CSE 568: ROBOTICS ALGORITHMS

(FALL 2018)

Dr. Kartik Dantu

University at Buffalo

LAB 3 REPORT

Submitted by-

ANIRUDDHA SINHA (UB Person No - 50289428)

Date: October 24, 2018

Aim

Implement some basic image processing algorithms to detect features to frame alignment using Octave / MATLAB.

Task Overview

The task given to us is to colorize the **Prokudin-Gorskii photo collection***.

Prokudin-Gorskii was a photographer, who during the early twentieth century used an early colour technology that involved recording three exposures of every scene onto a glass plate using a red, green, and blue filter, which created separate images of three different channels, red, green and blue. These channels were stacked together and displayed using a special projector to show a coloured image.

The objective of the current task is to extract the three colour channel images from the given set of images, place them on top of each other and align them so that they form a single RGB colour image. We have mainly brought to use three methods in image processing to achieve the task, namely, **Sum of Squared Differences (SSD), Normalized Cross Correlation (NCC) and Harris Corner Detection.** All these methods achieve the objective by employing the concept of template matching and feature detection, where we try to find the similarity between two or more pixels from the channel images and perform alignment accordingly.

1. Simple Colour Images

The input set of images provided to us consists of six images, each of which is a concatenation of three separate glass plate images. Each glass plate image represents a channelled image, R, G and B. Presented below are the results of the **unaligned colour image,** where for each image, the Red and Green channels have simply been placed above the Blue channel, without any alignment technique.

Fig1.1. Unaligned colour image -1

* (Adapted from Radhakrishna Dasari with permission, although originally designed by Aloysha Efros at CMU)

Fig1.2. Unaligned colour image -2

Fig1.3. Unaligned colour image -3

Fig1.4. Unaligned colour image -4

Fig1.5. Unaligned colour image -5

Fig1.6. Unaligned colour image -6

2. Alignment of images

As is evident from the images above, the channels aren't aligned with each other properly. In order to align them together, we have used the following methods –

- Sum of Squared Differences (SSD)
- Normalized Cross Correlation (NCC)
- Harris Corner Detection and Feature Matching using RANSAC

2.1 Method 1 - Sum of Squared Differences (SSD):

The Sum of Squared Differences is one measure of match that is based on pixel by pixel intensity differences between the two images. It calculates the summation of squared for the product of pixels subtraction between two images. With this similarity measure, matching point can be determined by considering the location of minimum value in the image matrices. Generally, SSD is directly the formulation of sum of square error, and is given as –

$$\boldsymbol{h[m,n]} = \sum_{\boldsymbol{k,l=-N}}^{\boldsymbol{N}} (\boldsymbol{g[k,l]} - \boldsymbol{f[m+k,n+l]})^{\boldsymbol{2}},$$

where, $h[m,n]$ = SSD pixel image
$g[k,l]$ = pixel location of the template, g (NxN dimensions)
$f[m+k,n+l]$ = pixel location of the main image to be compared

Process used:

For this task, the SSD has been performed using the following process on each of the images:

1. First, we crop out a portion of the reference channel image (Blue) and the channel image that needs to be aligned (Red and Green). I am using a 20% cropped out edges from both channels (reference and comparison).

2. Second, we take a window of [-15, 15] pixels and perform circular shift over the to-be-aligned cropped channel image using the updates in the window for each iteration.
3. The next step is to perform SSD over the circularly shifted channel image and the cropped out portion of the reference channel. The **SSD values are compared** for every shift and **the one with the minimum value is considered best**, and the corresponding value of the current iterations in the window are considered the shift by which the channel needs to be aligned.
4. Finally, we align both Red and Green channels by their corresponding best shift values and place them over the original Blue channel. This is the **aligned colour image.**

   Shown below are the SSD aligned images of the images in the given set, with a measure of their corresponding displacement vectors.

Fig2.1.1. Aligned image-1 by SSD

SSD Displacement vector for image-1
Red = [4, -1]
Green = [-5, -2]

Fig2.1.2. Aligned image-2 by SSD

SSD Displacement vector for image-2
Red = [5, 0]
Green = [-4, -2]

Fig2.1.3. Aligned image-3 by SSD

SSD Displacement vector for image-3
Red = [7, 2]
Green = [-7, -3]

Fig2.1.4. Aligned image-4 by SSD

SSD Displacement vector for image-4
Red = [9, 1]
Green = [-4, -1]

Fig2.1.5. Aligned image-5 by SSD

SSD Displacement vector for image-5
Red = [6, 1]
Green = [-5, -3]

Fig2.1.6. Aligned image-6 by SSD

SSD Displacement vector for image-6
Red = [5, 1]
Green = [0, 0]

2.2 Method 2 – Normalized Cross Correlation (NCC):

In signal processing, cross-correlation is a measure of similarity of two series as a function of the displacement of one relative to the other. Also known as dot product or inner product, correlation is commonly used for searching a long signal for a shorter, known feature.

