cmake\_minimum\_required(VERSION 3.5.1)

project(grid\_map\_pcl)

set(CMAKE\_CXX\_STANDARD 14)

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

set(CMAKE\_EXPORT\_COMPILE\_COMMANDS ON)

set(SRC\_FILES

src/GridMapPclConverter.cpp

src/GridMapPclLoader.cpp

src/helpers.cpp

src/PclLoaderParameters.cpp

src/PointcloudProcessor.cpp

)

set(CATKIN\_PACKAGE\_DEPENDENCIES

grid\_map\_core

grid\_map\_msgs

grid\_map\_ros

pcl\_ros

roscpp

)

find\_package(OpenMP QUIET)

if (OpenMP\_FOUND)

add\_compile\_options("${OpenMP\_CXX\_FLAGS}")

add\_definitions(-DGRID\_MAP\_PCL\_OPENMP\_FOUND=${OpenMP\_FOUND})

endif()

find\_package(PCL REQUIRED

COMPONENTS

common

features

filters

io

kdtree

segmentation

surface

)

## Find catkin macros and libraries

find\_package(catkin REQUIRED

COMPONENTS

${CATKIN\_PACKAGE\_DEPENDENCIES}

)

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

## 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

LIBRARIES

${PROJECT\_NAME}

yaml-cpp

CATKIN\_DEPENDS

${CATKIN\_PACKAGE\_DEPENDENCIES}

DEPENDS

PCL

)

###########

## Build ##

###########

# Library.

add\_library(${PROJECT\_NAME}

${SRC\_FILES}

)

add\_dependencies(${PROJECT\_NAME}

${catkin\_EXPORTED\_TARGETS}

)

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

include

)

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

${catkin\_INCLUDE\_DIRS}

${EIGEN3\_INCLUDE\_DIR}

${OpenMP\_CXX\_INCLUDE\_DIRS}

${PCL\_COMMON\_INCLUDE\_DIRS}

${PCL\_FEATURES\_INCLUDE\_DIRS}

${PCL\_FILTERS\_INCLUDE\_DIRS}

${PCL\_IO\_INCLUDE\_DIRS}

${PCL\_KDTREE\_INCLUDE\_DIRS}

${PCL\_SEGMENTATION\_INCLUDE\_DIRS}

${PCL\_SURFACE\_INCLUDE\_DIRS}

)

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

${catkin\_LIBRARIES}

${OpenMP\_CXX\_LIBRARIES}

yaml-cpp

)

# Node.

add\_executable(grid\_map\_pcl\_loader\_node

src/grid\_map\_pcl\_loader\_node.cpp

)

add\_dependencies(grid\_map\_pcl\_loader\_node

${PROJECT\_NAME}

)

target\_include\_directories(grid\_map\_pcl\_loader\_node PRIVATE

include

)

target\_include\_directories(grid\_map\_pcl\_loader\_node SYSTEM PUBLIC

${catkin\_INCLUDE\_DIRS}

${EIGEN3\_INCLUDE\_DIR}

)

target\_link\_libraries(grid\_map\_pcl\_loader\_node

${PROJECT\_NAME}

${catkin\_LIBRARIES}

)

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

## Install ##

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

install(

TARGETS

${PROJECT\_NAME}

grid\_map\_pcl\_loader\_node

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

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

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

)

install(

DIRECTORY

include/${PROJECT\_NAME}/

DESTINATION ${CATKIN\_PACKAGE\_INCLUDE\_DESTINATION}

FILES\_MATCHING PATTERN "\*.hpp"

)

install(

DIRECTORY

doc

config

launch

DESTINATION ${CATKIN\_PACKAGE\_SHARE\_DESTINATION}

)

install(

DIRECTORY

test

DESTINATION ${CATKIN\_PACKAGE\_SHARE\_DESTINATION}

FILES\_MATCHING PATTERN "\*.pcd"

)

install(

FILES

README.md

DESTINATION ${CATKIN\_PACKAGE\_SHARE\_DESTINATION}

)

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

## Testing ##

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

if(CATKIN\_ENABLE\_TESTING)

set(CMAKE\_CXX\_FLAGS "${CMAKE\_CXX\_FLAGS} -pthread")

## Add gtest based cpp test target and link libraries

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

test/test\_grid\_map\_pcl.cpp

test/GridMapPclLoaderTest.cpp

test/HelpersTest.cpp

test/PointcloudProcessorTest.cpp

test/test\_helpers.cpp

test/PointcloudCreator.cpp

)

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

include

)

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

${catkin\_INCLUDE\_DIRS}

${EIGEN3\_INCLUDE\_DIR}

${OpenMP\_CXX\_INCLUDE\_DIRS}

)

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

${PROJECT\_NAME}

${catkin\_LIBRARIES}

)

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

## 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()

endif(cmake\_clang\_tools\_FOUND)