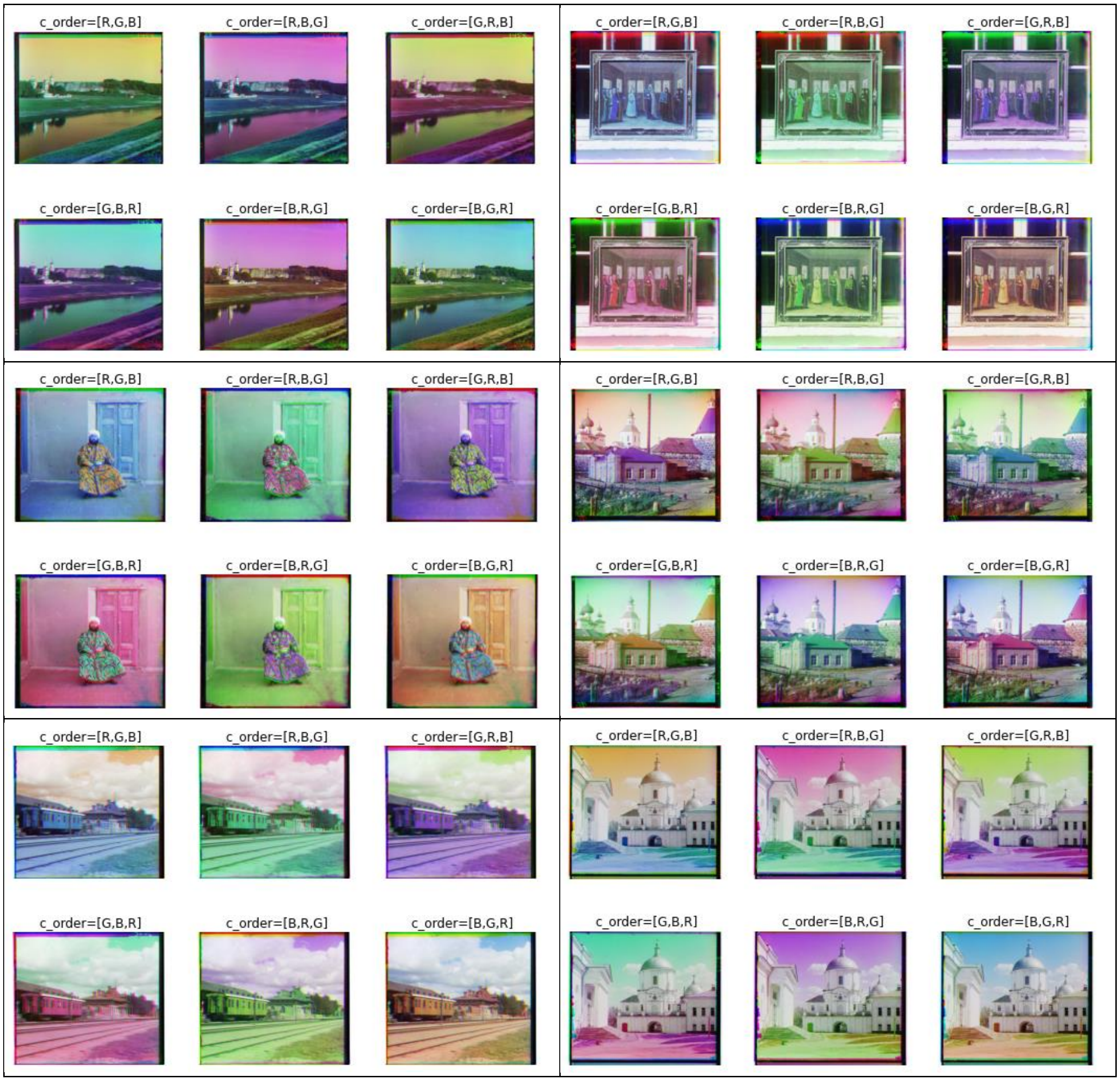
CS543/ECE549 Assignment 1

Name: Huey-Chii Liang

NetId: hcliang2

Part 1: Implementation Description

1. In order to make the pictures as close to each other as possible to reduce the calculation time, I cut off the edge blanks before dividing the original picture into three equal parts.
2. Next, I calculated the displacement relationship between the pictures through normalized cross-correlation (NCC). Among them I set the window size to 15.
3. Since we don't know which picture is red, green or blue at the beginning, I list every possible combination of stacked pictures and display the results, as shown below:



According to these pictures, we can know that the six pictures to be tested are all BGR from top to bottom because the [B, G, R] pictures are the most plausible one.

