Jordan Itzkovitz

260687254

COMP409

Wednesday February 6th 2019

Assignment 1

Q1)

The plot above describes the relative speedup of the image processing algorithm in accordance with the number of threads being used for computation. As shown in the graph, the algorithm increases in performance as the number of threads utilized increases up to 4-5 threads, at which point there is a slight decline in performance as the number of threads increase to 8. These experiments were performed on a quadcore processor, meaning that each of the 4 CPU’s are assigned their own thread when up to 4 threads are being used. For greater than 4 threads used there is an overhead associated to thread switching between for each CPU that is assigned more than 1 thread, thus causing a slow down in performance.

The algorithm assigns each thread a different position of the image to perform computations on. After completing such computations for the output image pixel, each thread is assigned a new position N positions away from it’s current one – N being the number of threads assigned to the overall computation. The Java object that contains the output image is inherently synchronized, however no two threads will ever be working on the same position in the image and thus does not require synchronization due to the algorithms implementation. The inherent locking of the output image may in fact decrease the efficiency of the algorithm as threads must still wait for permission to write to the file despite the fact that they are working on different positions. The other shared resources, including the input image and kernel, are only ever being read from, and thus do not require synchronization as well.