System programming hw2 report

202111030 김성우

1. Result

Table 1. Edge filter

Size \ Trial	1 (times)	2 (times)	3 (times)	4 (times)	5 (times)	Avg (times)
128	4.913594	4.890326	4.816756	5.010101	5.020145	4.93018
256	4.891617	4.998921	4.926097	5.024438	5.007533	4.96972
512	4.913842	4.901881	4.871654	4.877366	4.966161	4.90618
768	5.065100	4.905279	4.976670	4.900056	4.908890	4.95119
1024	5.855463	5.766138	5.682758	5.737021	5.725070	5.75329

Table 2. Gaussian filter

Size \ Trial	1 (times)	2 (times)	3 (times)	4 (times)	5 (times)	Avg (times)
128	4.932617	4.969545	4.960217	4.906286	5.100092	4.973750
256	5.332359	5.147327	5.129288	5.318136	5.003334	5.186088
512	5.263868	5.110775	5.295691	5.133480	5.097496	5.180662
768	5.109776	5.157315	5.101137	5.282735	5.118880	5.153169
1024	6.131223	5.934446	5.945989	6.020908	6.050896	6.016292

2. Applied optimization

The core optimization technique for speed improvement utilized the property that all filters used in convolution operations exhibit circular symmetry. Due to this circular symmetry, it is unnecessary to iterate through every pixel within the filter. Circular symmetry means that the top, bottom, left, and right values are identical, the four corner values are identical, and the filter has a central value. Thus, while iterating over every pixel in the image, we simultaneously access the top, bottom, left, and right pixels and the four corner pixels, performing the computation in one go. This simultaneous access is effectively equivalent to applying loop unrolling. By considering this symmetry, many redundant calculations can be ignored, leading to significant performance improvements.

The next optimization technique applied was the elimination of conditional statements through zero padding of the image. Conditional operations impose a substantial computational load. In the original code, every pixel was checked with a conditional statement to determine if it was an edge pixel. While this has minimal impact on small images, it significantly affects performance for larger images, such as those with sizes 768 or 1024. To achieve performance gains by eliminating conditional statements, zero padding was performed using memcpy(), and convolution operations were executed on the padded image.

Lastly, code motion was employed to minimize unnecessary computations within loops. Specifically, redundant index calculations were addressed by declaring local variables. Similarly, memory aliasing was minimized by declaring local variables wherever possible. Additionally, to leverage the spatial locality of the cache, computations were performed row-wise.