The cross-correlation is used for referring to the correlations between the entries of two random vectors X and Y to find a similarity measure. However, cross correlation is affected by the magnitude of intensity values. The solution to is to normalize all the values, and hence we use Normalized Cross Correlation, given by the following formula –

$$\boldsymbol{h[i]} = \frac{\sum_{i=-N}^{N}(\boldsymbol{F(i)I(x+i)})}{\sqrt{\sum_{i=-N}^{N}(\boldsymbol{I(x+i)})^2}\sqrt{\sum_{i=-N}^{N}(\boldsymbol{F(i)})^2}},$$

where, $F(i)$ = template pixel of template F (NxN dimensions)
$I(x+i)$ = image pixel

Shown next are the results of aligned images by NCC on the given set of images.

Process used:

The approach of Normalized Cross Correlation is different than Sum of Squared Differences and the common process employed for every image is described below:

1. Firstly, in this case, the main idea is that the entire channels have been chosen as templates for comparison and eventual alignment.

2. Second, we perform edge detection on each of the channels using `Canny Edge detection`. The need to find an Edge Detected image is to retain only the important details or features of the image, and lose the unnecessary information.
3. Next, after getting the edge detection channels, we try to find the shifts in the Red and Green channels w.r.t the Blue channel, using **`norxcorr2`.**
4. The **`norxcorr2`** function returns a matrix of normalized cross correlated coefficients between the template and the image which in this case are the to-be-aligned channel (Red and Green) and the reference channel (Blue) respectively. A **good similarity measure for NCC is represented by the coefficient with the maximum value.** Thus, we find the maximum in the entire **`norxcorr2`** coefficient matrix and the corresponding peak values and its indices.
5. The indices thus found represent the shift in each of the channels, Red and Green.
6. We align both channels and place them above the Blue channel. This is our **final aligned image using NCC**.

Shown below are the results of the aligned images using NCC for each of the image in the input set. Also given are the respective calculated displacement vectors for both Red and Green channels in all images.

Fig2.2.1. Aligned image-1 by NCC

NCC Displacement vector for image-1
Red = [7, 4]
Green = [-6, -2]

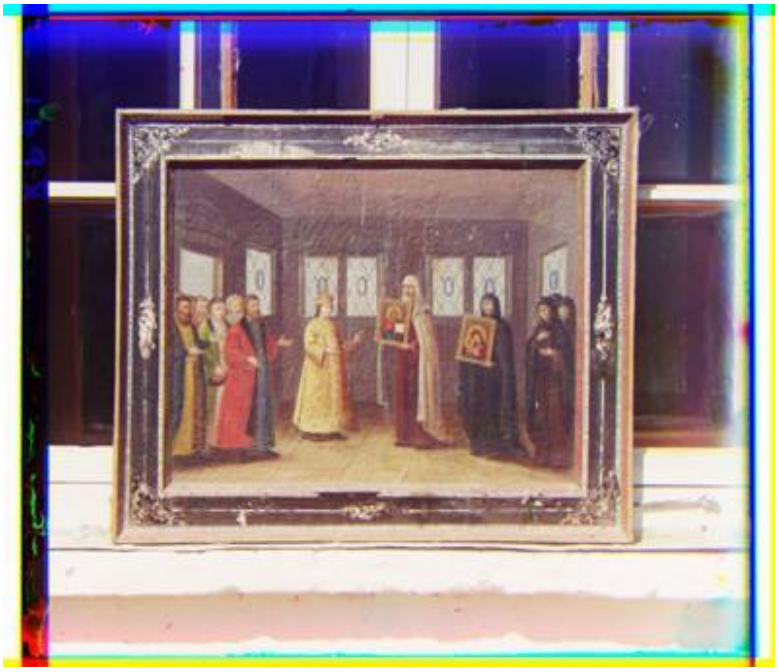


Fig2.2.2. Aligned image-2 by NCC

NCC Displacement vector for image-2
Red = [5, 3]
Green = [-4, -2]

Fig2.2.3. Aligned image-3 by NCC

NCC Displacement vector for image-3
Red = [8, 4]
Green = [-7, -3]

