Fred Hohman
May 14, 2014
MATH 4500 Image Compression Tips

For the Spring 2014 MATH 4510 class, we were given two primary methods for our image compression final project: linear gradient method and Principal Component Analysis (PCA). Linear gradient method attempts to reconstruct an image via blocks of (you guessed it!) linear gradients. To draw and create your own gradients search for digital design applications and make sure a gradient tool is included. Adobe's CS has many applications that can create sophisticated gradients. With that said, I did not attempt linear gradient method; I stuck to PCA.

PCA attempts to find a basis of image blocks based on whatever pictures it takes in. To use PCA effectively I suggest searching the internet for pre-made training image sets. These typically include pictures with a full range of contrast (pure white to pure black), depth of field, and sharp edges. In our class, most people elected to choose PCA, and most of the implementation code as very similar. That means the most significant factor between two students compression algorithms will be the training set used when performing PCA. Therefore it is important that your training set includes as many distinct photographic characteristics as possible.

If you are confused about the process behind PCA, take a look at this short step-by-step guide:

1. Read over Dr. Canterella's primer code and import the appropriate functions to convert images to vectors (and the reverse).
2. Gather a diverse training set of images.
3. Convert images to vectors and stack all the row and column 1024-vectors of every image into a new matrix. Call this matrix M. It's size should be something like BIGx1024.
4. Perform Transpose[M].M. Its size should be 1024x1024.
5. Take the singular value decomposition of this square matrix.
6. The column vectors of U now form a basis for the images in your training set. Take the appropriate numbers of column vectors and form a new matrix. This is your compression matrix.
7. Follow Dr. Cantarella's primer code to apply your compression matrix to an arbitrary image.

Final tip: It is a good idea to run your Mathematica matrix/list of vectors/etc. through Dimension[] after each step to verify that you are applying the correct transformations to the correct objects. Let Dimensions[] guide you!