Lab 3: Colorizing the Prokudin-Gorskii photo collection

Objective

Colorize the pictures using SSD, NCC and using feature matching by detecting the features(corners) with a Harris Corner detector.

SSD

Method *im_align1* takes the b, g & r channels and aligns r & g channel to b. We select a window of shifts from -15 to 15 and calculate the SSD for each shift of r and g to score that shift. The shift with the least score is selected. To calculate the SSD, we just take the element-wise square of the difference of b & g channel matrix intensity values and calculate the sum of all the elements in the resultant matrix.

Offset for each image :

Image 1

Shift offset for blue - green with SSD =
5
Shift offset for blue - red with SSD =
9

Image 2

Shift offset for blue - green with SSD =
4
Shift offset for blue - red with SSD =
9

Image 3

Shift offset for blue - green with SSD =
7
Shift offset for blue - red with SSD =
15

Image 4

Shift offset for blue - green with SSD =
15
Shift offset for blue - red with SSD =
13

Image 5

Shift offset for blue - green with SSD =
6
Shift offset for blue - red with SSD =
12

Image 6
Shift offset for blue - green with SSD =
0
Shift offset for blue - red with SSD =
7

NCC

Method *im_align2* uses NCC to align the r, g & b channels. Similar to our approach for SSD, we align g with b and r with b. We also go over the same -15 to 15 shift window and calculate NCC score for every shift. To calculate the coeffs, we use the below function -

$$\gamma(u,v) = \frac{\sum_{x,y}\left[f(x,y) - \overline{f}_{u,v}\right]\left[t(x-u,y-v) - \overline{t}\right]}{\left\{\sum_{x,y}\left[f(x,y) - \overline{f}_{u,v}\right]^2 \sum_{x,y}\left[t(x-u,y-v) - \overline{t}\right]^2\right\}^{0.5}}$$

Ref - http://scribblethink.org/Work/nvisionInterface/nip.pdf

We consider the max score to select the shift with the best alignment.

Offset for each image :

Image 1
Shift offset for blue - green with NCC =
6
Shift offset for blue - red with NCC =
10

Image 2
Shift offset for blue - green with NCC =
4
Shift offset for blue - red with NCC =
9

Image 3
Shift offset for blue - green with NCC =
15
Shift offset for blue - red with NCC =
11

Image 4
Shift offset for blue - green with NCC =
4
Shift offset for blue - red with NCC =

13

Image 5
Shift offset for blue - green with NCC =
6
Shift offset for blue - red with NCC =
11

Image 6
Shift offset for blue - green with NCC =
0
Shift offset for blue - red with NCC =
6

Feature Matching

Corner detection

We are using corners as features, so first we use the method *harris* to detect all the corners in all 3 channels. The method returns the corners as the 2 separate vectors of x & y coordinates of the detected corners.

To detect the corners, first we take the gradient of an image along x & y axis (say Ix & Iy). Then we calculate the dot products of the gradients (Ix.*Ix, Ix.*Iy & Iy.*Iy).
We apply the gaussian filter to soften the image and reduce the noise (sudden very high intensity change spikes in the gradient).

It was observed that the corner of the image had some falsely detected “image” corners, so we ignore the border pixels while detecting the corners. We detect the corner-ness of each pixel by first generating the below matrix for every pixel.

$$H(x,y) = \begin{bmatrix} S_{x2}(x,y) & S_{xy}(x,y) \\ S_{xy}(x,y) & S_{y2}(x,y) \end{bmatrix}$$

Above Sx2, Sxy & Sy2 are the sum of the convoluted matrix we generated by considering the -2 & +2 pixel around a pixel.

Harris corner detector ref : http://www.cse.psu.edu/~rtc12/CSE486/lecture06.pdf

Detected corners for every channel are saved as an image with the name *image*-detected-corner-*.jpg*
Some of them can be seen below –

