cmake\_minimum\_required(VERSION 3.5.1)

project(grid\_map\_filters)

# Better with parallelized algorithms.

#set(CMAKE\_CXX\_FLAGS "-std=c++11 ${CMAKE\_CXX\_FLAGS} -ffast-math")

# Better with serial algorithms.

set(CMAKE\_CXX\_STANDARD 11)

# We want performance (fast-math) but also need a representation for NaN values to represent missing values.

# Therefore, we disable the finite-math-only flag that was set by fast-math.

set(CMAKE\_CXX\_FLAGS "${CMAKE\_CXX\_FLAGS} -ffast-math -fno-finite-math-only")

add\_compile\_options(-Wall -Wextra -Wpedantic)

set(CMAKE\_EXPORT\_COMPILE\_COMMANDS ON)

# Other possible options.

#set(CMAKE\_CXX\_FLAGS "${CMAKE\_CXX\_FLAGS} -march=native -mtune=native -ffast-math")

#set(TARGET\_ARCHITECTURE "kaby-lake")

## Find catkin macros and libraries

find\_package(catkin REQUIRED COMPONENTS

grid\_map\_core

grid\_map\_ros

grid\_map\_msgs

filters

)

## System dependencies are found with CMake's conventions

find\_package(PkgConfig REQUIRED)

pkg\_check\_modules(TBB "tbb")

if (TBB\_FOUND)

add\_definitions(

${TBB\_CFLAGS}

${TBB\_CFLAGS\_OTHER}

)

else()

message([FATAL\_ERROR] "tbb module not found")

endif ()

find\_package(OpenCV REQUIRED)

###################################

## catkin specific configuration ##

###################################

## The catkin\_package macro generates cmake config files for your package

## Declare things to be passed to dependent projects

## INCLUDE\_DIRS: uncomment this if you package contains header files

## LIBRARIES: libraries you create in this project that dependent projects also need

## CATKIN\_DEPENDS: catkin\_packages dependent projects also need

## DEPENDS: system dependencies of this project that dependent projects also need

catkin\_package(

INCLUDE\_DIRS

include

${TBB\_INCLUDE\_DIRS}

LIBRARIES

${PROJECT\_NAME}

CATKIN\_DEPENDS

grid\_map\_ros

grid\_map\_core

grid\_map\_msgs

filters

DEPENDS

OpenCV

)

###########

## Build ##

###########

## Specify additional locations of header files

include\_directories(

include

SYSTEM

${catkin\_INCLUDE\_DIRS}

${TBB\_INCLUDE\_DIRS}

${OpenCV\_INCLUDE\_DIRS}

)

## Declare a cpp library

add\_library(${PROJECT\_NAME}

src/ThresholdFilter.cpp

src/MinInRadiusFilter.cpp

src/MeanInRadiusFilter.cpp

src/MedianFillFilter.cpp

src/MockFilter.cpp

src/NormalVectorsFilter.cpp

src/CurvatureFilter.cpp

src/NormalColorMapFilter.cpp

src/LightIntensityFilter.cpp

src/MathExpressionFilter.cpp

src/SlidingWindowMathExpressionFilter.cpp

src/DuplicationFilter.cpp

src/DeletionFilter.cpp

src/ColorFillFilter.cpp

src/ColorMapFilter.cpp

src/ColorBlendingFilter.cpp

src/SetBasicLayersFilter.cpp

src/BufferNormalizerFilter.cpp

)

target\_link\_libraries(${PROJECT\_NAME}

${catkin\_LIBRARIES}

${TBB\_LIBRARIES}

${OpenCV\_LIBRARIES}

)

#############

## Install ##

#############

# Mark executables and/or libraries for installation

install(

TARGETS ${PROJECT\_NAME}

ARCHIVE DESTINATION ${CATKIN\_PACKAGE\_LIB\_DESTINATION}

LIBRARY DESTINATION ${CATKIN\_PACKAGE\_LIB\_DESTINATION}

RUNTIME DESTINATION ${CATKIN\_PACKAGE\_BIN\_DESTINATION}

)

# Mark cpp header files for installation

install(

DIRECTORY include/${PROJECT\_NAME}/

DESTINATION ${CATKIN\_PACKAGE\_INCLUDE\_DESTINATION}

FILES\_MATCHING PATTERN "\*.hpp"

)

# Mark other files for installation

install(

FILES filter\_plugins.xml

DESTINATION ${CATKIN\_PACKAGE\_SHARE\_DESTINATION}

)

#############

## Testing ##

#############

if (CATKIN\_ENABLE\_TESTING)

catkin\_add\_gtest(${PROJECT\_NAME}-test

test/test\_grid\_map\_filters.cpp

test/median\_fill\_filter\_test.cpp

test/threshold\_filter\_test.cpp

)

target\_include\_directories(${PROJECT\_NAME}-test PRIVATE

include

)

target\_include\_directories(${PROJECT\_NAME}-test SYSTEM PUBLIC

${catkin\_INCLUDE\_DIRS}

)

target\_link\_libraries(${PROJECT\_NAME}-test

gmock

gtest

${PROJECT\_NAME}

)

###################

## Code\_coverage ##

###################

find\_package(cmake\_code\_coverage QUIET)

if(cmake\_code\_coverage\_FOUND)

add\_gtest\_coverage(

TEST\_BUILD\_TARGETS

${PROJECT\_NAME}-test

)

endif()

endif()

#################

## Clang\_tools ##

#################

find\_package(cmake\_clang\_tools QUIET)

if(cmake\_clang\_tools\_FOUND)

add\_default\_clang\_tooling(

DISABLE\_CLANG\_FORMAT

)

endif(cmake\_clang\_tools\_FOUND)