

## 1. Introduction

In this project, our target is to automatically colorize the Prokudin-Gorskii glass plate graph. These photo capture three exposures of ever scene using green, red and blue filter. Our goal is to find the best displacement between three channel, and make them can overlay to create an RGB color image.

## 2. Implementation procedure

### Step 1: split image & preprocessing

First we have to split the glass plate graph into three equal part, where they have the same width and roughly the one third of height of the original image. And the order from top to bottom will be blue, green and red.

### Step 2: Edge detection implementation

Edge detection plays a crucial role in our channel alignment process, as it helps identify significant features for matching across different color channels. We implement the Sobel edge detection algorithm, which calculates image gradients in both horizontal and vertical directions. The Sobel operator consists of two 3\*3 kernels designed to calculate gradients. These kernels perform discrete differentiation to approximate derivatives.

$$\text{vertical gradient}(G_y) = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$\text{horizontal gradient}(G_x) = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

The gradient magnitude is then calculated as

$$G = \sqrt{G_x^2 + G_y^2}$$

### Step 3: Multi-scale processing system

In order to improve the efficiency(especially in high resolution graph), we implemented downsampling strategy. After that we do the bilinear interpolation.

### Step 4: Channel alignment

We used NCC to evaluate the similarities between two images.

$$NCC = \frac{1}{n} \frac{xy}{\sigma_x \sigma_y}, \text{ where } x \text{ and } y \text{ are two images}$$





Then accomplish image shifting.

### Step 5: complete all of them

Using Numpy's function "dstack" to combine three different channel.

$$output = np.dstack((align_r, align_g, b)).astype(np.uint8)$$

## 3. Experimental result

Without align	align
	
	



## 4. Discussion

At the very first, we simply align the template with original R, G, B channel by performing NCC. But the result may be terrible in some picture (like the emir picture). So we try to calculate the edge of the image by using Sobel operator, and used NCC for aligning.

There are three way to combine  $G_x$  and  $G_y$

$$G = G_x + G_y \quad - \quad 1$$

$$G = |G_x| + |G_y| \quad - \quad 2$$

$$G = \sqrt{G_x^2 + G_y^2} \quad - \quad 3$$

Last, we use the method 3 cause it makes the results better.

## 5. Conclusion

In this homework we have learned that if you want to align channels, using Sobel filter is a very important way. Because without using it might easy get effect by the noise, and Sobel filter can highlight significant structural edges and contours in image. This create features that are invariant to lighting conditions and contrast variations.