Image-based information retrieval 
in an automated 
analysis of printed circuit boards

Von der Fakultät für Elektrotechnik und Informationstechnik 
der Rheinisch-Westfälischen Technischen Hochschule Aachen 
zur Erlangung des akademischen Grades eines Doktors 
der Ingenieurwissenschaften genehmigte Dissertation

vorgelegt von

Diplom-Ingenieur
Wei Li
aus
Hunan, China

Berichter:
Universitätsprofessor Dr.-Ing. Dorit Merhof
Universitätsprofessor Dr.-Ing. Jens-Rainer Ohm

Tag der mündlichen Prüfung: 29. Juni 2017

Diese Dissertation ist auf den Internetseiten der Universitätsbibliothek 
online verfügbar.
Abstract

Regarding the huge quantities of valuable materials and hazardous substances contained in waste Printed Circuit Boards (PCBs), an appropriate recycling of them is highly recommended. However, due to the high complexity and variability in their material composition, state-of-the-art industrial recycling systems commonly suffer from an unstable recycling performance resulting in a suboptimal eco-efficiency. The best solution for addressing this issue is a selective and dynamic recycling, where fractions containing different materials are sent to correspondingly configured processing routes. In order to facilitate such a dedicated recycling, an image-based analysis of PCBs for an automated determination of the material composition is gaining increasing attention. Nevertheless, due to the technical challenges imposed by the high complexity and variability of PCBs, a practical solution for realizing the intended information retrieval is not yet available.

In this thesis, a comprehensive, yet feasible PCB analysis is presented for the first time, where the overall complexity and diversity are substantially reduced by performing the analysis at the level of single components instead of entire PCBs. As confirmed by a quantitative evaluation, this establishes the necessary technical support for realizing indirect retrieval of the sought material composition. Moreover, a systematic investigation of all relevant aspects for achieving the desired analysis is performed, which covers the characterization of the employed imaging system, the localization and categorization of single components, as well as the extraction of text information. By this means, achievements, limitations and potentials are identified for each aspect, which provide a deep insight into the presented PCB analysis and open perspectives for further developments.
Due to the highly dynamic production of PCB components, new recycling targets steadily emerge in recycling processes. To address this issue, adaptations of the information retrieval system for the newly emerging targets become necessary throughout the service life of recycling systems. Benefiting from the well-designed implementation of the overall analysis and the resulting good extensibility, only a few data-driven models need to be retrained during adaptions, which is straightforward for given adequate training data. Explicit parameter tuning is in general unnecessary as the remaining parts of the overall analysis are either not affected or featured with automated parameter adaption as required.

In consideration of the associated high complexity and variability in size, shape, color and texture, PCBs and the mounted components generally give rise to a very challenging application scenario in computer vision. To deal with this problem, generic and adaptive analysis has been considered throughout the entire development process. As a result, most of the obtained methods, algorithms and approaches are not limited to the specific application on PCBs, but are also applicable in more generic tasks.
Acknowledgements

First of all, I would like to thank Prof. Dr.-Ing. Aach, who gave me the opportunity to start my doctoral research at the Institute of Imaging and Computer Vision (LfB), RWTH Aachen University. Although he unfortunately passed away after guiding me for the first year at LfB, I am deeply indebted to him for leading me into this exciting area of research and figuring out a proper way forward. Especially, I would like to express my sincere gratitude to my supervisor Prof. Dr.-Ing. Merhof for her consistent support and encouragement in conducting the presented research work, as well as her immense knowledge, valuable suggestions and great patience. Her necessary guidance and the given adequate freedom in research were indispensible for completing this thesis. Additionally, my sincere thank also goes to Prof. Dr.-Ing. Ohm for inspiring discussions and insightful comments. His support as provisional head was essential for the success of this thesis.

For all precious memories when looking back on my time at the institute, I am very grateful to all previous and current staffs of LfB. I would like to thank Achim for diverse discussions, Christine for the delicious muffins, Daniel for the perfect IT support, Matthias for the German grammar check, Peter for the small talks and social events, Julie for the organization of game nights, Joschka for sharing his professional knowledge, Kraisorn for the travel tips, and all other colleagues for their help and support in overcoming numerous obstacles: Alex, Yu, Maria, Gerli, Sebastian, Dorian, David, Simon, Marcin, Daniel Bug and Michael. I also want to thank my students for their cooperation and contributions to diverse topics: Dennis, Bernhard, Daniel Herchenbach, Stefan, Tarek, Tim, Martin, Anja, Georg, Guojie, Can, Chen, Yun, Wenting and Ziang.
Last but not least, I would like to thank my family for their unconditional and unwavering support throughout the time of my doctoral study and my life in general.

Wei Li
Contents

Lists of abbreviations i

List of symbols v

List of figure xiv

List of tables xvi

1 Introduction 1
   1.1 Motivation ........................................ 1
   1.2 Objectives ........................................ 2
   1.3 Contributions .................................... 3
   1.4 Structure of thesis ............................... 5

2 PCB recycling 7
   2.1 Material flow in recycling ....................... 9
   2.2 Image-based information retrieval .............. 12
   2.3 Challenges .................................... 14
3 Imaging system characterization

3.1 Modeling ........................................ 17
   3.1.1 Camera .................................... 17
   3.1.2 Thin lens ................................ 19
      3.1.2.1 Distortion ............................ 21
      3.1.2.2 Chromatic aberration (CA) .......... 23

3.2 Geometric calibration ............................... 25
   3.2.1 Offline calibration .......................... 26
      3.2.1.1 Planar homography $\mathcal{H}$ .......... 26
      3.2.1.2 The optimal estimate of $\mathcal{H}$ .... 28
      3.2.1.3 The absolute conic $\Omega_{\infty}$ ...... 29
      3.2.1.4 Vanishing points ...................... 30
      3.2.1.5 Solving intrinsic and extrinsic parameters 31
      3.2.1.6 Full parameter estimation .......... 33
   3.2.2 Online correction of lens distortion ........ 35
      3.2.2.1 Epipolar constraint .................. 36
      3.2.2.2 Model reformulation ................. 38
      3.2.2.3 Recovering distortion-free image coordinates 41
      3.2.2.4 Polynomial model ..................... 42
      3.2.2.5 Division model ....................... 48
      3.2.2.6 Rational model ....................... 48
      3.2.2.7 Parameter estimation ............... 49
   3.2.3 Online estimation of camera calibration matrix . 50
      3.2.3.1 Non-stationary scene ................ 52
      3.2.3.2 General constraints .................. 55
      3.2.3.3 Approaches with linear initialization . 60
4 Assembly print-guided segmentation of SMDs

4.1 Segmentation of small devices
  4.1.1 Border candidates
  4.1.2 Rotation estimation
  4.1.3 Hierarchical assessment

4.2 Segmentation of ICs

4.3 Evaluation
  4.3.1 Rotation estimation
  4.3.2 Segmentation
    4.3.2.1 Data and performance measures
    4.3.2.2 Results

4.4 Discussion
5 General 2D information-based analysis of components  117

5.1 State of the art ........................................ 119

5.2 Localization ........................................... 121
  5.2.1 Combinatory approach ............................... 121
    5.2.1.1 Diversification-driven proposal generation  122
    5.2.1.2 Region validation .............................. 128
  5.2.2 Compact approach ................................. 131
    5.2.2.1 Local variance-based segmentation .......... 132
    5.2.2.2 Bounding-box regression .................... 139
    5.2.2.3 Region merging and validation .............. 151

5.3 Categorization ....................................... 153

5.4 Evaluation ........................................... 153
  5.4.1 Localization ..................................... 154
    5.4.1.1 Data and performance measures ............ 154
    5.4.1.2 Combinatory approach ....................... 155
    5.4.1.3 Compact approach ............................ 160
  5.4.2 Categorization .................................. 171
    5.4.2.1 Data and performance measures ............ 171
    5.4.2.2 Results ..................................... 173

5.5 Discussion ......................................... 176

6 Text analysis 181

6.1 State of the art ..................................... 182

6.2 Pre-processing ....................................... 188
  6.2.1 Binarization ..................................... 189
  6.2.2 Skew correction and text line segmentation .... 191

6.3 Text recognition .................................... 194
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.4</td>
<td>Evaluation</td>
<td>197</td>
</tr>
<tr>
<td>6.4.1</td>
<td>Data and performance measures</td>
<td>197</td>
</tr>
<tr>
<td>6.4.2</td>
<td>Localizing text objects</td>
<td>198</td>
</tr>
<tr>
<td>6.4.3</td>
<td>Retrieving text information</td>
<td>200</td>
</tr>
<tr>
<td>6.5</td>
<td>Discussion</td>
<td>210</td>
</tr>
<tr>
<td>7</td>
<td>Conclusion</td>
<td>213</td>
</tr>
<tr>
<td>7.1</td>
<td>Summary</td>
<td>213</td>
</tr>
<tr>
<td>7.2</td>
<td>Outlook</td>
<td>217</td>
</tr>
<tr>
<td>A</td>
<td>Tool for generating reference data</td>
<td>219</td>
</tr>
<tr>
<td>A.1</td>
<td>State of the art</td>
<td>220</td>
</tr>
<tr>
<td>A.2</td>
<td>A better segmentation tool for PCBs</td>
<td>222</td>
</tr>
<tr>
<td>A.2.1</td>
<td>Concepts</td>
<td>222</td>
</tr>
<tr>
<td>A.2.2</td>
<td>Implementation</td>
<td>224</td>
</tr>
<tr>
<td>A.3</td>
<td>Evaluation and application</td>
<td>226</td>
</tr>
<tr>
<td>A.4</td>
<td>Discussion</td>
<td>228</td>
</tr>
<tr>
<td>B</td>
<td>Correction of lens distortion</td>
<td>229</td>
</tr>
<tr>
<td>B.1</td>
<td>Polynomial model using ( {k_1} )</td>
<td>229</td>
</tr>
<tr>
<td>B.1.1</td>
<td>Proof of ( \mathbf{h}_3^T = [0, 0, \eta] )</td>
<td>229</td>
</tr>
<tr>
<td>B.1.2</td>
<td>Simplified equations for ( \mathbf{h}_3^T = [0, 0, 1] )</td>
<td>231</td>
</tr>
<tr>
<td>B.2</td>
<td>Polynomial model using ( {k_1, p_1, p_2} )</td>
<td>232</td>
</tr>
<tr>
<td>B.2.1</td>
<td>Transformation between lifted coordinates</td>
<td>232</td>
</tr>
<tr>
<td>B.2.2</td>
<td>Ambiguities in estimated parameters</td>
<td>233</td>
</tr>
<tr>
<td>B.3</td>
<td>Polynomial model using ( {k_1, k_2, k_3, p_1, p_2} )</td>
<td>235</td>
</tr>
<tr>
<td>B.3.1</td>
<td>Transformed model for lifted coordinates</td>
<td>235</td>
</tr>
<tr>
<td>B.3.2</td>
<td>Constraint on ( x^i \cdot y^j ) for ( (i + j) = 6 )</td>
<td>236</td>
</tr>
<tr>
<td>B.4</td>
<td>Parameter constraints for division model</td>
<td>239</td>
</tr>
</tbody>
</table>
C Features and classifiers 241
   C.1 Features ................................................. 241
   C.2 Fundamental classifiers ................................. 242
   C.3 Ensemble methods ...................................... 243
   C.4 Deep learning .......................................... 244

Bibliography 245
## Lists of abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC</td>
<td>absolute conic</td>
</tr>
<tr>
<td>ACA</td>
<td>axial chromatic aberration</td>
</tr>
<tr>
<td>AdaBoost</td>
<td>adaptive boosting</td>
</tr>
<tr>
<td>ANN</td>
<td>artificial neural network</td>
</tr>
<tr>
<td>AOI</td>
<td>automated optical inspection</td>
</tr>
<tr>
<td>AP</td>
<td>average precision</td>
</tr>
<tr>
<td>AUC</td>
<td>area under the receiver-operating-characteristic curve</td>
</tr>
<tr>
<td>BoVW</td>
<td>bag-of-visual-words</td>
</tr>
<tr>
<td>BW</td>
<td>black-white</td>
</tr>
<tr>
<td>CA</td>
<td>chromatic aberration</td>
</tr>
<tr>
<td>CCR</td>
<td>correct classification rate</td>
</tr>
<tr>
<td>CNN</td>
<td>convolutional neural network</td>
</tr>
<tr>
<td>CNN-WRDF</td>
<td>convolutional neural network with recycled deep features</td>
</tr>
<tr>
<td>Conv</td>
<td>convolutional</td>
</tr>
<tr>
<td>DBN</td>
<td>deep belief network</td>
</tr>
<tr>
<td>DCM</td>
<td>division correction model</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------</td>
<td>------------</td>
</tr>
<tr>
<td>DF</td>
<td>division function</td>
</tr>
<tr>
<td>DL</td>
<td>deep learning</td>
</tr>
<tr>
<td>DT</td>
<td>distance transform</td>
</tr>
<tr>
<td>DTree</td>
<td>decision tree</td>
</tr>
<tr>
<td>EEE</td>
<td>electrical and electronic equipment</td>
</tr>
<tr>
<td>EMST</td>
<td>Euclidean minimum spanning tree</td>
</tr>
<tr>
<td>FC</td>
<td>fully connected</td>
</tr>
<tr>
<td>FCN</td>
<td>fully convolutional network</td>
</tr>
<tr>
<td>FED</td>
<td>fast explicit diffusion</td>
</tr>
<tr>
<td>FV</td>
<td>Fisher vector</td>
</tr>
<tr>
<td>GA</td>
<td>genetic algorithm</td>
</tr>
<tr>
<td>GLCM</td>
<td>gray-level co-occurrence matrix</td>
</tr>
<tr>
<td>GPU</td>
<td>graphics processing units</td>
</tr>
<tr>
<td>HOG</td>
<td>histograms of oriented gradients</td>
</tr>
<tr>
<td>IAC</td>
<td>image of the absolute conic</td>
</tr>
<tr>
<td>IC</td>
<td>integrated circuit</td>
</tr>
<tr>
<td>IoU</td>
<td>intersection-over-union</td>
</tr>
<tr>
<td>k-NN</td>
<td>$k$ nearest neighbors</td>
</tr>
<tr>
<td>KLT</td>
<td>Kanade–Lucas–Tomasi</td>
</tr>
<tr>
<td>LBP</td>
<td>local binary pattern</td>
</tr>
<tr>
<td>LCV</td>
<td>linear combination of views</td>
</tr>
<tr>
<td>LD</td>
<td>lens distortion</td>
</tr>
<tr>
<td>LoG</td>
<td>Laplacian of Gaussian</td>
</tr>
<tr>
<td>LV</td>
<td>local variance</td>
</tr>
</tbody>
</table>
ML  maximum likelihood
NIN  network in network
NMS  non-maximum suppression
OCR  optical character recognition
PCA  principal component analysis
PCB  printed circuit board
PCM  polynomial correction model
PF   polynomial function
PoP  post-regression-overlapping-pre-regression
PRC  precision-recall curve
RAD  ridge-based analysis of distributions
RANSAC  random sample consensus
RBM  restricted Boltzmann machine
RCM  rational correction model
ReLU rectified linear unit
RF   rational function
RF   random forest
RMS  root-mean-square
RMSE root-mean-square error
RNN  recurrent neural network
SA   simplex algorithm
SMD  surface-mounted device
SVM  support vector machine
TCA  transverse chromatic aberration
<table>
<thead>
<tr>
<th>THC</th>
<th>through-hole component</th>
</tr>
</thead>
<tbody>
<tr>
<td>WEEE</td>
<td>waste electrical and electronic equipment</td>
</tr>
</tbody>
</table>
Lists of symbols

\( \hat{\cdot} \) estimated value
\( \tilde{\cdot} \) homogeneous coordinates
\( \cdot \) variable/function after model transformation
\( a \) aspect ratio
\( b \) bounding box
\( C \) planar absolute conic
\( \cong \) equivalent to
\( \nabla_x \cdot I(\cdot) \) image gradient operator
\( D_{v_1}(\cdot) \) directional derivative operator
\( D_{v_1}^2(\cdot) \) second-order directional derivative operator
\( d \) distance
\( d(\cdot) \) distance function
\( \mathbb{E}^3 \) 3D Euclidean space
\( E_{\text{loss}} \) loss value
\( \tilde{e}_i \) epipole in the \( i \)-th view
\( F \) fundamental matrix
f  focal length

**FN**  set of false negatives

**FP**  set of false positives

**G(σ)**  Gaussian kernel with standard deviation σ

**∇²_{x_1} G(σ)**  Laplacian-of-Gaussian kernel with standard deviation σ

\( \mathcal{H} \)  2D/planar homography

\( H \)  3D homography

\( H_{\text{AM}} \)  homography from affine to metric reconstruction

\( H_{\text{AP}} \)  homography from affine to projective reconstruction

\( H_{\text{EA}} \)  homography from Euclidean to affine reconstruction

\( H_{\text{EM}} \)  homography from Euclidean to metric reconstruction

\( H_{\text{EP}} \)  homography from Euclidean to projective reconstruction

\( H_{\text{MA}} \)  homography from metric to affine reconstruction

\( H_{\text{ME}} \)  homography from metric to Euclidean reconstruction

\( H_{\text{PA}} \)  homography from projective to affine reconstruction

\( H_{\text{PE}} \)  homography from projective to Euclidean reconstruction

\( \mathcal{H}_{\infty} \)  homography induced by \( \tilde{\pi}_{E\infty} \)

\( \mathcal{H}_{\infty} \)  homography induced by \( \tilde{\pi}_{P\infty} \)

\( \mathcal{H}_{i,j,\infty}^{E} \)  homography from \( i \)-th to \( j \)-th view induced by \( \tilde{\pi}_{E\infty} \)

\( \mathcal{H}_{j,i,\infty}^{E} \)  homography from \( j \)-th to \( i \)-th view induced by \( \tilde{\pi}_{E\infty} \)

\( \mathcal{H}_{i,j,\infty}^{P} \)  homography from \( i \)-th to \( j \)-th view induced by \( \tilde{\pi}_{P\infty} \)

\( \mathcal{H}_{j,i,\infty}^{P} \)  homography from \( j \)-th to \( i \)-th view induced by \( \tilde{\pi}_{P\infty} \)

\( I \)  identity matrix

\( I \)  image
$I_{\text{LoG}}$ image after the Laplacian-of-Gaussian operation

$I_{LV}$ local-variance image

$i$ index of the $i$-th view/image

$j$ index of the $j$-th view/image

$K$ camera calibration matrix

$\kappa$ parameter of division model for correcting lens distortion

$L_{1}(\cdot)$ lens distortion function in image coordinates

$L_{1}^{-1}(\cdot)$ function for correcting lens distortion in image coordinates

$L_{N}(\cdot)$ lens distortion function in normalized camera coordinates

$L_{N}^{-1}(\cdot)$ function for correcting lens distortion in normalized camera coordinates

$l_{i}$ epipolar line in the $i$-th view

$M_{1}$ parameter matrix for correcting lens distortion in image coordinates using rational model

$M_{N}$ parameter matrix for correcting lens distortion in normalized camera coordinates using transformed rational model

