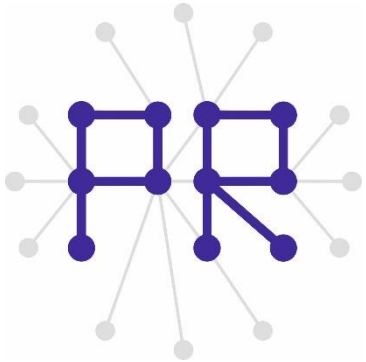
Multimedia Retrieval Exercise Course

10 Local Features: SIFT Feature Extraction by OpenCV

Kimiaki Shirahama, D.E.

Research Group for Pattern Recognition
Institute for Vision and Graphics
University of Siegen, Germany



Overview of Today's Lesson

- Install OpenCV from Source Files
  - ❑ Make OpenCV Source Files
  - ❑ Build OpenCV Modules
  - ❑ Test SIFT Feature Extraction

- More Information

Install OpenCV from Source Files

Download source files of OpenCV 3.4.0 and opencv_contrib 3.4.0

- https://github.com/Itseez/opencv/releases
- https://github.com/Itseez/opencv_contrib/releases

The version of OpenCV and the one of opencv_contrib must be the same

It is recommended to decompress OpenCV 3.4.0 and opencv_contrib 3.4.0 into the following folders:

- C:\opencv-3.4.0
- C:\opencv-3.4.0\opencv_contrib-3.4.0

NOTE: Since I have already installed OpenCV 3.4.0 and opencv_contrib 3.4.0 under these folders, I will use "C:\tmp\opencv-3.4.0" and "C:\tmp\opencv-3.4.0 \opencv_contrib-3.4.0" in the following discussion. Replace these with your own folders.

Download CMake which will be used to compile source files of OpenCV and opencv_contrib

- https://cmake.org/download/

Make OpenCV Source Files

1. Start CMake

2. Set these two folders
   (“build” folder will be automatically created)

2. Press “Configure”

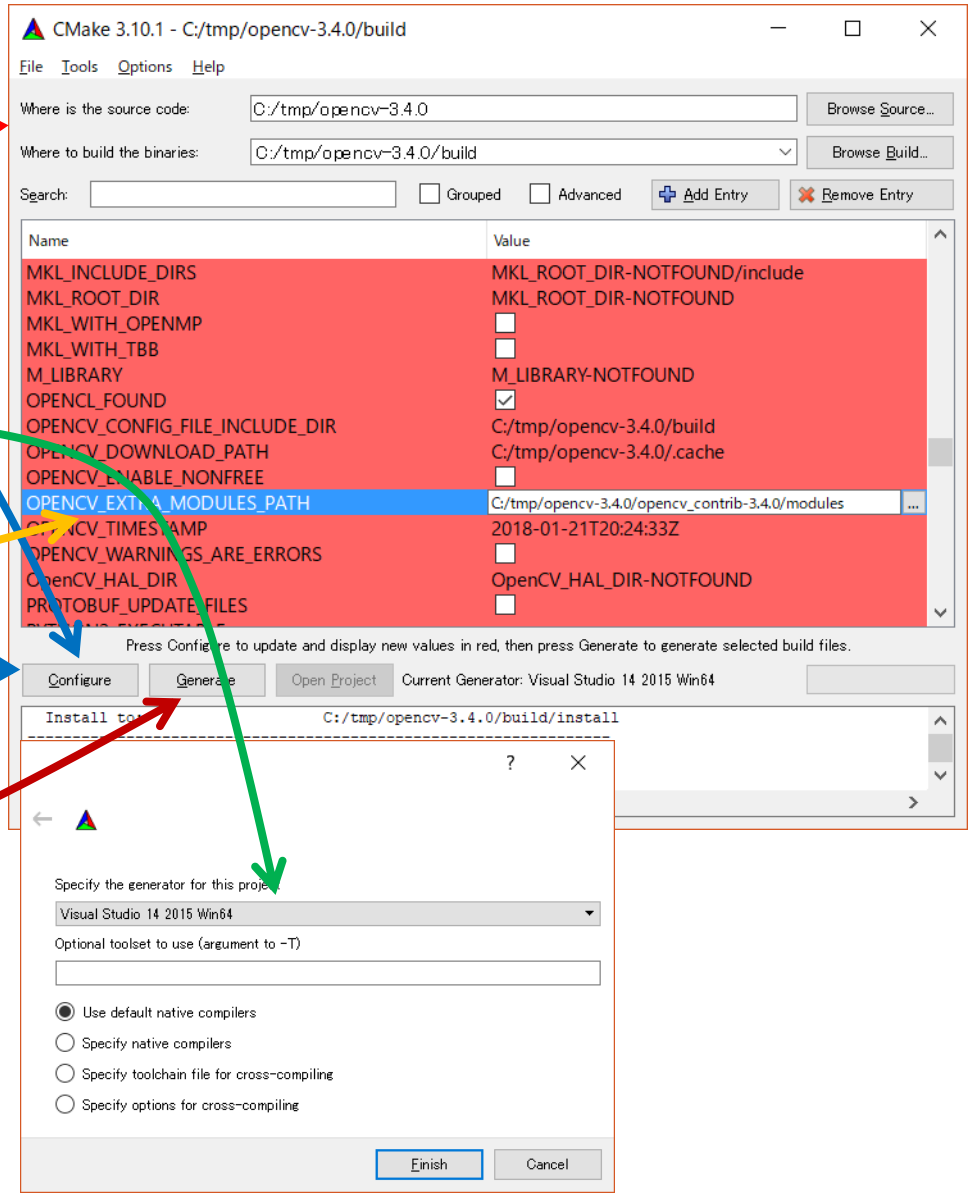
3. Choose “Visual Studio 14 2015 Win64”

