Preface

One of the first times I ever encountered video and image processing was in a semester project at my fourth year of studying. The aim of the project was to design a system that automatically located the center and size of mushrooms in an image. Given this information a robot should pick the mushrooms. I was intrigued by the notion of a "seeing computer". Little did I know that this encounter would shape most parts (so far) of my professional life.

I decided to study video and image processing in depth and signed up for a master's program focusing on these topics. I soon realized that I had made a good choice, but was puzzled by the fact that the wonders of digital video and image processing often were presented in a strict mathematical manner. While this is fine for hardcore engineers (including me) and computer scientists, it makes video and image processing unnecessarily difficult for others. I really felt this was a pity and decided to do something about it—that was 15 years ago.

In this book the concepts and methods are described in a less mathematical manner and the language is in general casual. In order to assist the reader with the math that *is* used in the book Appendix B is included. In this regards this textbook is self-contained. Some of the key algorithms are exemplified in C-code. Please note that the code is neither optimal nor complete and merely serves as an additional input for comprehending the algorithms.

Another aspect that puzzled me as a student was that the textbooks were all about image processing, while we constructed systems that worked with video. Many of the methods described for image processing can obviously also be applied to video data. But video data add the temporal dimension, which is often the key to success in systems processing video. This book therefore aims at not only introducing image processing but also video processing. Moreover, the last two chapters of the book describe the process of designing and implementing real systems processing video data. On the website for the book you can find detailed descriptions of other practical systems processing video: http://www.vip.aau.dk.

I have tried to make the book as concise as possible. This has forced me to leave out details and topics that might be of interest to some readers. As a compromise each chapter is ended by a "Further Information" section wherein pointers to additional concepts, methods and details are given.

For Instructors   Each chapter is ended by a number of exercises. The first exercise after each chapter aims at assessing to what degree the students have understood the main concepts. If possible, it is recommended that these exercises are discussed within small groups. The following exercises have a more practical focus where concrete problems need to be solved using the different methods/algorithms presented in the associated chapters. Lastly one or more so-called additional exercises are present. These aim at topics not discussed directly in the chapters. The idea behind these exercises is that they can serve as self-studies where each student (or a small group of students) finds the solution by investigating other sources. They could then present their findings for other students.

Besides the exercises listed in the book I strongly recommend to combine those with examples and exercises where real images/videos are processed. Personally I start with ImageJ for image processing and EyesWeb for video processing. The main motivation for using these programs is that they are easy to learn and hence the students can focus on the video and image processing as opposed to a specific programming language, when solving the exercises. However, when it comes to building real systems I recommend using OpenCV or openFrameworks (EyesWeb or similar can of course also be used to build systems, but they do not generalize as well). To this end students of course need to have a course on procedural programming before or in parallel with the image processing course. To make the switch from ImageJ/Eyesweb to a more low-level environment like OpenCV, I normally ask each student to do an assignment where they write a program that can capture an image, make some image processing and display the result. When the student can do this he has a framework for implementing "all" other image processing methods. The time allocated for this assignment of course depends on the programming experiences of the students.

Enjoy!

Viborg, Denmark   Thomas B. Moeslund