Robotics Algorithms (CSE568)
Project 3 Report

Srivenkata Krishnan Sesharamanujam (UBIT Name: srivenka)

October 24, 2018

1. Introduction

The aim of the project is to take the digitized Prokudin-Gorskii glass plate images and automatically produce a color image using the following aligning concepts: First, Sum of squared Differences. Second, Normalized cross correlation. Third, using ransac algorithm on the features obtained. The rest of the report is organized as follows: Section 2 outlines the theoretical concepts used in aligning the images and final section contains the output of aligned images and inferences.

2. Approach

2.1 What is given and how to go about ?

Initially, the images are loaded into MATLAB and cut into three separate images. They are then added into a RGB image using MATLAB's cat function, which concatenates each of the layers into the image.

2.2 Image alignment

The idea of the project is to align the images based on three methods,namely

- Sum of Squared Differences.
- Normalized cross correlation.
- Using Features of the image.

Before looking into the different methods, The images often have ugly and unwanted borders and colored lines around the edges. These lines are detected and removed.

2.3 Sum of Squared Differences

Sum of squared differences (SSD) is one of measure of match that 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 consider the location of minimum value in the image matrices. Generally, SSD is directly is given by the equation,

$$ssd(i,j) = \sum_{i=0}^{M} \sum_{j=0} N(f(i,j) - g(i+u, j+u))^2 \tag{2.1}$$

where M is size of rows in reference image and N is size of column while u and v are variable, shift component along x-direction and y-direction respectively.

The idea in our program is to keep the blue channel image as the base image and shift the red and green channel images in the range$[-15,15]$. Now, we will find the minimum ssd among the range. Finally, we get two displacement vectors $[x, y]$ for both red and green images. Now, apply the shift operator on the corresponding red and green images respectively. Now, concatenate the blue image with the new red and green images. This is how the aligned image is obtained from sum of squared differences.

2.4 Normalized cross correlation

Normalized Cross Correlation (NCC) is always chosen as similarity measure due to its robustness. Its computation is more complex if we compare to the previous as it involves numerous multiplication, division and square root operations. Technically, NCC determines the matching point between template and image by searching the location of maximum value in the image matrices.

The idea in our program is to keep the blue channel image as the base image and shift the red and green channel images in the range$[-15, 15]$. Now, we will find the maximum ssd among the range. Finally, we get two displacement vectors $[x, y]$ for both red and green images. Now, apply the shift operator on the corresponding red and green images respectively. Now, concatenate the blue image with the new red and green images. This is how the aligned image is obtained from normalized cross correlation.

2.5 Using Features

2.5.1 Harris Corners

The Harris Corner Detector is a mathematical operator that finds features (specifically corners) in an image. It is simple to compute, and is fast enough to work on computers. Also, it is popular because it is rotation, scale and illumination variation independent.

It is just a mathematical way of determining which windows produce large variations when moved in any direction. With each window, a score R is associated. Based on this score, you can figure out which ones are corners and which ones are not.

R is determined by,

$$R = det(M) - constant_k * (trace(M)^2); \tag{2.2}$$

where M is

$$\begin{pmatrix} I_x^2 & I_{xy}^2 \\ I_{xy}^2 & I_y^2 \end{pmatrix}$$

$I_x^2, I_{xy}^2, I_{xy}^2, I_y^2$ are the corresponding derivatives along x axis, yaxis and xy axis.

After calculating the score, it is checked if it greater that a threshold value to determine the feature(corner). The image with the corners are identified.

2.5.2 RANSAC algorithm on Features

The algorithm randomly picks a feature in say blue image channel and it aligns with a random feature in say green image. The pixel shift for this alignment is calulated and applied to every other feature . If you find a feature within the threshold window, you can count that as an inlier; else you it is not. Run the algorithm several times, and pick the best alignment. After getting the alignment, try to get the aligned image by concatenating the three images.

3. Results

3.1 Unaligned Image

The images 1, 2, 3

Figure 1: Unaligned Image - Image 1

Figure 2: Unaligned Image - Image 3

3.2 Aligned image using SSD

The images 4, 5, 6

3.3 Aligned image using NCC

The images 7, 8, 9

Figure 3: Unaligned Image - Image 6

Figure 4: aligned Image - Image 1 -ssd

3.4 Aligned image using features

Figure 5: aligned Image - Image 3 -ssd

Figure 6: aligned Image - Image 3 -ssd

Figure 7: aligned Image - Image 1 -ncc

Figure 8: aligned Image - Image 3 -ncc

Figure 9: aligned Image - Image 3 -ncc