



Introduction to Computer Vision and OpenCV

A significant share of the information that we get from the world while we are awake is through sight. Our eyes do a wonderful job of swiveling about incessantly and changing focus as needed to see things. Our brain does an even more wonderful job of processing the information stream from both eyes and creating a 3D map of the world around us and making us aware of our position and orientation in this map. Wouldn't it be cool if robots (and computers in general) could see, and understand what they see, as we do?

For robots, seeing in itself is less of a problem—cameras of all sorts are available and quite easy to use. However, to a computer with a camera connected to it, the camera feed is technically just a time-varying set of numbers.

Enter computer vision.

Computer vision is all about making robots intelligent enough to take decisions based on what they see.

Why Was This Book Written?

In my opinion, robots today are like personal computers 35 years ago—a budding technology that has the potential to revolutionize the way we live our daily lives. If someone takes you 35 years ahead in time, don't be surprised to see robots roaming the streets and working inside buildings, helping and collaborating safely with humans on a lot of daily tasks. Don't be surprised also if you see robots in industries and hospitals, performing the most complex and precision-demanding tasks with ease. And you guessed it right, to do all this they will need highly efficient, intelligent, and robust vision systems.

Computer vision is perhaps the hottest area of research in robotics today. There are a lot of smart people all around the world trying to design algorithms and implement them to give robots the ability to interpret what they see intelligently and correctly. If you too want to contribute to this field of research, this book is your first step.

In this book I aim to teach you the basic concepts, and some slightly more advanced ones, in some of the most important areas of computer vision research through a series of projects of increasing complexity. Starting from something as simple as making the computer recognize colors, I will lead you through a journey that will even teach you how to make a robot estimate its speed and direction from how the objects in its camera feed are moving.

We shall implement all our projects with the help of a programming library (roughly, a set of prewritten functions that can execute relevant higher-level tasks) called OpenCV.

This book will familiarize you with the algorithm implementations that OpenCV provides via its built-in functions, theoretical details of the algorithms, and the C++ programming philosophies that are generally employed while using OpenCV. Toward the end of the book, we will also discuss a couple of projects in which we employ OpenCV's framework for algorithms of our own design. A moderate level of comfort with C++ programming will be assumed.

OpenCV

OpenCV (Open-source Computer Vision, opencv.org) is the Swiss Army knife of computer vision. It has a wide range of modules that can help you with a lot of computer vision problems. But perhaps the most useful part of OpenCV is its architecture and memory management. It provides you with a framework in which you can work with images and video in any way you want, using OpenCV's algorithms or your own, without worrying about allocating and deallocating memory for your images.

History of OpenCV

It is interesting to delve a bit into why and how OpenCV was created. OpenCV was officially launched as a research project within Intel Research to advance technologies in CPU-intensive applications. A lot of the main contributors to the project included members of Intel Research Russia and Intel's Performance Library Team. The objectives of this project were listed as:

- Advance vision research by providing not only open but also optimized code for basic vision infrastructure. (No more reinventing the wheel!)
- Disseminate vision knowledge by providing a common infrastructure that developers could build on, so that code would be more readily readable and transferable.
- Advance vision-based commercial applications by making portable, performance-optimized code available for free—with a license that did not require the applications to be open or free themselves.

The first alpha version of OpenCV was released to the public at the IEEE Conference on Computer Vision and Pattern Recognition in 2000. Currently, OpenCV is owned by a nonprofit foundation called OpenCV.org.

Built-in Modules

OpenCV's built-in modules are powerful and versatile enough to solve most of your computer vision problems for which well-established solutions are available. You can crop images, enhance them by modifying brightness, sharpness and contrast, detect shapes in them, segment images into intuitively obvious regions, detect moving objects in video, recognize known objects, estimate a robot's motion from its camera feed, and use stereo cameras to get a 3D view of the world—to name just a few applications. If, however, you are a researcher and want to develop a computer vision algorithm of your own for which these modules themselves are not entirely sufficient, OpenCV will still help you a lot by its architecture, memory-management environment, and GPU support. You will find that your own algorithms working in tandem with OpenCV's highly optimized modules make a potent combination indeed.

One aspect of the OpenCV modules that needs to be emphasized is that they are highly optimized. They are intended for real-time applications and designed to execute very fast across a variety of computing platforms from MacBooks to small embedded fitPCs running stripped down flavors of Linux.

OpenCV provides you with a set of modules that can execute roughly the functionalities listed in Table 1-1.

Table 1-1. *Built-in modules offered by OpenCV*

Module	Functionality
Core	Core data structures, data types, and memory management
Imgproc	Image filtering, geometric image transformations, structure, and shape analysis
Highgui	GUI, reading and writing images and video
Video	Motion analysis and object tracking in video
Calib3d	Camera calibration and 3D reconstruction from multiple views
Features2d	Feature extraction, description, and matching
Objdetect	Object detection using cascade and histogram-of-gradient classifiers
ML	Statistical models and classification algorithms for use in computer vision applications
Flann	Fast Library for Approximate Nearest Neighbors—fast searches in high-dimensional (feature) spaces
GPU	Parallelization of selected algorithms for fast execution on GPUs
Stitching	Warping, blending, and bundle adjustment for image stitching
Nonfree	Implementations of algorithms that are patented in some countries

In this book, I shall cover projects that make use of most of these modules.

Summary

I hope this introductory chapter has given you a rough idea of what this book is all about! The readership I have in mind includes students interested in using their knowledge of C++ to program fast computer vision applications and in learning the basic theory behind many of the most famous algorithms. If you already know the theory, and are interested in learning OpenCV syntax and programming methodologies, this book with its numerous code examples will prove useful to you also.

The next chapter deals with installing and setting up OpenCV on your computer so that you can quickly get started with some exciting projects!