Globally Consistent Normal Orientation for Point Clouds by Regularizing the Winding-Number Field

Code of Globally Consistent Normal Orientation for Point Clouds by Regularizing the Winding-Number Field.

We will publish the code on GitHub after the paper is accepted.

Currently we have only tested our code on 64-bit windows systems and Visual Studio 2022 Professional.

Dependence

- CGAL
- Eigen3
- Boost

Please using vcpkg to install dependent libraries!!!

- .\vcpkg install boost:x64-windows
- .\vcpkg install cgal:x64-windows
- .\vcpkg install eigen:x64-windows
- .\vcpkg integrate install

MSVC on Windows

```
Download this project: NormalOrientation
```

Open Cmake-GUI

```
Where is the source code: NormalOrientation

Where to build the binaries: NormalOrientation/build
```

note: check the location of dependencies and install. It is recommended to use vcpkg to add dependencies.

Configure->Generate->Open Project

ALL_BUILD -> Build
Turn Debug to Release -> ALL_BUILD -> Build

Please set MAIN as Startup Project, and make the following changes:

Properties -> Configuration Properties -> C/C++ -> Code Generation ->

- Enable Parallel Code Generation : Yes
- Enable Enhanced Instruction Set : AVX2
- Floating Point Model : Fast

Properties -> Configuration Properties -> C/C++ -> Language -> Open MP Support : Yes

Test

The example is in 'MAIN'.

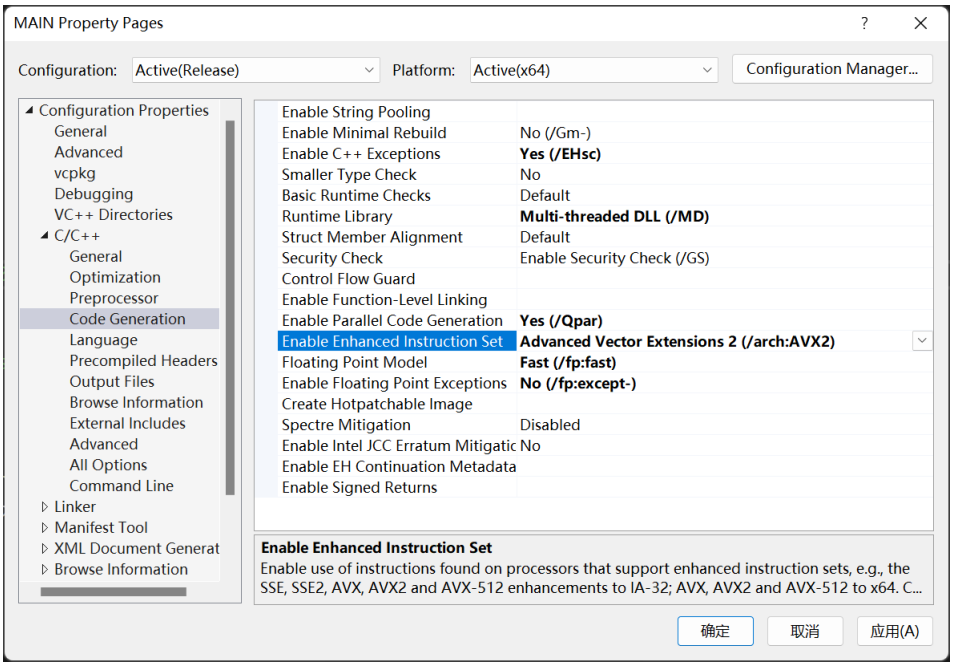
All the files is in 'NormalOrientation\data'.

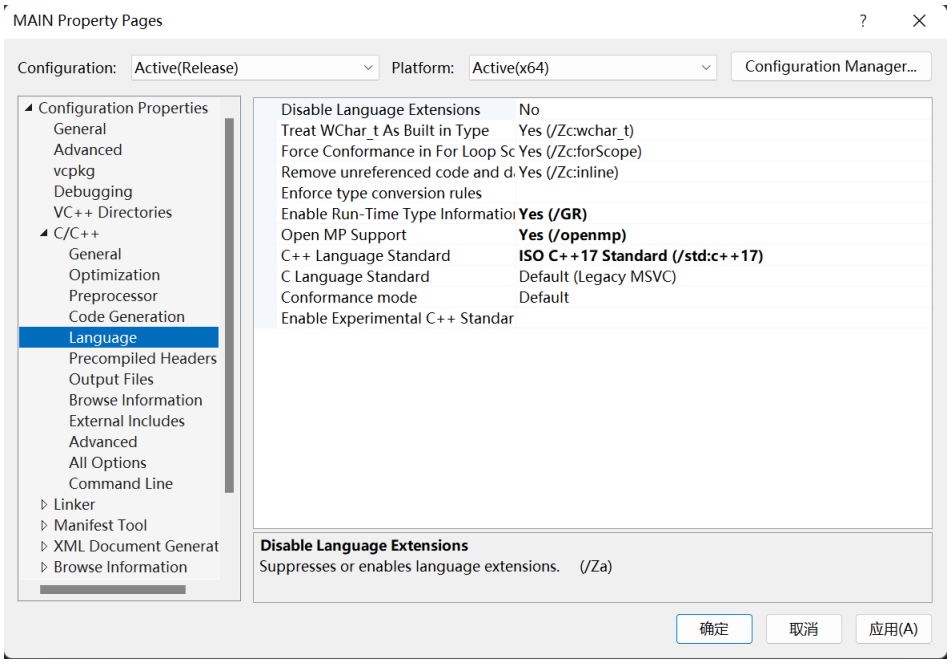
The output files is in 'NormalOrientation\data\Out'.

We put the result of our operation in 'NormalOrientation\data\MyResult', you can use it for comparison to know whether the program is running correctly.

IMPORTANT NOTE:

This code is not optimized for speed, but for clarity.
Please open Openmp and AVX2 in Visual Studio to speed up the code.
Please set the floating point model to fast in Visual Studio to speed up the code.
The default number of Openmp parallel threads is 28, set according to an AMD Ryzen 5950x CPU, please set different number of threads according to the CPU you use to get the best running effect.





Testing Platform

- Windows 10
- Visual Studio 2022 Professional
- AMD Ryzen 5950X
- 32GB Memory

Considering that most computers do not have this configuration, this commit does not support acceleration.

In order to allow you to view the optimization process in more detail,
we have not set the optimization stop condition, you can manually stop the optimization, and view all iteration results in the data\out folder