

Image Processing Basics in C++

*by Ismail Ari
October 5, 2008*

Introduction

This is a short guide which aims to make you familiar with implementing image processing (IP) applications in C++. In lectures, you use matrices as images and learn different IP algorithms to apply on these matrices. Now, you will learn how to do the basic operations in practice. The topics included are image input/output operations, getting and setting pixel values of an image, and applying some image processing algorithms. By these topics and your prior programming experience, you are supposed to develop your own solutions to the assigned problems.

Choosing an Image Processing Library

There are many different open source libraries available for image processing applications which you can find on the web. The outstanding ones are:

CIimg : A Templated Image Processing Library in C++

<http://cimg.sourceforge.net/>

The CIimg Library is a C++ toolkit providing simple classes and functions to load, save, process and display images in your own C++ code. It consists only of a single header file CIimg.h that must be included in your program source. It contains useful image processing algorithms for loading/saving, resizing/rotating, filtering, object drawing (text, lines, faces, ellipses, ...), etc. Images are instanced by a class able to represent images up to 4-dimension wide (from 1-D scalar signals to 3-D volumes of vector-valued pixels), with template pixel types. It depends on a minimal number of libraries : you can compile it with only standard C libraries. No need for exotic libraries and complex dependencies.

OpenCV: Open Source Computer Vision Library

<http://sourceforge.net/projects/opencvlibrary/>

This library is mainly aimed at real time computer vision. Some example areas would be Human-Computer Interaction (HCI); Object Identification, Segmentation and Recognition; Face Recognition; Gesture Recognition; Motion Tracking, Ego Motion, Motion Understanding; Structure From Motion (SFM); and Mobile Robotics.

In this guide, we are going to use CIimg library because of its simplicity and portability. Besides, its documentation is better than OpenCV.

Diving into CImg Library

Go to [CImg homepage](#) and download the compressed standard package. The package contains "CImg.h" header file which you will include in your code. If you like, also download the pre-compiled packages and see the demo to get an idea about the limits of the library.

Download the [reference manual](#). Copy the "Hello world" code (Sec. 2.1.2) into a file and try to build it using the instructions in Sec. 2.1.3. There are different options for different compilers on different platforms.

Read the introductory tutorial (Sec. 2.9) to see how to load images and display them. Then, learn to use image loops (Sec. 2.11) to get familiar with pixel handling, and read Sec. 2.13 to see how pixel data are stored with CImg.

Recall that `img.save("Output filename")` saves the image to the disk whereas `img.display("Output image")` shows the image in a display.

Finally, read Sec. 2.15 to see how to work with command line arguments.

We will cover "sample.cpp" and "tutorial.cpp" files together during P.S.

Other References

- [An Introduction to the CImg Library](#), Vicknesh Selvam.
- [Short Tutorial to the CImg & Argstream Libraries](#), ETHZ Computer Vision Laboratory