José A. Biescas González 28 de abril de 2020

Concurrency In Scala Report

There are two case classes found inside the ImageFilterServer, these are the parallel and median filter classes. The main difference between these two is that the median filter class removes noise from an image using a serial implementation of the Median Filter method found in the wikipedia page provided by the professor, while the parallel class, although also using the Median Filter implementation, incorporates Scala's concurrent collections. These concurrent collections provide the functions for creating parallel structures which in theory should speed up the execution time by some time. In essence, these parallel structures let me divide the image into two parts, then these two parts will be processed in a concurrent manner.

Out of 10 runs, the parallel implementation completely dominated the serial implementation all 10 runs, only once did the serial implementation do better on a single image. Which is very likely that certain dips in execution time could be due to hardware performance more than anything else. In overall, the parallel implementation was around 1.3 to 2 times faster. Although after reading more on the wikipedia page, it is possible to speed up even more our algorithm with the use of selection algorithms for optimally selecting the median in each window calculated from the algorithm.

In overall, the algorithm runs well, executing in less than 1,000ms for each image at a time. Although colored images seem to pose a problem, as the test images used did not result in an image with a clear picture compared to the original, although there was improvement. It did not compare to how well the algorithm could process black and white images. The reasons for this might have to do with the calculation of the median, but it is something that I personally do not understand well yet.