

Report on using CPP code for speedup

Li Husen

November 30, 2020

1 Introduction

The final step in the previous code on Re-Tiling and PPS, takes too much time to run. Therefore, according to the suggestion of Mr. Chen and the CPP code online, try to speed up the code. And use MEX programming to realize MATLAB and CPP file connection.

This report has two contents. First, matlab code and CPP code were used respectively to realize the last step of the Retiling, and CPP code was found to take longer time to realize. Second, call MEX file in MATLAB. However, I met some problems and spent a lot of time, still unresolved. The detailed problems are explained in the third section.

2 MATLAB or CPP

Matlab and CPP code time comparison:

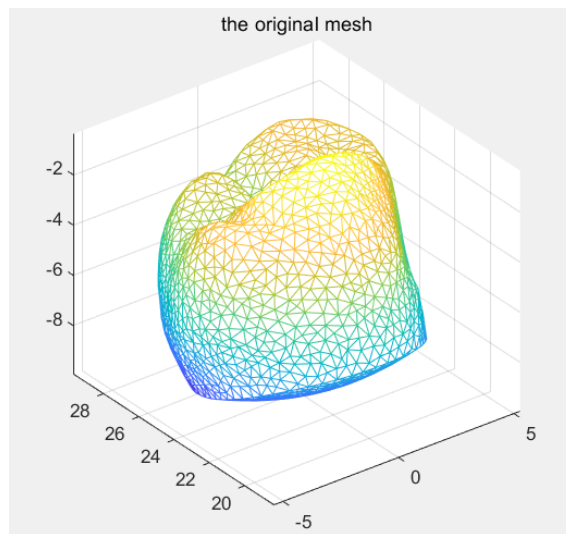


Figure 1

model name	number of vertex	number of face
model	1967	3930
matlab	3.450	(Unit: second)
CPP	72.305	

screenshot as follows: So the conclusion is that the CPP code is slower to

```
Removing Old Vertices...
历时 3.453115 秒。

done (0 vertices retained from removal)
The run time is: 72.305s
ROV完成。
```

Figure 2

implement.

3 mex programming

Since the CPP file is already debugged, we need to call it in MATLAB by writing an appropriate MEX file. Once we have written the MEX file, we can debug and generate dynamic libraries in VS2019.

However, when I connect the written CPP to the MEX file, there are some errors about the CGAL file (the CGAL library is used when the CPP code is implemented). These errors took a long time and were not resolved, so it is temporarily impossible to directly run the CPP code introduced in MATLAB.

Some of the wrong screenshots are as follows:

✖ C2589	"(:::"右边的非法标记	myMexVcTest	Dimension.h	28
✖ C2062	意外的类型"unknown-type"	myMexVcTest	Dimension.h	28
✖ C2059	语法错误:""	myMexVcTest	Dimension.h	28
⚠ C4003	类函数宏的调用"min"参数不足	myMexVcTest	hash_functions.h	60

Figure 3

4 Conclusion

There may be better ways to speed up with CPP, but the guy's code clearly doesn't.

This time I spent most of my time configuring the CGAL library, and modifying errors in the CPP file (there were some errors in the CPP file, which had to be modified before running).

怕老师不明白我蹩脚的英文，第一次英文报告，附上中文原文。

关于 CPP 代码提速的报告

李虎森

November 30, 2020

1 前言

在之前关于 Re-tiling 和 PPS 结合，网格重新三角化的代码中，最后一个步骤——移除原始顶点时花费时间较多，于是根据陈老师的建议，看看借鉴网上 CPP 代码是否能够给代码提速，并且利用 mex 编程实现 matlab 和 cpp 文件连接。

此次报告有两个内容，一是分别用 matlab 代码和 cpp 代码，实现算法最后一步——移除原始顶点，发现 cpp 代码实现耗时更长。二是关于在 matlab 中调用 mex 文件，由于在 VS2019 中已经实现了 cpp 主要算法，所以只需要写一个 mex 文件，就能实现在 matlab 中调用 cpp 文件，但是遇到点问题，花了很多时间也不知道怎么解决，详细问题在第 3 小结说明。

2 MATLAB or CPP

模型在 matlab 和 CPP 中耗时比较：

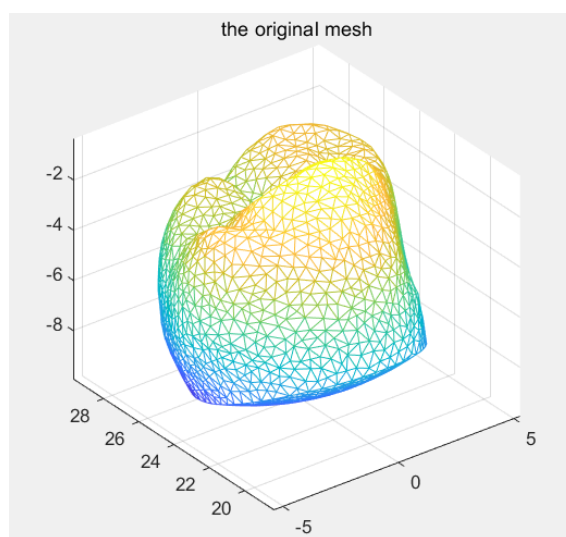


Figure 1

模型名称	顶点数	面数
modle	1967	3930
matlab 耗时	3.450	(单位: 秒)
CPP 耗时	72.305	

截图如下:

```
Removing Old Vertices...
历时 3.453115 秒。

done (0 vertices retained from removal)
The run time is: 72.305s
ROV完成。
```

Figure 2

所以暂时得出的结论是: 该 CPP 代码实现较慢。

3 mex 编程

因为 CPP 文件已经调试好了, 所以要在 MATLAB 中调用, 只需要写一个合适的 mex 文件。当写好 mex 文件过后, 可以先在 VS2019 中进行调试、生成动态库。

但是, 将写好的 CPP 与 mex 文件连接时, 出现了一些关于 CGAL 文件 (因为 CPP 代码实现时, 用到了 CGAL 库) 报错, 这些错误花了较长时间也没解决, 所以暂时不能直接在 MATLAB 中运行前面介绍的 CPP 代码。(因为单独运行 CPP 代码是可以的, 所以应该不是 CGAL 库的配置出错)。

一些错误的截图如下:

✖ C2589	"("::"右边的非法标记	myMexVcTest	Dimension.h	28
✖ C2062	意外的类型"unknown-type"	myMexVcTest	Dimension.h	28
✖ C2059	语法错误:""	myMexVcTest	Dimension.h	28
⚠ C4003	类函数宏的调用"min"参数不足	myMexVcTest	hash_functions.h	60

Figure 3

4 小结

关于移除原始顶点, 可能有更好的 CPP 提速思路, 但是这位大佬写的代码显然不能。

这次时间主要花在 CGAL 库的配置和修改 cpp 文件中的错误上 (cpp 文件中有一些错误, 要修改后才能运行)。