Fig2.2.4. Aligned image-4 by NCC

NCC Displacement vector for image-4
Red = [5, 1]
Green = [-4, 0]

Fig2.2.5. Aligned image-5 by NCC

NCC Displacement vector for image-5
Red = [6, 4]
Green = [-5, -3]

Fig2.2.6. Aligned image-6 by NCC

NCC Displacement vector for image-6
Red = [7, 2]
Green = [0, 0]

2.3 Method 3 – Harris Corner Detection and Feature Matching using RANSAC:

In image processing, a pixel of an image is considered a corner only if there are intensity changes in both the x and y directions, and not only in one direction. To calculate the cornerness of a pixel, we use the following formula –

$$\text{Cornerness, } \boldsymbol{C = \det(M) - \kappa.trace(M)^2},$$

where, M = auto-correlation matrix for every pixel
κ = sensitivity parameter determined empirically, usually between 0.04 and 0.15

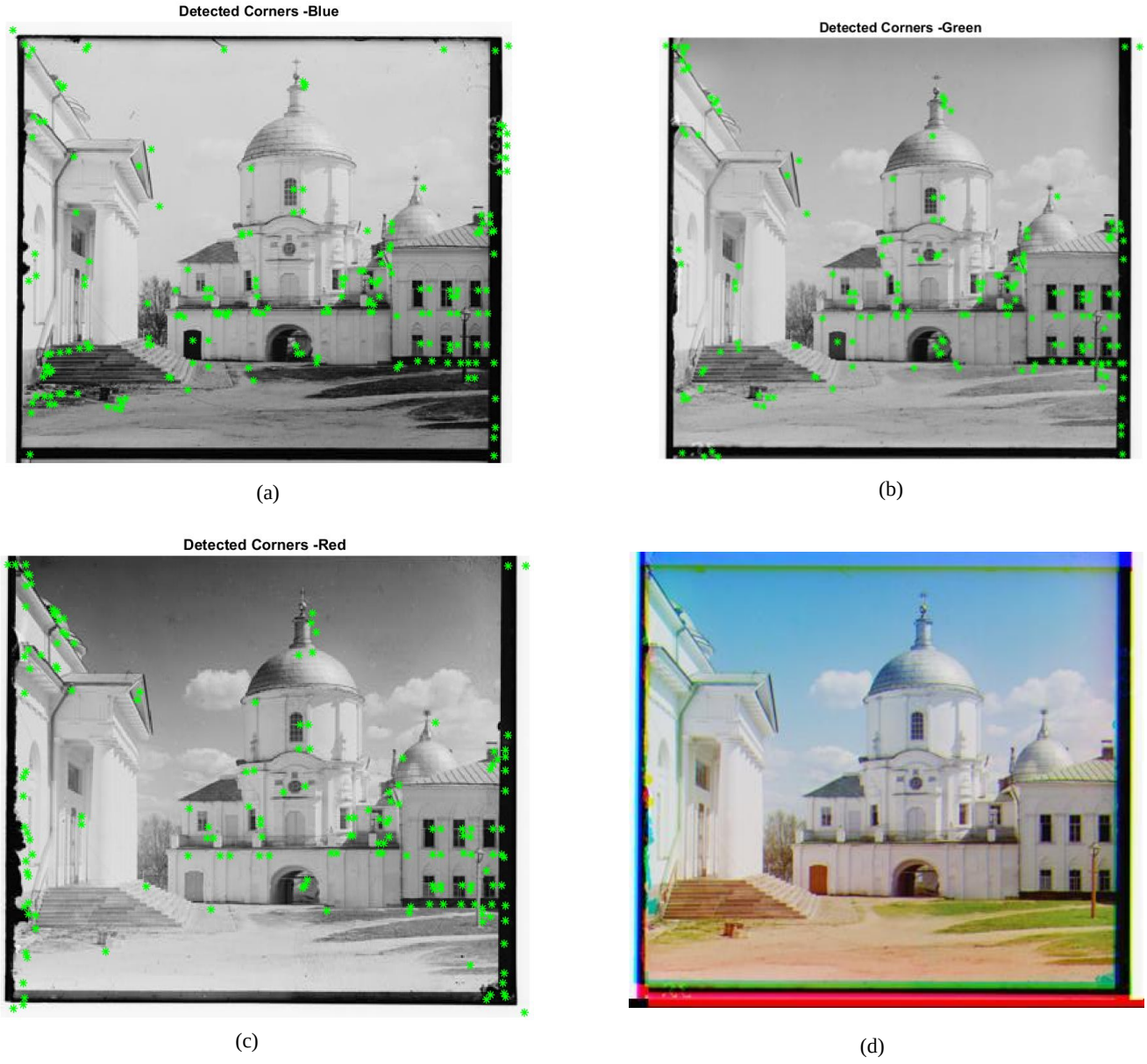
We thus extract the local maxima from the cornerness function.