image6-detected-corner-b.jpg

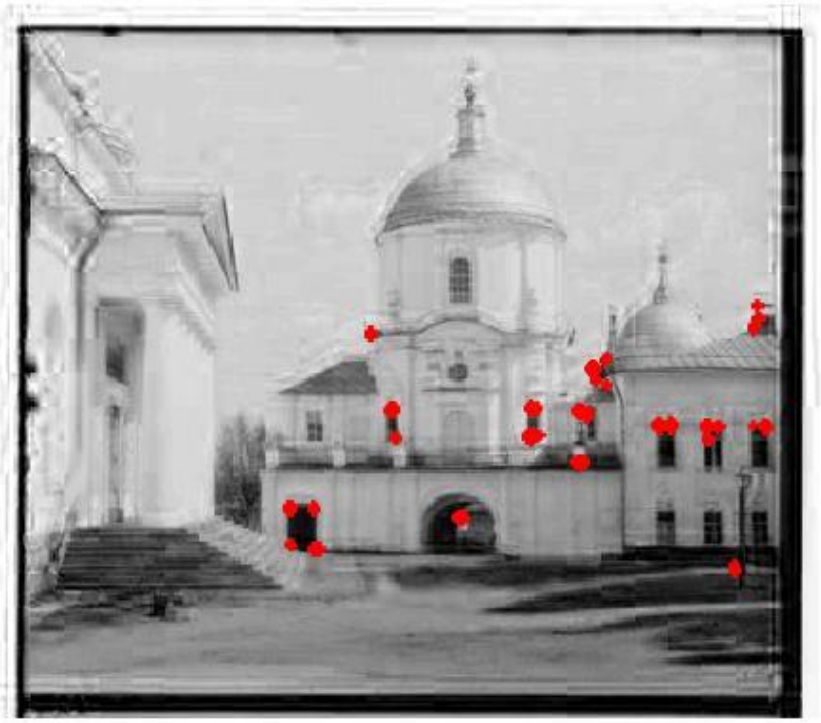


image6-detected-corner-g.jpg

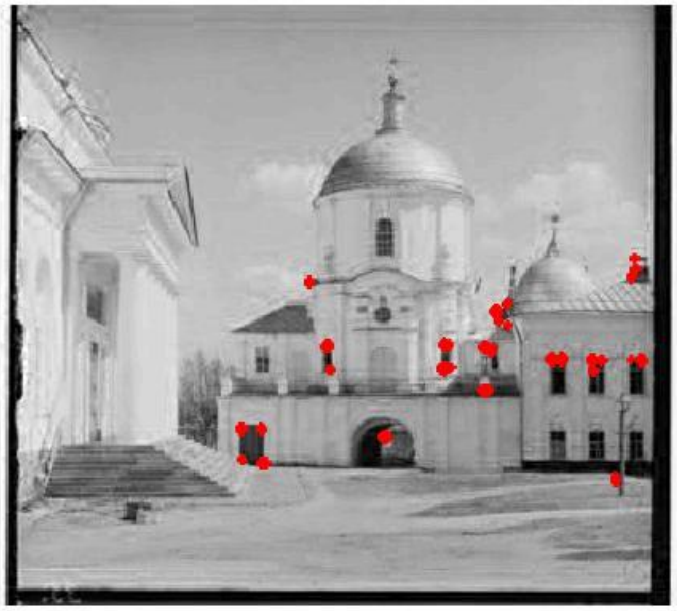
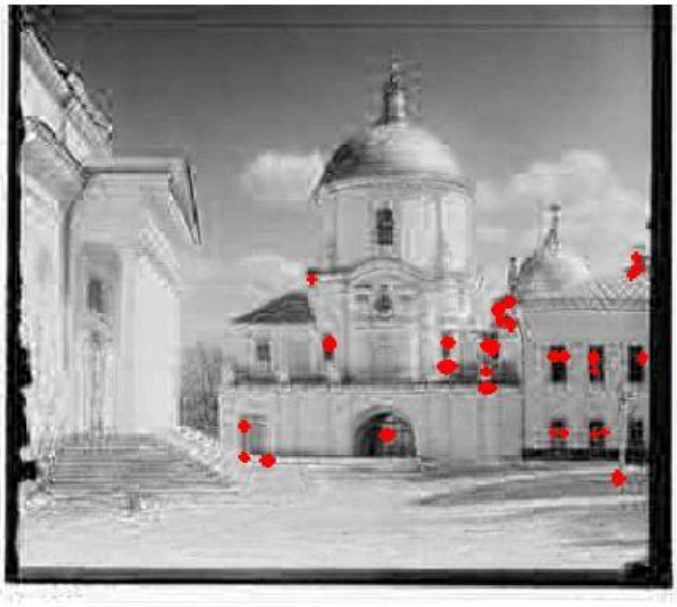


image6-detected-corner-r.jpg



RANSAC

Finally, to do find the right shift offset, we select the top 200 features detected in the descending order of their corner-ness values (since better corners are better features). We align g & r with b by select one feature from b and one from r. Then we shift all the detected corners in r by the offset of the difference between the selected features from x & r. We calculate all the inliers corners for this alignment by comparing with the shifted corners of r with the corners of b.

We run through different combination and select the one with the max inliers. This is our final offset.