



南开大学
Nankai University

南 开 大 学

计 算 机 学 院

并行程序设计实验报告

SIMD 编程报告

洪一帆 1811363

年级：2018 级

专业：计算机科学与技术

指导教师：王刚

2022 年 9 月 30 日

摘要

关键字: Parallel

目录

1

2 总结

每页底下的水印
如图??所示



图 1: Caption

表

N/n\Algo	naive-conv	naive-pool	omp-conv	omp-pool
64/2	0.0167	0.01255	0.04142	Effective Physical Core Utilization (utilized cores out of 4)
64/4	0.03599	0.0394	0.0458	0.0421

表 1: 性能测试结果 (4 线程)(单位:ms)

带单元格表格

<i>Cost</i>		To				
		<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>
From	<i>B</i>	7	0	1	3	8
	<i>C</i>	8	1	0	2	7
	<i>D</i>	8	3	2	0	5

表 2: 结点 C 距离向量表 (无毒性逆转)

伪代码

Algorithm 1 初始化 obj 文件信息——对应 MeshSimplify 类中 readfile 函数,Face 类 calMatrix 函数

Input: obj 文件, 顶点、边、面列表

Output: 是否读取成功

1: function calMatrix(*Face*)

2: $normal \leftarrow e1 \times e2$

3: $normal \leftarrow normal / normal.length$

4: $temp[] \leftarrow normal.x, normal.y, normal.z, normal \cdot Face.v1$

5: $Matrix[i][j] = temp[i] * temp[j]$

6: return *Matrix*

7: end function

8: 根据 obj 的 v 和 f 区分点面信息, 读取并加入列表

9: $scale \leftarrow$ 记录点坐标中距离原点最远的分量, 以便后续 OpenGL 进行显示

- 10: $ori \leftarrow$ 记录中心点, 便于 OpenGL 显示在中心位置, 避免有的 obj 偏移原点较多
- 11: 根据三角面片信息, 计算一个面的三条边
- 12: 计算每个面的矩阵 $\leftarrow calMatrix$
- 13: 将每个面的矩阵加到各点, 由点维护
- 14: return True

代码

逐列访问平凡算法

```

1 void ord()
2 {
3     double head,tail,freq,head1,tail1,timess=0; // timers
4     init(N);
5     QueryPerformanceFrequency((LARGE_INTEGER*)&freq);
6     QueryPerformanceCounter((LARGE_INTEGER*)&head);
7     for (int i=0; i<NN; i++)
8         for (int j=0; j<NN; j++)
9             col_sum[i] += (b[j][i]*a[j]);
10    QueryPerformanceCounter((LARGE_INTEGER*)&tail);
11    cout << "\nordCol:" <<(tail-head)*1000.0 / freq << "ms" << endl;
12 }

```

```

#include<stdio.h>
int main(){
int a,b;
// 输入变量
scanf("%d%d",&a,&b);
// 输出结果
printf("Hello World %d\n",a+b);
return 0;

```

参考文献 [?]

多行公式

$$a + b = a + b \quad (1)$$

$$\frac{a+b}{a-b} \quad (2)$$

行内公式: $\sum_{i=1}^N$

超链接 [YouTube](#)

带标号枚举

1. 1

2. 2

不带标号枚举

- 1
- 2

切换字体大小