School of Computer Science and Engineering

The University of New South Wales

COMP9517 - Assignment 1

Zachary Hamid

z5059915

1 Task 1

The Prokudin-Gorsky negatives constitute a collection of colour separated negative images that were taken by Sergey Prokudin-Gorsky. This was accomplished by taking three photos of a subject, each with a colour filter over the camera, namely red, green, and blue filters. These images were taken because Sergey believed that in the future, we would have the technology to reconstruct these images in colour. In this day and age, this is a relatively trivial task, all that needs to be done is the red, green, and blue-filtered negatives need to be stacked on top of each other and displayed as an RGB image. The only issue lies in aligning the negatives correctly so that the image that is displayed is as close to the original as possible.

The required alignment was achieved by first obtaining a misaligned RGB image (or for OpenCV, BGR) from the colour separated negative image. This was done by manually cropping the image by a percentage of it's height and width, in the case of this implementation, 3.5% of the width is cropped from either side, and 2% of the height is cropped off from both top and bottom. This newly cropped image then had rows removed from the bottom until the image height was divisible by three, then was simply divided by it's height equally in three pieces, these pieces were then stacked in the order of B, G, R from top to bottom.

From here, the correct alignment for the individual colour channels needed to be found. This was done by holding the blue channel still and moving the red and green channels over it ± some offset (in this case 30) both up/down and left/right until the optimal offset was found. This optimal offset was determined by giving a score to each offset and returning the offset with the best score. The exact procedure that was applied was the offsets were looped through both vertically and horizontally for the red and green channels separately, and for a given offset, a masked version of the red and green channels were obtained as outlined in the figure below (using an offset of (2,0) as an example).

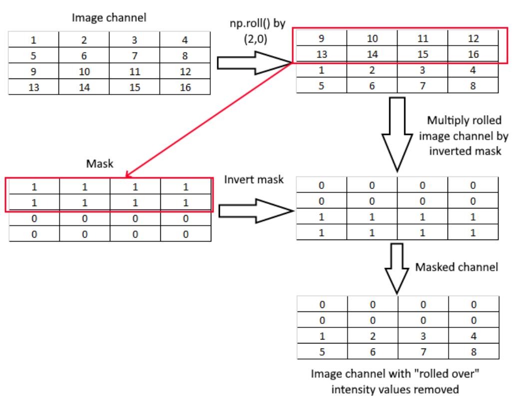
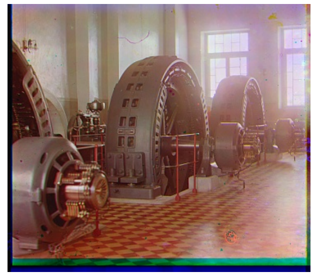


Figure 1: Process of Obtaining Masked Image Channel Using Offset of (2,0)

The new zero columns and rows of this masked channel were then trimmed off, and these same columns and rows were trimmed off of the reference blue channel. The result was that only the overlapping areas of both channels remained. This is the part that the metric is calculated on that gives the given offset a score to determine how optimal it is. From here, the masked channel that corresponded to the best score was returned along with the offsets, and this masked channel (both red and green) was stacked with the initial reference blue channel to form the correctly aligned image. The score was calculated using OpenCV's matchTemplate() function on both the trimmed blue channel and the trimmed red/green channels, and the chosen metric was the Normalized Correlation Coefficient (NCCoeff). It was found that this metric obtained more accurate alignment of the tested images than SSD, SAD and NCC using the above implementation. The advantage of using NCCoeff is that it actually penalises mismatches between the two images being matched and applies a negative score to areas that don't match in the image. This isn't the case for SSD, SAD and NCC and this means that NCCoeff is seemingly more strict with matching images and was able to correctly find the optimal alignment for all 5 given test images for Task 1. Alignment results for two of the five test images can be seen in the figures below, and it is clear that the alignment has been performed correctly.

(a) Pre-Alignment



(b) Post-Alignment
$G : (-8, 1), R : (-16, 1)$

Figure 2: Pre and post-alignment of image s1

(a) Pre-Alignment

(b) Post-Alignment
$G : (-8, 5), R : (-15, 8)$

Figure 3: Pre and post-alignment of image s2

2 Task 2

An image pyramid is a list of images that differ in resolution. Specifically, when the images are ordered such that the highest level images have the lowest resolution, and the lowest level images have the highest resolution (therefore, giving the list a "pyramid" shape).

