HW2

202111010 권도윤

- Result

	1회	2회	3회	평균
128	2.22	2.17	2.23	2.21
256	2.41	2.34	2.35	2.37
512	2.24	2.35	2.36	2.32
1024	2.64	2.62	2.76	2.67

The execution results are as shown in the table above. The speedup number output when running the program is rounded to the second digit.

The image appeared smallest when the size was 128 and largest when the image size was 1024.

When the image size is 512, the speedup score decreases slightly, but overall, you can see that the speedup score increases as the image size increases.

- Optimization Approaches
  - Remove unnecessary code
    Code that does not cause any problems during program operation has been removed.

    The dynamically allocated part of the filter_optimized() function was directly assigned to the output variable without dynamic allocation.

    The code that initializes memset in the Pixel_convolution() function was removed because it is not necessary for program operation.

  - Reduce repetitive operations
    When calculating the array index within a loop, unnecessary repetitive calculations were reduced by storing the repeated part in a local variable.

    In the Pixel_convolution() function, unnecessary repetitive calculations were reduced by storing the filter element in a local variable rather than accessing it every time.

  - Avoid declaring variables within loops
    If a variable is declared within a loop, declaration and destruction occur continuously, so the variable declaration part was written outside the loop as much as possible.

  - loop unrolling
    Speedup was attempted by unrolling the loop that performs the convolution operation in the Pixel_convolution() function.

    Since the total number of iterations in this function is relatively small at 9, it was written by unrolling all iterations.

  - type change
    In the Pixel_convolution() function, speedup was attempted by converting the types of double variables such as r, g, and b to int type.