$n$ normal vector

$O_{ND}$ distortion center

$ox_{ND}$ $x$ coordinate of the distortion center

$oy_{ND}$ $y$ coordinate of the distortion center

$\Omega^{*}$ absolute dual quadric

$\Omega_{\infty}$ absolute conic

$\omega$ image of the absolute conic

$\omega^{*}$ dual image of the absolute conic

$p$ translation along the $x$-axis in images
\( \mathbb{P}^2 \) 2D projective plane
\( \mathbb{P}^3 \) 3D projective space
\( P^A \) camera matrix undergoing affine transformation
\( P^E \) camera matrix undergoing Euclidean transformation
\( P^M \) camera matrix undergoing metric transformation
\( P^P \) camera matrix undergoing projective transformation
\( p_i \) principal point
\( px_i \) \( x \) coordinate of the principal point
\( py_i \) \( y \) coordinate of the principal point
\( \pi^E \) plane undergoing Euclidean transformation
\( \tilde{\pi}_\infty \) plane at infinity
\( \tilde{\pi}^E_\infty \) plane at infinity undergoing Euclidean transformation
\( \tilde{\pi}^P_\infty \) plane at infinity undergoing projective transformation
\( Q^* \) absolute dual quadric undergoing projective transformation
\( q \) translation along the \( y \)-axis in images
\( \mathbb{R} \) set of real numbers
\( R \) rotation matrix
\( r \) distance to the distortion center
\( \rho \) point depth
\( s \) skew factor
\( \mathcal{T} \) transformation matrix
\( t' \) translation vector
\( \Theta \) set of unknown parameters
\( TN \) set of true negatives
\( TP \) set of true positives
\( v_\infty \) vanishing point
\( \tilde{v}_i \) homogeneous image of the vanishing point \( v_\infty \)
\( v_{ul} \) the upper-left vertex of a rectangle
\( v_{ur} \) the upper-right vertex of a rectangle
\( v_{ll} \) the lower-left vertex of a rectangle
\( v_{lr} \) the lower-right vertex of a rectangle
\( \mathcal{W} \) local window
\( \mathcal{W}(x_i) \) local window of the image point \( x_i \)
\( w(x_{1,W} - x_1) \) weight assigned to the point \( (x_{1,W} - x_1) \) in the local window
\( x \) point
\( x_0 \) reference point of a ray expression
\( x_c \) point in camera coordinates
\( x_i \) point in image coordinates
\( x_{1,W} \) point in the local window \( \mathcal{W} \) of the image point \( x_1 \)
\( x_{1D} \) point in distorted image coordinates
\( x_N \) point in normalized camera coordinates
\( x_{ND} \) point in distorted normalized camera coordinates
\( x_W \) point in world coordinates
\( \tilde{x}_{W2D} \) point in homogeneous planar world coordinates
\( \tilde{x}_W^A \) point in homogeneous world coordinates undergoing affine transformation
\( \tilde{x}_W^E \) point in homogeneous world coordinates undergoing Euclidean transformation
\( \tilde{x}_W^M \) point in homogeneous world coordinates undergoing metric transformation

\( \tilde{x}_W^P \) point in homogeneous world coordinates undergoing projective transformation

\( \tilde{x}_{i,h,i} \) point on the horopter curve in homogeneous image coordinates of the \( i \)-th view

\( \tilde{x}_{i,h,j} \) point on the horopter curve in homogeneous image coordinates of the \( j \)-th view

\( \tilde{x}_{W,h}^E \) point on the horopter curve in homogeneous world coordinates undergoing Euclidean transformation

\( \tilde{x}_{W,h}^P \) point on the horopter curve in homogeneous world coordinates undergoing projective transformation

\( \tilde{x}_{W,h,\infty}^E \) point on the horopter curve and the plane at infinity in homogeneous world coordinates undergoing Euclidean transformation

\( \tilde{x}_{W,\infty}^P \) homogeneous world point at infinity undergoing projective transformation

\( x_\infty \) point at infinity

\( \tilde{x}_\infty^E \) homogeneous point at infinity undergoing Euclidean transformation

\( \tilde{x}_{1,\infty} \) homogeneous image of point at infinity

\( \chi_N \) point in lifted normalized camera coordinates

\( \chi_{ID} \) point in lifted distorted image coordinates

\( \chi_{ND} \) point in lifted distorted normalized camera coordinates
List of figures

2.1 Worldwide annual quantity of WEEE. ................. 8
2.2 Material and value compositions of PCBs. ............. 9
2.3 Proposed image-based retrieval of the material composition. 14
2.4 High complexity and diversity of PCBs in their assembly and mounted components. ......................... 15

3.1 Image formation of a pinhole camera. ............... 18
3.2 Imaging model of a thin lens. ......................... 20
3.3 Different types of lens distortion. ..................... 23
3.4 Chromatic aberration and its impact on image formation. 24
3.5 Typical planar patterns for offline camera calibration. ... 26
3.6 Epipolar geometry. ................................ 38
3.7 Stationary and non-stationary scenes................ 53
3.8 Multiple motions in PCB images. ..................... 64
3.9 Affine motion field assumed for KLT tracking. ....... 65
3.10 Tracked feature trajectories of multiple objects. ...... 66
3.11 Comparison between real and synthetic image feature trajectories. ......................... 69
3.12 Nearly planar structure of PCBs. .................... 71
3.13 The dominance of nearly coplanar feature points in PCB images. ........................................ 72
3.14 Detected geometric primitives for online camera calibration. 73
3.15 Rotations between the PCB and camera coordinate systems. 75
3.16 Global and local TCA corrections. ......................... 80
3.17 Feature matching in images with lens distortion. ........ 82
3.18 Online correction of lens distortion. ....................... 84
3.19 Distorted and undistorted images. ......................... 85
3.20 Online estimation of $K$ for non-stationary scenes........ 87
3.21 Motion segmentation results. ............................... 88
3.22 Geometry-based online calibration. ......................... 89
3.23 Complementary experimental setups for assessing the performance of global TCA correction. .................. 90
3.24 Visualization of local TCA correction. ..................... 93

4.1 Assembly print. .............................................. 98
4.2 Surface-mounted resistors. .................................. 99
4.3 Corner detection using structuring elements. ............... 101
4.4 Artifacts in rotation estimation using edge pixels. ......... 104
4.5 Hierarchical cancellation of false candidates in the segmentation of small devices. ......................... 105
4.6 Size validation of IC candidates. ............................. 107
4.7 Error in rotation estimation for diverse PCBs. ............. 108
4.8 Rotation estimation on challenging images. ................. 109
4.9 SMDs in images under different illuminations. ............. 111
4.10 Investigation of the quality threshold. ...................... 112
4.11 Segmentation results on the dataset “PCB SMDs”......... 114
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Objectives of the general 2D information-based component analysis.</td>
<td>118</td>
</tr>
<tr>
<td>5.2</td>
<td>Workflow of the combinatory localization approach.</td>
<td>122</td>
</tr>
<tr>
<td>5.3</td>
<td>Images arising from different scale spaces.</td>
<td>125</td>
</tr>
<tr>
<td>5.4</td>
<td>HOG features of PCB components.</td>
<td>129</td>
</tr>
<tr>
<td>5.5</td>
<td>Bag-of-visual-words model.</td>
<td>130</td>
</tr>
<tr>
<td>5.6</td>
<td>Workflow of the compact localization approach.</td>
<td>132</td>
</tr>
<tr>
<td>5.7</td>
<td>SIFT features and axes of symmetry detected in original and local-variância images.</td>
<td>135</td>
</tr>
<tr>
<td>5.8</td>
<td>Generation of LV image.</td>
<td>136</td>
</tr>
<tr>
<td>5.9</td>
<td>Positive proposals arising from the LV-based image segmentation.</td>
<td>140</td>
</tr>
<tr>
<td>5.10</td>
<td>Iterative bounding-box regression.</td>
<td>141</td>
</tr>
<tr>
<td>5.11</td>
<td>Design of CNNs.</td>
<td>143</td>
</tr>
<tr>
<td>5.12</td>
<td>CNN architectures for bounding-box regression.</td>
<td>145</td>
</tr>
<tr>
<td>5.13</td>
<td>A simple example of CNN-WRDF.</td>
<td>148</td>
</tr>
<tr>
<td>5.14</td>
<td>Augmentation of a region proposal.</td>
<td>151</td>
</tr>
<tr>
<td>5.15</td>
<td>Geometric augmentation of localization data for training region-regression and proposal-validation models.</td>
<td>154</td>
</tr>
<tr>
<td>5.16</td>
<td>Overall localization performance of the combinatory approach.</td>
<td>163</td>
</tr>
<tr>
<td>5.17</td>
<td>Test loss of different CNN architectures for bounding-box regression.</td>
<td>165</td>
</tr>
<tr>
<td>5.18</td>
<td>Improvement of the best IoU score through bounding-box regression.</td>
<td>168</td>
</tr>
<tr>
<td>5.19</td>
<td>Overall localization performance of the compact approach.</td>
<td>170</td>
</tr>
<tr>
<td>5.20</td>
<td>Localization results on a demo PCB.</td>
<td>172</td>
</tr>
<tr>
<td>5.21</td>
<td>Synthetic samples for training categorization classifiers.</td>
<td>173</td>
</tr>
</tbody>
</table>
6.1 Relevant text information on PCBs. 182
6.2 Different application scenarios for OCR systems. 184
6.3 Local text images. 188
6.4 Stroke width estimation. 190
6.5 Skew correction using subregions. 193
6.6 Segmentation of single text lines. 194
6.7 Random samples of synthetic text images. 198
6.8 Initial proposals for localizing text objects. 199
6.9 Final precision-recall curve for localizing text objects. 200
6.10 Text recognition results on images with one single text line. 206
6.11 Text recognition results on images with multiple text lines. 207
6.12 Text recognition results on rotated binary images with one single text line. 208
6.13 Text recognition results on rotated binary images with multiple text lines. 209
A.1 An overview of the developed ABeST. 225
List of tables

3.1 The reference camera model for online calibration.

3.2 RMS deviation from the mean feature position using different global TCA correction methods.

3.3 Energy distribution of edge pixel values along three principal components after local TCA correction.

4.1 Statistics of the dataset “PCB SMDs”.

5.1 Performance of single pixel-grouping algorithms.

5.2 Performance of single pixel-grouping algorithms using different configurations of scale and parameter setting.

5.3 Performance of combined pixel-grouping algorithms using multiple scales and parameter settings.

5.4 Performance of RF classifiers in region validation.

5.5 Performance of SVM classifiers in region validation.

5.6 Comparison between proposal generation algorithms.

5.7 Comparison between fundamental feature-classifier combinations.

5.8 Comparison between advanced feature-classifier combinations.
6.1 Resulting case-insensitive F-score of diverse pre-processing approaches. ................................. 203

6.2 Constrained word recognition performance in the ICDAR 2015 robust reading competition. .................. 210

A.1 A qualitative comparison between segmentation tools. . . 226

A.2 Statistics of the dataset “PCB General”. .................... 227
Chapter 1

Introduction

1.1 Motivation

There is an increasing global concern about waste electrical and electronic products due to the rapid growth of annual production and their common short life spans. Despite the widely introduced legislation on e-waste, only a small fraction of the overall mass is currently officially collected and appropriately treated, whereas the remainder undergoes an insufficient recycling of valuable materials and an inadequate management of hazardous substances. Combined with the huge quantity of e-waste, these issues give rise to an inefficient utilization of resources on a large scale and result in great negative impacts on the environment and human health.

As the central components for realizing the desired functionality, Printed Circuit Boards (PCBs) are integral constituents of electrical and electronic products. Due to the general presence of highly valuable metals and hazardous substances, PCBs are of great interest in recycling e-waste. Nevertheless, they are commonly associated with a high complexity arising from sophisticated circuits and densely mounted devices. Depending on their functionality and the applied technologies in manufacturing processes, a high variability in their material composition is also observed. As a result of a missing online material analysis in commercial recycling systems, an optimal processing of PCBs is not possible using state-of-the-art recycling approaches. This leads to an unstable recycling performance.
and makes the high investment of industrial infrastructure less affordable owing to the varying eco-efficiency. Motivated by the manual recovery of valuable metals while discarding the cost-intensive recycling equipment, waste PCBs are often transported with other e-waste from developed to developing countries with lower labor costs. Through a crude recycling using simple methods, a fraction of target metals (e.g., gold, copper, etc.) is extracted and the remaining parts including toxic gases, liquids and solid scraps are released into nature without undergoing further appropriate processing. These facts cause not only a significant loss of raw materials, but also irreversible pollution of the air, the land and water bodies, which imposes potential threats to life and health.

In order to enlarge the profit margin of industrial PCB recycling and to reduce the environmental footprint of recycling processes, a selective and dynamic recycling for maximizing the eco-efficiency is required, where single PCBs or their fractions are sent to processing routes optimally configured for the contained materials. Such recycling is achievable only if the material information about the currently processed PCB can be acquired. In comparison to the cost-intensive and inflexible spectral analysis, image-based information retrieval combined with a comprehensive data bank covering the material composition of mainstream PCB components is able to provide an economic and flexible solution for retrieving the required material information. By matching recognized components to the reference samples in the data bank, the sought material composition can be read from the saved information. By this means, reliable recycling performance can be realized and the overall eco-efficiency can be stabilized for encouraging large-scale industrial application while suppressing substandard treatment with low efficiency and environmental issues.

1.2 Objectives

Although the demand for a machine vision-based analysis of PCBs was already identified more than one decade ago, no realistic and systematic solution had been provided before this thesis to the best of the author’s knowledge. Previous research resulted in rudimentary analysis approaches only, which merely underwent feasibility studies on some selected samples. A comprehensive and quantitative performance evaluation regarding the
high complexity and variability of PCBs was not presented so far. Moreover, a reliable analysis at the level of components is missing. The only technically mature method was the recognition of single PCBs by matching them to the samples saved in a data bank [1,2]. Its performance has been verified through a matching test between the images of the sample PCBs with different orientations. However, the reliability of such recognition remains questionable for practical applications as PCBs found in recycling are unlikely identical with the reference samples in the data bank. A further concern is the applicability of this analysis method considering the hardly manageable diversity in the appearance of PCBs from different manufacturers and for different purposes.

To address the challenges arising from the high complexity and variability of PCBs as well as the missing prior knowledge of target objects, a first practical solution based on the analysis of single components is proposed and presented in this thesis. In contrast to the highly variable appearance of PCBs, the appearance of mainstream devices (e.g. capacitors, resistors, slots, connectors, integrated circuits, etc.) exhibit much lower variability. Using appropriate detection approaches, relevant components on PCBs are first localized and then assigned to the corresponding groups. After reading the related label information of the detected components, they can be matched to reference samples in the data bank with the help of their appearance, category assignment and label information. For the purpose of investigating the performance of developed methods, algorithms and approaches, a quantitative and comprehensive evaluation of the results obtained on datasets comprising diverse PCBs and devices is provided.

The focus of this thesis is on image processing approaches for waste PCB analysis. A data bank covering mainstream devices and their material composition is thus out of scope. Consequently, its implementation and the related matching between test and reference samples in the data bank need to be considered in future work.

1.3 Contributions

The most important contribution of this thesis is the first realistic solution for achieving an image-based analysis of waste PCBs. Since the production of PCB components is highly dynamic, the adaption of the overall
analysis for new recycling targets becomes essential for facilitating the desired information retrieval throughout the service life of recycling systems. In consideration of this fact, methods, algorithms and approaches are primarily developed for generic purposes. By this means, the requirement on target and application-specific parameter tuning is eliminated. The necessary adaption is realized through the modification of a few data-driven models, where the required retraining of these models is straightforward for given adequate training data. As a result of this well-designed implementation, the image-based analysis of PCBs can be extended with minimal effort to deal with newly emerging targets.

In comparison to previous publications [3] – [9], this thesis provides a systematic solution and a more comprehensive evaluation. As essential extensions, more advanced approaches leading to an improved analysis performance are presented. For a better overview of the most significant technical novelties, they are listed below according to the resulting workflow of the intended PCB analysis.

- To improve the performance of the epipolar constraint-based correction of lens distortion, a reformulation of state-of-the-art correction models is proposed, which reveals the underestimated power for a simultaneous parameter estimation of both lens and camera models. Through a detailed investigation, general guidance for selecting their appropriate parameter configurations is also provided (Chapter 3).

- State-of-the-art camera self-calibration approaches are extended for the case of non-stationary scenes, where rigid objects undergo independent motions. Regarding the degenerate motions and structures observed in PCB recycling, an alternative approach using geometric constraints is developed, which combines the conventional constraint with the specific constraint available on PCBs (Chapter 3).

- A widely applicable method for globally correcting transverse chromatic aberration is realized through a generic formulation of lens distortion (Chapter 3). It can be integrated into standard camera calibration workflows without great effort.

- To recover the desired horizontal or vertical orientation of PCBs, an effective and reliable approach for estimating their orientation in images is designed (Chapter 4).
• For PCBs arising from mass production, a carefully designed segmentation algorithm uses assembly print as guidance for localizing surface-mounted small devices (Chapter 4).

• A generic proposal generation algorithm is proposed for localizing general PCB components and thus discards additional parameter tuning (Chapter 5). It relies on a novel local variance-based method for exploiting image content and on a non-parametric adaptive thresholding.

• A novel architecture is designed for improving the localization performance of convolutional neural networks (Chapter 5). With the help of the recycled deep features, more accurate results are obtained in both tasks of localizing components and reading text.

• To address the challenges imposed by varying color, font, size, orientation and layout in text recognition, novel methods for realizing adaptive binarization of text objects and reliable skew correction as well as text line segmentation are developed (Chapter 6). They are indispensible for achieving the desired retrieval of label information.

1.4 Structure of thesis

This thesis comprises five parts according to the different steps in the execution of the related research work. The first two chapters state the background, motivation and objectives studied in the initialization step. This introductory part is then followed by three consecutive parts presented in Chapter 3 to 6, which are focused on the implementation of the proposed PCB analysis, including the imaging system characterization, the localization and categorization of single components as well as the text recognition. Comprehensive evaluation and in-depth discussion are always conducted in the corresponding chapters for confirming the desired performance and gaining a deep insight into the involved methods, algorithms and approaches. In the last and conclusive part, the overall performance is summarized and directions of future work are proposed. In more detail, the remaining chapters of this thesis are organized as follows:

Chapter 2 provides introductory details on PCB recycling as well as on the unsolved problems in state-of-the-art recycling systems. The image-
based information retrieval is then introduced as the proposed solution for stabilizing the recycling performance and maximizing the overall eco-efficiency. Through a brief literature review, technical challenges in the desired analysis of waste PCBs are also identified.

Chapter 3 exclusively deals with the characterization of the employed imaging system. Due to their better flexibility, approaches facilitating simultaneous image acquisition and lens/camera calibration are preferred. By decoupling the entire imaging process into several successive steps, the spatial and color information in images is recovered through the correction of lens distortion, the determination of the camera’s intrinsic parameters and the correction of chromatic aberration.

Chapter 4 aims at the segmentation of surface-mounted devices, which are widely used on mass-produced PCBs. Assembly print is utilized as the guidance for localizing small devices, whereas Integrated Circuits (ICs) exhibiting homogeneous surface are segmented using color information. To improve the reliability of the proposed segmentation, all PCB images are automatically rotated back to a horizontal or vertical orientation using an effective and reliable approach developed in this thesis.

