# 1、分形算法Mandelbrot并行化

使用带-DRelease参数的cmake构建（-O3优化），选定图像长宽为2000像素（M、N），迭代轮数2000（MAX\_ITER）。测试机器为macOS 10.15.1，Intel(R) Core(TM) i7-8559U CPU @ 2.70GHz。单线程和多线程使用时间如下：

单线程（mandelbrot.cpp）：用时13170.165ms。运行截图见mandelbrot.png

多线程（mandelbrot-multi.cpp）：运行截图见mandelbrot-multi.png

|  |  |
| --- | --- |
| OpenMP线程数 | 用时（ms） |
| 1 | 12927.618 |
| 2 | 10077.319 |
| 4 | 5923.231 |
| 8 | 3354.226 |
| 16 | 2839.002 |
| 32 | 2774.719 |
| 64 | 2953.224 |
| 128 | 3045.823 |

从以下图像上可以看出，多线程和单线程的运算结果完全相同。即程序使用OpenMP修改的结果是正确的。随着多线程版本中的线程数逐渐增多，运行时间逐渐降低并稳定在3000ms左右，这是因为CPU的运算能力是有限的，此时瓶颈不再是CPU并行程度。另外，当线程数从1增加到2时，运行时间并未折半，同样，从4变为8、8变为16等同样也没有将运行时间成倍缩短。这是因为在并行化时，虽然计算任务被并行了，但是却引入了线程见/进程间调度的成本，而这在并行程度较低时（如2线程）表现最为明显。