In order to generate an image pyramid for iterative offset searching, the number of levels of the pyramid had to be decided on first. Finding a balance for the number of levels can be challenging, since if you use too many levels, low resolution images will have too low of a resolution at the highest level of the pyramid. Conversely, if too few levels are used, we aren't taking full advantage of the computation time that is saved by using an image pyramid in the first place. Therefore, it was decided that rather than using a hard-coded value for this, the number of levels in the pyramid is calculated based on the size of the original image. This is calculated as

$$\text{Number of levels } = \log_2\left(\frac{\text{Image Width}}{\text{Minimum Width}}\right) \tag{1}$$

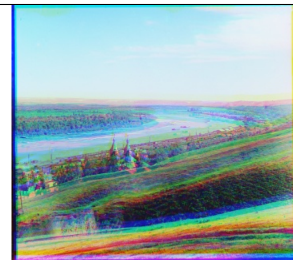
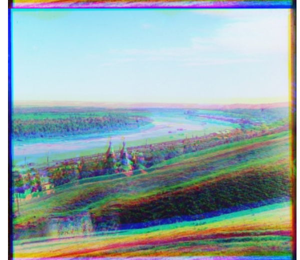
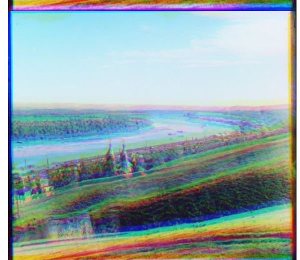
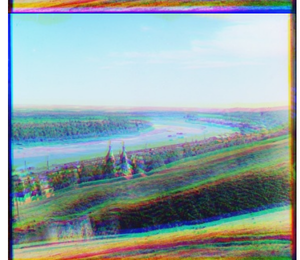
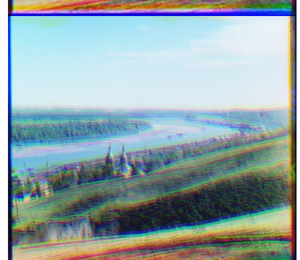
Where Minimum Width was set to be 50, this meant that image pyramids for the low level images were 3 levels deep, and for the high resolution images, 6 levels deep. This value was determined to obtain correct alignment for both the low and high resolution images given for the assignment.

Once the number of layers was decided, the method for lowering the resolution of the image needed to be determined. For this assignment, a simple approach was applied of using OpenCV's resize() function with a height and width reduction of 50% at each layer, and using the default bilinear interpolation method.

Now, to apply this image pyramid to the image alignment problem, the following procedure was implemented:

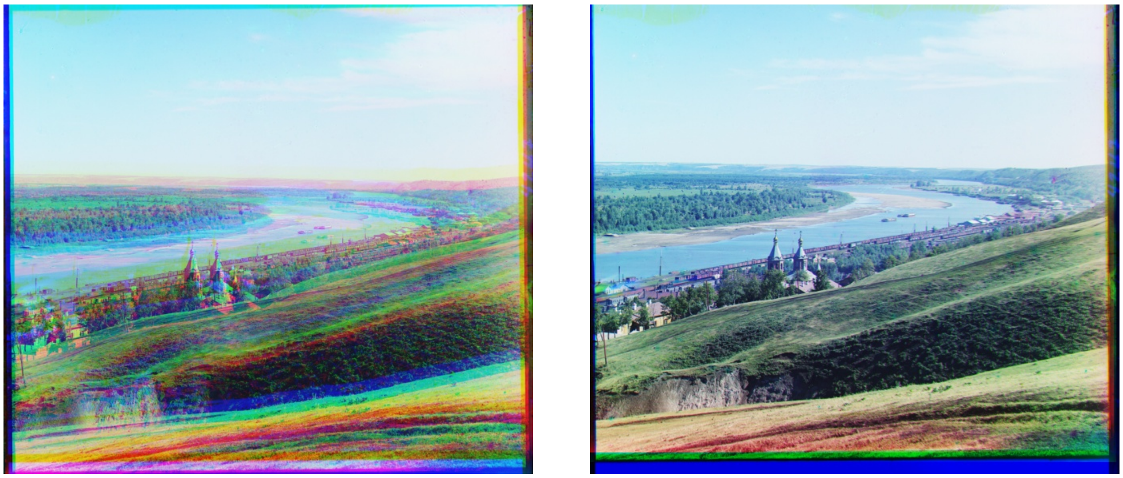
1. Calculate number of layers needed for image pyramid
2. Construct image pyramid of input image of depth equal to number of layers calculated
3. Initialise starting offset to 0 for i and j
4. Iterate through image pyramid layers starting at highest layer (lowest resolution image)
   - 4.1. For current layer, calculate optimal alignment starting at start offsets multiplied by 2 and looping through ± search range (± 10 was chosen) using same process as Task 1.
   - 4.2. Set start offset to this found optimal alignment and go to next layer and repeat the step above.