Chapter 5 presents, in contrast to Chapter 4, an unconstrained analysis of general components, where objects of interest are first localized and then categorized into the corresponding component groups. Alternative to a combinatorial localization approach employing the diversification strategy, a more compact approach relying on novel local-variance analysis and on bounding-box regression provides more accurate localization results. After an in-depth review of convolutional neural networks, a novel architecture is designed to further boost the overall localization performance.

Chapter 6 investigates the text recognition on PCBs, especially on mounted components. Besides state-of-the-art text spotting approaches for whole images, novel pre-processing methods are developed to support reading label information on components. Text in the pre-processed images is extracted through a convolutional neural network implemented in the proposed architecture, which is significantly superior to conventional implementations of optical character recognition engines.

Chapter 7 summarizes the performance of individual steps and draws a conclusion regarding the overall performance and the intended objectives. With respect to the open questions and further developments, perspectives for future work are also presented.
Chapter 2

PCB recycling

Benefiting from the rapid technological developments, a steady expansion of electrical and electronic products emerges. However, as a result of the continuously lowering prices and their fast replacement with more advanced functionality, the life spans of such products also decrease dramatically. This leads consequently to a huge annual quantity of Waste Electrical and Electronic Equipment (WEEE), which needs to be appropriately recycled, for both economic and environmental reasons. In a representative study [10] of WEEE performed by the United Nations University, the global quantity of e-waste has been estimated by applying a harmonized measurement method on empirical data collected from all countries, which resulted in a more accurate and more detailed analysis of WEEE in comparison to those studies based on region-specific data using varying definitions and methodologies. In the graphical representation of the estimated quantity of WEEE depicted in Figure 2.1, the huge absolute amount and a steady growth in weight can be recognized. Although the overall value of WEEE is expected to be up to 48 billion Euros, only around 15.5% of globally generated e-waste were formally documented and fully recycled in 2014. The inadequate collection and treatment lead not only to a great economic loss, but also to an undesired environmental impact as most of the contained hazardous substances are inappropriately processed and released into nature (e.g. the air, the land and water bodies).
As the essential constituents for realizing the desired functionality of most electrical and electronic products, PCBs are intimately related to WEEE and contribute to a significant fraction of the overall mass. According to a wide range of publications [11, 12, 13, 14, 15], PCBs are assumed to give rise to around 3% of overall WEEE by weight. Besides the dominance (around 70% by weight) of non-metallic materials contained in PCBs, many valuable metal elements are also involved in their production and can thus be recycled through appropriately defined processing routes. Regarding the value and the toxicity of these metals, they are categorized into three major groups: base, precious and toxic. In the first group, copper (Cu) is the most significant metal, which is commonly used as the conducting material in circuits. Further typical base metals are the housing materials iron (Fe) and aluminium (Al), as well as the soldering material tin (Sn). Precious metals, on contrary to base metals, occur only in small amounts in PCBs. Nevertheless, they contribute to the most value in recycling [16, 17, 18]. Usually, waste PCBs are classified with respect to their gold (Au) content and are sold to collection agencies. In the analysis of the overall value distribution, palladium (Pd) and silver (Ag) are further important sources for achieving an economic recycling, whose values are comparable to the value associated with copper. Besides base and precious metals, there are also toxic metal elements contained in
2.1. Material flow in recycling

As stated above, there are large quantities of valuable elements available in waste PCBs and they are considered as one of the important sources for obtaining secondary materials from urban mining. In contrast to natural ores, urban waste exhibits in general higher concentrations of target elements and the contained materials can be extracted with less effort due to their higher purity in comparison to the raw materials available as mixtures in nature. These facts can be verified in the case of PCBs by considering the elements gold and copper. Natural ores contain on average 1.4–8.3g gold per metric ton, whereas the concentrations of gold in electrical and electronic PCBs are assessed to be 17–81g and 100–1300g
per metric ton, respectively [18]. Copper as the common material for realizing electrical connections in circuits is extensively used in PCBs and contributes to 16–35% of the total mass [15]. Especially, a direct recovery of highly pure copper is achievable if conducting layers are for example extracted from the multilayer structure of PCBs through appropriate thermal and mechanical processing.

Considering the substantially reduced energy consumption and cost associated with the materials recovered from waste PCBs, they are of great interest for the recycling industry. In a further consideration of the controlled environmental footprint in urban mining and the appropriate treatment of hazardous materials, the recycling of PCBs together with other WEEE components is also gaining public attention. Currently, national and regional e-waste legislation has been widely introduced. In the European Community Directive 2012/19/EU [23] on WEEE, targets are set for collection, recycling and recovery with respect to electrical and electronic products. In China and India as the countries with the largest population, official WEEE management systems have been established. According to [10], in total four billion people are covered by national treatment legislation.

Although attempts have been made to achieve economic and environmental treatment of waste PCBs, the current collection and recycling remain far from sufficient [10]. Only a small fraction of the overall quantity is officially documented and recycled using the highest standards. Often, WEEE is transported from developed to developing countries as second-hand products and for raw materials. In the case of waste PCBs, they usually undergo a rude “backyard” recycling based on manual processing. Using simple acid leaching and smelting, a fraction of base and precious metals (copper, gold, silver, etc.) is recovered. Since health and environmental protections are rarely considered, toxic gases are commonly released into the air during the combustion of non-metallic fractions and the waste disposal into nature is usually preferred, which in turn leads to pollution of the land and water bodies.

In commercial recycling systems, to increase the efficiency of recycling and also to reduce the irreversible damage to nature and human health, automated processing based on mechanical as well as metallurgical processes has been implemented, where pyrometallurgical solutions are well-developed for large-scale industrial recycling, whereas chemical and bi-
2.1. Material flow in recycling

ological approaches are generally still subject to ongoing research [15]. In a typical recycling system, waste PCBs are first manually dismantled from WEEE and then mechanically pre-processed for separating diverse materials. After shredding or crushing processes, PCBs are broken into fractions of tiny sizes. By this means, metals and non-metallic materials are liberated and fed into the successive processing units. Ferrous components are typically extracted through a magnetic separation, while aluminium is concentrated through an eddy-current separation. With the help of an additional gravity separation in specific fluid or through controlled airflow, copper combined with precious metals is separated from plastic and other light materials. This especially valuable metallic mixture subsequently undergoes metallurgical processing, where copper, gold and silver are smelted, refined and recovered using metal smelters. As clarified in [16], the material liberation based on mechanical methods bears the risk of losing a significant fraction of precious metals before smelting and leads to a less beneficial recycling. A better solution is to utilize integrated metal smelters with more advance equipment, which does not require additional pre-processing and can deal with highly toxic gases arising from the combustion of halogenated/brominated flame retardants. However, this processing route is also associated with some considerable drawbacks. Even when the extremely high investment of advanced smelters is omitted, which in the case of [16] was over one billion U.S. dollars, appropriately mixed feed materials regarding their chemical and physical properties are explicitly required to achieve an eco-efficient recycling. For iron and aluminium contained in PCBs, since there is no pre-separation conducted, they cannot be directly recovered. Due to the selective smelting, they are collected in the slag and need to be recovered through further metallurgical processing. Furthermore, most non-metallic fractions with economic values [14] are irreversibly converted into other forms and cannot be recovered any more.

In state-of-the-art PCB recycling described above, strategic metals, for instance indium (In) and tantalum (Ta), remain unconsidered. Such elements are only produced in a few countries, but are in general highly technical relevant for realizing desired circuits. Although there exist technical and political motivations for recovering strategic metals from waste PCBs, an economic recycling of them is currently hardly feasible due to their extremely low concentrations and the pre-defined processing routes optimized for the generally available materials with the highest values.
2.2 Image-based information retrieval

Apparently, despite the comprehensive recovery of valuable elements and the good control of hazardous substances realized through automated PCB recycling, it suffers from a suboptimal recycling performance arising from the highly complex and variable material composition of PCBs. Since state-of-the-art recycling systems uniformly process components and PCBs comprising very different materials without dynamic adaption, a competition is unavoidable in recovering diverse materials. Moreover, considering the variation in prices of materials and the rapid development of technologies for manufacturing PCBs, a significant value loss combined with undesired environmental impact is possible. This prevents in general a wide application of automated recycling systems as the high investment of the processing equipment becomes less affordable in the absence of optimized eco-efficiency.

Through the analysis conducted above, the two essential problems of current PCB recycling are identified: the uniform treatment of diverse material carriers (PCB components) despite their highly variable material composition and the non-adaptive processing routes without considering the dynamic of material values for the overall optimal eco-efficiency. In principle, these two problems can be addressed without great effort if the material composition and the price information are provided. Using state-of-the-art technologies for automated separation of material carriers, e.g. thermal disassembly, laser-based desoldering and precise water jet cutting combined with industrial robots, PCB fractions carrying precious metals are selectively separated and fed into smelters, whereas the remaining parts undergo mechanical separation. In such a recycling, possible loss of valuable materials in the pre-processing is avoided and standard metal smelters requiring much lower investment are applicable in recovering copper, gold and silver, where toxic gases are substantially suppressed as non-metallic fractions are primarily collected through the mechanical processing route. Also with the help of the selective separation, components containing significant fractions of strategic metals are correspondingly collected to enrich their concentrations and an economic recovery of these metals becomes thus feasible. For materials resulting in a competitive recovery or associated with high-cost processing, the overall recycling route is dynamically configured regarding the maximal added value achievable in the current material flow. This is important for low-
cost recycling systems with limited ability in separating diverse materials since the desired profit margin is only realizable with the optimally configured processing routes and the maximized benefit for the waste PCBs currently undergoing recycling.

In order to provide the necessary information for facilitating selective separation and dynamic configuration, methods for automatically retrieving the material composition of PCBs and the mounted components are required, while the price information of diverse materials can be acquired from markets. The most straightforward method for determining the material composition is using active spectral or thermal analysis. As reviewed in [11], atomic absorption spectroscopy, energy-dispersive X-ray fluorescence spectrometer and thermogravimetry analysis can be employed in analyzing materials contained in PCBs. However, such analysis generally relies on cost-intensive equipment and requires well-controlled test conditions. Consequently, they are suited for studying sampled materials and cannot be applied for continuously assessing the composition of the material flow in commercial recycling. In state-of-the-art PCB recycling, manual inspection of components and PCBs is performed by trained personnel and qualitatively assesses the concentration of gold. Considering the low efficiency and the limited reliability of manual assessment based on visual information, an image-based automated retrieval of the material composition is preferable, which leads not only to increased efficiency and reliability, but also to a more comprehensive material analysis in addition to the gold content. To better clarify the process of the proposed image-based analysis, its schematic depiction is presented in Figure 2.3.

After acquiring images of the currently to be recycled PCBs, all relevant components are localized and assigned to the corresponding categories. Combining the obtained categorization results with the recognized label information and further properties (e.g. appearance), materials contained in these components are qualitatively or quantitatively determined with the help of a comprehensive data bank covering a great number of devices from diverse manufacturers. The retrieved material composition with the spatial information of these components is forwarded to the control unit of the recycling system, which realizes selective separation of PCB fractions and dynamically configures the overall processing route for achieving the optimal eco-efficiency. To establish the necessary technical support for the intended information retrieval and the desired dynamic recycling, this
Chapter 2. PCB recycling

Figure 2.3: Proposed image-based retrieval of the material composition.

thesis is focused on solutions for image-based analysis of components and text/label information. A comprehensive material study of components and the generation of the data bank in Figure 2.3 are beyond the scope of this thesis and are thus excluded.

2.3 Challenges

Despite the widely recognized demand for an automated analysis of PCBs and the mounted components [11,24,25], a realistic solution, especially using image-based analysis, for retrieving relevant information on PCBs is unavailable so far. Although the disassembly systems introduced in [24,25] have considered the application of visual recognition and Automated Optical Inspection (AOI), the corresponding subsystems for performing appropriate image analysis were regarded as black-box/off-the-shelf components, where no further implementation and performance information was provided. The absence of image-based information retrieval was a natural consequence of the high complexity and variability of PCBs, which is partially illustrated in Figure 2.4. For realizing highly integrated and sophisticated circuits, a great number of diverse components are usually involved and densely mounted on the surface, where the applied components depend on the intended functionality of the circuits and the cost-performance consideration, which exhibit in general a great variety be-
2.3. Challenges

Figure 2.4: High complexity and diversity of PCBs in their assembly and mounted components.

between different products and manufacturers. These facts lead to great challenges to state-of-the-art methods, algorithms and approaches in image analysis and computer vision.

In the field of AOI, numerous visual analysis systems have been developed for assessing the quality of produced PCBs [26,27,28,29,30]. Nevertheless, these AOI functions are only suited for applications in manufacturing processes as they require well-controlled conditions and prior knowledge of the target products. For instance, the quality control systems introduced in [29] and [30] are both for the purpose of inspecting surface-mounted devices. To reduce the complexity of localizing sought objects, carefully designed illumination is employed to give rise to predefined appearance of components. Besides the specific illumination, clean surface of solder joints and components is also required, which is, however, not necessarily the case on waste PCBs due to erosion and soiling. Moreover, horizontal or vertical placement of devices as well as the computer-added design
files with the reference circuits are prerequisites. Since these conditions are commonly not satisfied in the application scenarios associated with recycling, a direct adoption of the image analysis designed for AOI is thus improper. There also exist image-based auto-teaching systems for assisting in automated production [31,32]. Similar to AOI systems, such visual assistant programs usually rely on specific assumptions which are violated in PCB recycling.

In the context of this thesis, the proposed practical information retrieval is introduced and quantitatively evaluated on datasets comprising a wide spectrum of PCBs collected from diverse sources. This essentially distinguishes this thesis from previous publications [1,2,33,34,35] in this field. The early research work had often been focused on targets with less complexity and variability (for example multispectral analysis of simple surface materials [33,34]) or on solutions with limited applicability in recycling (for example recognizing PCBs instead of components [1,35,35]). A further common issue of these publications is their inadequate evaluation with respect to the high variability of PCBs. To address the challenges imposed by waste PCBs, the proposed information retrieval is conducted on the level of components, where relevant label information is also extracted for improving the reliability of the overall analysis. By this means, the great diversity in the production of PCBs can be better compensated, as the mainstream devices for constructing circuits have much lower variability. Even for an unknown PCB, the desired material analysis is still achievable as long as the involved components are recognized and matched to the corresponding samples in the data bank. It should be emphasized that the proposed methods, algorithms and approaches result in a comprehensive analysis of all components, which provides a general solution for the case of PCB recycling.
Chapter 3

Imaging system characterization

As the desired information retrieval is merely based on the acquired PCB images, an accurate imaging system is hence an essential prerequisite for achieving the intended analysis. Furthermore, to maintain and to retrieve the correct spatial information of sought objects, it is also necessary to describe the imaging system using appropriate lens and camera models with known parameters. For the purposes of correcting chromatic aberration [36,37] (CA) and lens distortion [38,39,40,41,42] (LD), as well as modeling the lens and the camera [43,44,45], different methods for characterizing the employed imaging system are investigated in this chapter. The corresponding evaluation results are also provided for conducting a comprehensive comparison.

3.1 Modeling

3.1.1 Camera

A pinhole camera is the most basic model for describing the projection of 3D world points onto the 2D image plane. Usually, to avoid the result-
Figure 3.1: Image formation of a pinhole camera.

ing inverted image in the original pinhole camera model, the image plane is virtually placed between the pinhole and the world points for a more convenient model description. In Figure 3.1, the pinhole is located at the origin $o_C$ of the camera frame $x_C$-$y_C$-$z_C$. The image plane is perpendicular to the optical axis, which is in this case the $z_C$-axis, and $o_I$ denotes the origin of the image plane $x_I$-$y_I$. The optical axis intersects the image plane at the principal point $p_I$ with the image coordinates $[px_I, py_I]^T$ and with the camera coordinates $[0, 0, fp]^T$. For an arbitrary world point $x_W$ with the world coordinates $[x_W, y_W, z_W]^T$, coordinates of its image in the camera and in the image frame are $x_C$ and $x_I$, respectively. Mathematically, 2D and 3D Euclidean spaces $\mathbb{E}^2$ and $\mathbb{E}^3$ can be extended to projective spaces $\mathbb{P}^2$ and $\mathbb{P}^3$, respectively, by appending an additional entry of the value one to the coordinate vector $x$ [46]. The resulting new coordinates are designated as homogeneous coordinates denoted by $\tilde{x}$. Given the world coordinates $x_W = [x_W, y_W, z_W]^T$, the corresponding homogeneous world coordinates are $\tilde{x}_W = [x_W, y_W, z_W, 1]^T$. Moreover, $c \cdot \tilde{x}_W$ represents the same point as $\tilde{x}_W$, where $c$ is a non-zero scale factor. If not stated otherwise, homogeneous coordinate vectors are scaled to have the last entry equal to one in this thesis. If the rotation matrix $R$ and the translation vector $t$ between the world and the camera frames are known, the 2D projection $x_C$ of the 3D point $x_W$ is written as

$$x_C = (fp/\rho) \cdot [R \begin{bmatrix} R \end{bmatrix} \begin{bmatrix} -R \cdot t \end{bmatrix} \cdot \tilde{x}_W = (fp/\rho) \cdot [R \begin{bmatrix} t' \end{bmatrix} \cdot \tilde{x}_W, \quad (3.1)$$

where $t' = -R \cdot t$. The variable $\rho$ denotes the point depth of $x_W$ along the optical axis and takes the value $([0, 0, 1] \cdot [R \begin{bmatrix} t' \end{bmatrix} \cdot \tilde{x}_W)$. As $R$ and $t'$ are

18
3.1. Modeling

external to the camera and vary correspondingly to the selected world and camera frames, they are referred to as the camera’s extrinsic parameters.

In reality, physical devices, typically Charge-Coupled Devices (CCDs) and Complementary Metal-Oxide-Semiconductor (CMOS) sensors consisting of matrix-arranged photon-sensitive cells, are used for converting the incoming photons of each cell into the corresponding image pixel value. Due to the inaccuracy in manufacturing and in assembly, the axes $x_i$ and $y_i$ are in general differently scaled and the angle $\alpha$ between these two axes differs from the ideal value $\pi/2$. To also incorporate this 2D mapping of the projection on the image plane:

$$x_C = [x_C, y_C, f_p] \rightarrow \tilde{x}_I = [x_I, y_I, 1]^T,$$

Equation 3.1 is extended with the camera calibration matrix $K$ for localizing the image point in homogeneous image coordinates:

$$\tilde{x}_I = K \cdot x_N = K \cdot (1/f_p) \cdot x_C = K \cdot (1/\rho) \cdot [R|t'] \cdot \tilde{x}_W,$$  \hfill (3.2)

where $x_N = [x_N, y_N, 1]^T$ are the normalized camera coordinates of the image point with $x_N = (1/f_p) \cdot x_c$ and

$$K = \begin{bmatrix} f & f \cdot s & px_i \\ 0 & f \cdot a & py_i \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} f & f \cdot (-\cot \alpha) & px_i \\ 0 & f' \cdot (1/\sin \alpha) & py_i \\ 0 & 0 & 1 \end{bmatrix}.$$ \hfill (3.3)

In Eq. 3.3, the variables $s$ and $a$ denote the skew factor and the aspect ratio, respectively. Furthermore, the depth $f_p$ of the image plane in the camera frame is converted into values $f$ and $f'$ in terms of pixels along the two axes $x_i$ and $y_i$. Apparently, $f$, $s$, $a$, $px_i$ and $py_i$ are intrinsically related to the camera and independent of the selection of the world and camera coordinate systems. In contrast to $R$ and $t'$, they are referred to as the camera’s intrinsic parameters.

