

(New cropping ratio: p=0.15)

* The results of the improved SSD method with best background channel are here:

*For 00125v.jpg the best improved SSD is which have the background channel R*





*For 00153v.jpg the best improved SSD is which have the background channel R*



*For 00149v.jpg the best improved SSD is which have the background channel G*



*For 00398v.jpg the best improved SSD is which have the background channel R*

*For 00351v.jpg the best improved SSD is which have the background channel B*



*For 01112v.jpg the best improved SSD is which have the background channel R*



1. NCC: Using the normalized cross-correlation (NCC) to overcome the problem of brightness. this function manages us to find the best alignment of one of the channels respects to other we realize that the image did not have the same brightness so to match them we should use matric like normalized cross-correlation (NCC), which is simply the dot product between the two images normalized to have zero mean and unit norm.

* NCC function definition:
* Input Arguments:
* best\_alignment: is one of the three RGB channels.
* reference: is the channel that we do the alignment respect to it

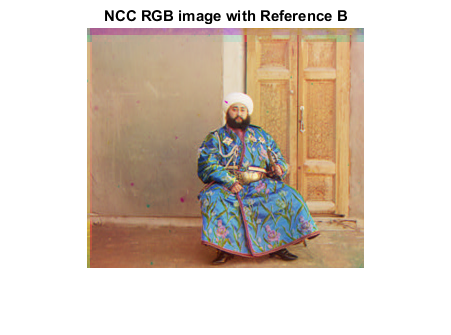
normalized cross-correlation (NCC) is simply the dot product between the two normalized images to have zero mean and unit norm the NCC give us the ability to match the difference in the brightness in each channel this function manages us to find the best alignment of one of the channels respects to other by finding the maximum correlation value.

* It will give those results:

r = 15

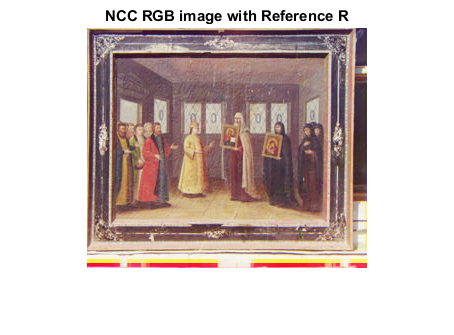


*The best NCC 00125v.jpg when G is the reference and R, B is the alignment channels*



*The best NCC 00153v.jpg when B is the reference and R, G is the alignment channels*

*The best NCC 00149v.jpg when R is the reference and B, G is the alignment channels*





*The best NCC 00398v.jpg when B is the reference and R, G is the alignment channels*

*The best NCC 00351v.jpg when R is the reference and B, G is the alignment channels*



*The best NCC 00398v.jpg when G is the reference and R, B is the alignment channels*



*The best NCC 01112v.jpg when B is the reference and R, G is the alignment channels*

1. [BONUS POINT] pyramid approach: is a multiscale alignment by using an image pyramid to implement a faster search procedure such as an image pyramid represents the image at multiple scales (usually scaled by a factor of 2) and the processing is done sequentially starting from the coarsest scale (smallest image) and going down the pyramid, updating our estimate as we go.

* Pyramid function definition:

Input Arguments:

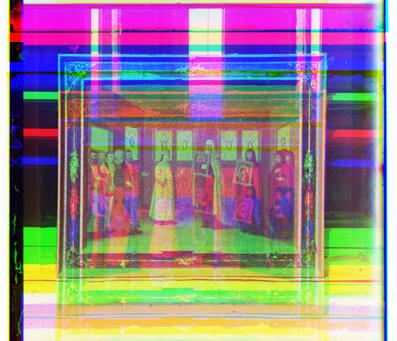
* x\_channel: is one of the three RGB channels.
* reference: is the channel that we do the alignment respect to it.
* It will out the following results:

*All pyramids output for 00153v.jpg*



*All pyramids output for 00149v.jpg*

*All pyramids output for 00125v.jpg*





*All pyramids output for 01112v.jpg*



*All pyramids output for 00398v.jpg*



*All pyramids output for 00351v.jpg*



1. **Conclusion:**

From the previous results which we obtained it from make a combination of the three channels, using the alignment through using the sum of squared differences (SSD) the normalized cross-correlation (NCC) or using the pyramid approach we find that no certain method give us best quality for all the Prokudin-Gorskii glass plate images but every method have its cons and pros in certain area such as using the sum of squared differences (SSD) on the image( 00125v.gif) it gives results close to the results which obtained from the normalized cross-correlation (NCC) but in the pyramid the results have less quality then the two methods. So, it is up to you to choose whose is the best depend on your input images.

1. **References:**
2. <http://6.869.csail.mit.edu/fa19/lectures/notes_lecture_7.pdf>
3. <https://learn.lboro.ac.uk/archive/olmp/olmp_resources/pages/workbooks_1_50_jan2008/Workbook30/30_4_mtrx_norms.pdf>
4. <https://inst.eecs.berkeley.edu/~cs194-26/fa17/upload/files/proj1/cs194-26-abq/>