TECHNISCHE UNIVERSITÄT CHEMNITZ

Chapter 1: Grayscale

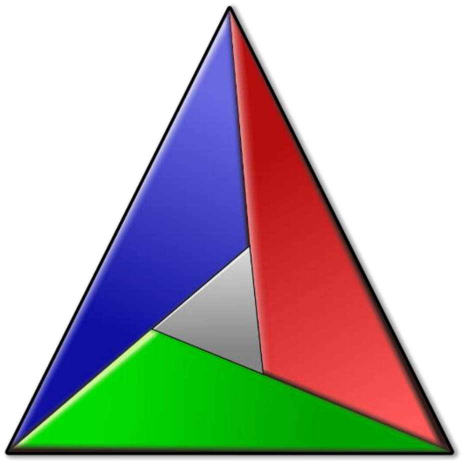
Tools: CMake, OpenCV

TECHNISCHE UNIVERSITÄT CHEMNITZ

CMake:

- Free open-source cross-platform tool to manage the build process of software
- Automatically creates the build environment for you
- Collects all libraries for your project
- Generates the project for your IDE (e.g. Makefile, Visual Studio, Qt Creator, Xcode, KDevelop), so you can use CMake on every OS to create a valid project file
- Needs a project description file “CMakeLists.txt”
- For the complete documentation please look at:

  http://www.cmake.org/documentation/



CMake example

Example for CMakeLists.txt

```
cmake_minimum_required(VERSION 2.8.10)

# set project name
project(Introduction)

# set compile flags
set(CMAKE_CXX_FLAGS "-std=c++11")
set(CMAKE_BUILD_TYPE "Release")

# find libraries
find_package(OpenCV REQUIRED)

# set include directories
include_directories(${OpenCV_INCLUDE_DIR})

# make executable
add_executable(${PROJECT_NAME} main.cpp)

# link against libraries
target_link_libraries(${PROJECT_NAME} ${OpenCV_LIBS})
```

Note:

The CMakeLists.txt files will be given to you for all exercises. You do not need to create or change them.

OpenCV

- Free open-source computer vision framework
- Is the standard in computer vision

Download:

https://opencv.org/downloads.html

Documentation:

https://docs.opencv.org/4.0.1/

OpenCV: Basic functions

cv::Mat class:

- stores the data of an image

- List of important members:
  - int cols: number of columns
  - int rows: number of rows
  - uchar* data: pointer to the image data

- List of important methods:
  - void Mat::**create**(int **rows**, int **cols**, int **type**):
    - **rows**: New number of rows.
    - **cols**: New number of columns.
    - **type**: New matrix type (CV_8U = grayscale image, CV_8UC3 = 24 bit color image)

  - template<typename T> T& Mat::**at**(int **i**, int **j**)
    - Access the data of a image
    - **i**: Index along the dimension 0 (rows)
    - **j**: Index along the dimension 1 (columns)

- https://docs.opencv.org/4.0.1/d3/d63/classcv_1_1Mat.html

OpenCV: Basic functions

TECHNISCHE UNIVERSITÄT CHEMNITZ

Accessing image data:

- Example:

```
cv::Mat img = cv::imread("lena.tiff")

for (int r = 0; r < rows; ++r)
{
      for (int c = 0; c < cols; ++c)
      {
            std::cout << img.at<uchar>(r, c) << std::endl;
      }
 }
```

- A full reference of cv::Mat is available here: https://docs.opencv.org/4.0.1/d3/d63/classcv_1_1Mat.html

OpenCV: Basic functions

Read an image from file:

- Mat **imread**(const string& **filename**, int **flags=1**)
  - **filename**: Name of file to be loaded.
  - **flags**: Flags specifying the color type of the loaded image:
    - IMREAD_ANYDEPTH - If set, return 16-bit/32-bit image when the input has the corresponding depth, otherwise convert it to 8-bit.
    - IMREAD_COLOR - If set, always convert image to color
    - IMREAD_GRAYSCALE - If set, always convert image to grayscale

- https://docs.opencv.org/4.0.1/ → imgcodecs. Image file reading and writing

Show an image:

- void **imshow**(const string& **winname**, InputArray **mat**)
  - **winname**: Name of the window.
  - **mat**: matrix (image) to be shown.

