Homework 4

1) Form pairs to write XML-RPC server/client programs (one student writes each).

   • The server offers 3 methods providing lossless image manipulation (be creative). Each server method must have descriptive documentation accessible by the client.

   • The client calls each method on the server machine with an array from an image. The client must then reverse the server methods to recover the original image.

   •Client saves the original image, modified image received from the server, and then reconstructed image. Server saves image received from the client and modified image.

   •Include example images in your solutions, and note your partner in your README. Do not discuss the nature of the server methods outside of the in-line documentation (client author can pester server author for more explicit documentation if it is not adequate).

Homework 4

2) Write a program that identifies musical notes from sound (AIFF) files.
- Run it on the supplied sound files (12) and report your program's results. Use the labeled sounds (4) to make sure it works correctly.
- The provided sound files contain 1-3 simultaneous notes from different organs.
- Save copies of any example plots to illustrate how your program works.

http://www.piddingworth.com/last_post_sheet_music.gif

Homework 4 Hints

1) Form pairs in class and decide on a time to work together. This can be done over the internet, but make sure to exchange contact info so you can get the right IP address and port number and set up a time to run the programs together. Be creative with the server methods, but make sure the operations can be explained simply in English. An example might be to switch color channels and reverse the order of the values in one channel. Make sure to think about properly dealing with differently shaped image arrays (methods should be compatible with both 3-color and grayscale). Lastly, XMLRPC won't transmit arrays, so you'll have to convert to another data structure.

2) You'll want to decompose the sound into a frequency power spectrum. Use a Fast Fourier Transform. Be care about "unpacking" the string hexcode into python data structures. The sound files use 32 bit data. Play around with what happens when you convert the string data to other integer sizes, or signed vs unsigned integers. Also, beware of harmonics.