3.1.2 Thin lens

To reduce the long exposure time associated with pinhole cameras, thin lenses are usually employed and placed in front of the imaging devices (CCDs or CMOS sensors). On the one hand, by enlarging the aperture, the number of incoming photons is dramatically increased, while the required exposure time substantially decreases. On the other hand, with a negligible thickness in comparison to the radii of the spherical lens surfaces, optical effects caused by the thickness of lenses are insignificant and
Chapter 3. Imaging system characterization

The ray tracing calculation is thus simplified. According to the lensmaker’s equation with the paraxial approximation [47, 48], the focal length $F$ of the thin lens in Figure 3.2 is approximated by

$$\frac{1}{F} \approx (n - 1) \cdot \left( \frac{1}{r_1} - \frac{1}{r_2} \right),$$

where $n$ is the refractive index of the lens material and the focal length $F$ is defined in air or vacuum. The radius $r_1$ is with a positive value as the spherical surface is convex. In contrast, the radius $r_2$ is with a negative value due to the concave surface. By virtually moving the image plane to the same side of the lens as the object, Eq. 3.1 and 3.2 remain valid for the obtained virtual image at the distance $f_p$ from the lens center.

A fundamental difference between a pinhole camera and a camera with a thin lens is that the focused image of an object is only obtained at a certain distance from the lens, while the pinhole camera does not suffer from any defocusing issue. The relationship among the point depth $\rho$, the image plane distance $f_p$ and the focal length $F$ is written as [48]

$$\frac{1}{\rho} + \frac{1}{f_p} - \frac{1}{F} = 0.$$  \hspace{1cm} (3.5)

Given a world point $\mathbf{x}_w$ and a thin lens with the focal length $F$, the image plane distance $f_p$ can be determined. For a lens system consisting of multiple lenses, an equivalent focal length $F$ exists, which is adjustable and

![Figure 3.2: Imaging model of a thin lens. The image plane is virtually moved from $-f_p$ to $f_p$ with respect to the lens center.](image)

object

virtual image

sensor

$r_1 > 0$

$r_2 < 0$

$\rho$

$f_p$

$-f_p$

image

0

focal point
3.1. Modeling

also obeys Eq. 3.5. In practice, to simultaneously obtain nearly focused images of objects with different $\rho$ values, the focal length $F$ of the applied lens or lens system is selected to fulfill $F \ll \rho_{max}$ so that $f_p \approx F$.

3.1.2.1 Distortion

In Eq. 3.4, the approximate focal length $F$ is only valid for paraxial rays close to the optical axis. As a result, for object points far off the optical axis, monochromatic aberrations, e.g. spherical aberration, coma, astigmatism, curvature of field and distortion, become significant and need to be considered. Among them, the first four aberrations lead to blurred images, whereas the latter gives rise to geometric deformation [49,50]. Unfortunately, it is impossible to avoid all these aberrations at the same time, even through an appropriate lens design. Thus, all lenses or lens systems suffer from certain aberrations. Commonly, to reduce spherical aberration or astigmatism, an aperture stop (a non-transparent plate with a circular orifice at the center) for limiting the amount of light reaching the lens surface or reaching the imaging device is employed, which gives rise to additional distortion. Depending on the position of the aperture stop, different types of distortion can be observed. If the stop is placed behind the thin lens and in front of the real image in Figure 3.2, only the light from the object and refracted at the upper part of the lens reaches the image plane. For non-paraxial rays, the focal length decreases along with increasing distance from the lens center. Therefore, the refracted rays cross the optical axis at a nearer location to the lens center than the original focal point and are consequently focused at a more peripheral position than the original image. As a result, pincushion distortion is observed due to the increased distance between the image and the optical axis. Contrarily, barrel distortion with reduced magnification of the image is observed for the stop placed between the object and the thin lens. Since these two types of lens distortion arise from non-paraxial rays, the induced geometric distortion is radial. Moreover, for a perfectly centered aperture stop, the change of the image magnification is isotropic with respect to the optical axis.

As stated in [38], further types of lens distortion comprising radial and tangential components arise from imperfections in manufacturing and in assembly, which are typically inaccurately manufactured lens surfaces, deviations from strictly collinear lens centers and slight tilt of lens elements.
or the imaging device. For a better understanding, a visual comparison between different forms of distortion is presented in Figure 3.3. Regarding the conducted analysis of total lens distortion above, the resulting geometric deformation is actually defined for normalized camera coordinates through the general distortion function \( \mathcal{L}_N(\cdot) \): 

\[
\begin{bmatrix}
x_{\text{ND}} \\
y_{\text{ND}}
\end{bmatrix} = \begin{bmatrix} x_N \\
y_N
\end{bmatrix} + (k_1 \cdot r^2 + k_2 \cdot r^4 + k_3 \cdot r^6 + \cdots) \begin{bmatrix} \tilde{x}_N \\
\tilde{y}_N
\end{bmatrix} + (1 + p_3 \cdot r^2 + p_4 \cdot r^4 + \cdots) \begin{bmatrix} p_1 \cdot (r^2 + 2 \tilde{x}_N \cdot \tilde{y}_N) \\
p_2 \cdot (r^2 + 2 \tilde{y}_N^2) + 2p_1 \cdot \tilde{x}_N \cdot \tilde{y}_N
\end{bmatrix} + (1 + s_3 \cdot r^2 + s_4 \cdot r^4 + \cdots) \begin{bmatrix} s_1 \cdot r^2 \\
s_2 \cdot r^2
\end{bmatrix},
\]

(3.6)

where \( \tilde{x}_N = x_N - o_{\text{ND}}x \) and \( \tilde{y}_N = y_N - o_{\text{ND}}y \). \( r = \sqrt{\tilde{x}_N^2 + \tilde{y}_N^2} \) denotes the radial distance between the distortion center \( o_{\text{ND}} \) and the distortion-free image position \( x_N \). In Eq. 3.6, the distorted image position consists of different fractions: the original coordinates \( x_N \) and \( y_N \), the isotropic radial distortion defined by \( \{k_1, k_2, k_3, \cdots\} \), the decentering distortion defined by \( \{p_1, p_2, p_3, p_4, \cdots\} \), as well as the thin prism distortion defined by \( \{s_1, s_2, s_3, s_4, \cdots\} \). It should be noted that occasionally the decentering distortion is also referred to as “tangential distortion” in the literature. This is erroneous since the decentering distortion contains both radial and tangential components. By also integrating distortion into the image formation in Eq. 3.2, a more comprehensive model is obtained as follows:

\[
\tilde{x}_{\text{ID}} = K \cdot \mathcal{L}_N(\mathcal{L}_N(x_N, o_{\text{ND}}) = K \cdot \mathcal{L}_N((1/\rho) \cdot [R^t] \cdot \tilde{x}_W, o_{\text{ND}}),
\]

(3.7)

where \( \tilde{x}_{\text{ID}} \) denotes the distorted homogeneous image coordinates of the image point. Further lens distortion models and a comprehensive evaluation of their performance can be found in the literature [42].
3.1. Modeling

(a) pincushion  (b) barrel
(c) anisotropic radial  (d) tangential

**Figure 3.3:** Different types of lens distortion. Green solid lines denote the original geometric primitives, while red dashed lines are used for visualizing their distorted shapes. Moreover, the distortion center is marked by the black dot in each image. As illustrated in (a) and (b), the pincushion and barrel distortion components are both radial and isotropic. On the contrary, the radial and tangential components in (c) and (d) are anisotropic and arising from the decentering and thin prism fractions in Eq. 3.6.

3.1.2.2 Chromatic aberration (CA)

Besides monochromatic aberrations, there is also another common artifact in images: chromatic aberration (CA), which arises from the varying refractive index of transparent material for polychromatic light containing radiation of multiple wavelengths. This aberration can be easily derived from Eq. 3.4. The focal length $F$ is not only related to the shape of the lens, but also related to the refractive indices of air/vacuum and the lens material. In case of an object reflecting or emitting polychromatic light, a varying focal length is obtained for the multiple color fractions in light. Consequently, multiple images of the same object are observed, which correspond to the different color fractions. These non-coincident projections lead to two undesirable effects in image formation: axial chromatic aberra-
Figure 3.4: Chromatic aberration and its impact on image formation. In (a), the green image is located on the sensor and thus focused. For the red image, its axial displacement with respect to the green image gives rise to ACA, while the lateral displacement parallel to the image plane results in TCA. Due to CA, color fringes are observed in (b) and (c) along the boundaries between bright and dark regions.

(a) chromatic aberration (CA)  
(b) ACA  
(c) TCA

tion (ACA) as longitudinal displacement between images along the optical axis and transverse chromatic aberration (TCA) as lateral displacement between images on the image plane. For a better understanding, the ACA and TCA between green and red fractions are visualized in Figure 3.4 (a). Moreover, their impact on image formation is also demonstrated in Figure 3.4 (b) and (c). Although both ACA and TCA give rise to color fringes along the boundaries between bright and dark regions, there are essential differences between them. On the one hand, color fringes related to ACA are with the same colors at different positions on the boundaries. On the other hand, color fringes related to TCA are with different colors (commonly red and blue since images are usually focused for the green fraction of polychromatic light) on the opposite sides of the boundaries.

To limit CA, sophisticated lens designs utilizing multiple lens elements and special filling media between them have been invented. Achromats with achromatic doublets [49,50] are the most commonly used solution and CA can be corrected for two wavelengths (usually red and blue). Further improved CA correction can be achieved if apochromats and su-
perchromats are applied, which account for three and four wavelengths, respectively. However, due to the costly manufacturing and the residual aberration for other wavelengths, as well as the highly limited tolerance to deviations from the optimal design and even from the intended working temperature, CA is still observed for many lenses, especially for those low-cost variants. Since Eq. 3.7 varies for inconstant focal length, there are in general different imaging models for individual color fractions in polychromatic light.

3.2 Geometric calibration

As stated above, a color image acquired using a camera can be considered as the superposition of images corresponding to single wavelengths. Thus, a cascaded characterization of imaging systems is adopted in this thesis, which decomposes the entire procedure into two consecutive steps: calibration of monochromatic camera/lens models and correction of CA. This decoupling reduces the complexity of the modeling required in each step and also gives a better overview of the system characterization.

In the step of geometric calibration, the intrinsic parameters of the involved camera and lens models are determined. Given the pinhole camera model-based image formation in Eq. 3.7, all sought parameters are those defining the camera calibration matrix \( K \) and the distortion function \( L_n(\cdot) \). For the purpose of solving unknown parameters, well-designed calibration objects combined with elaborate offline calibration approaches are usually employed. Benefiting from the specifically constructed patterns, accurate and reliable results can be obtained, while regular working processes are inevitably interrupted and calibration operations should be scheduled in advance. In addition to offline calibration approaches and for a more flexible solution, it is also feasible to simultaneously conduct an online calibration during regular image acquisition, where the rigidity of the scene or some special geometric entities, e.g. circles and orthogonal line segments, are explicitly considered. In this section, offline calibration is first reviewed and defines the standard approach. It is then followed by the introduction of online calibration with novel extensions to the state of the art, which have been partially published in [8] and presented in the student theses [52,53,54].
3.2.1 Offline calibration

The most commonly used offline calibration has its origin in Zhang’s famous paper [45] and relies on the practical constraints imposed by planar patterns, where the requirement of known plane motion is eliminated in comparison the Tsai’s technique [43]. This approach is usually extended with the model from [44] and has found wide applications in computer vision [55, 56, 57]. Two typically used planar patterns are illustrated in Figure 3.5, where square corners and circle centers are employed as control points for retrieving the sought geometric structure in images.

3.2.1.1 Planar homography $\mathcal{H}$

For an imaging system with an ideal lens without distortion, Eq. 3.7 is simplified into Eq. 3.2. Since the obtained image is defined in homogeneous coordinates, $\bar{x}_i$ is equivalent to $\rho \cdot \bar{x}_i$, despite the non-zero scale factor $\rho$. Let $\cong$ denote this equivalence, Eq. 3.2 can thus be rewritten as

$$\bar{x}_i \cong \rho \cdot \bar{x}_i = K \cdot [R|t'] \cdot \bar{x}_w.
$$

(3.8)

If the planar pattern is located on the $x$-$y$ plane of the world frame, the $z$-coordinates of all control points are zero and $\bar{x}_w = [x_w, y_w, 0, 1]^T$. To omit the irrelevant $z$-coordinates of the planar control points, a new homogeneous vector $\bar{x}_{w2d} = [x_w, y_w, 1]^T$ is used to denote the coordinates of any point located on the $x$-$y$ plane. Setting $z_w = 0$ and considering the

---

**Figure 3.5**: Typical planar patterns for offline camera calibration.
single columns \( r_1, r_2, r_3 \) of the rotation matrix \( R = [r_1 | r_2 | r_3] \), Eq. 3.8 is further transformed into

\[
\tilde{x}_1 \cong \rho \cdot \tilde{x}_1 = K \cdot [r_1 | r_2 | t'] \cdot \tilde{x}_{w2d} = \mathcal{H} \cdot \tilde{x}_{w2d},
\]

(3.9)

where a \( 3 \times 3 \) planar homography \( \mathcal{H} \) with

\[
\mathcal{H} = [\bar{h}_1 | \bar{h}_2 | \bar{h}_3]^T = [h_1 | h_2 | h_3] = K \cdot [r_1 | r_2 | t']
\]

(3.10)

is obtained and defines the bijection mapping the planar control points to their images in projective space \( \mathbb{P}^2 \). Given the known coordinates \( \tilde{x}_{w2d} \) of each control point and the located projection \( \tilde{x}_i \) on the image plane, it is straightforward to solve the corresponding homography \( \mathcal{H} \). Regarding the equality in Eq. 3.9 up to the non-zero scale factor \( \rho \), Eq. 3.9 is rewritten as \( (\bar{h}_i^T \cdot \tilde{x}_{w2d}) \cdot \tilde{x}_1 = \mathcal{H} \cdot \tilde{x}_{w2d} \) with \( \bar{h}_3^T \cdot \tilde{x}_{w2d} = \rho \). Apparently, two linear equations in the parameters of \( \mathcal{H} \) can be obtained:

\[
\begin{bmatrix}
\bar{h}_1^T - x_1 \bar{h}_3^T \\
\bar{h}_2^T - y_1 \bar{h}_3^T
\end{bmatrix} \cdot \tilde{x}_{w2d} = 0
\]

(3.11)

with \( \tilde{x}_1 = [x_1, y_1, 1]^T \). Equation 3.11 is notably also valid for any homography \( \mathcal{H}' \) with \( \mathcal{H}' = [h'_1 | h'_2 | h'_3] = \eta \cdot \mathcal{H} \), where \( \eta \) is an arbitrary non-zero scale factor. If no less than four non-collinear control points are located in the image, a system of linear equations \( \eta \cdot \mathbf{A} \cdot [\bar{h}_1^T, \bar{h}_2^T, \bar{h}_3^T]^T = 0 \) and the additional Frobenius norm constraint \( \| \eta \cdot [\bar{h}_1^T, \bar{h}_2^T, \bar{h}_3^T]^T \|_F = 1 \) are sufficient for solving \( \eta \cdot \mathcal{H} \), where \( \mathbf{A} \) is the coefficient matrix determined according to Eq. 3.11. For simplicity, the scale factor \( \eta \) is not explicitly considered and \( \mathcal{H} \) still denotes the resulting homography satisfying Eq. 3.11 and the Frobenius norm constraint. In a later analysis, it will be shown that this simplification does not affect the intended calibration.

In practice, the magnitude of coefficients in \( \mathbf{A} \) varies over a wide range and \( \mathbf{A} \) is therefore poorly conditioned. This could lead to a significant decrease in accuracy or even to numeric instability in parameter estimation. To achieve reliable performance while solving \( \mathcal{H} \), the coefficients in matrix \( \mathbf{A} \) are computed for the normalized planar and image coordinates, which can be obtained using the invertible transformation matrices \( T_1 \) and \( T_{w2d} \) following the data normalization algorithm in [58]. The resulting homography \( \mathcal{H}_N \) on the normalized data can be derived from Eq. 3.9:

\[
T_1 \tilde{x}_1 \cong T_1 \mathcal{H} \cdot T_{w2d}^{-1} T_{w2d} \cdot \tilde{x}_{w2d} = \mathcal{H}_N \cdot T_{w2d} \cdot \tilde{x}_{w2d},
\]

(3.12)

where \( \mathcal{H}_N = T_1 \mathcal{H} \cdot T_{w2d}^{-1} \). If \( \mathcal{H}_N \) is solved, the sought homography \( \mathcal{H} \) is also obtained as \( \mathcal{H} = T_1^{-1} \mathcal{H}_N \cdot T_{w2d} \).
3.2.1.2 The optimal estimate of $H$

Since the image coordinates $x_i$ of any projected control point are measured using some image processing techniques, they are in general disturbed by measurement errors. As a result, only an approximation $\hat{H}$ of the true $H$ can be obtained. Apparently, the least-square estimate obtained so far with respect to the coefficient matrix $A$ does not give the optimal solution for $H$ and it is conceivable that a more accurate estimate of $H$ is presented if the inaccuracy of the measurements is appropriately considered. However, an exact determination of the inaccuracy is unrealistic due to the unknown true positions of the projected control points. To deal with this problem, the optimization of $\hat{H}$ is conducted statistically, where the likelihood regarding the observed measurements is maximized by the optimal $\hat{H}$ [46]. Formally, the likelihood $L(H|X_i)$ with the variable $H$ and the observation $X_i$ equals the probability density $f(X_i|H)$ of the observation $X_i$ given the parameter $H$ of the probability density function $f(\cdot)$ [59], where the observation $X_i$ is the projected control points $\{x_{i,1}, x_{i,2}, \ldots, x_{i,N}\}$. By assuming an independent error term for each measurement and a zero-mean Gaussian error model with the variance $\sigma^2$, the joint probability density $f(X_i|H)$ of the observed measurements is rewritten as $f(x_{i,1}|H) \cdot f(x_{i,2}|H) \cdots f(x_{i,N}|H)$. Thus, the optimal solution with the Maximum Likelihood (ML) is

$$\hat{H} = \arg \max_H L(H|X_i) = \arg \max_H \prod_{j=1}^N e^{-\|\hat{x}_{1,j} - \tilde{x}_{1,j}\|^2/(2\sigma^2)}/2\pi \cdot \sigma^2,$$ (3.13)

where $\hat{x}_{1,j} \approx H: x_{w2D,j}$ with $\hat{x}_{1,j}$ of the form $[\hat{x}_{1,j}, \hat{y}_{1,j}, 1]^T$. $x_{w2D,j}$ and $\hat{x}_{1,j}$ denote the homogeneous planar coordinates of the $j$-th control point and its corresponding image, respectively. Considering the monotonic increase of the logarithm for increasing input value, the ML estimation in Eq. 3.13 is equivalent to the estimation maximizing the logarithm of the likelihood $L(H|X_i)$:

$$\hat{H} = \arg \max_H \ln(L(H|X_i)) = \arg \min_H \sum_{j=1}^N \|\hat{x}_{1,j} - \tilde{x}_{1,j}\|^2.$$ (3.14)

In Eq. 3.14, the cost term $\|\hat{x}_{1,j} - \tilde{x}_{1,j}\|^2$ associated with the $j$-th measurement also has its geometric interpretation: it is the squared distance
3.2. Geometric calibration

between the observed image $\tilde{x}_{i,j}$ and the virtual image $\hat{\tilde{x}}_{i,j}$ obtained using the estimated planar homography $\hat{\mathcal{H}}$. The optimal estimate of $\mathcal{H}$ is then obtained if the squared distance is minimized for all control points. This nonlinear minimization problem is in the form of bundle adjustment [60] and can be solved using the Levenberg-Marquardt algorithm [61]. It is also possible to extend Eq. 3.14 with an appropriate weighting matrix, where all distance values are mutually related through certain weighting factors. The significance of this extension is twofold: the introduced weighting factors indicate the confidence/relevance of corresponding measurements on the one hand and embed the correlation/covariance between measurements in the optimization on the other.