4. Set “OPENCV_EXTRA_MODULES_PATH” to
“C:/opencv-3.4.0/opencv_contrib/modules/”

5 Press “Configure”

6. If “BUILD_opencv_hdf” is marked, remove it

7. Press “Generate



Build OpenCV Modules

1. Open C:\tmp\opencv-3.4.0\build\OpenCV.sln with Visual C++

2. In the “Solution Explorer”, right-click “INSTALL” in “CMakeTargets” and press “Set to start-up project”.
(Since my Visual C++ is Japanese, these translations may be different, but I guess you can find the above kind of entries)

3. Change “Debug” to “Release”

4. Press “Build solution” in “Build” in the menu bar.
This “Build” takes time. When everything is successful, a lot of OpenCV library files are generated under “C:\tmp\opencv-3.4.0\build\bin\Release”.

---

5. Add “C:\tmp\opencv-3.4.0\build\bin\Release” to “PATH” in environmental variables

6. Re-start the PC

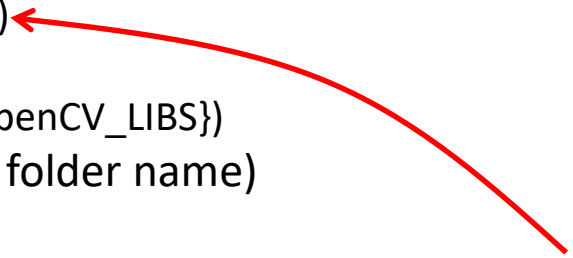
Test SIFT Feature Extraction (1/3)

1. Create a folder (in the following example, I create “SIFTTest”)

2. Under this folder, create “CMakeLists.txt” with the following content:

```
cmake_minimum_required(VERSION 2.8)
project(SIFTTest)
find_package( OpenCV REQUIRED)
include_directories( ${OpenCV_INCLUDE_DIRS})
add_executable(SIFTTest test.cpp)
set(CMAKE_BUILD_TYPE Release)
target_link_libraries(SIFTTest ${OpenCV_LIBS})
```

(Replace “SIFTTest” with your own folder name)



3. Under the folder at Step 1, create a source file (I create “test.cpp”) that has the content in the next slide.

Test SIFT Feature Extraction (2/3)

test.cpp


```
#include <iostream>
#include <opencv2/core.hpp>
#include <opencv2/highgui.hpp>
#include <opencv2/xfeatures2d.hpp>

using namespace std;
using namespace cv;
using namespace cv::xfeatures2d;

int main(int argc, char** argv) {

    string img_filename = "C:\\Users\\公章\\Documents\\MMR_exercise\\101_ObjectCategories\\accordion\\image_0001.jpg";
    cout << ">> Read " << img_filename << endl;
    Mat img = imread(img_filename, cv::IMREAD_GRAYSCALE);
    if (!img.data) {
        cout << "Cannot read the image" << endl;
        int a; cin >> a; return -1;
    }

    Ptr<SIFT> detectorSIFT = SIFT::create();
    vector<KeyPoint> keypoints;
    detectorSIFT->detect(img, keypoints);
    cout << ">> # of detected keypoints = " << keypoints.size() << endl;

    Mat img_keypoints;
    drawKeypoints(img, keypoints, img_keypoints, Scalar::all(-1), DrawMatchesFlags::DRAW_RICH_KEYPOINTS);

    imshow("SIFT", img_keypoints);
    waitKey(0);

    return 0;
}
```

