Computational Photography
Homework 1 - Colorizing the Prokudin-Gorskii Collection

Amiya R Panda
UIN - 727006179

Method

1. Initially, the images are loaded into MATLAB and cut into three separate images of same height as the inputs were three channels placed one below the other. The first piece was used as the blue channel, the second as red and the final piece as the green channel.
2. In order to align the image, only the center-most 81 % of the image is used. In other words, 10 % around each edge is cut out. This is to make sure that the borders do not affect the image alignment.
3. The cropped images were then aligned and colorful images were produced by placing one on top of another. They are then added into a RGB image using MATLAB's cat function, which concatenates each of the layers into the image.
4. The above process is then optimized to get a perfect alignment. This is done by an experimental method of shifting the pixels and then checking the alignment. The metric used to compare the accuracy of the alignment was Normalized cross correlation.
5. Two approaches were used to produce the result. The first approach involved a brute-force search algorithm. 'FindShift()' function in the code uses this approach. The demerit of this approach is that it requires a large amount of time for computation.
6. To make the computation faster, the images were recursively down-sampled by a factor of 2 and then displacements were calculated at each level and these displacements were propagated back every level. This approach is called "Image Pyramid" and it reduced the time taken for each image drastically to a maximum of 34 seconds per image. The end condition in this case was taken as 400px height where a basic search was performed. 'Findpyramid_shift()' function in the code implements this approach.

OBSERVATION

We have observed that for jpg images, the time taken for alignment is approximately 3 seconds whereas for TIF images it varies from 24 seconds to 34 seconds. Moreover, the "shift_window" for shift for the lowermost level required for all the images except "village" and "self_potrait" is

15 pixels. For these cases, it require 25 pixels. Tweaking is required to achieve the optimum output as after a certain point the images get misaligned. i.e. the curve between the "shift_window" and the alignment reaches the maximum at 15 pixels for some images and 25 for the rest. We can also approach the problem by changing the pixel size of the last level case and decreasing the shift_window size accordingly.

The following are the final X, Y shifts and the time taken for each image using the IMAGE PYRAMID approach.