4. After knowing what the red, green and blue pictures are, I used the displacement results just calculated to combine the pictures. The results are shown in Part 2.

Part 2: Basic Alignment Outputs

A. Channel Offsets

Using channel **Blue** as base channel:

Image	Green (h,w) offset	Red (h,w) offset
00125v.jpg	[-1, -1]	[-2, -4]
00149v.jpg	[1, -3]	[1, -5]
00153v.jpg	[2, -1]	[3, 3]
00351v.jpg	[0, -3]	[-1, 0]
00398v.jpg	[-1, -1]	[-2, -3]
01112v.jpg	[1, -7]	[2, -9]

Using channel **Green** as base channel:

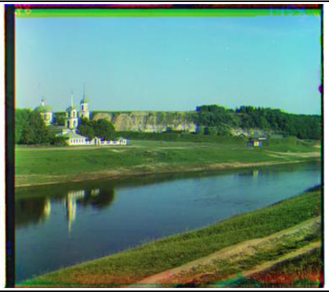
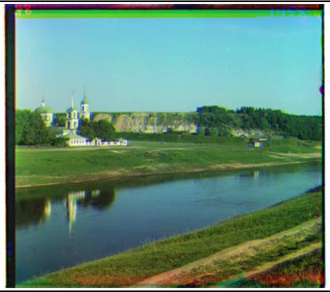
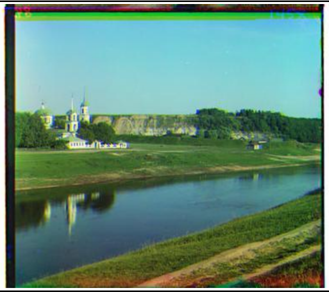
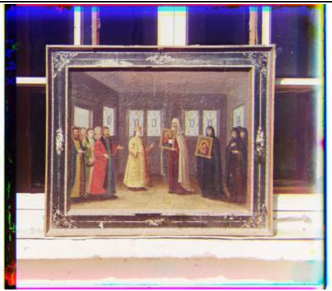
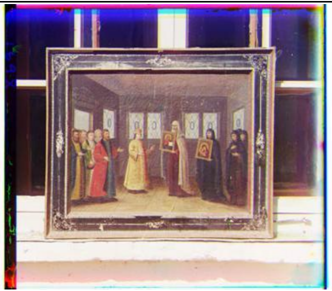
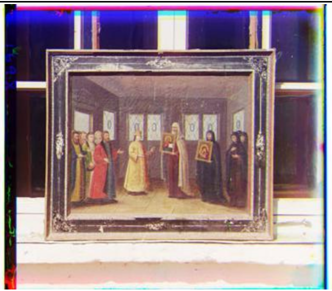
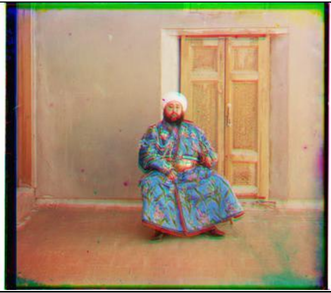
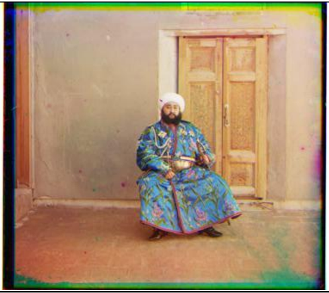
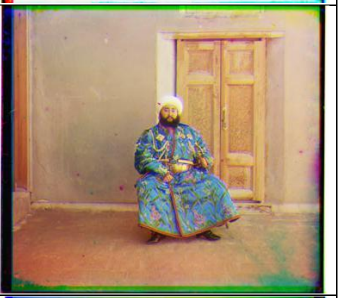
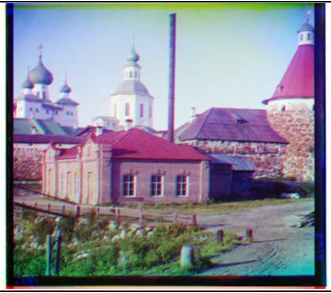
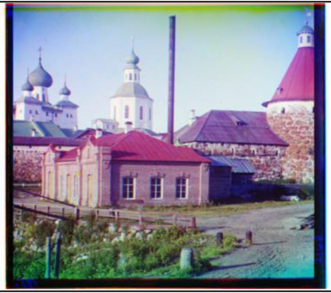
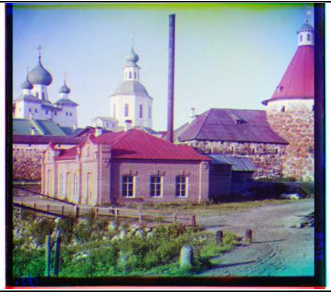
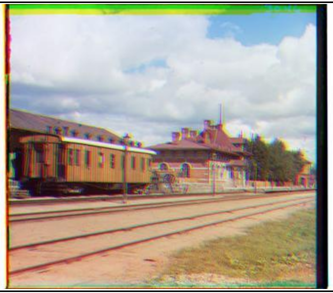
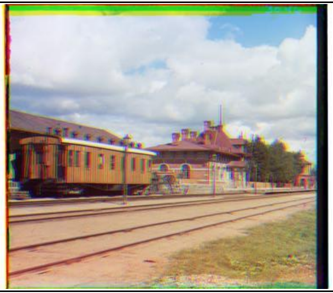
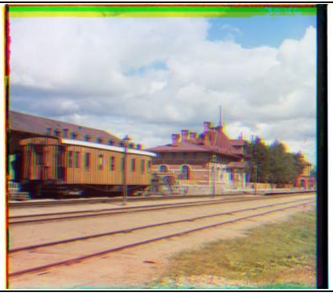
Image	Blue (h,w) offset	Red (h,w) offset
00125v.jpg	[1, 1]	[-1, -3]
00149v.jpg	[-1, 3]	[0, -2]
00153v.jpg	[-2, 1]	[2, 0]
00351v.jpg	[0, 3]	[0, 2]
00398v.jpg	[1, 1]	[-1, -1]
01112v.jpg	[-1, 7]	[1, -2]

