202111126 이나애

Section 1.

Table 1. Optimization result

Four strategies were combined and used, and the speed increased by approximately 1.3 times each strategy was added. The original code was optimized by improving the memory access pattern, eliminating dynamic allocation, using float instead of double, and dividing the convolution function into middle and edge parts to remove unnecessary conditional statements in the for loop. Additionally, forloops were unrolled to reduce conditional statements. The code achieved a 2.5 times speed increase.

Section 2.

To optimize the code, several strategies were considered, and ultimately, four strategies were combined.

Strategy 1: Memory Access Pattern Optimization

To enhance cache efficiency, the code was changed to a row-major order. When accessing data in a 2D array, processing all columns in one row first and then moving to the next row makes the memory access pattern sequential, thereby increasing cache efficiency.

Strategy 2: Elimination of Unnecessary Dynamic Memory Allocation

The variable p, which was allocated within the for loop, was eliminated. Instead of dynamically allocating memory the result is directly stored in the output array. This eliminates the process of dynamic memory allocation and deallocation within the loop, and also removes the need for memset(), thus improving speed.

Strategy 3: Blocking Technique - Discarded

Cache blocking is a technique to enhance cache efficiency by changing memory access patterns. It involves dividing large data structures into smaller blocks and optimizing memory access patterns. However, this method did not significantly improve speed in this case.

Blocking is a method used to efficiently process large data structures. If the data is not large enough, cache efficiency might not be significantly improved. Additionally, there might be a possibility that the code implemented for blocking did not work correctly.

Strategy 4: Using float instead of double

Strategy 5: Replacing for loops and if conditions with direct conditions.

The convolution function contained if statements to check the edges. Since for loops would pass through these if statements unnecessarily, removing these if statements and unrolled the for loops improves speed.