cathedral.jpg

blue-green disp : x: 5 y: 2

blue-red disp : x: 12 y: 3

Time(sec): 3.29285

emir.tif

blue-green disp : x: 48 y: 24

blue-red disp : x: 104 y: 56

Time(sec): 25.3362

harvesters.tif

blue-green disp : x: 56 y: 16

blue-red disp : x: 120 y: 16

Time(sec): 26.0757

icon.tif

blue-green disp : x: 40 y: 16

blue-red disp : x: 88 y: 24

Time(sec): 25.7478

lady.tif

blue-green disp : x: 48 y: 8

blue-red disp : x: 112 y: 8

Time(sec): 24.1461

monastery.jpg

blue-green disp : x: -3 y: 2

blue-red disp : x: 3 y: 2

Time(sec): 3.19294

nativity.jpg

blue-green disp : x: 3 y: 1

blue-red disp : x: 7 y: 0

Time(sec): 3.24016

self_portrait.tif

blue-green disp : x: 80 y: 32

blue-red disp : x: 176 y: 40

Time(sec): 31.6799

settlers.jpg

blue-green disp : x: 7 y: 0

blue-red disp : x: 14 y: -1

Time(sec): 3.40309

three_generations.tif

blue-green disp : x: 56 y: 16

blue-red disp : x: 112 y: 8

Time(sec): 30.3685

train.tif

blue-green disp : x: 40 y: 8

blue-red disp : x: 88 y: 32

Time(sec): 24.7865

turkmen.tif

blue-green disp : x: 56 y: 24

blue-red disp : x: 112 y: 24

Time(sec): 26.7382

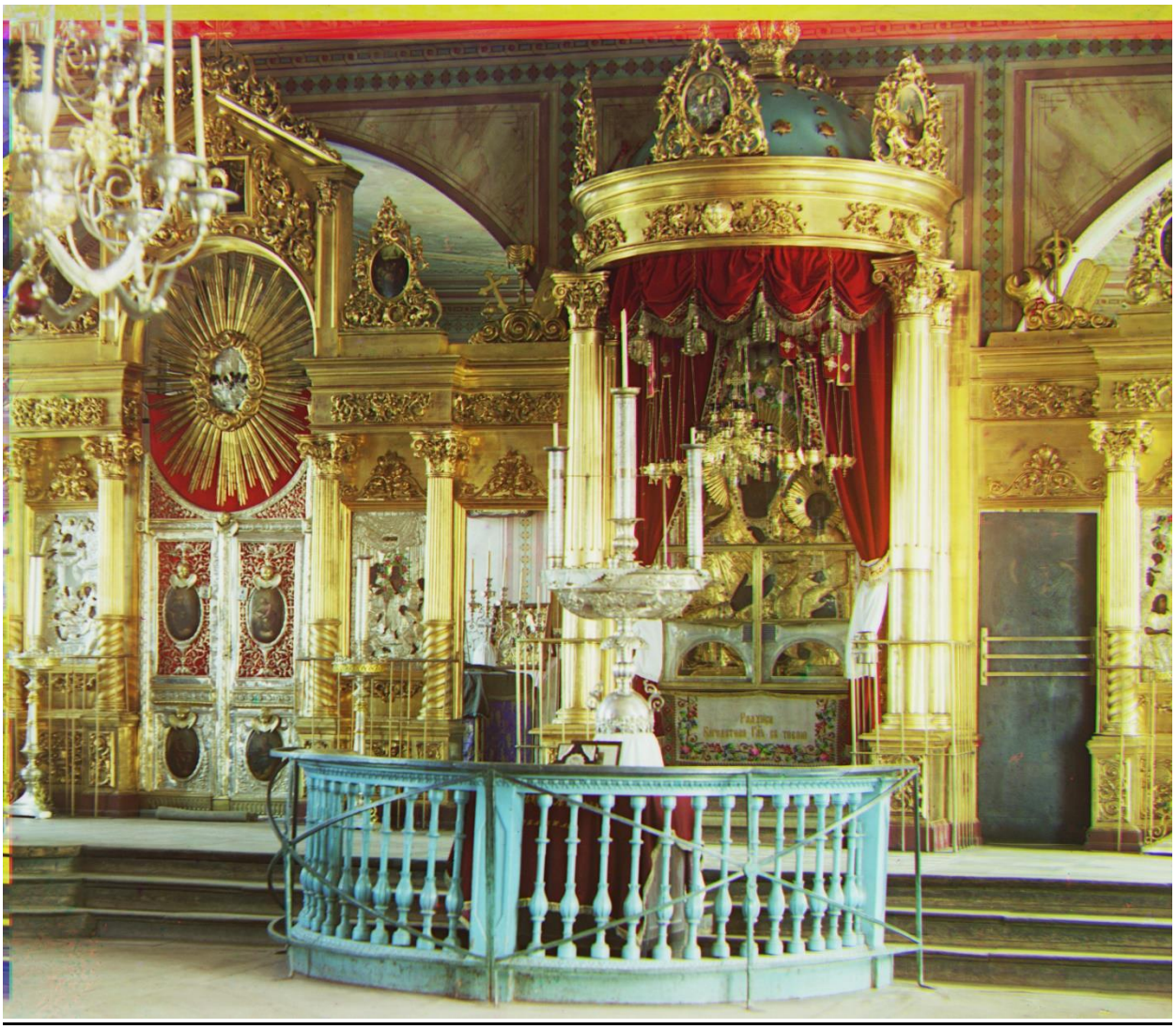
village.tif

blue-green disp : x: 64 y: 16

blue-red disp : x: 136 y: 24

Time(sec): 32.3596

OUTPUT IMAGES




Аб 132