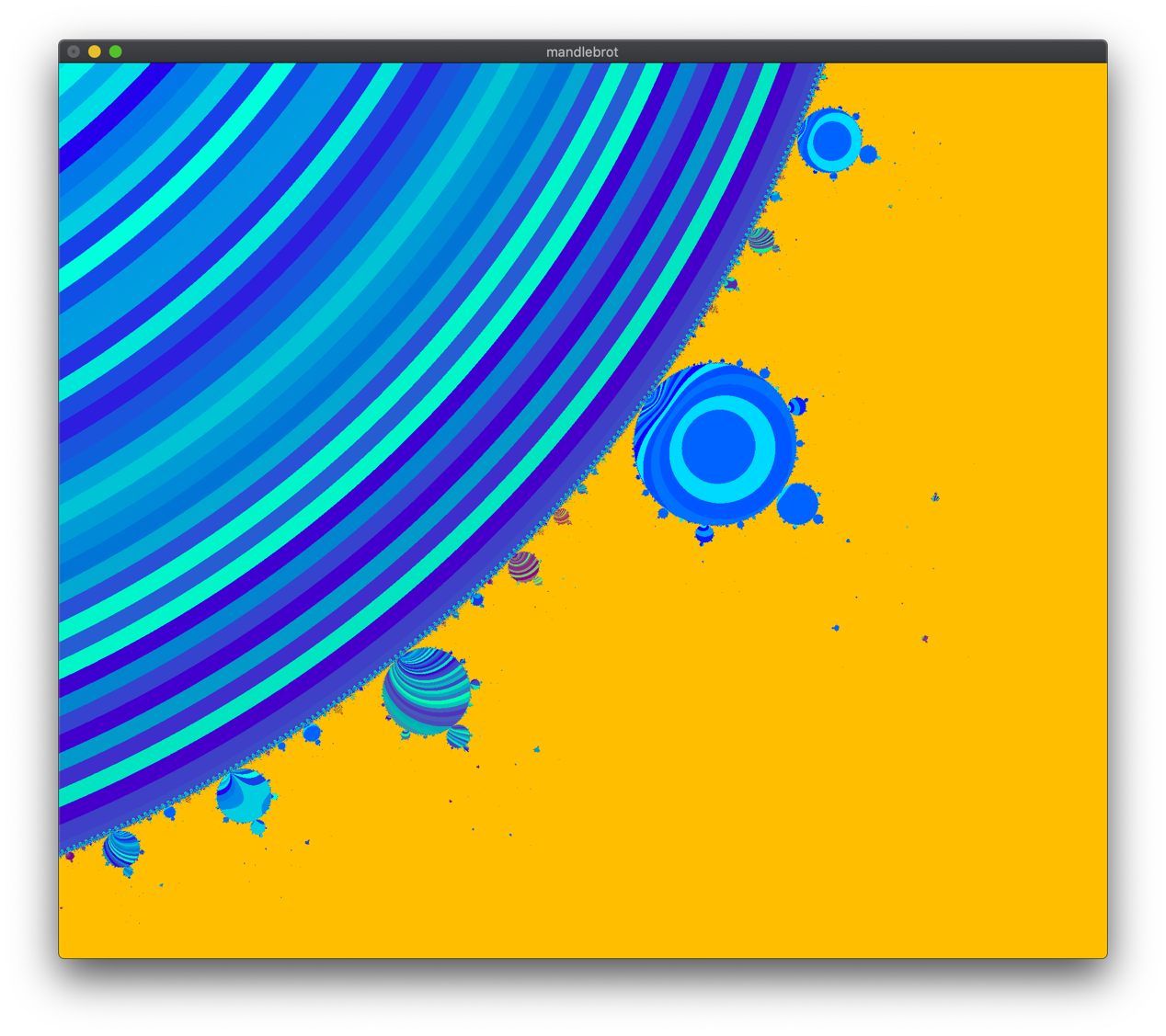


Figure 1单线程运行结果

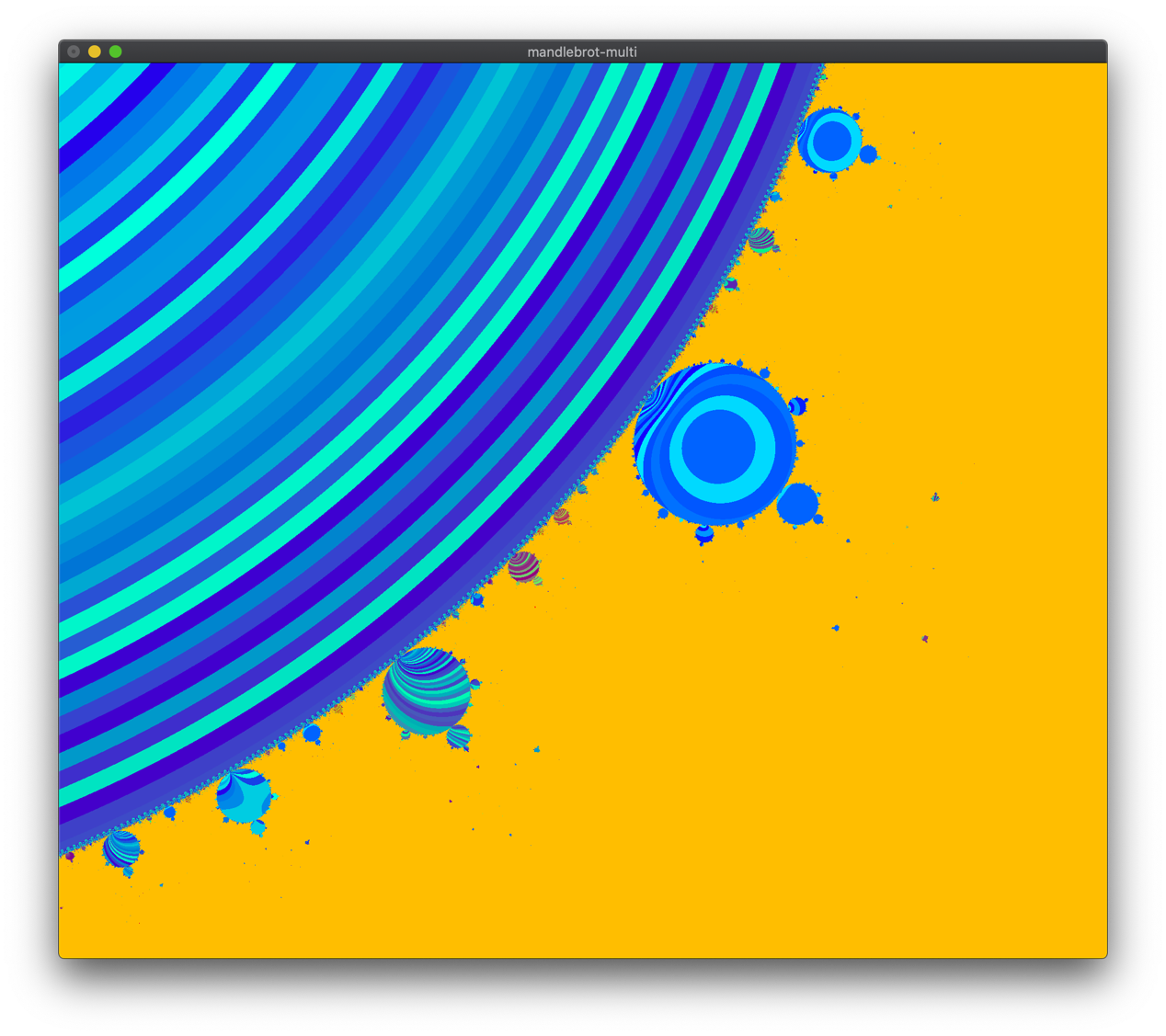


Figure 2多线程运行结果