Comparison performance report

github.com/blazingSummerSun

December 8, 2024

To evaluate the performance using different amounts of threads, the following settings were applied (less iterations, more points):

Resolution: 1920x1080
Number of iterations: 10.000
Number of initial points: 2.000
Number of linear transformations: 5
Pattern of non-linear transformation: Diamond
Symmetry parameter: 4
Number of threads: 1-5 (since >5 threads don't increase performance)

To compare the 1st performance, the following procedure was chosen: generate 10 similar fractals in a row with the specific settings, and find t_{avg}:

$$t_{avg} = \frac{t_{total}}{10}$$

Results:
1 thread: $t_{avg} = 17306ms$
2 threads: $t_{avg} = 11396ms(-35\%)$
3 threads: $t_{avg} = 10096ms(-12\%)$
4 threads: $t_{avg} = 9536ms(-6\%)$
5 threads: $t_{avg} = 12756ms(+33\%)$

To evaluate the performance using different amounts of threads, the following settings were applied (more iterations, less points):

Resolution: 1920x1080
Number of iterations: 30.000
Number of initial points: 20
Number of linear transformations: 5 (since >5 threads don't increase performance)
Pattern of non-linear transformation: Diamond
Symmetry parameter: 4
Number of threads: 1-5

To compare the 2nd performance, the following procedure was chosen: generate 20 similar fractals in a row with the specific settings, and find t_{avg}:

$$t_{avg} = \frac{t_{total}}{20}$$

Results:
1 thread: $t_{avg} = 868ms$
2 threads: $t_{avg} = 741ms(-15\%)$
3 threads: $t_{avg} = 668ms(-10\%)$
4 threads: $t_{avg} = 872ms(+30\%)$
5 threads: $t_{avg} = 917ms(+5\%)$

Observation: the more points there are in a fractal, the greater the performance boost will be when running multiple threads.