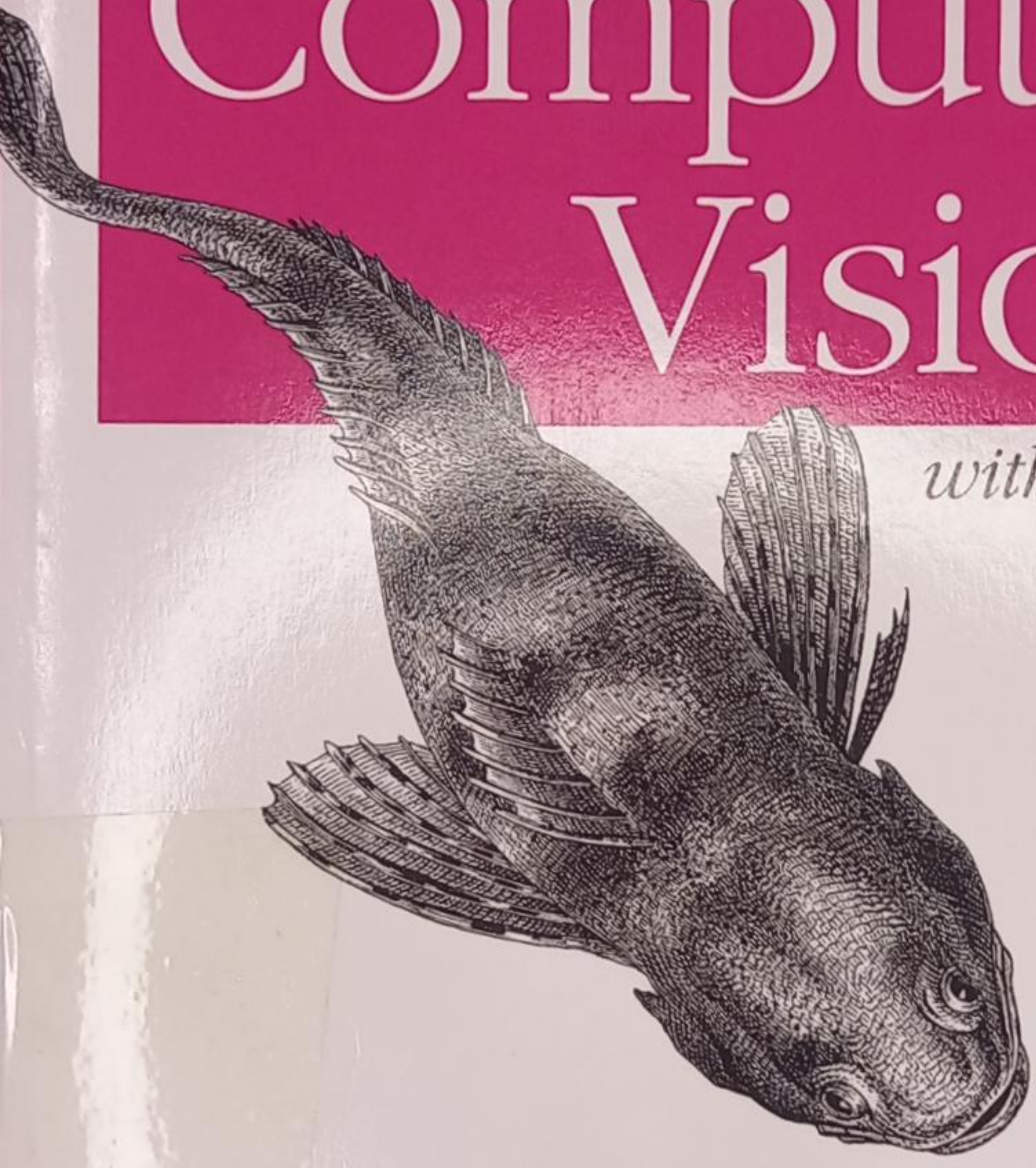
Tools and Algorithms for Analyzing Images

Programming Computer Vision with Python



O'REILLY®

Jan Erik Solem

Programming Computer Vision with Python

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI
TRUNG TÂM THÔNG TIN THƯ VIỆN
· 02 07
· 02427

