Summary

In this chapter, we saw how we could use multiple computer vision algorithms to perform a bigger task and implemented a system similar to Microsoft's Office Lens. This algorithm can be extended and made better using better segmentation and corner detection algorithms. Also, once you have the page in the resulting image, you can apply machine learning algorithms to detect the text on the page.

Index

A

adaptive thresholding
about 20, 21
adaptive method 20
block size 20
C 21
affine transformation 121
Android NDK
download link 138
setting up 138, 139
automatic panoramic straightening 134

B

basic 2D transformations
about 120, 121
affine 121
projective 122
rigid 121
translation 121
best practices
about 169
data, handling between multiple activities 172
images, handling in Android 170
BRIEF
about 71
correlation 73
steered BRIEF 72
variance 72
BRISK (Binary Robust Invariant Scalable Keypoints)
about 74
in OpenCV 78
keypoint description 76
scale-space keypoint detection 74, 75
bundle adjustment 134

C

Canny Edge detection
about 32, 33
edge selection, through hysteresis thresholding 32
gradient of image, calculating 32
image, smoothing 32
non-maximal supression 32
Canny Edge detector
about 32
reference 32
cascade classifiers
about 83, 84
Haar cascades 84, 85
LBP cascades 85, 86
used, for face detection 86-93
cautions, for building application
duplicate data 169
limited computational capacity 170
memory leaks 169
network usage 170
Contour detection
implementation 42, 43
Contours
about 42
reference, for hierarchies 44
custom kernels
creating 15, 16

D

data, handling between multiple activities
about 172
database, using 174
data, transferring via Intent 173
file, using 174
static fields, using 173
Difference of Gaussian (DoG) 29-31, 52
dilation
about 16
applying 17
distance between vectors
defining 151
document scanning app
algorithm 177, 178
developing 175-177
implementing, on Android 179-189

E

Edge detection and Corner detection
about 28
Canny Edge detector 32, 33
Difference of Gaussian (DoG) 29-31
Harris Corner detection 36-38
Sobel operator 34-36
erosion
about 18
applying 18
errors, troubleshooting
about 165
code, debugging with Logcat 168
permission errors 165-167

F

face detection
performing, cascade classifier used 86-93
FAST
about 70
FAST detector 70
orientation, by intensity centroid 71
fast Hessian detector 65
Fast Library for Approximate Nearest Neighbors. *See* **FLANN**
Fast Retina Keypoint (FREAK)
about 79
coarse-to-fine descriptor 80
in OpenCV 81
orientation 81
retinal sampling pattern 79
saccadic search 80
feature description 48
feature detection 48
feature matching 47
features 47
Features App
creating 23-28
FLANN 60

G

gain compensation 135
Gaussian blur 12, 13
GaussianBlur function 13
Gaussian kernel
about 12, 13
reference 13
Gaussian pyramid
about 112, 113
creating, in OpenCV 114-120
global motion estimation 122-124

H

Haar cascades 84, 85
Happy Camera project
about 96, 97
faces and smiles, correlating 97
happy images, tagging 97
image, saving 97
smile detector, adding 97
Harris corner detection
about 36
implementing 37, 38
Harris corner detector 36, 53
Hessian matrix 54
Histogram of Oriented Gradients (HOG) descriptors
about 93
cells, combining to form blocks 94

classifier, building 94
gradient, computing 93
orientation binning 94
using 94-96
working 93
Hough transformations
about 38
Hough circles 40
Hough circles implementation 41, 42
Hough lines 38-40

I

illumination dependence 57
image matching
about 132
homography estimation, RANSAC used 132
verification, using probabilistic model 132, 133
image pyramids
about 104, 111
expand operation 112
Gaussian pyramids 112, 113
Laplacian pyramids 114
reduce operation 112
images
effects, applying 2
storing, in OpenCV 4
images, handling in Android
about 170
images, loading 170
images, processing 171
image stitching
about 129
Android NDK, setting up 138, 139
automatic panoramic straightening 134
bundle adjustment 134
C++ code 143-146
feature detection 130, 131
gain compensation 135
image matching 132
implementing 137
Java code, writing 140-142
layout 139
multi-band blending 136
OpenCV, used 137
performing 129
integral images
reference link 85
Intent class 173

K

Kanade-Lucas-Tomasi (KLT) tracker
about 125
implementing 125
implementing, on OpenCV 125-127
keypoint description
about 76
descriptor, building 77
sampling pattern and rotation estimation 76, 77
k-nearest neighbors (KNN) 150

L

Laplacian pyramids
about 114
creating, in OpenCV 114-120
Least Square Error 103
linear filters
about 5, 6
adaptive thresholding 20
custom kernels, creating 15, 16
Gaussian blur 12, 13
mean filter 6-11
median blur 14
morphological operations 16
thresholding 19
Local Binary Patterns (LBP) cascades 85, 86
Logcat
reference 169
Log class
reference 169

M

machine learning 149
Mat object 4
matching features
  about 59
  brute-force matcher 60
  FLANN based matcher 60
  objects, detecting 64, 65
  points, matching 60-63
mean filter
  about 6-10
  applying 11
median blur
  about 14
  applying 14
menus in Android
  reference 24
MNIST database
  about 153
  URL 153
morphological operations
  about 16
  dilation 16, 17
  erosion 18
multi-band blending 136

O

object tracking
  about 99
  in videos 99
OCR, using k-nearest neighbors
  about 150
  camera application, building 151, 152
  digits, recognizing 158-160
  training data, handling 153-157
oFAST 70
OpenCV
  about 2
  linear filters 5
  setting up 2, 3
OpenCV4Android SDK
  URL 3
Optical Character Recognition (OCR)
  about 149, 150
  k-nearest neighbors, used 150, 151
  Support Vector Machines (SVMs), used 160-162
optical flow
  about 99, 100
  Horn and Schunck method 101
  implementing, on Android 105-110
  Lucas and Kanade method 101-104
Oriented FAST and Rotated BRIEF (ORB)
  about 70
  contributions 70
  in OpenCV 73
  oFAST 70
  rBRIEF 71

P

permission errors
  about 165-167
  common permissions 167, 168
Prewitt operator
  reference 36
projective transformation 122
pseudo-inverse 103

R

rBRIEF 71
rigid transformation 121
rotation dependence 56

S

Scale Invariant Feature Transform (SIFT)
  about 48
  keypoint descriptor 55-57
  keypoint localization 52-54
  orientation assignment 54, 55
  properties 48
  scale-space extrema detection 49-52
  setting up, in OpenCV 57-59
  URL 48
  working 49
Sobel operator
  about 34
  using 34-36
Speeded Up Robust Features (SURF)
  about 65
  in OpenCV 69
  URL 66
Sudoku puzzle project
  digits, recognizing 162-164
  puzzle, detecting in image 44-46
  puzzle, solving 162
Support Vector Machines (SVM) 150, 160
SURF descriptor
  about 67
  based on Haar wavelet responses 68
  orientation assignment 67, 68
SURF detector 65, 66

T

thresholding
  about 19
  constants 19
  reference 20
translation transformation 121

U

U-SURF 67