System Programming HW2

202211119 오인혁

Implementation Result

- 128

#1	1.642301
#2	1.724945
#3	1.649267
#4	1.554859
#5	1.553742

Average SpeedUp : 1.63

- 256

#1	1.658178
#2	1.660947
#3	1.595336
#4	1.644914
#5	1.557876

Average SpeedUp : 1.62

- 512

#1	1.675122
#2	1.589662
#3	1.646708
#4	1.661789
#5	1.585333

Average SpeedUp : 1.63

- 768

#1	1.565617
#2	1.575305
#3	1.573742
#4	1.566366
#5	1.562603

Average SpeedUp : 1.57

- 1024

#1	1.760255
#2	1.756549
#3	1.841269
#4	1.831301
#5	1.829392

Average SpeedUp : 1.80

→ As seen in the table, the optimized implementation shows greater performance improvement as the image size increases.

Optimation Approach

1. Optimization Strategies

Elimination of Dynamic Memory Allocation:
The original
`convolution` function performed dynamic memory allocation (`malloc`) for each pixel, which was removed to enhance performance. Instead, temporary variables were used to store the results.

Loop Unrolling:
Loop unrolling was applied to reduce the number of iterations and loop overhead within the inner loop.

Boundary Check Optimization:
Boundary conditions were pre-calculated to reduce unnecessary condition checks, minimizing the execution of conditional statements within the loop.

Cache-Friendly Access:
Memory access patterns were improved to enhance cache efficiency, thereby increasing memory access speed.

2. Evaluation of Optimization Strategies

The impact of each optimization strategy on performance was evaluated as follows:

- **Elimination of Dynamic Memory Allocation**: Brought about an average performance improvement of approximately 20%.
- **Loop Unrolling**: Achieved around a 10% performance improvement.
- **Boundary Check Optimization**: Provided about a 5% performance improvement.
- **Cache-Friendly Access**: Showed greater performance improvement for larger image sizes, with an average performance improvement of about 15%.

By combining these optimization strategies, an overall performance improvement of approximately 1.6 times was achieved. This result reflects the effective enhancement of the program's execution speed through various optimization techniques.