University at Buffalo

Computer Science and Engineering

CSE: 468/568 : Robotics Algorithm

Lab3 : Colorizing the Prokudin – Gorskii photo Collection

By: Akhil Goyal (50415687)

Lab Overview:

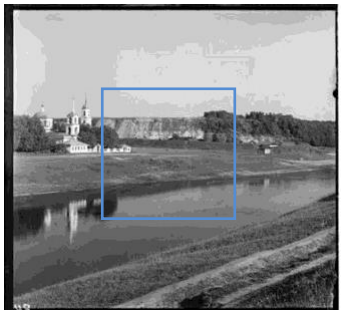
- Extract color filter form the digitized Prokudin-Gorskii glass image.
- Align and save using different algorithms.

Alignment method:

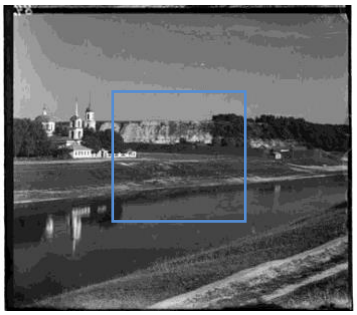
1. Sum of squared differences(SSD): Alignment involve creating a reference window and comparing its elements with the target image by calculating the sum square of the difference (sum((image1 – image2)^2)).

The method of comparing in this lab is as follows:

1. Create a reference window of size s say 101 from the center of the first image.

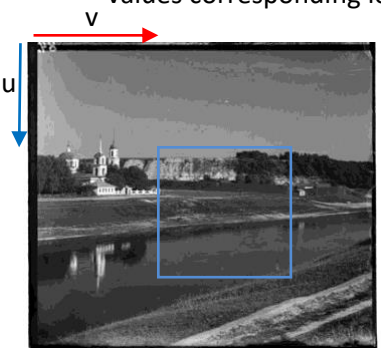


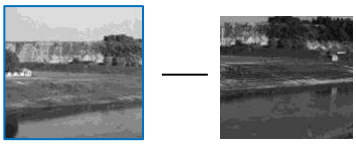
2. Create another window of same size and location after shifting it by some pixels.



3. Compare the two images by taking difference and calculate sum of the square of the difference.

4. Shift the second image again repeating 2 and 3 step until the least difference is found. Shift values corresponding least SSD value is the best match.





5. Used the shift value for the best match to generate colored image.

2. Normalized cross – correlation (NCC): NCC algorithm uses same concept as SSD but uses correlation coefficient for comparing the images. The formula for the correlation coefficient is given below. Similar images have higher correlation coefficient, thus the objective will be shift value for highest coefficient.

$$\rho(A,B) = \frac{1}{N-1}\sum_{i=1}^{N}\left(\frac{A_i-\mu_A}{\sigma_A}\right)\left(\frac{B_i-\mu_B}{\sigma_B}\right),$$

3. Feature based alignment(using corners):

Robert Collins
CSE486, Penn St

Harris Corner Detection Algorithm

1. Compute x and y derivatives of image
$$I_x = G_\sigma^x * I \quad I_y = G_\sigma^y * I$$
2. Compute products of derivatives at every pixel
$$I_{x2} = I_x.I_x \quad I_{y2} = I_y.I_y \quad I_{xy} = I_x.I_y$$
3. Compute the sums of the products of derivatives at each pixel
$$S_{x2} = G_{\sigma\prime} * I_{x2} \quad S_{y2} = G_{\sigma\prime} * I_{y2} \quad S_{xy} = G_{\sigma\prime} * I_{xy}$$
4. Define at each pixel (x, y) the matrix
$$H(x,y) = \begin{bmatrix} S_{x2}(x,y) & S_{xy}(x,y) \\ S_{xy}(x,y) & S_{y2}(x,y) \end{bmatrix}$$
5. Compute the response of the detector at each pixel
$$R = Det(H) - k(Trace(H))^2$$
6. Threshold on value of R. Compute nonmax suppression.

- Used sobel filter for x and y derivative.

```
sobel_y= [1,2,1;0 0 0;-1 -2 -1]/8;
sobel_x=[-1,0,1;-2,0,2;-1,0,1]/8;
```

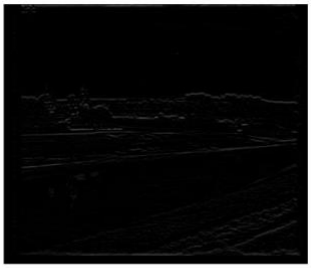
- Size of window used for sum of the product of derivatives is 3x3.
- Sorted the R matrix in descending order and extracted value for Nth from top for threshold.
- Used the threshold for corner detection(R(u,v) > Threshold), pixels having value more than the threshold are corners.
- Used ransac on corner index to complute shift.

Ransac:

1. Extract corner indices using harries corner detection of two images (set 1 and set 2).

2. Random select indices from the two data set.
3. Assume the difference of the selected indices be the shift of the image.
4. Add this shift to the second data set and count the matches with first data set.
5. Repeat step 2 to 4 for maximum count and Select the shift value with the maximum count

