

Table of Contents

Preface	1
Chapter 1: Cartoonifier and Skin Changer for Android	7
Accessing the webcam	9
Main camera processing loop for a desktop app	10
Generating a black-and-white sketch	11
Generating a color painting and a cartoon	12
Generating an "evil" mode using edge filters	14
Generating an "alien" mode using skin detection	16
Skin-detection algorithm	16
Showing the user where to put their face	17
Implementation of the skin-color changer	19
Porting from desktop to Android	24
Setting up an Android project that uses OpenCV	24
Color formats used for image processing on Android	25
Input color format from the camera	25
Output color format for display	26
Adding the cartoonifier code to the Android NDK app	28
Reviewing the Android app	30
Cartoonifying the image when the user taps the screen	31
Saving the image to a file and to the Android picture gallery	33
Showing an Android notification message about a saved image	36
Changing cartoon modes through the Android menu bar	37
Reducing the random pepper noise from the sketch image	40
Showing the FPS of the app	43
Using a different camera resolution	43
Customizing the app	44
Summary	45

Chapter 2: Marker-based Augmented Reality on iPhone or iPad	47
Creating an iOS project that uses OpenCV	48
Adding OpenCV framework	49
Including OpenCV headers	51
Application architecture	52
Marker detection	62
Marker identification	64
Grayscale conversion	64
Image binarization	65
Contours detection	66
Candidates search	67
Marker code recognition	72
Reading marker code	72
Marker location refinement	74
Placing a marker in 3D	76
Camera calibration	76
Marker pose estimation	78
Rendering the 3D virtual object	82
Creating the OpenGL rendering layer	82
Rendering an AR scene	85
Summary	92
References	92
Chapter 3: Marker-less Augmented Reality	93
Marker-based versus marker-less AR	94
Using feature descriptors to find an arbitrary image on video	95
Feature extraction	95
Definition of a pattern object	98
Matching of feature points	98
PatternDetector.cpp	99
Outlier removal	100
Cross-match filter	101
Ratio test	101
Homography estimation	102
Homography refinement	104
Putting it all together	107
Pattern pose estimation	108
PatternDetector.cpp	108
Obtaining the camera-intrinsic matrix	110
Pattern.cpp	113
Application infrastructure	114
ARPipeline.hpp	115
ARPipeline.cpp	115
Enabling support for 3D visualization in OpenCV	116

Creating OpenGL windows using OpenCV	118
Video capture using OpenCV	118
Rendering augmented reality	119
ARDrawingContext.hpp	119
ARDrawingContext.cpp	120
Demonstration	122
main.cpp	123
Summary	126
References	127
Chapter 4: Exploring Structure from Motion Using OpenCV	129
Structure from Motion concepts	130
Estimating the camera motion from a pair of images	132
Point matching using rich feature descriptors	132
Point matching using optical flow	134
Finding camera matrices	139
Reconstructing the scene	143
Reconstruction from many views	147
Refinement of the reconstruction	151
Visualizing 3D point clouds with PCL	155
Using the example code	158
Summary	159
References	160
Chapter 5: Number Plate Recognition Using SVM and Neural Networks	161
Introduction to ANPR	161
ANPR algorithm	163
Plate detection	166
Segmentation	167
Classification	173
Plate recognition	176
OCR segmentation	177
Feature extraction	178
OCR classification	181
Evaluation	185
Summary	188
Chapter 6: Non-rigid Face Tracking	189
Overview	191
Utilities	191
Object-oriented design	191

Data collection: Image and video annotation	193
Training data types	194
Annotation tool	198
Pre-annotated data (The MUCT dataset)	198
Geometrical constraints	199
Procrustes analysis	202
Linear shape models	205
A combined local-global representation	207
Training and visualization	209
Facial feature detectors	212
Correlation-based patch models	214
Learning discriminative patch models	214
Generative versus discriminative patch models	218
Accounting for global geometric transformations	219
Training and visualization	222
Face detection and initialization	224
Face tracking	228
Face tracker implementation	229
Training and visualization	231
Generic versus person-specific models	232
Summary	233
References	233
Chapter 7: 3D Head Pose Estimation Using AAM and POSIT	235
Active Appearance Models overview	236
Active Shape Models	238
Getting the feel of PCA	240
Triangulation	245
Triangle texture warping	247
Model Instantiation – playing with the Active Appearance Model	249
AAM search and fitting	250
POSIT	253
Diving into POSIT	253
POSIT and head model	256
Tracking from webcam or video file	257
Summary	259
References	260
Chapter 8: Face Recognition using Eigenfaces or Fisherfaces	261
Introduction to face recognition and face detection	261
Step 1: Face detection	263
Implementing face detection using OpenCV	264
Loading a Haar or LBP detector for object or face detection	265
Accessing the webcam	266

Detecting an object using the Haar or LBP Classifier	266
Detecting the face	268
Step 2: Face preprocessing	270
Eye detection	271
Eye search regions	272
Step 3: Collecting faces and learning from them	281
Collecting preprocessed faces for training	283
Training the face recognition system from collected faces	285
Viewing the learned knowledge	287
Average face	289
Eigenvalues, Eigenfaces, and Fisherfaces	290
Step 4: Face recognition	292
Face identification: Recognizing people from their face	292
Face verification: Validating that it is the claimed person	292
Finishing touches: Saving and loading files	295
Finishing touches: Making a nice and interactive GUI	295
Drawing the GUI elements	297
Checking and handling mouse clicks	306
Summary	308
References	309
Index	311
