

OpenMP 加速实现图像边缘检测

一、要求

Sobel 常用于图像的边缘检测，计算公式如下：

$$Gx = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * A \quad Gy = \begin{bmatrix} -1 & -2 & 1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * A$$

$$G = \sqrt{Gx^2 + Gy^2}$$

其中 A 是二维图像， G 为检测到的梯度强度。

请用 OpenMP 加速方法计算边缘检测（边缘强度），评估采用 OpenMP 带来的加速比。自选图像，要求图像像素数大于 2,000,000。

二、调试与结果

2.1 调试过程

在 Visual Studio 2019 集成开发环境下，建立 OpenMP 工程，并结合 OpenCV 库进行实验，所选图像的像素数为 $3943 \times 2628 = 10,362,204$ 。注意，应当在 Debug 模式下运行代码，因为 Release 模式下编译器会对代码进行优化，部分语句并行执行以提升运行速度。

边缘检测的函数定义如第三部分的 edgeDetection() 所示，一共有两层循环。若使用 OpenMP 进行并行加速并且在循环外部声名 Gx、Gy，需要将 Gx、Gy 变量用 private 关键字声明成线程私有的变量，其他线程无法访问私有副本，否则各线程使用的 Gx、Gy 会存储在同一内存空间中，各线程读写访问均会改变变量的值，造成最终计算错误。具体语句为：“#pragma omp parallel for private(Gx, Gy)”。

2.2 运行结果

（1）边缘检测结果如图 2 所示，可以看到，原始图片的边缘信息已经被较好地检测出来。

（2）进行三次实验，对比 OpenMP 带来的加速效果，结果如表 1 所示。计算可知，本例中使用 OpenMP 带来的加速比约为 3.4。



图 1 原始图像



图 2 边缘检测结果

表 1 OpenMP 加速效果对比

	无 OpenMP 加速	OpenMP 加速	加速比
第一次	2.105 s	0.633 s	3.325
第二次	1.978 s	0.535 s	3.697
第三次	2.111 s	0.636 s	3.319