3.2.1.3 The absolute conic $\Omega_\infty$

In projective space $\mathbb{P}^3$ with homogeneous coordinates, there are some ideal points with the last entry of the coordinate vector equal to zero: $\tilde{x}_\infty = [x_\infty, y_\infty, z_\infty, 0]^T$. There are no corresponding points existing in 3D Euclidean space $\mathbb{E}^3$ since the coordinates $x_\infty$, $y_\infty$, and $z_\infty$ are divided by 0, which results in infinite values. $\tilde{x}_\infty$ thus refers to a point at infinity and all points at infinity lie on the plane $\tilde{\pi}_\infty$, where $\tilde{\pi}_\infty$ is of the form $[0, 0, 0, 1]^T$ and gives $\tilde{\pi}_\infty^T \cdot \tilde{x}_\infty = 0$. It should be noted that the given forms of $\tilde{x}_\infty$ and $\tilde{\pi}_\infty$ above are defined for the correspondence $(x = [x, y, z]^T) \leftrightarrow (\tilde{x} = c \cdot [x, y, z, 1]^T)$ (3.15) between $\mathbb{E}^3$ and $\mathbb{P}^3$, where $c$ is an arbitrary non-zero scale factor and usually set to one. If this correspondence is not given, $\tilde{x}_\infty$ and $\tilde{\pi}_\infty$ in general differ from the forms $[x_\infty, y_\infty, z_\infty, 0]^T$ and $[0, 0, 0, 1]^T$. To avoid any confusion, the homogeneous coordinates obtained according to Eq. 3.15 is denoted by $\tilde{x}_E$. Similarly, a point at the infinity and the plane at the infinity are written as $\tilde{x}_E^\infty$ and $\tilde{\pi}_E^\infty$, respectively. In the sense of camera calibration, a special planar geometric entity $C$ is located on $\tilde{\pi}_E^\infty$ and any planar point $x_E^\infty$ on this conic satisfies $(x_E^\infty)^T \cdot C \cdot x_E^\infty = 0$, where $C$ is in fact the identity matrix $I$ and $x_E^\infty = [x_E^\infty, y_E^\infty, z_E^\infty]^T$. In $\mathbb{P}^3$, $C$ is denoted by $\Omega_\infty$ and referred to as the absolute conic (AC) [46]. By substituting $\tilde{x}_w$ with $\tilde{x}_E^\infty$ in Eq. 3.8, the projection $\tilde{x}_{1,\infty}$ of a point $\tilde{x}_E^\infty$ at infinity is

$$\tilde{x}_{1,\infty} \cong K \cdot [R | t'] \cdot \tilde{x}_E^\infty = K \cdot R \cdot x_E^\infty \cong \mathcal{H}_\infty \cdot x_E^\infty$$ (3.16)
with $\mathcal{H}_\infty \cong K \cdot R$ defining a planar homography. Consequently, the equation of points on $AC$ is further transformed into

$$0 = (x^E_\infty)^T \cdot C \cdot x^E_\infty \equiv \tilde{x}^T_{1,\infty} \cdot \mathcal{H}^T_\infty \cdot C \cdot \mathcal{H}_\infty \cdot \tilde{x}_{1,\infty} \cong \tilde{x}^T_{1,\infty} \cdot \omega \cdot \tilde{x}_{1,\infty}, \quad (3.17)$$

where $R^T \cdot I \cdot R^{-1} = I$ and $\omega = K^T \cdot K^{-1} \equiv \mathcal{H}^T_\infty \cdot C \cdot \mathcal{H}_\infty$. As illustrated in Eq. 3.16 and 3.17, any point $x^E_\infty$ on $AC$ is projected to the image $\tilde{x}_{1,\infty}$ satisfying $\tilde{x}^T_{1,\infty} \cdot \omega \cdot \tilde{x}_{1,\infty} = 0$. Thus, $\omega$ is the Image of the Absolute Conic (IAC) after the $\tilde{x}_w \rightarrow \tilde{x}_1$ projection.

### 3.2.1.4 Vanishing points

Under a projective transformation in Eq. 3.8, a set of parallel lines in $\mathbb{E}^3$ could have their images intersecting at a certain point on the image plane. This point is referred to as the vanishing point of these parallel lines in $\mathbb{P}^2$ as they appear to converge at it in the image. For $\mathbb{P}^3$ originating from Eq. 3.15, points on a line parallel to the direction vector $v_\infty$ in $\mathbb{E}^3$ are governed by the expression $(\tilde{x}^E_0 + t \cdot \tilde{v}^E_\infty)$, where $\tilde{x}^E_0 = [x^T_0, 1]^T$ is a reference point and $\tilde{v}^E_\infty = [v^T_\infty, 0]^T$ is the homogeneous direction vector. Thus, any point on this line is uniquely defined with the corresponding scale factor $t$. By moving the point $x_0$ over $\mathbb{E}^3$, all lines parallel to $v_\infty$ are obtained. Actually, the direction vector $\tilde{v}^E_\infty$ in $\mathbb{P}^3$ also defines an intersection point at infinity on $\tilde{\pi}^E_\infty$, considering $(\tilde{x}^E_0 + t \cdot \tilde{v}^E_\infty) \cong \tilde{v}^E_\infty$ while $t \rightarrow \infty$. This point is independent of the individual reference point $\tilde{x}^E_0$ and therefore lying on all lines parallel to $v_\infty$. As a result, the vanishing point $\tilde{v}_1$ is the image of this ideal point $\tilde{v}^E_\infty$ and written as

$$\tilde{v}_1 \cong K \cdot [R \cdot t'] \cdot \tilde{v}^E_\infty = K \cdot R \cdot v_\infty. \quad (3.18)$$

Without loss of generality, $v_\infty$ can be defined in the camera frame with $R = I$ and $t' = 0$. As a result, there is $\tilde{v}_1 \cong K \cdot v_\infty \leftrightarrow v_\infty \cong K^{-1} \cdot \tilde{v}_1$. Apparently, if the camera calibration matrix $K$ is determined, a camera modeled with Eq. 3.2 can be used to retrieve the direction of the line passing through the camera center $\tilde{x}^E_0 = [0, 0, 0, 1]^T$ and the points $(\tilde{x}^E_0 + t \cdot \tilde{v}^E_\infty)$ which result in the same projection $\tilde{v}_1 \cong K \cdot v_\infty$ on the image plane.

For two sets of parallel lines with known direction vectors $v_{\infty,1}$ and $v_{\infty,2}$, there is $\cos \phi = v^T_{\infty,1} \cdot v_{\infty,2}/(\|v_{\infty,1}\|_2 \cdot \|v_{\infty,2}\|_2)$ for the angle $\phi$ between these two vectors. By substituting $v_\infty$ with $R^{-1} \cdot K^{-1} \cdot \tilde{v}_1$ from Eq. 3.18,
3.2. Geometric calibration

this directional relationship can be rewritten as

\[
\cos \phi = \frac{\mathbf{v}_{\infty,1}^T \cdot \mathbf{v}_{\infty,2}}{\|\mathbf{v}_{\infty,1}\|_2 \cdot \|\mathbf{v}_{\infty,2}\|_2} = \frac{(\tilde{v}_{i,1})^T \cdot K^{-T} \cdot R^T \cdot R^{-1} \cdot \tilde{v}_{i,2}}{\|R^{-1} \cdot K^{-1} \cdot \tilde{v}_{i,1}\|_2 \cdot \|R^{-1} \cdot K^{-1} \cdot \tilde{v}_{i,2}\|_2}
\]

\[
= \frac{(\tilde{v}_{i,1})^T \cdot K^{-T} \cdot R^T \cdot R^{-1} \cdot K^{-1} \cdot \tilde{v}_{i,2}}{((\tilde{v}_{i,1})^T \cdot K^{-T} \cdot K^{-1} \cdot \tilde{v}_{i,1} \cdot (\tilde{v}_{i,2})^T \cdot K^{-T} \cdot K^{-1} \cdot \tilde{v}_{i,2})^{1/2}}
\]

\[
= \frac{(\tilde{v}_{i,1})^T \cdot \omega \cdot \tilde{v}_{i,2}}{((\tilde{v}_{i,1})^T \cdot \omega \cdot \tilde{v}_{i,1} \cdot (\tilde{v}_{i,2})^T \cdot \omega \cdot \tilde{v}_{i,2})^{1/2}},
\]

\[(3.19)\]

where vanishing points \(\tilde{v}_{i,1}\) and \(\tilde{v}_{i,2}\) are the images of the two ideal points \(\tilde{v}_{\infty,1} = [v_{\infty,1}^T, 0]^T\) and \(\tilde{v}_{\infty,2} = [v_{\infty,2}^T, 0]^T\), respectively. A special case in Eq. 3.19 is orthogonal directions with \(\phi = \pi/2\), which provides a linear constraint \((\tilde{v}_{i,1})^T \cdot \omega \cdot \tilde{v}_{i,2} = 0\) on \(\omega\).

3.2.1.5 Solving intrinsic and extrinsic parameters

In Eq. 3.10, the first and second columns of both the rotation matrix \(R\) and the planar homography \(H\) are related through the camera calibration matrix \(K\). Since the columns of \(R\) are orthonormal, two constraints on the sought parameters in \(K\) are derived from Eq. 3.10 [62]:

\[
\begin{align*}
\mathbf{r}_1^T \cdot \mathbf{r}_2 &= 0 \Rightarrow h_1^T \cdot \mathbf{h}_2 = 0, \\
\mathbf{r}_1^T \cdot \mathbf{r}_1 - \mathbf{r}_2^T \cdot \mathbf{r}_2 &= 0 \Rightarrow h_1^T \cdot \mathbf{h}_1 - h_2^T \cdot \mathbf{h}_2 = 0.
\end{align*}
\]

\[(3.20)\]

Alternatively, two similar constraints on \(\omega\) can be obtained by applying orthogonal direction vectors in Eq. 3.19. As illustrated in Figure 3.5, two groups of mutually orthogonal directions are available on each planar pattern: horizontal \(\perp\) vertical, and diagonals top-left to bottom-right \(\perp\) top-right to bottom-left [55]. Let \(\tilde{v}_{i,1} - \tilde{v}_{i,4}\) denote the vanishing points related to the four directions, Eq. 3.19 is simplified into linear equations

\[
\begin{align*}
(\tilde{v}_{i,1})^T \cdot \omega \cdot \tilde{v}_{i,2} &= 0, \\
(\tilde{v}_{i,3})^T \cdot \omega \cdot \tilde{v}_{i,4} &= 0.
\end{align*}
\]

\[(3.21)\]
Regarding Eq. 3.3, the $3 \times 3$ symmetric matrix $\omega$ can be expanded as

$$
\omega = \begin{bmatrix}
\omega_{11} & \omega_{12} & \omega_{13} \\
\omega_{12} & \omega_{22} & \omega_{23} \\
\omega_{13} & \omega_{23} & \omega_{33}
\end{bmatrix} = K^T \cdot K^{-1} = (K \cdot K^T)^{-1} \tag{3.22}
$$

Applying $\{\omega_{11}, \omega_{12}, \omega_{13}, \omega_{22}, \omega_{23}, \omega_{33}\}$ as unknown parameters in Eq. 3.20 or Eq. 3.21, two linear equations in these parameters are imposed by the corresponding constraints. Let $D$ denote the number of images acquired from the planar pattern at different positions and poses with respect to the camera, $\{\omega_{11}, \omega_{12}, \omega_{13}, \omega_{22}, \omega_{23}, \omega_{33}\}$ can be uniquely determined if $D \geq 3$. Actually, the six parameters are combinations of the five intrinsic parameters $\{f, s, a, px_1, py_1\}$ and another non-zero scale factor $\nu$, since globally scaling all elements in $\omega$ does not violate Eq. 3.20 and Eq. 3.21. Therefore, these equations are generally solved for $\nu \cdot \omega$ rather than $\omega$.

The necessary step for extracting the sought intrinsic parameters and the scale factor from $\{(\nu \cdot \omega_{11}), (\nu \cdot \omega_{12}), (\nu \cdot \omega_{13}), (\nu \cdot \omega_{22}), (\nu \cdot \omega_{23}), (\nu \cdot \omega_{33})\}$ is as follows:

$$
\begin{align*}
a &= ((\nu \cdot \omega_{22})/(\nu \cdot \omega_{11}) - (\nu \cdot \omega_{12})^2/(\nu \cdot \omega_{11})^2)^{-1/2}, \\
s &= -a \cdot (\nu \cdot \omega_{12})/(\nu \cdot \omega_{11}), \\
py_1 &= -(a^2 \cdot (\nu \cdot \omega_{23}) + a \cdot s \cdot (\nu \cdot \omega_{13}))/(\nu \cdot \omega_{11}), \\
px_1 &= py_1 \cdot s/a - (\nu \cdot \omega_{13})/(\nu \cdot \omega_{11}), \\
\nu &= (\nu \cdot \omega_{33}) - (px_1^2 + py_1^2 \cdot (1 + s^2))/a^2 - 2px_1 \cdot py_1 \cdot s/a \cdot (\nu \cdot \omega_{11}), \\
f &= (\nu/(\nu \cdot \omega_{11}))^{1/2}.
\end{align*} \tag{3.23}
$$

It should be emphasized that the first and second column vectors of the homography $H$ only depend on the rotation matrix $R$ in Eq. 3.10, given a constant camera calibration matrix $K$. It implies the requirement of significant rotations of the planar pattern between different images. Otherwise, no sufficient equations in $\{\omega_{11}, \omega_{12}, \omega_{13}, \omega_{22}, \omega_{23}, \omega_{33}\}$ can be obtained even if a great number of images are provided. This is also the necessary condition for using the alternative constraints deduced from vanishing points.
3.2. Geometric calibration

Besides all intrinsic parameters in $K$, it is also possible to recover the corresponding extrinsic parameters $R = [r_1 | r_2 | r_3]$ and $t'$ of the planar pattern for each image from Eq. 3.10:

$$
\begin{align*}
    r_1 &= K^{-1} \cdot \frac{h_1}{\|K^{-1} \cdot h_1\|_2}, \\
    r_2 &= K^{-1} \cdot \frac{h_2}{\|K^{-1} \cdot h_2\|_2}, \\
    r_3 &= r_1 \times r_2, \\
    t' &= K^{-1} \cdot \frac{h_3}{(\|K^{-1} \cdot h_1\|_2 \cdot \|K^{-1} \cdot h_2\|_2)^{1/2}}.
\end{align*}
$$  \hspace{1cm} (3.24)

Any scaling of the column vectors in $H$ is canceled out by the normalization with respect to the length of the vectors. The sought extrinsic parameters are then exactly solved even when the homography $H$ was determined up to an unknown non-zero scale factor $\eta$. In consideration of inaccuracy in measurements, the resulting $r_1$ and $r_2$ vectors are not necessarily orthonormal. To enforce this constraint, a new rotation matrix $\hat{R}$ with the least Frobenius distance $\|\hat{R} - R\|_F$ to the solved rotation matrix $R$ from Eq. 3.24 is determined as $\hat{R} = U \cdot \Sigma \cdot V^T = U \cdot V^T$ [62], where $R = U \cdot \Sigma \cdot V^T$ is the singular value decomposition of the $3 \times 3$ matrix $R$.

3.2.1.6 Full parameter estimation

So far, camera calibration has been considered only for the imaging systems without lens imperfections. In case of significant lens distortion, an appropriate distortion model $L_N(\cdot)$ and the associated parameters should also be determined. In practice, the polynomial model defined in Eq. 3.6 is predominantly employed in combination with the distortion parameters $\{k_1, k_2, k_3, p_1, p_2\}$ [44], while the remaining insignificant parameters are set to zero and thus discarded.

A difficulty in estimating the distortion parameters $\{k_1, k_2, k_3, p_1, p_2\}$ is the nonlinearity of $L_N(\cdot)$. By transforming an image point with the normalized image coordinates $x_N = [x_N, y_N, 1]^T$ into another point $\chi_N$ in the eleven-dimensional space with

$$
\chi_N = [(\tilde{x}_N \cdot r^6), (\tilde{y}_N \cdot r^6), (\tilde{x}_N \cdot r^4), (\tilde{y}_N \cdot r^4), (\tilde{x}_N \cdot r^2), (\tilde{y}_N \cdot r^2), (r^2 + 2\tilde{x}_N^2), (r^2 + 2\tilde{y}_N^2), (2\tilde{x}_N \cdot \tilde{y}_N), x_N, y_N]^T,
$$  \hspace{1cm} (3.25)

a linear function for describing the distortion process is obtained as

$$
\begin{bmatrix}
    x_{ND} \\
    y_{ND}
\end{bmatrix} =
\begin{bmatrix}
    k_1 & 0 & k_2 & 0 & k_3 & 0 & p_1 & 0 & p_2 & 1 & 0 \\
    0 & k_1 & 0 & k_2 & 0 & k_3 & 0 & p_2 & p_1 & 0 & 1
\end{bmatrix} \cdot \chi_N.
$$  \hspace{1cm} (3.26)

33
Chapter 3. Imaging system characterization

where the coordinates $\tilde{x}_N$ and $\tilde{y}_N$ are defined with respect to the distortion center $o_{ND} = [ox_{ND}, oy_{ND}, 1]^T$ with $\tilde{x}_N = x_N - ox_{ND}$ and $\tilde{y}_N = y_N - oy_{ND}$. For convenience, $r$ denotes the distance to the distortion center and $r^2 = \tilde{x}_N^2 + \tilde{y}_N^2$. In modeling lens distortion, the distortion center is commonly assumed to be physically coincident with the principal point and hence $[ox_{ND}, oy_{ND}] = [0, 0]$. However, the employed distortion model in general deviates from the actual physical model. Regarding this, the distortion center can be explicitly considered as being independent of the principal point for a better calibration performance.

If the coincidence between the distortion center and the principal point is adopted, the distortion parameters can be solved from the linear equations in Eq. 3.26 given the distorted and distortion-free coordinates of sufficient image points. In [45], an iterative full estimation of $K$, $\{R, t'\}$ and $L_N(·)$ using two alternating steps was suggested. At initialization, the correspondence between the control points on the planar pattern and their images on the image plane is obtained for each view using the known geometric constraints, e.g. appearance and spatial distribution of the control points. After rewriting Eq. 3.7 as $\tilde{x}_{ID} = \mathcal{L}_i(K/p[R|t']\cdot \tilde{x}_W)$ for $p_1 = p_2 = 0$, the camera calibration matrix $K$ and the extrinsic parameters $\{R, t'\}$ for all $D$ views are solved consecutively in the first step, where the inverse of the currently estimated distortion function $\hat{L}_i(·)$ on image coordinates is required. The difference between the observed image $\tilde{x}_{ID}$ of a control point $\tilde{x}_W$ and its virtual projection $\hat{x}_{ID} = \mathcal{L}_i(\hat{K}/\hat{p} [\hat{R} | \hat{t}' ] \cdot \tilde{x}_W)$ is considered as the consequence of the inaccuracy in $\hat{L}_i(·)$. Subsequently and as a complementary step, the distortion parameters are updated according to Eq. 3.26. By repeating these two steps and thus refining the estimated parameters, the reprojection error $\|\tilde{x}_{ID} - \hat{x}_{ID}\|_2$ is globally minimized for all control points observed in $D$ views.