Test SIFT Feature Extraction (3/3)

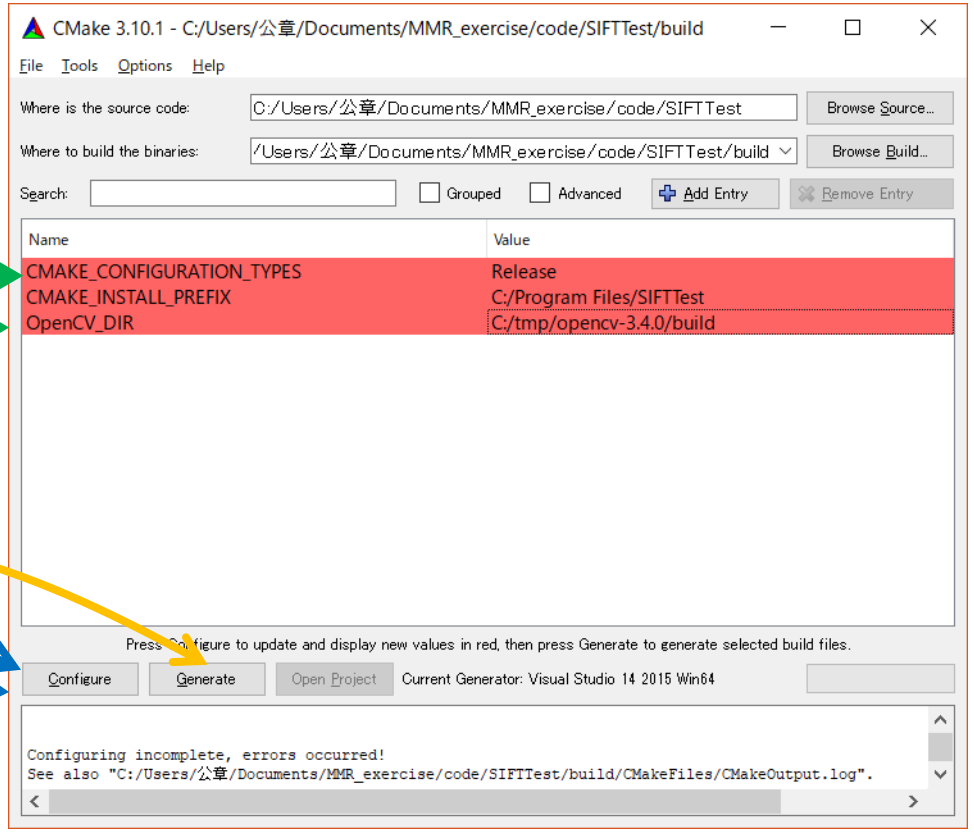
1. Set these folders as the one created at Step 1 and the one extended with “build”.

2. Press “Configure”

3. Set “CMAKE_CONFIGURATION_TYPES” to “Release” and set “OpenCV_DIR” to <your OpenCV folder/build>

4. Press “Configure”

5. Press “Generate”



6. Open SIFTTest.sln in the “build” folder with Visual C++
7. In “Solution Explorer”, right-click “SIFTTest” and press “Set to start-up project”.
8. In “Solution Explorer”, open “SIFTTest/Source files/test.cpp”.
9. Build and run the solution.

More Information

For Linux users, I think the install-instruction in opencv_contrib should work
https://github.com/opencv/opencv_contrib

Please refer to the official OpenCV website for more information about SIFT feature extraction
https://docs.opencv.org/trunk/d2/dca/group__xfeatures2d__nonfree.html

Especially, the information about Keypoint (local region) and its description can be found on the following pages:
https://docs.opencv.org/trunk/d0/d13/classcv_1_1Feature2D.html
https://docs.opencv.org/trunk/d5/dde/tutorial_feature_description.html

In the next lesson, we will study about “Feature Matching” in this tutorial page:
https://docs.opencv.org/trunk/d9/d97/tutorial_table_of_content_features2d.html