

5.4) How to unwarp omnidirectional images

unwarpImage.cpp

```
#include "unwrapper.h"
#include "config.h"

int main(int argc, char* argv[])
{
    if(argc==2)
    {
        // Initialisation step
        unwrapper UW; //Object creation
        UW.setCenter(658, 475); //Center of the image
        UW.setInnerAndOuterRadius(84, 468); //Inner and outer radius
        UW.setScale(.5, .5); //Output image scale in x and y direction (1, 1) gives the unscaled
        image.
        UW.initialize(); // Initializes the object. Precomputes the mapping coordinates and
        uses for the subsequent images.

        // Unwarp the images
        cv::Mat inputImg,outputImg;
        inputImg = cv::imread(argv[1], CV_LOAD_IMAGE_COLOR); // Read the file

        UW.unwrapImage(inputImg, outputImg, 1);

        // save as png
        std::vector<int> compression_params;
        compression_params.push_back(CV_IMWRITE_PNG_COMPRESSION);
        compression_params.push_back(9);
        cv::imwrite("unwarpImage.png", outputImg, compression_params);
    }

    return 0;
}
```

Explanation :

- An unwrapper object is created with the estimated parameters for our dataset.

```
unwrapper UW;
UW.setCenter(658, 475);
UW.setInnerAndOuterRadius(84, 468);
UW.setScale(.5, .5);
UW.initialize();
```

- Then, the object is initialize : the mapping coordinates is precomputed.

```
UW.initialize();
```

- This object can be used to unwarp images.

CMakeLists.txt

```
project(ImageUnwarping)
cmake_minimum_required(VERSION 2.6)

add_definitions( -Wall )

set(libIPDS_include_dir "/home/courbon/IPDS/IPDSCodes/IPDSLib/include/")
set(libIPDS_lib_dir "/home/courbon/IPDS/IPDSCodes/IPDSLib/build/lib/")

# TRY TO FIND OpenCV
set(CMAKE_MODULE_PATH ${CMAKE_MODULE_PATH} "${CMAKE_SOURCE_DIR}/Modules/")
find_package ( OpenCV)

IF(NOT OpenCV_FOUND)
  # TRY TO FIND OpenCV2
  find_package ( OpenCV2)

  IF(OpenCV2_FOUND)
    MESSAGE(STATUS "OpenCV2 found")
    include_directories(${OpenCV2_INCLUDE_DIRS})
  endif()
else()
  include_directories(${OpenCV_INCLUDE_DIR})
endif()

IF(OpenCV_FOUND OR OpenCV2_FOUND)
  #Inclusion de la bibliothèque
  include_directories(${libIPDS_include_dir})

  link_directories(${libIPDS_lib_dir})

  add_executable ( ${PROJECT_NAME} unwarplImage.cpp)

  IF(OpenCV2_FOUND)
    target_link_libraries ( ${PROJECT_NAME} IPDS ${OpenCV2_CORE_LIBRARY} $
{OpenCV2_IMGPROC_LIBRARY} ${OpenCV2_HIGHGUI_LIBRARY})
  ELSE(OpenCV2_FOUND)
    target_link_libraries ( ${PROJECT_NAME} IPDS ${OpenCV_LIBS})
  ENDIF(OpenCV2_FOUND)

endif()
```

Note that the paths to the IPDS include and library (libIPDS_include_dir and libIPDS_lib_dir) have to be modified to suit your set-up.

(the source code is in the archive ImageUnwarping.zip)