Table 1: Image Pyramid Intermediate Results of Image 00911u

Pyramid Level	Green Offset	Red Offset	Image With Applied Offset
0	(-1, -1)	(-2, -1)	
1	(-3, -1)	(-4, -2)	
2	(-5, -2)	(-8, -4)	
3	(-10, -3)	(-17, -4)	
4	(-19, -5)	(-34, -9)	
5	(-38, -10)	(-68, -17)	

From Table 1 above, the iterative improvement of offset assumption can be clearly seen when comparing the image alignment as the pyramid is traversed. It can also be seen that the improvement is almost exponential, with a larger leap occurring between level 3 and 4, and level 4 and 5. This is also expected, since the offsets double when they are passed to the next level in the pyramid (since the resolution has doubled). Movement of the channels, when it did occur, only happened with $\pm$ 1 or 2 pixels. So the lower levels of the pyramid were able to hone in quite early on the rough location of the correct alignment, and higher levels were able to use this and fine tune it.

Some high resolution alignment results can be seen below to show the effectiveness of the implemented approach.



(a) Pre-Alignment

(b) Post-Alignment
$G : (-38, -10), R : (-68, -17)$

Figure 4: Pre and post-alignment of image 00549u

(a) Pre-Alignment

(b) Post-Alignment
$G : (-56, -4), R : (-62, -6)$

Figure 5: Pre and post-alignment of image 00911u

3 Task 3

The original images produced by aligning the Prokudin-Gorsky negatives result in images that can have low quality and colour balance. This could potentially be attributed to the age of the film itself or perhaps many other factors. This, however, leaves room for improvement of the images to attempt to restore them to a state that is closer to what might have been experienced in person. Initially, a simple histogram equalisation was applied to the images, and whilst it did improve the quality of some of them, it also made the colours unbalanced in others, sometimes amplifying a specific channel more than it should, or sometimes completely losing the original colour palette of the image. It also made artifacts in the negatives more pronounced. Therefore, an improvement of this method was applied instead, a method known as Contrast Limited Adaptive Histogram Equalisation (CLAHE). The improvements this method offers over standard histogram equalisation is that it calculates multiple histograms, with each corresponding to a distinct area of the image, and redistributes lightness values using these. The reason standard AHE wasn't used is because it has the potential to overamplify noise in some images. The application of Contrast Limiting prevents this overamplification from becoming too noticeable.

(a) Unaltered Aligned Image (b) Histogram Equalized Aligned Image (c) CLAHE Aligned Image

Figure 6: Effect of HE and CLAHE on Aligned Image 00911u

(a) Unaltered Aligned Image (b) Histogram Equalized Aligned Image (c) CLAHE Aligned Image

Figure 7: Effect of HE and CLAHE on Aligned Image 00549u

In both of the cases above, the application of CLAHE improved sharpness of the images and reduced the "faded" appearance. Comparing to histogram equalization, CLAHE retained the original colour balance of the image, and didn't allow the red channel to become overprominent. Additionally, HE of the image darkened it too much, providing no improvement over the standard unaltered image. CLAHE, on the other hand, upped the contrast and due to the contrast-limiting feature, this contrast increase doesn't become overbearing in the image, and actually helps to improve the quality.