

Assignment Report

202211187 정해주

I. Implementation Results

The summary of the representative results of applying various sizes of images to the code is as follows.

Image file	img_128.bmp	img_256.bmp	img_512.bmp	img_768.bmp	img_1024.bmp
Speed up	1.809576	1.722309	1.701620	2.023258	2.153487

Table 1. Results using '*sharpen_filter*'

As shown in Table 1, the performance improved as the size of the image files increased. On average, there was an improvement of 1.88 times, but the 768 and 1024 sizes exceeded this value, showing results of over two times. However, considering that the speedup might not be accurately measured if the image file is too small due to a very short execution time, the actual speedup could be expected to be above two times.

II. Optimization Approach

Based on the main objective to implement a program logic efficiently, most of optimization strategies were primarily referenced from 'Extra - Program Optimization'. The methods employed in this approach are detailed as follows.

A. Reduction in Strength

Complex expressions and variables that require extensive calculations and processing were declared outside the loop or simplified. By doing so, these pre-computed variables are used within the loop, thereby reducing the computational overhead and improving performance.

B. Sharing Common Subexpressions

In operations where similar expressions were used, they were consolidated into a single variable. This variable was then reused wherever necessary, reducing redundancy and enhancing efficiency.

C. Uniform Type Declaration

To avoid time delays caused by type conversions, the calculation process was standardized to the float type. This uniformity ensures consistent performance and minimizes the overhead associated with handling multiple data types such as integer.

D. Rearrangement of Nested loops

Variables used within the innermost loop were rearranged to be accessed consecutively. This rearrangement optimizes cache utilization and reduces memory access time, leading to faster execution of the nested loops.

Each of these optimization approaches contributed to a more efficient implementation of the program logic, ensuring optimal performance and resource utilization.