A considerable drawback of the aforementioned two-step optimization is the associated slow convergence of $\|\tilde{x}_{ID} - \hat{x}_{ID}\|_2$. Since the update of the parameters is realized in two separate steps for each iteration, the resulting global reprojection error does not necessarily decrease or the resulting descent is often insufficient. To accelerate the parameter estimation, a bundle adjustment-based full parameter estimation on the observations across all $D$ views is preferred:

$$\hat{\Theta} = \arg \min_{\Theta} \sum_{i=1}^{D} \sum_{j=1}^{N} \|\tilde{x}_{ID, ij} - \hat{x}_{ID, ij}\|_2^2; \quad (3.27)$$

34
3.2. Geometric calibration

where $\tilde{x}_{ID,ij}$ and $\hat{x}_{ID,ij}$ denote the homogeneous image coordinates of the observed image and the virtual projection of the $j$-th control point $\tilde{x}_{w,j}$ in the $i$-th view, respectively, while $\Theta$ denotes the set of all unknown parameters. Moreover, the virtual projection $\hat{x}_{ID,ij}$ is determined by the estimated parameters $\{\hat{K}, \hat{R}_i, \hat{t}'_i, \hat{L}_N(\cdot)\}$ as $\hat{x}_{ID,ij} = \hat{K} \cdot \hat{L}_N(1/\hat{\rho} \cdot [\hat{R}_i][\hat{t}'_i] \cdot \tilde{x}_{w,j})$.

The initial estimate of $\Theta$ for conducting an iterative optimization is solved using the linear equations derived in Section 3.2.1.5 and temporarily setting all distortion parameters to zero.

### 3.2.2 Online correction of lens distortion

Relying on the carefully designed patterns and the accordingly organized calibration procedures, accurate and reliable results are readily accessible using the offline approach described above. However, the use of specific calibration objects also reduces the flexibility of such calibration approaches and thus limits their applicability in practice, especially for cases with difficulties in acquiring additional calibration images without interrupting the regular image acquisition. Simultaneous online calibration during the image acquisition is thus very desirable and provides an attractive alternative to the offline solution.

Similarly to performing the bundle adjustment-based offline parameter estimation in Section 3.2.1.6, a simultaneous estimation of both lens distortion $L_N(\cdot)$ and the camera’s intrinsic parameters in $K$ can also be achieved for online calibration, where the reprojection error in Eq. 3.27 computed for the control points on known planar patterns is replaced by some more general error measures. As demonstrated in [40] and [63], this kind of online imaging system characterization can be realized using either dedicated lens distortion models (e.g. division model) or iterative optimization with reliable initial estimate (e.g. EXIF tags of images). If distortion models other than the specifically suited variants are preferred and the access to a reasonable initial estimate is not provided, the intended parameter estimation turns out to be difficult or even completely infeasible. To remove this restriction, the estimation of $L_N(\cdot)$ and $K$ is hence decoupled into two separate stages, where lens distortion is corrected in advance of estimating the camera calibration matrix $K$. If required, an optional global adjustment of all parameters can also be considered, which gives rise to an additional simultaneous parameter estimation.
Aiming at modeling lens distortion in images, straight lines are extremely useful regarding the related invariance under projective transformations: their images must also be straight. Relying on this “plumb-line” constraint and exploiting line segments in scenes, a series of methods [64, 65, 66, 67, 68, 69] have been developed for recovering the original geometry in distorted images. However, such methods assume the existence of significant line structures and are usually with limited application due to specific distortion models, e.g. polynomial or division model only with isotropic radial components. Apart from such special entities, more general constraints on the projection of 3D points can be induced by their multiple views acquired using the same camera, where the epipolar [70] and trilinear [71] constraints on point correspondences are especially favorable for characterizing lens distortion. In the literature, [39, 40, 41, 72] are the classic methods for recovering distortion-free image coordinates by linking point correspondences across multiple views.

Since the correction of lens distortion should be followed by the estimation of $K$ with the camera’s intrinsic parameters, an exact recovery of the original image coordinates $\tilde{x}_I$ from $\tilde{x}_{ID}$ is therefore indispensable for achieving the intended calibration. Rather than modeling the physical distortion, state-of-the-art methods mostly focus on 2D mapping functions between the distorted and distortion-free image coordinates, while the desired recovery of $\tilde{x}_I$ is not necessarily enforced. Regarding this and employing the widely applicable imaging system model in Eq. 3.7, the entire image formation process of a finite projective camera combined with a non-ideal lens is explicitly investigated in this subsection for revealing the intrinsic correlation between $\tilde{x}_I$ and $\tilde{x}_{ID}$. Furthermore, under a close consideration of the two views-induced epipolar constraint on point correspondences, the feasibility of exactly recovering $\tilde{x}_I$ is carefully reviewed for diverse image correction models, which has not been addressed in the literature so far. As a major extension to a previous publication [8], a more comprehensive and detailed analysis is conducted here.

### 3.2.2.1 Epipolar constraint

As illustrated in Figure 3.6, a 3D point with the homogeneous world coordinates $\tilde{x}_W^E$ is projected to $\tilde{x}_{I,i}$ and $\tilde{x}_{I,j}$ in the two views $i$ and $j$ acquired using the same camera but at different positions and poses. The
rotation matrices and translation vectors between the world frame and the corresponding camera frames of the two views are denoted by $R_i$, $R_j$ and $t'_i$, $t'_j$, respectively. Let $\pi^E = [-(n^T/d), 1]^T$ define a plane in $\mathbb{P}^3$ satisfying $(\pi^E)^T \tilde{x}^E = 0$ for any point $\tilde{x}^E$ on $\pi^E$, where $n \in \mathbb{R}^{3 \times 1}$ denotes the normal vector and $d$ is the distance from the origin of the world frame to this plane. If the camera centers $\tilde{c}_i$ and $\tilde{c}_j$ in the two views are not located on $\pi^E$, the ray starting at $\tilde{c}_j$ and passing through $\tilde{x}^E_w$ intersects the plane $\pi^E$ at a point with the homogeneous planar coordinates $\tilde{x}^E_{\pi 2D,j}$. According to Eq. 3.10, there is a planar homography $H_j$ defining the projection of points on $\pi^E$ in the view $j$ and thus $\tilde{x}_{1,j} \cong H_j \cdot \tilde{x}^E_{\pi 2D,j}$. Since the point $\tilde{x}^E_{\pi 2D,j}$ falls on the line joining the camera center $\tilde{c}_j$ and the 3D point $\tilde{x}^E_w$, its image $H_i \cdot \tilde{x}^E_{\pi 2D,j}$ in the view $i$ is also on the line passing through the epipole $\tilde{e}_i$ and the image point $\tilde{x}_{1,i}$, where $\tilde{e}_i$ is the image of $\tilde{c}_j$ and $H_i$ denotes the planar homography between the plane $\pi^E$ and the view $i$. Following the line definition $\tilde{e}_i \times \tilde{x}_{1,i}$ [46] of the epipolar line $l_i$ joining two points $\tilde{c}_i$ and $\tilde{x}_{1,i}$ on $\mathbb{P}^2$, this collinear constraint on the three image points or the epipolar constraint is written as

$$l_i^T \cdot (H_i \cdot \tilde{x}^E_{\pi 2D,j}) \cong (H_i \cdot \tilde{x}^E_{\pi 2D,j})^T \cdot (\tilde{e}_i \times \tilde{x}_{1,i}) = 0$$

$$\cong (H_i \cdot H_j^{-1} \cdot \tilde{x}_{1,j})^T \cdot (\tilde{e}_i \times \tilde{x}_{1,i})$$

$$\cong (H_j^i \cdot \tilde{x}_{1,j})^T \cdot ([\tilde{e}_i]_x) \cdot \tilde{x}_{1,i}$$

$$= \tilde{x}_{1,j}^T \cdot (H_j^i)^T \cdot ([\tilde{e}_i]_x) \cdot \tilde{x}_{1,i}$$

$$\cong \tilde{x}_{1,j}^T \cdot F \cdot \tilde{x}_{1,i},$$

where $H_j^i \cong H_i \cdot H_j^{-1}$ and the $3 \times 3$ skew-symmetric matrix $([\tilde{e}_i]_x)$ of rank two transforms the cross product $\tilde{e}_i \times \tilde{x}_{1,i}$ into the equivalent form $([\tilde{e}_i]_x) \cdot \tilde{x}_{1,i}$. The $3 \times 3$ matrix $F \cong (H_j^i)^T \cdot ([\tilde{e}_i]_x)$ of rank two is referred to as the fundamental matrix between the two views $i$ and $j$, which gives the epipolar constraint in Eq. 3.28 between point correspondences. If $\pi^E$ is selected as the plane $\pi^E_{\infty}$ at infinity, the fundamental matrix is rewritten as $F \cong K^{-T} \cdot R_j^T \cdot R_i^T \cdot K^T \cdot ([\tilde{e}_i]_x)$ since there are $H_i \cong K \cdot R_i$ and $H_j \cong K \cdot R_j$ from Eq. 3.16. Through an analogous analysis and regarding the epipolar line $l_j \cong \tilde{e}_j \times (H_j \cdot \tilde{x}^E_{\pi 2D,i})$ in the view $j$, the collinear constraint $\tilde{x}_{1,j}^T \cdot l_j = 0$ leads to an alternative expression of the fundamental matrix: $F \cong ([\tilde{e}_j]_x) \cdot H_j^i \cong ([\tilde{e}_j]_x) \cdot K \cdot R_j \cdot R_i^{-1} \cdot K^{-1}$.
Figure 3.6: Epipolar geometry. There exist two collinear constraints: \((\bar{e}_i \times \bar{x}_{1,i})^T \mathcal{H}_i \cdot \bar{x}^E_{\pi 2D,j} = 0\) and \((\bar{e}_j \times \bar{x}_{1,j})^T \mathcal{H}_j \cdot \bar{x}^E_{\pi 2D,i} = 0\).

### 3.2.2.2 Model reformulation

In the bundle adjustment Eq. 3.27 of offline calibration, the error to be minimized is computed with respect to the observed image coordinates \(\tilde{x}_{1D}\) with lens distortion. Therefore, the lens distortion function \(\mathcal{L}_N(\cdot)\) is preferred, as the virtual image \(\hat{x}_{1D}\) can be directly obtained by substituting the estimated distortion-free image \(\hat{x}_N = \hat{K}^{-1} \tilde{x}_I\) in \(\mathcal{L}_N(\cdot)\). But for the epipolar constraint-guided parameter estimation, the inverse distortion function \(\mathcal{L}_N^{-1}(\cdot)\) for \(x_N = \mathcal{L}_N^{-1}(x_{ND}, o_{ND})\) is much more convenient since the collinearity error with respect to Eq. 3.28 is minimized by recovering the original point correspondences \(\tilde{x}_{1,i} \leftrightarrow \tilde{x}_{1,j}\) from the distorted image coordinates \(\tilde{x}_{1D,i}\) and \(\tilde{x}_{1D,j}\).

Out of the numerous publications in the literature for correcting lens distortion, the polynomial [39] (PCM), division [40] (DCM) and rational [41] (RCM) models have found their dominant application in practice. As the inverse formula to the distortion model in Eq. 3.6 and in general with different model parameters for the correction purpose, Zhang con-
ducted the recovery of image coordinates in [39] using a polynomial function (PF) consisting of isotropic radial and decentering fractions while omitting insignificant thin prism component. In the distorted image coordinates, this model [51,73] is expressed as

\[
\begin{bmatrix}
\tilde{x}_I \\
\tilde{y}_I
\end{bmatrix} = \begin{bmatrix}
x_{ID} \\
y_{ID}
\end{bmatrix} + (k_1 \cdot r^2 + k_2 \cdot r^4 + k_3 \cdot r^6 + \ldots) \begin{bmatrix}
\tilde{x}_{ID} \\
\tilde{y}_{ID}
\end{bmatrix} + (1 + p_3 \cdot r^2 + \ldots) \begin{bmatrix}
p_1 \cdot (r^2 + 2\tilde{x}_{ID}^2) + 2p_2 \cdot \tilde{x}_{ID} \cdot \tilde{y}_{ID} \\
p_2 \cdot (r^2 + 2\tilde{y}_{ID}^2) + 2p_1 \cdot \tilde{x}_{ID} \cdot \tilde{y}_{ID}
\end{bmatrix},
\]

(3.29)

where \( \tilde{x}_{ID} = x_{ID} - o_x \) and \( \tilde{y}_{ID} = y_{ID} - o_y \). \( r = \sqrt{\tilde{x}_{ID}^2 + \tilde{y}_{ID}^2} \) denotes the radial distance between the distortion center \( o_{ID} = [o_x, o_y, 1]^T \) in image coordinates and the distorted image position \( \tilde{x}_{ID} \). Especially, this image coordinates-based correction is derived for the case where the distortion center and the principal point are coincident. Two further assumptions are the known aspect ratio \( a \) and the skew factor \( s = 0 \). To reduce the time budget required for achieving the desired correction, Fitzgibbon employed a division function (DF) with one single correction parameter \( \kappa \):

\[
\begin{bmatrix}
x_I \\
y_I
\end{bmatrix} = \begin{bmatrix}
o_x \\
o_y
\end{bmatrix} + \frac{1}{1 + \kappa \cdot r^2} \begin{bmatrix}
\tilde{x}_{ID} \\
\tilde{y}_{ID}
\end{bmatrix},
\]

(3.30)

For general lens distortion, Claus and Fitzgibbon introduced a rational function (RF) similar to Eq. 3.26 by lifting any image point \( \tilde{x}_{ID} \) to another point \( \chi_{ID} = [x_{ID}^2, x_{ID}, y_{ID}^2, x_{ID}, y_{ID}, 1]^T \) in a six-dimensional space:

\[
\tilde{x}_{1} \cong M_{1} \cdot \chi_{ID},
\]

(3.31)

where \( M_{1} \) is a \( 3 \times 6 \) matrix for mapping \( \tilde{x}_{ID} \) to \( \tilde{x}_{1} \) up to a non-zero scale factor.

Considering the physical fundamentals of lens distortion introduced in Section 3.1.2.1 and aiming at revealing the intrinsic correlation between \( \tilde{x}_{1} \) and \( \tilde{x}_{ID} \) rather than a non-metric mapping function for \( \tilde{x}_{ID} \mapsto \tilde{x}_{1} \), lens distortion is corrected on normalized image coordinates and the recovered image coordinates from \( \tilde{x}_{ID} \) are obtained as

\[
\tilde{x}_{1} = K \cdot x_N \cong K \cdot \mathcal{L}_N^{-1}(x_{ND}, o_{ND}) = K \cdot \mathcal{L}_N^{-1}(K^{-1} \cdot \tilde{x}_{ID}, o_{ND}).
\]

(3.32)
Chapter 3. Imaging system characterization

The resulting image correction models $L_N^{-1}(\cdot)$ using PF (PCM), DF (DCM) and RF (RCM) are in the same forms as Eq. 3.29, 3.30 and 3.31, respectively, but with different parameters.

There arises a question before moving forward to estimating unknown parameters in Eq. 3.32: which parameters can be determined given the epipolar constraint? For PCM and DCM, the aspect ratio $a$ and the skew factor $s$ in $K$ lead to non-metric and affine transformations of the mapping $x_{ND} \mapsto \tilde{x}_{ID}$. These additional geometric transformations change the shape of lens distortion on the image plane and thus give rise to further geometric distortion in images. Moreover, the absolute position of the distortion center is irrelevant as it can be canceled by appropriately transforming coordinate systems. In this context, a reduction of the parameter space is necessary and serves to prevent the estimation of unconstrained parameters. By moving the origin of the normalized camera frame to $[ox_{ND}, oy_{ND}, 0]^T$ and scaling the resulting coordinates by the focal length $f$, a point with the distorted coordinates $x_{ND}$ is transformed into $x_{ND} = [x_N, y_N, 1]^T = [(x_{ND} - ox_{ND}) \cdot f, (y_{ND} - oy_{ND}) \cdot f, 1]^T$. Consequently, the image correction in Eq. 3.32 is also transformed into

$$\tilde{x}_1 = K \cdot x_N \cong K \cdot L_N^{-1}(x_{ND}) = K \cdot L_N^{-1}(K^{-1} \cdot \tilde{x}_{ID}), \quad (3.33)$$

where $x_N = [x_N, y_N, 1]^T = [(x_N - ox_{ND}) \cdot f, (y_N - oy_{ND}) \cdot f, 1]^T$ and

$$K = \begin{bmatrix} 1 & s & p \\ 0 & a & q \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & s & px_1 + (ox_{ND} + oy_{ND} \cdot s) \cdot f \\ 0 & a & py_1 + oy_{ND} \cdot a \cdot f \\ 0 & 0 & 1 \end{bmatrix} \quad (3.34)$$

with $p = px_1 + (ox_{ND} + oy_{ND} \cdot s) \cdot f$ and $q = py_1 + oy_{ND} \cdot a \cdot f$. The new correction model $L_N^{-1}(\cdot)$ is of the same form as $L_N^{-1}(\cdot)$, but with different correction parameters. For instance, the new DCM is written as

$$\begin{bmatrix} x_N \\ y_N \end{bmatrix} = \frac{1}{1 + \kappa \cdot r^2} \cdot \begin{bmatrix} x_{ND} \\ y_{ND} \end{bmatrix}, \quad (3.35)$$

where $r^2 = x_{ND}^2 + y_{ND}^2$ and the new correction parameter is $\kappa = \kappa / f^2$.

In the case of RCM, the lifting of the point $\tilde{x}_{ID}$: $\tilde{x}_{ID} \mapsto \chi_{ID}$ becomes $K \cdot x_{ND} \mapsto \chi_{ID}$ due to the fact $\tilde{x}_{ID} = K \cdot x_{ND}$. As all entries in the vector $\tilde{x}_{ID} = [x_{ID}, y_{ID}, 1]^T$ are linear combinations of the entries in the vector $x_{ND} = [x_{ND}, y_{ND}, 1]^T$ defined by $K$, the point $x_{ND}$ can also be lifted to another point $\chi_{ND} = [x_{ND}^2, x_{ND} \cdot y_{ND}, y_{ND}^2, x_{ND}, y_{ND}, 1]^T$ in the corresponding
six dimensional space so that there is \( \chi_{ID} = B \cdot \chi_{ND} \) with a 6×6 invertible matrix \( B \). Through the correction of lens distortion on normalized camera coordinates with \( x_N \cong M_N \cdot \chi_{ND} \), the distortion-free image coordinates \( \tilde{x}_i \) are recovered according to \( \tilde{x}_i \cong K \cdot M_N \cdot B^{-1} \cdot \chi_{ID} \). Comparing this expression to Eq. 3.31, the 3×6 parameter matrix \( M_I \) is actually equivalent to the product of the three matrices \( K \), \( M_N \) and \( B^{-1} \). However, if only \( M_I \) is obtained from the epipolar constraint on image point correspondences, a decomposition \( M_I \cong K \cdot M_N \cdot B^{-1} \) for determining \( M_N \) and the camera’s intrinsic parameters is in general infeasible.

