
CSI 394

Week 2: Into the PPM format!

1. We'll discuss the PPM format in class! In specific, P3. There are some sample PPMs on the course web page.
2. Read in a color image in the PPM format (make this a square image with a size that is the power of 2. On the website, mine are 256x256).
3. Average every pair of colors (groups of 3 numbers as we'll discuss)
4. Output the result in PPM format. You've just performed a kind of blur and/or sampling.
5. Try making a ppm using every other entry instead. How does this sampling change the look/feel of the result?
6. Try a different kind of sampling of your own design. Describe it. Show the three results side by side. Compare and contrast the results.
7. Other thoughts: You might find some other python image library. That's fine if you want to use that to read in some images or output them. If you decide to use this, remember to reference and share with us in class. However, to receive full credit on this code, perform this averaging and "skipping" manipulation yourself. This way we can actually dig into the images and do what we want with them directly rather than relying on a library's functions. You might note that windows doesn't natively accept this ASCII PPM format. So it might be a little hard to examine and/or check your work otherwise. You can download a free imaging tool called gimp to open your ppms. It is a little heavy for this work, but writing and reading ppms is much easier when compared to writing in other image formats. As another option, you can upload your images to sam and run the convert algorithm on them to get them in another format. For example, if you'd like to know what check.ppm looks like, you might run:
`convert check.ppm check.jpg` to get a converted jpg of that ppm. You can create your own ppms this way as well, but you might have to dig into the convert program to figure out how to create P3 PPMs in specific.
8. <https://stackoverflow.com/questions/6583573/how-to-read-numbers-from-file-in-python>
- Big Hint for reading ints from a file