

Programming for Performance: Assignment 2

Patrick Lam

Due: February 10, 2011

The primary goal of this assignment is to get you parallelizing C code for a simple task with minimal inter-thread communication. I recommend using one of the parallelization patterns that I described in Lectures 7 and 8.

The Task

You will create a program that reads JPEG files, rotates them according to their EXIF data¹, and creates a montage of the files.

You can find `exiftran`, a good starting point, in the `fbida` project:

<http://linux.bytesex.org/fbida/>

You can write your solution in any language you like; it's probably best to follow our solution and write it in C. Spawn parallel threads and process JPEG in the threads.

Input and output

Your program should take command-line parameters: an optional parameter for the number of threads (e.g. `-t 5`), a mandatory parameter for the output directory (e.g. `-out outputDir`), and then a list of JPEG files as input. It should output the transformed files into the output directory, along with a file `montage.jpg` containing thumbnails for all of the input files. The montage should contain 50×50 thumbnails for the input files, in rows of five inputs.

Tips

All of the JPEG library code that you need should be present in `fbida`. In particular, look at the code in `exiftran.c` and `genthumbnail.c`. I will post a single-threaded C solution on the course website.

Evaluation and deliverables

Submit your code. Indicate required libraries (we'll have `libjpeg6b` installed). Include a paragraph explaining your design structure (may be in the form of code comments). If your program satisfies the specification, you'll get full marks. Parallelizing the rotation will account for 70 points, and parallelizing the montage will account for 30 points.

¹Good digital cameras have an orientation sensor, so that you don't have to rotate all of your JPEGs manually.