### 3.2.2.3 Recovering distortion-free image coordinates

The objective of the epipolar constraint-guided image correction is to find the optimal estimate \( \hat{K} \) and \( \hat{L}_N^{-1} \) for minimizing the residual error with respect to Eq. 3.28 on corresponding image points across multiple views. For two views \( i \) and \( j \), the images of a 3D point \( \tilde{x}_w \) obey the constraint

\[
\tilde{x}_{1,j}^T \cdot F \cdot \tilde{x}_{1,i} = \tilde{x}_{1,j}^T \cdot K^{-T} \cdot K^T \cdot F \cdot K \cdot K^{-1} \cdot \tilde{x}_{1,i} = 0
\]

\[
= \tilde{x}_{N,j}^T \cdot H_j \cdot \hat{F} \cdot H_i \cdot x_N, i
\]

\[
\cong (L_N^{-1}(K^{-1} \cdot \tilde{x}_{ID,j}))^T \cdot \hat{F} \cdot L_N^{-1}(K^{-1} \cdot \tilde{x}_{ID,i}),
\]

where \( F = K^T \cdot F \cdot K \). In image correction, \( K \), \( L_N^{-1}(\cdot) \) and \( F \) are unknown variables being simultaneously estimated. It is also possible to introduce two 3×3 invertible matrices \( H_i \) and \( H_j \) into the equation above:

\[
x_{N,j}^T \cdot F \cdot x_N, i = x_{N,j}^T \cdot H_j \cdot H_j^{-T} \cdot F \cdot H_i^{-1} \cdot H_i \cdot x_N, i = 0
\]

\[
= x_{N,j}^T \cdot \hat{F} \cdot H_i \cdot x_N, i
\]

\[
\cong (L_N^{-1}(K^{-1} \cdot \tilde{x}_{ID,j}))^T \cdot \hat{F} \cdot L_N^{-1}(K^{-1} \cdot \tilde{x}_{ID,i}),
\]

where \( \hat{F} = H_j^{-T} \cdot F \cdot H_i^{-1} \). Apparently, if the left and the right side of \( \hat{F} \) in the last row of Eq. 3.37 can be rewritten as

\[
H_j \cdot L_N^{-1}(K^{-1} \cdot \tilde{x}_{ID,j}) \cong \hat{L}_N^{-1}(K^{-1} \cdot \tilde{x}_{ID,j}),
\]

\[
H_i \cdot L_N^{-1}(K^{-1} \cdot \tilde{x}_{ID,i}) \cong \hat{L}_N^{-1}(K^{-1} \cdot \tilde{x}_{ID,i}),
\]

ambiguous solutions \( \hat{K} \) and \( \hat{L}_N^{-1}(\cdot) \) could exist different from the original \( K \) and \( L_N^{-1} \), but still satisfy the epipolar constraint in the form.
(\hat{\mathbf{L}}^{-1}_N(\hat{\mathbf{K}}^{-1} \cdot \hat{x}_{\text{id},j}))^T \cdot \hat{\mathbf{F}} \cdot \mathbf{L}^{-1}_N(\hat{\mathbf{K}}^{-1} \cdot \hat{x}_{\text{id},i})$, where the original distortion-free image coordinates \( \hat{x}_{i,i} \) and \( \hat{x}_{i,j} \) cannot be recovered anymore. Regarding the equivalence between the two equations due to their symmetric forms on the two views \( i \) and \( j \), only the generalized form of Eq. 3.38 without any dependence on a specific view is further considered for investigating the existence of ambiguous solutions.

By substituting \( \hat{\mathbf{K}}^{-1} \cdot \hat{x}_{\text{id}} \) with \( x_{\text{ND}} \) and \( \hat{\mathbf{K}}^{-1} \cdot \hat{x}_{\text{id}} \) with \( \hat{x}_{\text{ND}} \), the generalized form of Eq. 3.38 is simplified and turns into

\[
\mathbf{H} \cdot \mathbf{L}^{-1}_N(x_{\text{ND}}) \approx \hat{\mathbf{L}}^{-1}_N(\hat{x}_{\text{ND}}) = \hat{\mathbf{L}}^{-1}_N(T \cdot x_{\text{ND}}),
\]

where \( \mathbf{H} = [h_1, h_2, h_3]^T \) with row vectors \( \{h_1^T, h_2^T, h_3^T\} \) and

\[
\hat{x}_{\text{ND}} = T \cdot x_{\text{ND}}
\]

with

\[
T = \begin{bmatrix}
1 & t_s & t_p \\
0 & t_a & t_q \\
0 & 0 & 1
\end{bmatrix} = \hat{\mathbf{K}}^{-1} \cdot \mathbf{K}
\]

\[
= \begin{bmatrix}
1 & s - \hat{s} \cdot \hat{a}/\hat{\hat{a}} & p - q \cdot \hat{s}/\hat{\hat{a}} - \hat{\hat{p}} + \hat{\hat{q}} \cdot \hat{s}/\hat{\hat{a}} \\
0 & \hat{a}/\hat{\hat{a}} & q/\hat{\hat{a}} - \hat{\hat{q}}/\hat{\hat{a}} \\
0 & 0 & 1
\end{bmatrix} \quad (3.41)
\]

Depending on the concrete form of the inverse function \( \mathbf{L}^{-1}_N \), the recovery of \( \mathbf{K} \) and \( \mathbf{L}^{-1}_N \) is in general different for each of the image correction models \( \text{PCM, DCM, and RCM} \). For this reason, a detailed analysis on the feasibility of recovering the original distortion-free image coordinates \( \hat{x}_1 \) is conducted for each of the three models individually.

### 3.2.2.4 Polynomial model

In the case of \( \text{PCM} \), a hierarchical analysis for \( \mathbf{PF} \) with different terms, e.g. isotropic radial component of different orders and decentering component, is desired, given their wide application in diverse image correction methods. At first, the basic form of \( \text{PCM} \) using the \( k_1 \)-term is considered and applied in Eq 3.39:

\[
(1/\eta) \cdot \mathbf{H} \cdot x_N = (1/\eta) \cdot \mathbf{H} \cdot \mathbf{L}^{-1}_N(x_{\text{ND}}) = \hat{\mathbf{L}}^{-1}_N(\hat{x}_{\text{ND}}) = \hat{x}_N
\]

\[
\iff \quad (1/\eta) \cdot \mathbf{H} \cdot \begin{bmatrix} x_{\text{ND}} \cdot (1 + \hat{k}_1 \cdot \hat{r}^2) \\ y_{\text{ND}} \cdot (1 + \hat{k}_1 \cdot \hat{r}^2) \\ 1 \end{bmatrix} = \begin{bmatrix} \hat{x}_{\text{ND}} \cdot (1 + \hat{k}_1 \cdot \hat{r}^2) \\ \hat{y}_{\text{ND}} \cdot (1 + \hat{k}_1 \cdot \hat{r}^2) \\ 1 \end{bmatrix},
\]

\[
(3.42)
\]
where \( r^2 = x_{ND}^2 + y_{ND}^2 \) and \( \hat{r}^2 = \hat{x}_{ND}^2 + \hat{y}_{ND}^2 \). The equivalence “\( \cong \)” between \( \mathcal{H} \cdot \mathcal{L}_N^{-1}(\mathbf{x}_{ND}) \) and \( \tilde{\mathcal{L}}_N^{-1}(\hat{\mathbf{x}}_{ND}) \) is replaced with the equality “\( = \)” between \( (1/\eta) \mathcal{H} \cdot \mathcal{L}_N^{-1}(\mathbf{x}_{ND}) \) and \( \tilde{\mathcal{L}}_N^{-1}(\hat{\mathbf{x}}_{ND}) \) by introducing a non-zero scale factor \( \eta \). As proven in Appendix B.1.1, the last row \( \mathbf{h}_3^T \) of the matrix \( \mathcal{H} \) is of the form \( \mathbf{h}_3^T = [0, 0, \eta] \) for distorted images with \( k_1 \neq 0 \). If not stated otherwise, \( \mathcal{H} \) is always scaled by the factor \( (1/\eta) \) for computational convenience so that \( \mathbf{h}_3^T \) becomes \([0, 0, 1]\) and \( \eta \) cancels out in Eq. 3.42:

\[
\begin{bmatrix}
\mathbf{h}_{11}^T \\
\mathbf{h}_{12}^T \\
\mathbf{h}_{13}^T
\end{bmatrix}
\begin{bmatrix}
x_{ND} \cdot (1 + k_1 \cdot \hat{r}^2) \\
y_{ND} \cdot (1 + k_1 \cdot \hat{r}^2) \\
1
\end{bmatrix}
= 
\begin{bmatrix}
\hat{x}_{ND} \cdot (1 + \hat{k}_1 \cdot \hat{r}^2) \\
\hat{y}_{ND} \cdot (1 + \hat{k}_1 \cdot \hat{r}^2)
\end{bmatrix}
\tag{3.43}
\]

with \( \mathbf{h}_1^T = [\mathbf{h}_{11}, \mathbf{h}_{12}, \mathbf{h}_{13}] \) and \( \mathbf{h}_2^T = [\mathbf{h}_{21}, \mathbf{h}_{22}, \mathbf{h}_{23}] \).

As derived in Appendix B.1.2, the equations above can be expanded and simplified into:

\[
\mathbf{h}_{11} \cdot \hat{k}_1 \cdot x_{ND}^3 + \mathbf{h}_{12} \cdot \hat{k}_1 \cdot x_{ND}^2 \cdot y_{ND} + \mathbf{h}_{11} \cdot \hat{k}_1 \cdot x_{ND} \cdot y_{ND}^2 + \mathbf{h}_{12} \cdot \hat{k}_1 \cdot y_{ND}^3 + \\
\mathbf{h}_{11} \cdot x_{ND} + \mathbf{h}_{12} \cdot y_{ND} + \mathbf{h}_{13} = \\
\hat{k}_1 \cdot x_{ND}^3 + 3 \hat{k}_1 \cdot t_s \cdot x_{ND}^2 \cdot y_{ND} + \hat{k}_1 \cdot (t_a^2 + 3 \cdot t_s^2) \cdot x_{ND} \cdot y_{ND}^2 + \\
\hat{k}_1 \cdot (t_a^2 \cdot t_s + t_s^3) \cdot y_{ND}^3 + 3 \hat{k}_1 \cdot t_p \cdot x_{ND}^2 + \hat{k}_1 \cdot (2 \cdot t_a \cdot t_q + 6 \cdot t_p \cdot t_s) \cdot x_{ND} \cdot y_{ND} + \\
\hat{k}_1 \cdot (t_a^2 \cdot t_p + 2 \cdot t_a \cdot t_q \cdot t_s + 3 \cdot t_p \cdot t_s^2) \cdot y_{ND}^2 + (3 \hat{k}_1 \cdot t_p^2 + \hat{k}_1 \cdot t_q^2 + 1) \cdot x_{ND} + \\
(3 \hat{k}_1 \cdot t_s^2 \cdot t_s + 2 \hat{k}_1 \cdot t_a \cdot t_p \cdot t_q + \hat{k}_1 \cdot t_q \cdot t_s + t_s) \cdot y_{ND} + \\
\hat{k}_1 \cdot t_p^3 + \hat{k}_1 \cdot t_p^2 \cdot t_q + t_p,
\]

and

\[
\mathbf{h}_{21} \cdot \hat{k}_1 \cdot x_{ND}^3 + \mathbf{h}_{22} \cdot \hat{k}_1 \cdot x_{ND}^2 \cdot y_{ND} + \mathbf{h}_{21} \cdot \hat{k}_1 \cdot x_{ND} \cdot y_{ND}^2 + \mathbf{h}_{22} \cdot \hat{k}_1 \cdot y_{ND}^3 + \\
\mathbf{h}_{21} \cdot x_{ND} + \mathbf{h}_{22} \cdot y_{ND} + \mathbf{h}_{23} = \\
\hat{k}_1 \cdot t_a \cdot x_{ND} \cdot y_{ND} + 2 \hat{k}_1 \cdot t_a \cdot t_s \cdot x_{ND} \cdot y_{ND}^2 + \hat{k}_1 \cdot (t_a^2 + t_a \cdot t_s^2) \cdot y_{ND}^3 + \\
\hat{k}_1 \cdot t_q \cdot x_{ND}^2 + \hat{k}_1 \cdot (2 \cdot t_a \cdot t_p + 2 \cdot t_q \cdot t_s) \cdot x_{ND} \cdot y_{ND} + \\
\hat{k}_1 \cdot (3 \cdot t_a \cdot t_q + 2 \cdot t_a \cdot t_p \cdot t_s + t_q \cdot t_s^2) \cdot y_{ND}^2 + 2 \hat{k}_1 \cdot t_p \cdot t_q \cdot x_{ND} + \\
(\hat{k}_1 \cdot t_a \cdot t_p^2 + 2 \hat{k}_1 \cdot t_p \cdot t_q \cdot t_s + 3 \hat{k}_1 \cdot t_a \cdot t_q^2 + t_a) \cdot y_{ND} + \\
\hat{k}_1 \cdot t_p^2 \cdot t_q + \hat{k}_1 \cdot t_q^3 + t_q,
\]

3.2. Geometric calibration
where \( \hat{x}_{ND} = [\hat{x}_{ND}, \hat{y}_{ND}, 1]^T = \mathcal{T} \cdot \hat{x}_{ND} \) with \( \hat{x}_{ND} = [x_{ND}, y_{ND}, 1]^T \) are written as linear combinations of elements in \( \{x_{ND}, y_{ND}, 1\} \). The epipolar constraint is valid for the projections of an arbitrary 3D point on two views. As a result, the entries \( x_{ND} \) and \( y_{ND} \) in the coordinate vector \( \hat{x}_{ND} \) can take arbitrary values. This implies an important constraint on the terms on both sides of Eq. 3.44 and 3.45: summing up the corresponding coefficients of \( x_{ND}^i \) or \( y_{ND}^i \) must give the same value on the left and on the right side of the corresponding equation. For instance, the total coefficients of \( x_{ND} \) and \( y_{ND} \) in Eq. 3.44 and 3.45 are \( \mathbf{h}_{11} \cdot \mathbf{k}_1 \) on the left side of Eq. 3.44 and \( \mathbf{k}_1 \cdot (t_a^2 + 3 \cdot t_s^2) \) on the right side. To enforce \( \mathbf{x}_{ND} \cdot \mathbf{h}_{11} \cdot \mathbf{k}_1 = \mathbf{x}_{ND} \cdot \mathbf{k}_1 \cdot (t_a^2 + 3 \cdot t_s^2) \), there must be \( \mathbf{h}_{11} \cdot \mathbf{k}_1 = \mathbf{k}_1 \cdot (t_a^2 + 3 \cdot t_s^2) \). Recalling the existence of \( \mathcal{H}^{-1} \) for \( \mathcal{H} \), the first two entries \( \mathbf{h}_{11} \) and \( \mathbf{h}_{12} \) in the row vector \( \mathbf{h}_1^T \) cannot take the value zero at the same time, otherwise \( \mathbf{h}_1^T \equiv \mathbf{h}_3^T \) and \( \det(\mathcal{H}) = 0 \). From the coefficient constraint on \( x_{ND}^3 \) and \( y_{ND}^3 \) in Eq. 3.44, the estimate \( \hat{k}_1 \) of the distortion parameter obeys \( \hat{k}_1 \neq 0 \) since \( \mathbf{k}_1 \neq 0 \). It is notable that the second-order terms \( \{x_{ND}^2, x_{ND} \cdot y_{ND}, y_{ND}^2\} \) are permanently missing on the left side of both Eq. 3.44 and 3.45. The coefficients of \( x_{ND}^2 \) on the right side are thus equal to zero: \( 3 \cdot \hat{k}_1 \cdot t_p = \hat{k}_1 \cdot t_q = 0 \), which gives the solution for \( t_p \) and \( t_q \) with \( t_p = t_q = 0 \). Consequently, the term \( x_{ND} \) vanishes in Eq. 3.45 and the corresponding coefficients enforce \( \mathbf{h}_{21} = 0 \). A direct solution for \( t_s \) and \( t_a \) cannot be derived from the currently available results, but the intrinsic relationship \( t_s \cdot t_a = 0 \) is obtained as \( 2 \cdot \hat{k}_1 \cdot t_a \cdot t_s = \mathbf{h}_{21} \cdot \mathbf{k}_1 = 0 \) for the term \( x_{ND} \cdot y_{ND}^2 \) in Eq. 3.45. With a further consideration of the equality \( \mathbf{h}_{22} = (\hat{k}_1 \cdot t_a \cdot t_p^2 + 2 \cdot \hat{k}_1 \cdot t_p \cdot t_q \cdot t_s + 3 \cdot \hat{k}_1 \cdot t_a \cdot t_q^2 + t_a) \) between the coefficients of \( y_{ND} \) in the same equation, there is \( \mathbf{h}_{22} = t_a \). Apparently, \( t_a \) must be different from zero for denying \( \mathbf{h}_3 \equiv \mathbf{h}_3^T \) and \( \det(\mathcal{H}) = 0 \), which leads to \( t_s = 0 \). With the help of the solved variables, the determination of the remaining unknown values of \( \{\hat{k}_1, t_a, \mathbf{h}_{11}, \mathbf{h}_{12}, \mathbf{h}_{13}, \mathbf{h}_{22}, \mathbf{h}_{23}\} \) is straightforward. Finally, \( \hat{K}, \mathcal{T}, \mathcal{H} \) and \( \hat{\mathcal{L}}^{-1}_{N}(\cdot) \) are obtained as

\[
\hat{K} = \begin{bmatrix}
1 & \pm s & p \\
0 & \pm a & q \\
0 & 0 & 1
\end{bmatrix}, \quad \mathcal{T} = \mathcal{H} = \begin{bmatrix}
1 & 0 & 0 \\
0 & \pm 1 & 0 \\
0 & 0 & 1
\end{bmatrix}, \quad \hat{\mathcal{L}}^{-1}_{N}(\cdot) = \mathcal{L}^{-1}_{N}(\cdot). \quad (3.46)
\]

It is interesting to see that there is an ambiguity in the sign of \( s \) and \( a \). Depending on the conducted parameter estimation, the corrected coordinates \( \hat{x}_{N} = \hat{\mathcal{L}}^{-1}_{N}(\hat{K}^{-1} \cdot \hat{x}_{ID}) \) are identical to the original distortion-free coordinates \( x_{N} = \mathcal{L}^{-1}_{N}(K^{-1} \cdot x_{ID}) \) or rotated by 180 degrees about the x-axis with \( \hat{x}_{N} = x_{N} \) and \( \hat{y}_{N} = -y_{N} \).
3.2. Geometric calibration

If the basic form of PCM is further extended with the $k_2$ and the $k_3$-term hierarchically, the same results can be obtained through a similar analysis on the more complex models. Besides the isotropic radial component defined by the parameters $\{k_1, k_2, k_3\}$, the decentering component defined by the parameters $\{p_1, p_2\}$ is often also involved in PCM for a better correction performance. To this end, the extended form of PCM relying on the $k_1$, $p_1$ and $p_2$ parameters is considered in addition to the models merely regarding the isotropic radial component. The resulting equation based on Eq. 3.39 is expressed as

$$
\mathcal{H} \cdot \begin{bmatrix}
\mathcal{X}_{ND} \cdot (1 + k_1 \cdot r^2) + p_1 \cdot (r^2 + 2 \mathcal{Y}_{ND} \cdot \mathcal{X}_{ND} \cdot \mathcal{Y}_{ND}) + 2 p_2 \cdot \mathcal{X}_{ND} \cdot \mathcal{Y}_{ND} \\
\mathcal{Y}_{ND} \cdot (1 + k_1 \cdot r^2) + p_2 \cdot (r^2 + 2 \mathcal{X}_{ND} \cdot \mathcal{X}_{ND} \cdot \mathcal{Y}_{ND}) + 2 p_1 \cdot \mathcal{X}_{ND} \cdot \mathcal{Y}_{ND}
\end{bmatrix} = \begin{bmatrix}
\hat{\mathcal{X}}_{ND} \cdot (1 + \hat{k}_1 \cdot \hat{r}^2) + \hat{p}_1 \cdot (\hat{r}^2 + 2 \hat{\mathcal{Y}}_{ND} \cdot \hat{\mathcal{X}}_{ND} \cdot \hat{\mathcal{Y}}_{ND}) + 2 \hat{p}_2 \cdot \hat{\mathcal{X}}_{ND} \cdot \hat{\mathcal{Y}}_{ND} \\
\hat{\mathcal{Y}}_{ND} \cdot (1 + \hat{k}_1 \cdot \hat{r}^2) + \hat{p}_2 \cdot (\hat{r}^2 + 2 \hat{\mathcal{X}}_{ND} \cdot \hat{\mathcal{X}}_{ND} \cdot \hat{\mathcal{Y}}_{ND}) + 2 \hat{p}_1 \cdot \hat{\mathcal{X}}_{ND} \cdot \hat{\mathcal{Y}}_{ND}
\end{bmatrix} \cdot 1.
$$

\hspace{1cm} (3.47)

After an analysis similar to Appendix B.1.1, it can be proven that the last row $\mathbf{h}_3^T$ of $\mathcal{H}$ is of the special form $[0, 0, \eta]$ again.

