Computer Vision - VU 183.585

Exercise Part: Assignment 0

Colorizing Images

Computer Vision Lab
Institute of Visual Computing and
Human-Centered Technology

Marco Peer - mpeer@cvl.tuwien.ac.at
Aline Schaufelberger - e12142881@student.tuwien.ac.at

WS 2024/25

General

This document summarizes the tasks and technical requirements for exercise 0 of the lecture and excercises (VU) Computer Vision at TU Wien in the winter term 2024/2025.

Due: 28.10.2024, 23:59 (CET)
Total achievable points: 4

In the winter term 2024/2025 the Computer Vision tasks of exercise 0 have to be implemented using *python 3* and *Jupyter Notebook*, a browser-based code editor. We recommend to install **pip**, a package manager for Python that helps you install and manage Python libraries and packages. You can find the installer based on your OS on the official pip website [1]. If you already have pip installed on your computer, skip the installation.

Environment Setup

In this exercise a custom virtual environment is required to be setup for each assignment separately (A1.txt / A2.txt). An environment contains a python installation of a defined version and a collection of packages. Different virtual environments can be created for different applications and have to be activated before their use. Follow this step-wise guide to make sure your environment is ready and all dependencies are installed correctly:

- Create a new virtual environment.

  ```
  python3 -m venv A1
  ```

- Activate the new virtual environment.

  For Windows: `A1\Scripts\activate`

  For Linux: `source A1/bin/activate`

- Use the **.txt** file included within the assignment to install all the python libraries and dependencies for the respective task.

  ```
  python3 -m pip install -r A1.txt
  ```

  1. Start the **jupyter notebook** file and now you should be able to start coding

     ```
     jupyter notebook [PATH_TO_FILE].ipynb
     ```

Recommended IDE For coding, we strongly recommend an IDE such as Visual Studio Code[2] which enables you to directly open Jupyter notebooks (refer to Working with Python in VS Code and Working with Jupyter Notebooks in VSCode for in-depth tutorials). Additionally, plenty of extensions for other programming languages/file types are available.

Read the instructions and implement the tasks in the notebook file `TASK[X].ipynb`. Use the hints and descriptions mentioned in the comments in the code and refrain from installing other libraries, not included in the environment .txt file.

[1] https://packaging.python.org/guides/installing-using-pip-and-virtual-environments/
[2] https://code.visualstudio.com/

NOTE: We will provide support for issues that might arise with your setup, but if you have problems setting up your environment in general use the TUWEL exercise forums, contact the tutor or attend the tutoring sessions.

Deliverables

You will hand in the assignments in a .zip file through **TUWEL**. Make sure all questions in the notebook files are answered and the code runs without errors. If asked, save your image results to your directory. Questions that are not sufficiently answered in the final submission or the missing of result images, will lead to the deduction of points. Do not forget to put your name at the top of the notebook as well as in all Python files.

TASK0 - Colorizing Images (4 Points)

The purpose of this assignment is to gain some initial experience with Python and Jupyter notebook, and to understand how to work with images by taking single color channel images and assemble them to RGB images. You will use digitized glass plate images from the Prokudin-Gorskii collection (see Figure 1 as example).

Background

Sergei Mikhailovich Prokudin-Gorskii (1863-1944) was a photographer who traveled the Russian empire and took thousands of photos of everything he saw, between the years 1909-1915. He used an early color technology that involved recording three exposures of every scene onto a glass plate using a red, green, and blue filter. Back then, there was no way to print such photos, and they had to be displayed using a special projector. Prokudin-Gorskii left Russia in 1918. His glass plate negatives survived and were purchased by the Library of Congress in 1948. Today, a digitized version of the Prokudin-Gorskii collection is available online.

