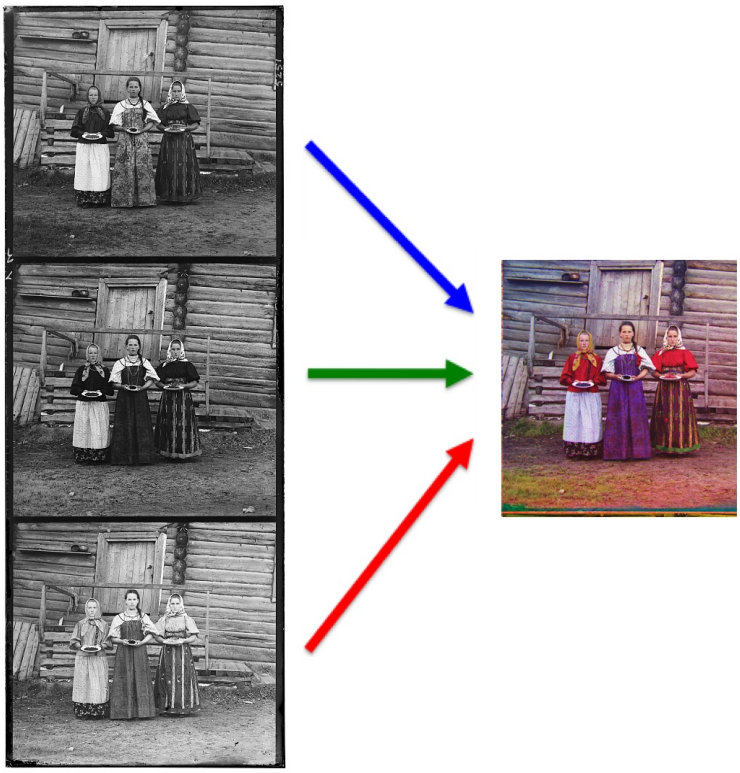
BIL 415 - Image Processing Practicum

HACETTEPE UNIVERSITY
Department of Computer Engineering

Problem Set 1
Fall 2015-2016
Dr. Erkut Erdem
TA. Aysun Kocak

Due Date: 23:59 pm on Friday, October 22nd, 2015

PART 1: Images of the Russian Empire: Colorizing the Prokudin-Gorskii photo collection [1]



Background

Sergei Mikhailovich Prokudin-Gorskii (1863-1944) [1], a Russian chemist and photographer, is considered as a pioneer in color photography. With Tsar's special permission and funding, he traveled across the Russian Empire, and took color photographs of people, daily life, buildings, landscapes, etc., documenting the early 20th-century Russia. He produced these color photographs by using a simple but ingenious idea: recording three exposures of every scene onto a glass plate using a blue, a green, and a red filter. These negative plates is now in possession by the Library of Congress of U.S.A., and have been recently digitized. They are available on-line at [2].

Overview

The goal of this part of assignment is to automatically produce color images from the digitized Prokudin-Gorskii glass plate images by applying some simple image processing techniques. For that, you will need to extract the red/green/blue color channels from the digitized plate images, and find the best alignment so that they form an RGB color image with few visual artifacts. You can download a starter code and some sample plate images from the course webpage.

[1] Credits: This assignment is a simplified version of the one that Alexei A. Efros developed for his Computational Photography class.

Details

Your program will take a glass plate image as input and produce a related color image as output. It should have the following structure:

1. Divide the input image into three equal parts corresponding to RGB channels.
2. Align the second and the third parts (G and R channels) to the first one (B channel). The simplest strategy to align the parts is to perform an exhaustive search over a window of possible displacements (e.g. [-15,15] pixels), and take the displacement that gives the best matching score.

   Some possible matching metrics are as follows:

   - Sum of squared differences (SSD) (sum(sum((image1-image2).ˆ2))
   - Normalized cross-correlation (NCC) (see Matlab function normxcorr2)
3. Improve the quality of the aligned image.

   Even if you've find a nearly correct alignment, the resulting color image may have a poor quality, i.e. it may suffer from low or high contrast, or the colors may not be so vivid. You can enhance the contrast of the images by applying Laplacian filtering, gamma correction, and histogram equalization, or a combination of those. Additionally, you can choose to work on a different color space, e.g. HSV color space (see Matlab function rgb2hsv), and accordingly manipulate the appropriate channels to obtain a more brighter image.

   Some important remarks:

   - Recall that the color filter order from top to bottom is BGR, not the typical RGB.
   - Assume that the negatives are evenly divided into 3 plates (i.e., each plate is in exactly 1/3 of the negative).
   - Assume that a simple x,y translation model is sufficient for proper alignment.
   - In Matlab, you can easily shift a matrix by using circshift function.
   - For a better matching score, you may ignore the image borders and use only the internal pixels.
   - You can devise and utilize an algorithm to automatically crop the borders in order to eliminate the artifacts in the boundaries and obtain a better quality result.

PART 2 : Color Spaces

Overview

The goal of this part of assignment is to implement a function that converts an L*a*b* image to HSV.

Details

- Implement formula(s) for conversion(s), **do not use MATLAB functions for conversion between color spaces** (e.g. rgb2hsv , lab2rgb)
- If there is no direct conversion between L*a*b and HSV color spaces, use other color spaces as intermediate steps
- L*a*b* images will be given as mat-file format. You should load the image and convert it to an HSV image.

Grading

The assignment will be graded out of 100:

- **70 (part 1):** CODE: 0 (no implementation), 10 (an attempt at a solution), 20 (a partially correct solution), 40 (a correct solution) and REPORT: 30
- **30 (part 2):** CODE: 0 (no implementation), 10 (a partially correct solution), 20 (a correct solution) and REPORT: 10

What to Hand In

You are required to submit all your code along with a short report. For that purpose, prepare a folder containing

- README.txt (text file containing details about your project, version of MATLAB)
- code/*.m (directory containing all your code)
- report/report.pdf

Archive this folder as studentid_pset1.zip and submit to dersler.cs.hacettepe.edu.tr , BBM415 course. Your report should contain;

- a brief overview of the problem,
- the details of your approach,
- the results of your algorithm on at least 8 negative images with your comments and show the results of all of the main steps (unaligned result, result after alignment, results of the enhancement)(for part 1),
- the result images (HSV images) and the intermediate results if there is anyfor all given images(for part 2).
- If your algorithm failed to give a satisfactory result on a particular image, provide a brief explanation of the reason(s).

Please, give all references that you used.

References

[1] http://en.wikipedia.org/wiki/Prokudin-Gorskii
[2] http://www.loc.gov/exhibits/empire/gorskii.html