Ix

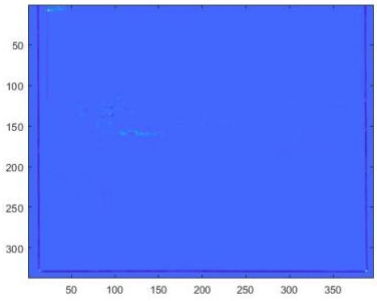


Iy

Ix^2

Iy^2

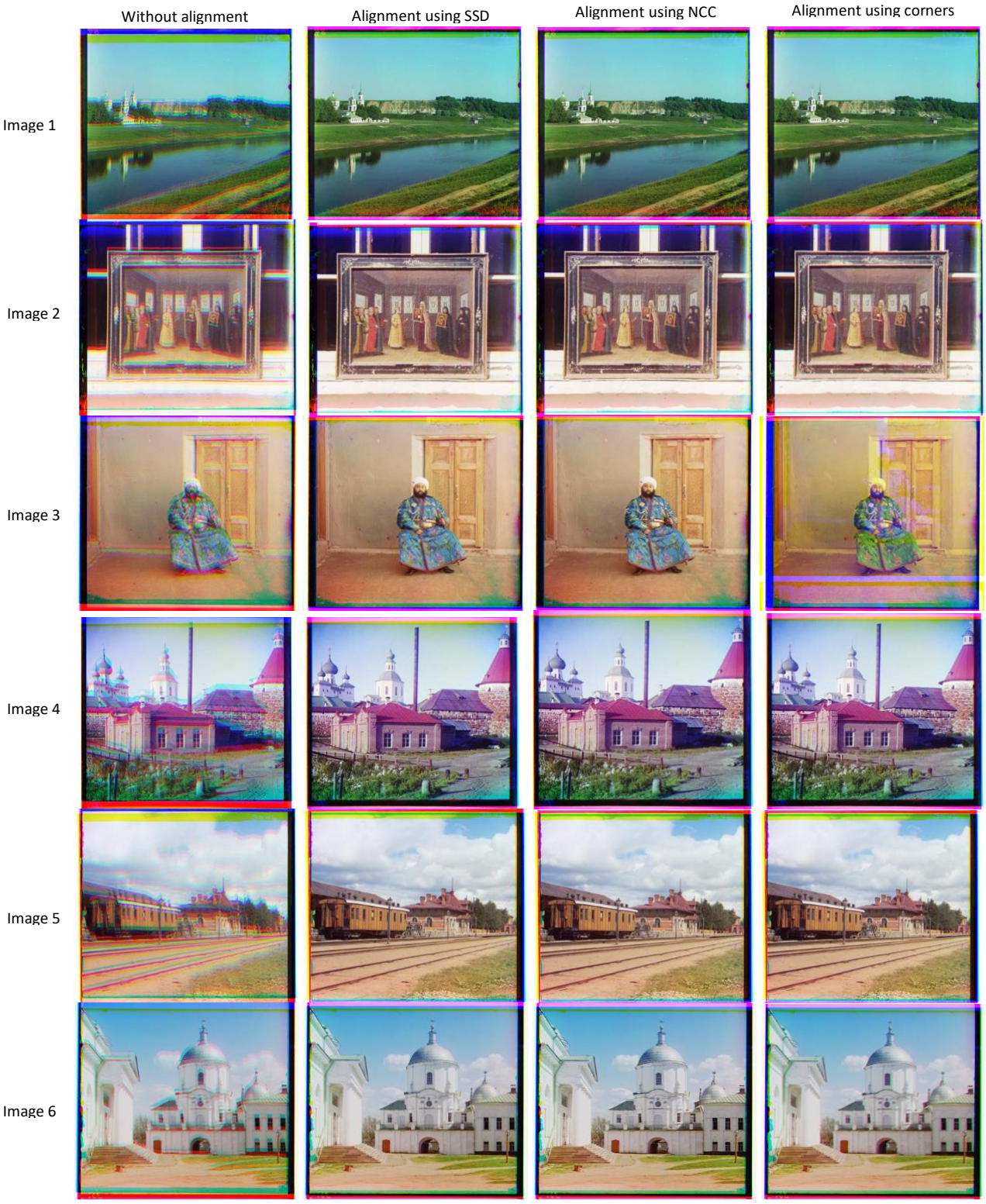
Ixy



R

Result from harries corner detection:

Result:


Without alignment
Alignment using SSD
Alignment using NCC
Alignment using corners
Image 1
Image 2
Image 3
Image 4
Image 5
Image 6

		SSD	NCC	Corners
Image 1	rShift	[5 , -1]	[5 , -1]	[5 , -1]
	gShift	[0 , 0]	[0 , 0]	[0 , 0]
	bShift	[-5 , -2]	[-5 , -2]	[-5 , -2]
Image 2	rShift	[6 , 0]	[6 , 0]	[7 , 0]
	gShift	[0 , 0]	[0 , 0]	[0 , 0]
	bShift	[-4 , -2]	[-4 , -2]	[-4 , -1]
Image 3	rShift	[8 , 2]	[8 , 2]	[8 , 2]
	gShift	[0 , 0]	[0 , 0]	[0 , 0]
	bShift	[-7 , -3]	[-7 , -3]	[-61 , 11]
Image 4	rShift	[10 , 1]	[10 , 1]	[10 , 1]
	gShift	[0 , 0]	[0 , 0]	[0 , 0]
	bShift	[-4 , -1]	[-4 , -1]	[-4 , -1]
Image 5	rShift	[7 , 1]	[7 , 1]	[7 , 1]
	gShift	[0 , 0]	[0 , 0]	[0 , 0]
	bShift	[-5 , -2]	[-5 , -2]	[-5 , -2]
Image 5	rShift	[6 , 1]	[6 , 1]	[6 , 2]
	gShift	[0 , 0]	[0 , 0]	[0 , -1]
	bShift	[0 , 0]	[0 , 0]	[0 , 0]

References:

- https://www.mathworks.com/help/matlab/ref/corrcoef.html
- http://www.cse.psu.edu/~rtc12/CSE486/lecture06.pdf