LaToya McIntyre

CS 276

HW 1

Image Processing: Grayscale and Negative

This experiment was designed to measure the speed of Processing’s image modification, specifically grayscale and negativize, with various sized images. The images are categorized as small (less than 40,000 pixels), medium (between 100,000 and 600,000 pixels), and large (over a million pixels). For each image, the time (in milliseconds) to complete the grayscale or negativize function was calculated and recorded in the table below. Each column represents a different implementation of the specified function. The normal function is indirect pixel array access through a nested for-loop. The modified nested loop is a direct access of the image’s pixel array with nested for-loops. While the modified single loop function, is a direct access of the pixel array through a single for-loop.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Time (milliseconds) | Normal Function | | Modified (nested loop) Function | | Modified (single loop) Function | |
| Image Size |  | Grayscale | Negativize | Grayscale | Negativize | Grayscale | Negativize |
| Small  (250 x 250) | 24 | 20 | 13 | 12 | 13 | 13 |
| Medium  (500 x 500) | 25 | 22 | 12 | 11 | 10 | 13 |
| Large  (1000 x 1000) | 41 | 38 | 25 | 26 | 28 | 28 |

From the information above, there is a significant difference in the time it takes to complete both the grayscale and negativize function based on whether the pixel array is directly or indirectly accessed; with the direct access taking approximately half the time of the indirect access. However, there is not a significant difference in completion time based on the number of loops, as long as both functions access the pixel array directly. Thus, the optimal way to implement a function involving image modification is definitively through direct pixel array access, with room for preference in terms of single for-loops versus nested for-loops.