CoE 4TN4 Project
Camera Image Signal Processing (ISP) Pipeline

January 12, 2024

1 Overview

This course project is about the design and implementation of algorithms in image signal processing (ISP) pipeline, which is the core of all digital cameras, and is taught in class and has many on-line references (e.g., *en.wikipedia.org/wiki/Image_processor*).

The project has two phases. The tasks of phase 1, which is due on Feb. 18, 2024, are to

- design and implement their own demosaicing algorithms (the minimum method would be simple interpolation).
- design and implement a white balance algorithm, possibly with some human input (optional in phase 1).
- choose and implement a simple contrast enhancement method (e.g., CLAHE).
- carry out experiments and discuss their observations, identify existing problems and suggest possible methods to improve the visual quality of the ISP results.

In phase 2, students will strive to solve more challenging cases, such as denoising, automatic white balance, restoration of low light images, and etc.

2 Phase 1 (due Feb. 18, 2024): color demosaicing

Photo sensors are in general sensitive to a wide range of visible light spectrum, unable to distinguish between colours. To achieve colour imaging, modern digital cameras employ colour filter array (CFA) allowing each pixel to sense only one of the three primary colours. A raw image captured by this type of camera is a mosaic of colour pixels laid out in the so-called Bayer Pattern. Full colour images we commonly use are digitally restored from such raw images using demosaicing algorithms. In this project, your task is to implement a highly effective demosaicing algorithm based on linear regression.

Most natural images are piecewise smooth with high correlations among adjacent pixels and among colour channels. Thus, each missing component in a mosaic image is approximately a linear combination of the surrounding known pixels. For instance, given a mosaic patch X as follows,

$$
\left\langle \overset{A}{\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & a_{1,5} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} & a_{2,5} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} & a_{3,5} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} & a_{4,5} \\ a_{5,1} & a_{5,2} & a_{5,3} & a_{5,4} & a_{5,5} \end{bmatrix}}, \overset{X}{\begin{bmatrix} x_{1,1} & x_{1,2} & x_{1,3} & x_{1,4} & x_{1,5} \\ x_{2,1} & x_{2,2} & x_{2,3} & x_{2,4} & x_{2,5} \\ x_{3,1} & x_{3,2} & x_{3,3} & x_{3,4} & x_{3,5} \\ x_{4,1} & x_{4,2} & x_{4,3} & x_{4,4} & x_{4,5} \\ x_{5,1} & x_{5,2} & x_{5,3} & x_{5,4} & x_{5,5} \end{bmatrix}} \right\rangle \approx \overset{g}{\square}
$$

it is possible to approximate the missing green component g at the centre using the inner product of some coefficient matrix A and X, i.e.,

$$g \approx \tilde{g} = \langle A, X \rangle = \sum_{i=1}^{5} \sum_{j=1}^{5} a_{i,j} x_{i,j}.$$

The optimal A can be learned from a large number of sample patches similar to X. Suppose that we have n such mosaic patches $X_1, X_2, \ldots, X_n$ and their corresponding ground truth missing centre green component $g_1, g_2, \ldots, g_n$, then the optimal A is a matrix that minimizes the approximation error as follows,

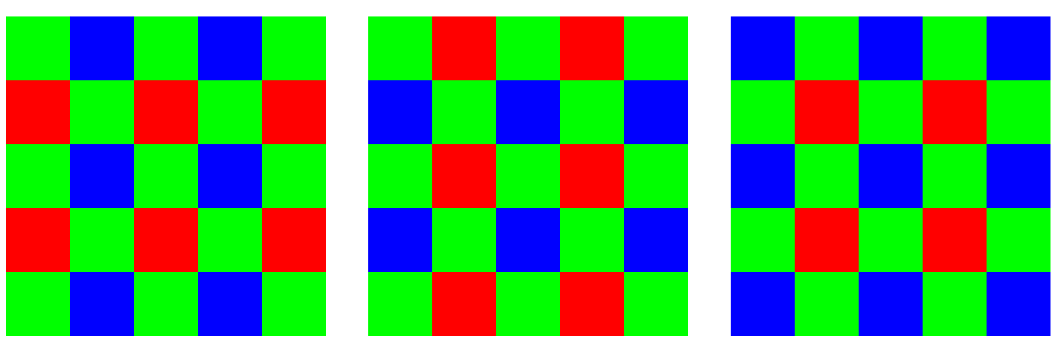
$$\min_A \sum_{k=0}^{n} (\langle A, X_k \rangle - g_k)^2.$$

This is a tractable linear least square problem. Please note that matrix A only predicts the green component; for the missing blue component, a different matrix, denoted by B, should be computed by the same approach,

$$\min_B \sum_{k=0}^{n} (\langle B, X_k \rangle - b_k)^2.$$

where $b_1, b_2, \ldots, b_n$ are the corresponding ground truth centre blue components.

Note that A, B only apply to the mosaic pattern as X above. For each of the other three different mosaic patterns as illustrated below,



you need to use different coefficient matrices to predict the missing colour components. Therefore, 8 coefficient matrices are required in total.

3 Tasks and Requirements

To implement the linear regression based demosaicing algorithm, you can follow the steps below.

1. Create a training dataset by simulating the 4 types of mosaic patches from full-colour images.
2. Solve the linear least square problem for each case and get the 8 optimal coefficient matrices.
3. Apply the matrices on each patch of a simulated mosaic image to approximate the missing colours.
4. Measure the RMSE between the demosaiced image and the ground truth.
5. Run your program on test raw mosaic data of our choice (to be released prior to deadline), and record the process and output results in video, and submit this demo video.

You need to write a report detailing your implementation and experimental results. You should compare the performance of the algorithm with the builtin `demosaic(...)` function in Matlab.