Jan Erik Solem

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

Table of Contents

Preface vii

1. Basic Image Handling and Processing 1
1.1 PIL—The Python Imaging Library 1
1.2 Matplotlib 3
1.3 NumPy 7
1.4 SciPy 16
1.5 Advanced Example: Image De-Noising 23
Exercises 26
Conventions for the Code Examples 27

2. Local Image Descriptors 29
2.1 Harris Corner Detector 29
2.2 SIFT—Scale-Invariant Feature Transform 36
2.3 Matching Geotagged Images 44
Exercises 51

3. Image to Image Mappings 53
3.1 Homographies 53
3.2 Warping Images 57
3.3 Creating Panoramas 70
Exercises 77

4. Camera Models and Augmented Reality 79
4.1 The Pin-Hole Camera Model 79
4.2 Camera Calibration 84
4.3 Pose Estimation from Planes and Markers 86
4.4 Augmented Reality 89
Exercises 98

5. Multiple View Geometry **99**
5.1 Epipolar Geometry 99
5.2 Computing with Cameras and 3D Structure 107
5.3 Multiple View Reconstruction 113
5.4 Stereo Images 120
Exercises 125

6. Clustering Images **127**
6.1 *K*-Means Clustering 127
6.2 Hierarchical Clustering 133
6.3 Spectral Clustering 140
Exercises 145

7. Searching Images **147**
7.1 Content-Based Image Retrieval 147
7.2 Visual Words 148
7.3 Indexing Images 151
7.4 Searching the Database for Images 155
7.5 Ranking Results Using Geometry 160
7.6 Building Demos and Web Applications 162
Exercises 165

8. Classifying Image Content **167**
8.1 *K*-Nearest Neighbors 167
8.2 Bayes Classifier 175
8.3 Support Vector Machines 179
8.4 Optical Character Recognition 183
Exercises 189

9. Image Segmentation **191**
9.1 Graph Cuts 191
9.2 Segmentation Using Clustering 200
9.3 Variational Methods 204
Exercises 206

10. OpenCV **209**
10.1 The OpenCV Python Interface 209
10.2 OpenCV Basics 210
10.3 Processing Video 213
10.4 Tracking 216
10.5 More Examples 223
Exercises 226

A. Installing Packages 227
A.1 NumPy and SciPy 227
A.2 Matplotlib 228
A.3 PIL 228
A.4 LibSVM 228
A.5 OpenCV 229
A.6 VLFeat 230
A.7 PyGame 230
A.8 PyOpenGL 230
A.9 Pydot 230
A.10 Python-graph 231
A.11 Simplejson 231
A.12 PySQLite 232
A.13 CherryPy 232

B. Image Datasets 233
B.1 Flickr 233
B.2 Panoramio 234
B.3 Oxford Visual Geometry Group 235
B.4 University of Kentucky Recognition Benchmark Images 235
B.5 Other 235

C. Image Credits 237
C.1 Images from Flickr 237
C.2 Other Images 238
C.3 Illustrations 238

References 239
Index 243

Preface

Today, images and video are everywhere. Online photo-sharing sites and social networks have them in the billions. Search engines will produce images of just about any conceivable query. Practically all phones and computers come with built-in cameras. It is not uncommon for people to have many gigabytes of photos and videos on their devices.

Programming a computer and designing algorithms for understanding what is in these images is the field of computer vision. Computer vision powers applications like image search, robot navigation, medical image analysis, photo management, and many more.

The idea behind this book is to give an easily accessible entry point to hands-on computer vision with enough understanding of the underlying theory and algorithms to be a foundation for students, researchers, and enthusiasts. The Python programming language, the language choice of this book, comes with many freely available, powerful modules for handling images, mathematical computing, and data mining.

When writing this book, I have used the following principles as a guideline. The book should:

- Be written in an exploratory style and encourage readers to follow the examples on their computers as they are reading the text.
- Promote and use free and open software with a low learning threshold. Python was the obvious choice.
- Be complete and self-contained. This book does not cover all of computer vision but rather it should be complete in that all code is presented and explained. The reader should be able to reproduce the examples and build upon them directly.
- Be broad rather than detailed, inspiring and motivational rather than theoretical.

In short, it should act as a source of inspiration for those interested in programming computer vision applications.