- https://docs.opencv.org/4.0.1/ → highgui. High-level GUI

OpenCV: Basic functions

Color conversion:

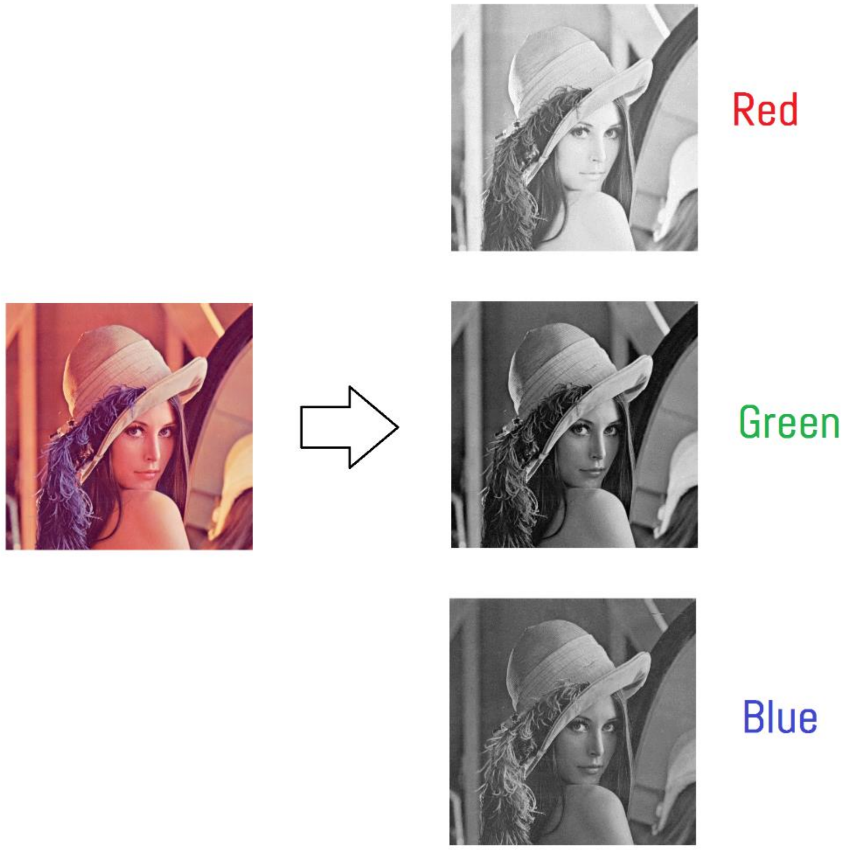
- void **cvtColor**(InputArray **src**, OutputArray **dst**, int **code**, int **dstCn=0**)
  - Converts an image from one color space to another.

  - **src**: input image: 8-bit unsigned, 16-bit unsigned (CV_16UC...), or single-precision floating-point
  - **dst**: output image of the same size and depth as **src**
  - **code**: color space conversion code (COLOR_BGR2GRAY, COLOR_RGB2GRAY, COLOR_GRAY2BGR, COLOR_GRAY2RGB)
  - **dstCn**: number of channels in the destination image; if the parameter is 0, the number of the channels is derived automatically from **src** and **code**

Exercise

First Exercise

- Compute the Grayscale Image of an RGB-Image
- In OpenCV, the channel weights for RGB-to-grayscale conversion are:
  - R*0.299 (R * 77/256)
  - G*0.587 (G * 150/256)
  - B*0.114 (B * 29/256)
- Weighting is done to account for human color perception
  → most sensitive to green, then red, then blue



Note:

This exercise is meant as a tutorial on how to get the files, use CMake, compile the code and start the program. It is very easy and you have to write only very few C++ code.

Exercise

Expected Output

Original RGB-Image

2 Grayscale Images