Process used:

1. First, we begin with finding the corners in the image. For that, we first create the masks in x and y direction and a Gaussian filter of size 7x7.
2. Next, we filter the image in x and y direction using the Gaussian filter and calculate the gradients in both directions.
3. Then, the squares of the gradients in either directions are calculated and the joined gradients in both directions. Thereafter, all these entities are again filtered using the x and y-direction masks and the Gaussian filter.
4. Now, we begin searching for Harris corners in the image. The **threshold is set equal to $2x10^9$** and the sensitivity parameter, $\boldsymbol{\kappa = 0.08}$**.** We find the auto-correlation matrix for every pixel and calculate the cornerness value using the above formula. Subsequently, **we check if the cornerness value is greater than the threshold. If True, this pixel is a corner**, and we save the pixel location.
5. Next, using the concept of *non-max suppression*, we detect every feature / corner only once.
6. Display the corners on the image.

7. The next steps involve using RANSAC algorithm for channel alignment.
8. We select a feature at random in the blue, red and green images.
9. Then, find the offset or pixel shift for a random feature in blue v/s green channels and blue v/s red, assuming they match.
10. Shift all other features in the image w.r.t the new shifts calculated.
11. The inlier count is updated by searching for corresponding feature matching in both channels within a window.
12. Finally, we calculate the best alignment for the red and green channels w.r.t the blue channel.

Shown below are the results of the Harris corner detection and image alignment for image-6.jpg.



(a) (b)

(c) (d)

Fig2.3.1. (a) Corners on Blue channel image, (b) Corners on Green channel image, (c) Corners on Red channel image, (d)Aligned Image by Corner Detection

Results

Table 1 lists all the required displacement vectors for both NCC and SSD methods of image alignment.

Image	SSD Displacement Vector				NCC Displacement Vector			
	Red Channel		Green Channel		Red Channel		Green Channel	
	Y-axis	X-axis	Y-axis	X-axis	Y-axis	X-axis	Y-axis	X-axis
image-1	4	-1	-5	-2	7	4	-6	-2
image-2	5	0	-4	-2	5	3	-4	-2
image-3	7	2	-7	-3	8	4	-7	-3
image-4	9	1	-4	-1	5	1	-4	0
image-5	6	1	-5	-3	6	4	-5	-3
image-6	5	1	0	0	7	2	0	0

Table 1. Displacement vectors of green and red channels for each image using SSD and NCC

Table 2 lists all the displacement vector for Red and Green channels calculated using Corner Detection.

Image	Harris Corner Detection Displacement Vector			
	Red Channel		Green Channel	
	Y-axis	X-axis	Y-axis	X-axis
image-1	-11	0	-7	-1
image-2	-10	-1	-5	-1
image-3	-15	-3	-8	-2
image-4	-14	0	-5	1
image-5	-12	-3	-6	-2
image-6	-6	0	-1	1

Table 2. Displacement vectors of green and red channels for each image after corner detection.

Conclusion

From the implementation of this assignment, we get to see how the metrics, Sum of Squared Differences (SSD) and Normalized Cross Correlation (NCC) can be used for feature matching and check the similarity between any two images. Another approach brought to use was Harris Corner Detection and RANSAC for feature matching which is quite useful, however sub-optimal as many times it does not always give accurate results like NCC or SSD do.

From my observation, NCC seems to work better than SSD and Harris Corner detection for the given set of images. The difference in displacement vectors of all three techniques if feeble, yet presents visually inaccurate results sometimes.

References

- Mohamad, Badrul & Yaakob, Shahrul & A. Raof, Rafikha Aliana & B. A. Nazren, A & Nasrudin, Mohd Wafi. (2015). Template Matching using Sum of Squared Difference and Normalized Cross Correlation. 100-104. 10.1109/SCORED.2015.7449303.4
- https://www.mathworks.com/help/images/ref/normxcorr2.html
- https://en.wikipedia.org/wiki/Cross-correlation
- https://www.mathworks.com/help/matlab/ref/saveas.html
- https://www.coursera.org/lecture/convolutional-neural-networks/non-max-suppression-dvrjH