Bayer Pattern (Adapted from Svetlana Lazebnik)
COMP 776, Fall 2017
Due: September 20, 2017
Assignment Data: https://drive.google.com/open?id=0BzP2GHNy5xrfaXFrRy0zV25kMWc

Summary

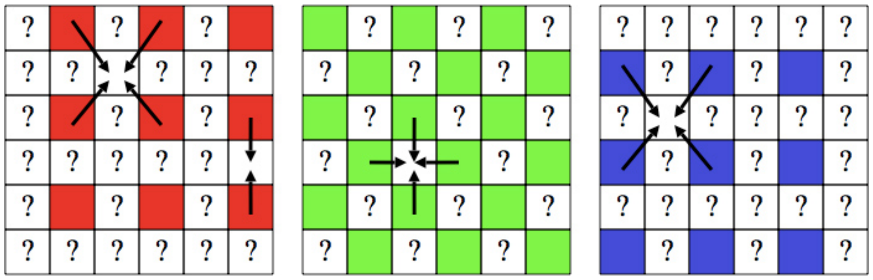
The goal of the assignment is to get started with image processing in Python by implementing a very simple demosaicing algorithm. The "mosaic" image ("crayons_mosaic.bmp") was created by taking the original color image and keeping only one color component for each pixel, according to the standard Bayer pattern:

```
R  G  .  .  .
G  B
.
.
.
```

So, once you read the Bayer image into a matrix, entry (0,0) will be the value of the red component for pixel (0,0), entry (0,1) will be green, etc.

Assignment and Submission

Implement a very simple linear interpolation approach for demosaicing: for each pixel, fill in the two missing channels by averaging either the four or the two neighboring known channel values:



The above method, being very simple, does not work perfectly. You can see where it makes mistakes by computing a map of squared differences (summed over the three color components) between the original ("crayons.jpg") and reconstructed color value for each pixel. Compute such a map and display it using matplotlib's imshow function,[1] along with the colorbar function. Consider using numpy's percentile function or a log-scale for the error visualization, and mention in your submission what you did to best visualize the error. In addition, report the average and maximum per-pixel errors for the image. Finally, show a close-up of a patch of the reconstructed image where the artifacts are particularly apparent and explain the cause of these artifacts.

Submit your images and write-up in a single PDF file. Also in your PDF, please include your name and a link to your code in your department Google Drive account. Be sure to share the folder with Jan-Michael Frahm, Marc Eder, and True Price.

[1] See the documentation to figure out how to scale/bound the values for good visibility. Hint: Use interpolation="nearest" to prevent matplotlib from automatically smoothing the image during visualization.