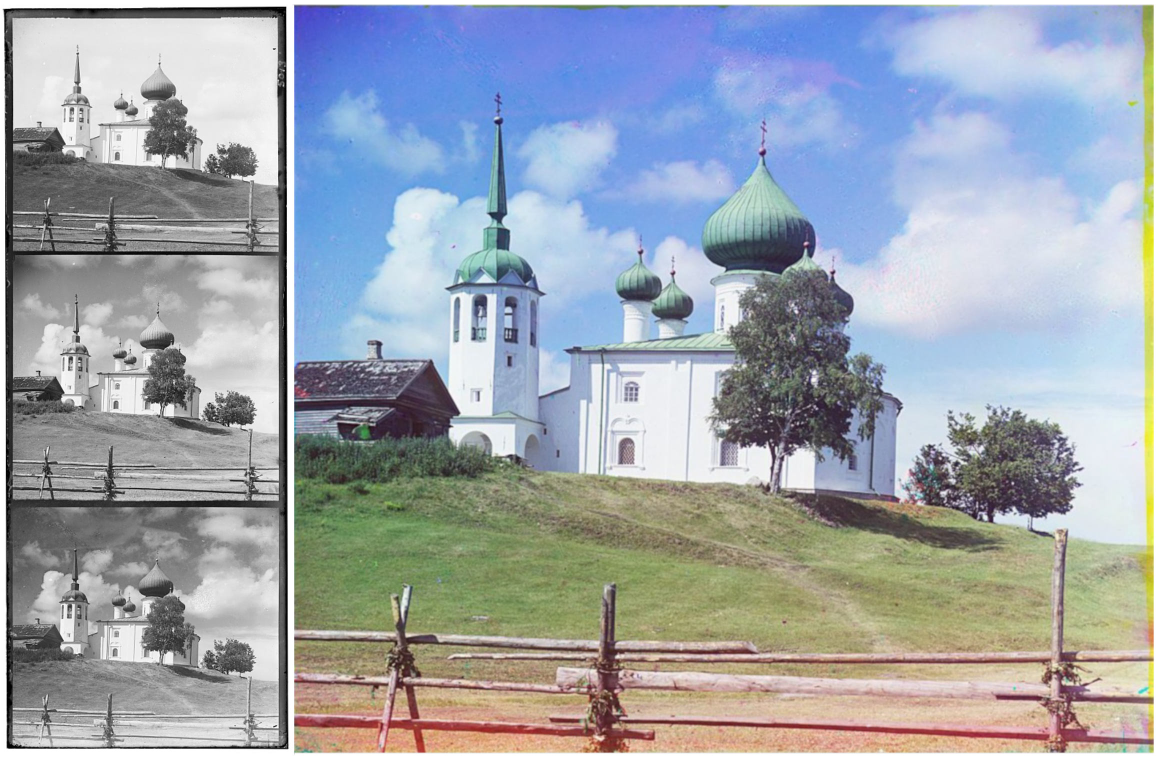

Figure 1: Glass plate scans of photographs with blue, green and red color filter (left) and colorized digital image (right).

Instructions

1 - Data import and preparation

There are six sets of glass plate images in the data folder, where the R, G and B channels are labeled by the suffixes **'_R', '_G'** and **'_B'**, respectively. Complete the first code section in the notebook by importing the three-channel images, further details and useful hints are pointed out in the code.

2 - Colorizing Images

The three channels are not perfectly aligned, thus you will complete the method `channels.align(..)`, which aligns them automatically. To do this, exhaustively search over a window of possible displacements (e.g. [-15,15] pixels), score each one using an image-matching metric, and take the displacement with the best score. Although the images do not actually have the same brightness values (they are different color channels), the **normalized cross-correlation (NCC)** is a sufficient choice for the image matching metric. The NCC between two images I_1 and I_2 is defined as

$$NCC(I_1, I_2) = \frac{\sum_{x,y}(I_1(x,y) - \bar{I_1}) * (I_2(x,y) - \bar{I_2})}{\sqrt{\sum_{x,y}(I_1(x,y) - \bar{I_1})^2 * \sum_{x,y}(I_2(x,y) - \bar{I_2})^2}} \tag{1}$$

where $\bar{I_1}$ and $\bar{I_2}$ are the mean values of the images I_1 and I_2, respectively. Use the equation to complete `corr2d(..)` in `channels.py`.

Deliverable

Please submit the the files in a folder called `Task0_[MATRICULAR_NUMBER]`, including `three` colorized images. Put your matricular number on the image results and in the Python files.