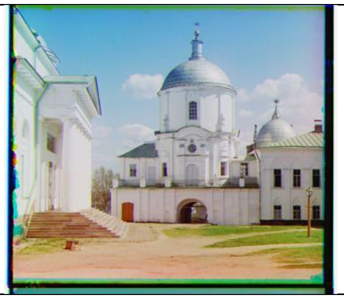
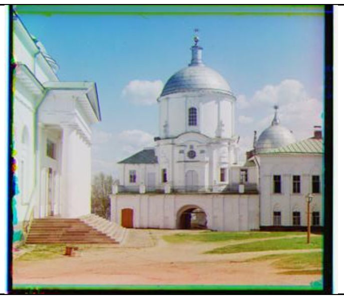
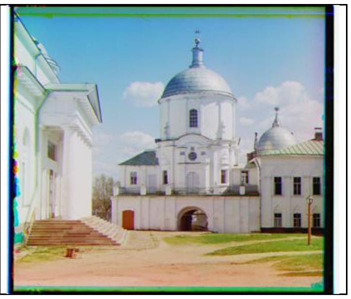
Using channel **Red** as base channel:

Image	Blue (h,w) offset	Green (h,w) offset
00125v.jpg	[2, 4]	[1, 3]
00149v.jpg	[-1, 5]	[0, 2]
00153v.jpg	[-3, -3]	[-2, 0]
00351v.jpg	[1, 0]	[0, -2]
00398v.jpg	[2, 3]	[1, 1]
01112v.jpg	[-2, 9]	[-1, 2]

B. Output Images

Insert the aligned colorized outputs for each image below (in compressed jpeg format):

Image	Blue as base channel	Green as base channel	Red as base channel
00125v			
00149			
00153			
00351			
00398			

01112			

Part 3: Multiscale Alignment Outputs

A. Channel Offsets

Using channel **Blue** as base channel:

Image	Green (h,w) offset	Red (h,w) offset
01047u.tif	[0, -39]	[1, -55]
01657u.tif	[-6, -9]	[-17, -26]
01861a.tif	[24, 5]	[47, 19]

