System Programing HW2 Report

202211040 김우진

Implementation Result

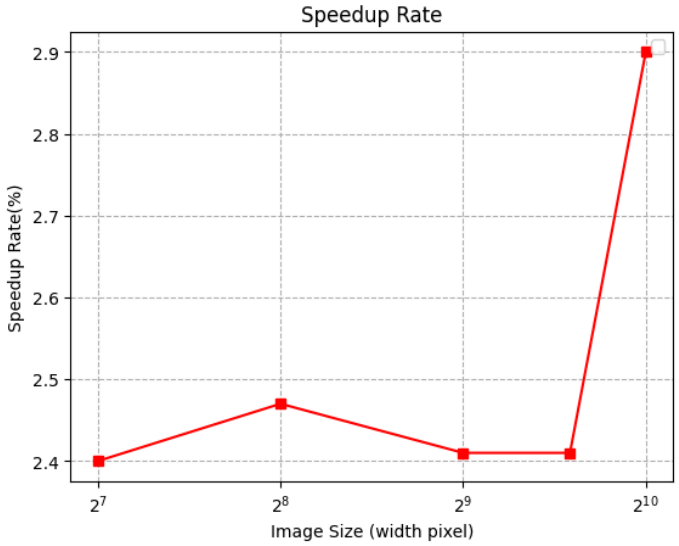

Image Size	128x128	256x256	512x512	768x768	1024x1024
Speedup Rate	2.40	2.47	2.41	2.41	2.90

The optimized code showed the best performance for images of size 1024*1024.

Issues with Existing Code (filter_baseline)

1. The current code uses dynamic allocation and free() within a for loop, which is unnecessary. Avoiding dynamic allocation is a better optimization strategy if possible.
2. The convolution function can be optimized to reduce redundant calculations, such as (x + dx) and (y + dy).
3. Using functions introduces call stack overhead. Therefore, avoiding functions can be beneficial for optimization.
4. In the existing code, the output array is accessed column-wise, which significantly reduces cache accessibility. Accessing the array row-wise is necessary.

Optimization Strategy

1. The first step is to resolve the aforementioned issues.
2. To further optimize, loop unrolling should be used. Loops introduce a significant number of conditional branch (beq) instructions, which can cause unnecessary stalls in the hardware. Unrolling the loops as much as possible is beneficial. Loop unrolling has proven to be the most effective strategy for reducing execution time.

Strategy for Loop Unrolling

1. First, remove the convolution function and implement its logic within the filtered_optimized function. This reduces overhead by avoiding call stacks.
2. Increment the indices of the double for loop, which selects the image pixels, by 2. Although increasing the index by 3, 4, or more is possible, it would make the code too lengthy and potentially lose spatial locality. Hence, an increment of 2 was chosen.
3. Increasing the y index by 2 has its advantages. While it may reduce cache accessibility, it halves the number of loop iterations. Experimental results showed performance improvements with an increment of 2.
4. Completely unrolling the 3*3 matrix multiplication part could yield greater performance improvements, but it would make the code excessively long. Nevertheless I tried. (As a result, the code has become nearly 300 lines long)