Using channel **Green** as base channel:

Image	Blue (h,w) offset	Red (h,w) offset
01047u.tif	[0, 39]	[12, -16]
01657u.tif	[6, 9]	[-9, -14]
01861a.tif	[-24, -5]	[23, 12]

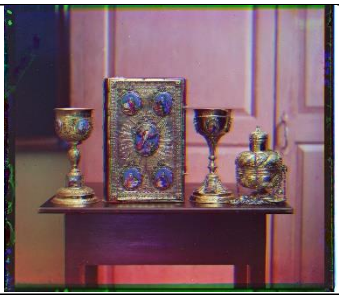
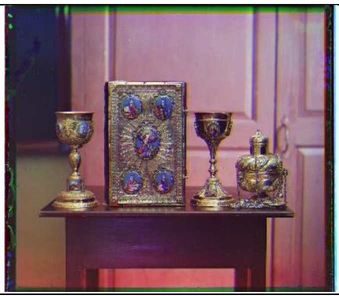
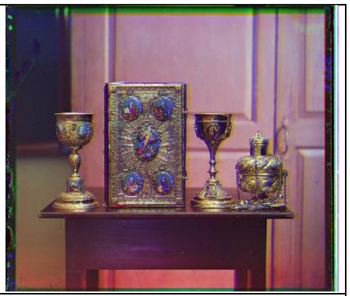
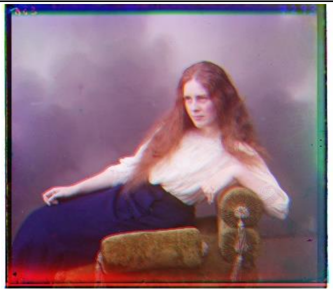
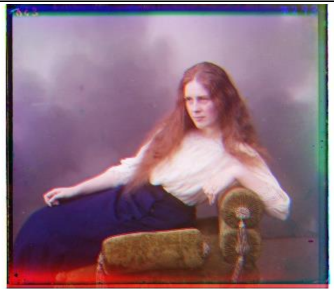
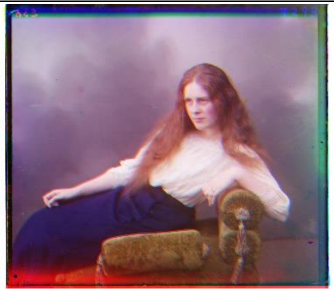
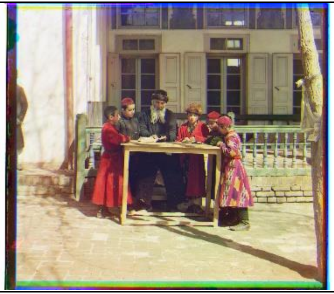
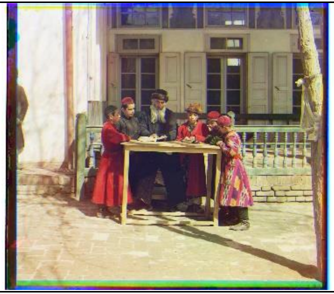
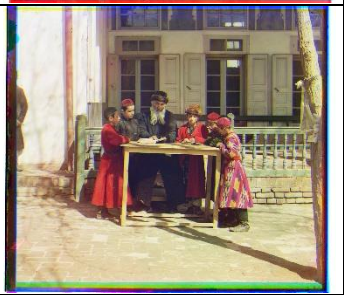
Using channel **Red** as base channel:

Image	Blue (h,w) offset	Green (h,w) offset
01047u.tif	[-1, 55]	[-12, 16]
01657u.tif	[17, 26]	[9, 14]
01861a.tif	[-47, -19]	[-23, -12]

B. Output Images

Insert the aligned colorized outputs for each image below (in compressed jpeg format):

Image	Blue as base channel	Green as base channel	Red as base channel

01047u			
01657u			
01861a			

C. Multiscale Running Time Improvement

Report improvement for the multiscale solution in terms of running time (feel free to use an estimate if the single-scale solution takes too long to run). For timing, you can use the python time module, as described in the assignment instructions.

Single-scale running time > 1 hour

Multiscale running time $= 170$ sec