Due to the increased complexity of the above equations in comparison to Eq. 3.42, the image corrections $\mathbf{x}_N = \mathcal{L}_N^{-1}(\mathbf{x}_{ND})$ and $\hat{\mathbf{x}}_N = \hat{\mathcal{L}}_N^{-1}(\hat{\mathbf{x}}_{ND})$ are rewritten using the lifted coordinates $\mathbf{X}_{ND}$ and $\hat{\mathbf{X}}_{ND}$ with

$$
\mathbf{X}_{ND} = [x^3_{ND}, x^2_{ND}, y^3_{ND}, y^2_{ND}, x^2_{ND}, y^2_{ND}, x_{ND}, y_{ND}, 1]^T;
\hat{\mathbf{X}}_{ND} = [\hat{x}^3_{ND}, \hat{x}^2_{ND}, \hat{y}^3_{ND}, \hat{y}^2_{ND}, \hat{x}^2_{ND}, \hat{y}^2_{ND}, \hat{x}_{ND}, \hat{y}_{ND}, 1]^T.
$$

\hspace{1cm} (3.48)

for better readability:

$$
\mathbf{x}_N = \begin{bmatrix}
k_1 & 0 & k_1 & 0 & 3p_1 & 2p_2 & p_1 & 1 & 0 & 0 \\
0 & k_1 & 0 & k_1 & p_2 & 2p_1 & 3p_2 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix} \cdot \mathbf{X}_{ND},
$$

\hspace{1cm} (3.49)

$$
\hat{\mathbf{x}}_N = \begin{bmatrix}
\hat{k}_1 & 0 & \hat{k}_1 & 0 & 3\hat{p}_1 & 2\hat{p}_2 & \hat{p}_1 & 1 & 0 & 0 \\
0 & \hat{k}_1 & 0 & \hat{k}_1 & \hat{p}_2 & 2\hat{p}_1 & 3\hat{p}_2 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix} \cdot \hat{\mathbf{X}}_{ND}.
$$
Chapter 3. Imaging system characterization

Consequently, Eq. 3.47 with $b_3^T = [0, 0, 1]$ and $\eta = 1$ is also transformed into

$$\mathcal{H} \cdot M_N \cdot \chi_{ND} = \hat{M}_N \cdot \hat{\chi}_{ND} = \hat{M}_N \cdot B_N \cdot \chi_{ND}$$

$$\implies (b_1^T \cdot M_N) \cdot \chi_{ND} = (\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \cdot M_N) \cdot B_N \cdot \chi_{ND}$$

$$\iff M_N' = \hat{M}_N' \cdot B_N$$

for arbitrary $x_{ND}$ and $y_{ND}$, where $B_N$ defines the transformation between the lifted coordinates $\hat{x}_{ND}$ and $\chi_{ND}$ with $\hat{x}_{ND} = B_N \cdot \chi_{ND}$ corresponding to $\tilde{x}_{ND} = T \cdot x_{ND}$. Thus, the 10×10 invertible matrix $B_N$ can be derived from the coordinate transformation matrix $T$ and its explicit form is given in Eq. B.11 and B.12. Apparently, sufficient equations in the unknown parameters of $b_1$, $b_2$, $T$ and $\hat{L}_N^{-1}(\cdot)$ can be obtained from Eq. 3.50 for each entry in $M_N'$. By solving these equations provided in B.2.2, ambiguous solutions, especially for the parameters $\{t_p, t_q, \hat{p}_1, \hat{p}_2\}$, are obtained. This reveals the fact that it is infeasible to recover the original PCM and distortion-free image coordinates without ambiguities if only the $k_1$, $p_1$ and $p_2$ parameters are considered for the correction model $L_N^{-1}(\cdot)$.

The most interesting PCM for practical applications is the variant with the full parameters $\{k_1, k_2, k_3, p_1, p_2\}$. Again, the distortion-free coordinates $x_N$ and $\hat{x}_N$ are obtained as $x_N = M_N \cdot \chi_{ND}$ and $\hat{x}_N = \hat{M}_N \cdot \hat{\chi}_{ND}$ according to Eq. 3.49, where $\chi_{ND}$, $\hat{\chi}_{ND}$, $M_N$ and $\hat{M}_N$ are significantly extended due to the additionally introduced parameters $\{k_2, k_3\}$, which also result in additional higher order terms $x_{iND} \cdot y_{jND}$ and $\hat{x}_{iND} \cdot \hat{y}_{iND}$ with $(i + j) \in \{5, 7\}$ in the correction models $L_N^{i}(\cdot)$ and $\hat{L}_N^{i}(\cdot)$, respectively. Unfortunately, as derived in Section B.3.1, the variables $\chi_{ND}$, $\hat{\chi}_{ND}$, $M_N$ and $\hat{M}_N$ are now of huge sizes. For examples, the vector $\chi_{ND}$ with the lifted coordinates has 36 entries. As a result, solving all unknown parameters in Eq. 3.50 becomes extremely difficult if the equality constraint $M_N' = \hat{M}_N' \cdot B_N$ is considered for each entry in the matrix $M_N'$.

To reduce the complexity in solving Eq. 3.50 in the case of the full parameters $\{k_1, k_2, k_3, p_1, p_2\}$, the entries $\{x_{iND} \cdot y_{iND} | (i + j) = 6\}$ in $\chi_{ND}$ are explicitly considered on both sides of the equation

$$\mathcal{H} \cdot M_N \cdot \chi_{ND} = \hat{M}_N \cdot B_N \cdot \chi_{ND}.$$  

(3.51)
3.2. Geometric calibration

As stated in Eq. B.16, these entries vanish on the left side of the equation above since the corresponding coefficients in the row vectors of $M_N$ are all zero. Regarding the arbitrariness of $x_{ND}$ and $y_{ND}$, this imposes that the same entries must also vanish on the right side of the equation. From Eq. 3.40 with $[\hat{x}_{ND}, \hat{y}_{ND}, 1]^T = \mathbf{T} \cdot [x_{ND}, y_{ND}, 1]^T$, it is straightforward to see that $\hat{x}_{ND}$ and $\hat{y}_{ND}$ are linear combinations of elements in $\{x_{ND}, y_{ND}, 1\}$. Therefore, the entries $\{x_{i}^{\prime} \cdot y_{j}^{\prime} \mid i + j = 6\}$ are only related to the entries $\{\hat{x}_{i} \cdot \hat{y}_{j} \mid (i + j) \in \{6, 7\}\}$ in $\hat{x}_{ND}$. Substituting $B_N \cdot x_{ND} = \hat{x}_{ND}$ into Eq. 3.51, its alternative expression is written as

$$\mathcal{H} \cdot M_N \cdot x_{ND} = \hat{M}_N \cdot \hat{x}_{ND}. \quad (3.52)$$

In the same form as $M_N$, $\hat{M}_N$ also has all coefficients corresponding to the entries $\{\hat{x}_{i} \cdot \hat{y}_{j} \mid (i + j) = 6\}$ in $\hat{x}_{ND}$ equal to zero. Thus, it is sufficient to investigate only the entries $\{\hat{x}_{i}^{\prime} \cdot \hat{y}_{j}^{\prime} \mid (i + j) = 7\}$ in $\hat{x}_{ND}$ on the right side of the equation above and enforce that all entries $\{x_{i}^{\prime} \cdot y_{j}^{\prime} \mid (i + j) = 6\}$ vanish after substituting $\hat{x}_{ND} = B_N \cdot x_{ND}$ back into Eq. 3.52. Following the detailed analysis in Section B.3.2, linear constraints, for instance Eq. B.20, are obtained and substantially simplify the remaining process of solving unknown parameters.

In the case of significant high-order correction terms related to $k_2$ and $k_3$, the recovering of the original distortion-free image coordinates is feasible, just as the same results provided in Eq. 3.46. Nevertheless, the sign of $\hat{p}_2$ is determined by $\hat{a}$: $\hat{p}_2 = -p_2$ if $\hat{a} = -a$; otherwise $\hat{p}_2 = p_2$. In the case of $k_2$ and $k_3$ equal or, more practically, close to zero, things turn out differently: the parameters $t_p$ and $t_q$ are not or only weakly constrained since $k_2$ and $k_3$ are insignificant. Due to the inaccuracy in determined point correspondences across multiple views, the equalities in Eq. B.20 do not exactly hold and there are residual terms on the right side of these equations. As a severe consequence, uncontrolled drift of these parameters from their real values is thus possible in the parameter estimation process and can lead to dramatically decreased correction performance. A possible solution is to dismiss the parameters $p_1$ and $p_2$ related to the decentering components. By this means, the correction model is simplified into the basic form only with the isotropic radial component and the image correction process is thus stabilized. Furthermore, the deviation from the original full-parameter model can be well compensated, as all other parameters, especially $a$ and $s$ in $K$, can be accordingly adapted through an appropriate optimization.
3.2.2.5 Division model

Through a similar analysis as for PCM, equations on unknown parameters in the case of DCM can be obtained by substituting Eq.3.35 into Eq.3.39:

\[
\begin{bmatrix}
  h_1^T \\
  h_2^T \\
  h_3^T
\end{bmatrix}
\begin{bmatrix}
  \frac{x_{\text{ND}}}{y_{\text{ND}}} \\
  1 + \kappa \cdot r^2
\end{bmatrix}
\cdot (1 + \hat{\kappa} \cdot \hat{r}^2) =
\begin{bmatrix}
  1 + \kappa \cdot r^2
\end{bmatrix}
\begin{bmatrix}
  \frac{\hat{x}_{\text{ND}}}{\hat{y}_{\text{ND}}}
\end{bmatrix}
\]

\[
\begin{bmatrix}
  \frac{x_{\text{ND}}}{y_{\text{ND}}} \\
  1 + \kappa \cdot r^2
\end{bmatrix}
\begin{bmatrix}
  \frac{\hat{x}_{\text{ND}}}{\hat{y}_{\text{ND}}}
\end{bmatrix}
\]

\[
\Rightarrow
\begin{bmatrix}
  \frac{x_{\text{ND}}}{y_{\text{ND}}} \\
  1 + \kappa \cdot r^2
\end{bmatrix}
\begin{bmatrix}
  \frac{x_{\text{ND}}}{y_{\text{ND}}} \\
  1 + \kappa \cdot r^2
\end{bmatrix}
\cdot (1 + \hat{\kappa} \cdot \hat{r}^2) =
\begin{bmatrix}
  \frac{\hat{x}_{\text{ND}}}{\hat{y}_{\text{ND}}}
\end{bmatrix}
\]

\[
(3.53)
\]

\[
\begin{bmatrix}
M_{\text{N,L}} \cdot \chi_{\text{ND}}
\end{bmatrix}
= \begin{bmatrix}
M_{\text{N,R}} \cdot \chi_{\text{ND}}
\end{bmatrix}
\Rightarrow
M_{\text{N,L}} = M_{\text{N,R}},
\]

where \( r^2 = x_{\text{ND}}^2 + y_{\text{ND}}^2 \) and \( \hat{r}^2 = \hat{x}_{\text{ND}}^2 + \hat{y}_{\text{ND}}^2 \). Using the linear coordinate transformation \([\hat{x}_{\text{ND}}, \hat{y}_{\text{ND}}, 1]^T = \mathcal{T} \cdot [x_{\text{ND}}, y_{\text{ND}}, 1]^T\), both sides of the equation above are only depending on the originally normalized coordinates \( x_{\text{ND}} \) with lens distortion. \( M_{\text{N,L}} \) and \( M_{\text{N,R}} \) are the coefficient matrices for the entries in the lifted coordinate vector \( \chi_{\text{ND}} \). After explicitly solving each unknown parameter in Section B.4, the same results as in Eq. 3.46 are obtained again and the original distortion-free image coordinates are exactly recovered.

3.2.2.6 Rational model

As discussed in the last part of Section 3.2.2.2, the estimation of the matrix \( \mathbf{K} \) is in general infeasible for RCM. Thus, the ambiguity between the actual and the estimated correction model differs from the expression in Eq. 3.39 and obeys

\[
\mathbf{H} \cdot \mathbf{M}_1 \cdot \chi_{\text{ID}} \simeq \hat{\mathbf{M}}_1 \cdot \chi_{\text{ID}},
\]

(3.54)

where \( \hat{\mathbf{M}}_1 \) is the estimated rational model for image correction. Although the actual RCM is unique and defined by \( \mathbf{M}_1 \), the estimated RCM also depends on an arbitrary homography \( \mathbf{H} \) on the left side of Eq. 3.54. As the consequence of the ambiguity in \( \hat{\mathbf{M}}_1 \), the recovery of the original distortion-free image coordinates relying on \( \hat{x}_i \simeq \mathbf{M}_1 \cdot \chi_{\text{ID}} \) is thus denied.
3.2. Geometric calibration

3.2.2.7 Parameter estimation

So far, the relationship between the actual model \( L^{-1}N(\cdot) \) and the estimated model \( \hat{L}^{-1}N(\cdot) \) for image correction has been investigated in detail. However, how to conduct the desired parameter estimation still remains an open question. To figure out a practical algorithm for this purpose, it is necessary to first come back to the fundamental epipolar constraint. Generally, the equation

\[
(\hat{L}^{-1}N(\hat{K}^{-1}\cdot \hat{x}_{ID,i}))^T \hat{F} \hat{L}^{-1}N(\hat{K}^{-1}\cdot \hat{x}_{ID,j}) = \hat{x}_{N,j}^T \hat{F} \hat{x}_{N,i} = 0
\]  

(3.55)

derived from 3.37 and 3.38 implies the only constraint on \( \hat{K} \) and \( \hat{L}^{-1}N(\cdot) \), given the point correspondence \( \hat{x}_{ID,i} \leftrightarrow \hat{x}_{ID,j} \) for each image pair of all \( D \) views. With the help of an appropriately defined distance function \( d(\hat{x}_{N,i}, \hat{x}_{N,j}, \hat{F}) \) and the corresponding cost function \( d^2(\hat{x}_{N,i}, \hat{x}_{N,j}, \hat{F}) \), the unknown \( \hat{F} \), \( \hat{K} \) and \( \hat{L}^{-1}N(\cdot) \) can be optimally estimated by globally minimizing the residual error related to Eq. 3.55. As Zhang demonstrated in [74], in the case of determining the fundamental matrix from given point correspondences, the symmetric epipolar distance

\[
d(\hat{x}_{N,i}, \hat{x}_{N,j}, \hat{F}) = \left( \frac{\hat{x}_{N,j}^T \hat{F} \hat{x}_{N,i}}{\hat{x}_{N,j}^T \hat{F} \hat{x}_{N,j} + \hat{x}_{N,i}^T \hat{F} \hat{x}_{N,i}} \right)^{1/2}
\]  

(3.56)

and the first-order approximation of the geometric distance

\[
d(\hat{x}_{N,i}, \hat{x}_{N,j}, \hat{F}) = \left| \frac{\hat{x}_{N,j}^T \hat{F} \hat{x}_{N,i}}{(\hat{x}_{N,j}^T \hat{F} \hat{x}_{N,j} + \hat{x}_{N,i}^T \hat{F} \hat{x}_{N,i})^{1/2}} \right|
\]  

(3.57)

are usually preferred for better estimation performance in comparison to the algebraic distance \( |\hat{x}_{N,j}^T \hat{F} \hat{x}_{N,i}| \).

Since the distortion-free projections \( \hat{x}_{N,i}, \hat{x}_{N,j} \) and the epipolar lines \( \hat{F} \cdot \hat{x}_{N,i}, \hat{F}^T \cdot \hat{x}_{N,j} \) depend on the estimated affine transformation matrix \( \hat{K} \) and the estimated correction model \( \hat{L}^{-1}N(\cdot) \), the distance functions defined in Eq. 3.56 and 3.57 are not suited for the parameter estimation in the context of image correction. An illustrative example for clarifying possible issues is \( a \to \infty \) for the estimated aspect ratio \( \hat{a} \). With the special form of \( \hat{F} \): \( \hat{F} = [0, 1, 0; 1, 0, 0; 0, 0, 0] \), all standard distance functions result in negligible residual error due to \( \hat{x}_{N,j}^T \hat{F} \cdot \hat{x}_{N,i} \approx 0 \) and thus cannot provide...
the desired guidance in parameter estimation. Regarding the uncertainty of $\hat{x}_{N,i}$, $\hat{x}_{N,j}$ and $\hat{F}$, an appropriate distance function should rely on some quantities independent of the unknown $\hat{K}$ and $\hat{L}^{-1}_N(\cdot)$. To this end and inspired by Fitzgibbon’s work [40], the cost function employed for the purpose of online lens correction is selected as

$$d^2(\hat{x}_{N,i}, \hat{x}_{N,j}, \hat{F}) = ||\hat{K} \cdot \hat{L}_N(\hat{x}_{N,i}) - \hat{K} \cdot \hat{L}_N(\hat{x}_{c,N,i})||_2^2 + ||\hat{K} \cdot \hat{L}_N(\hat{x}_{N,j}) - \hat{K} \cdot \hat{L}_N(\hat{x}_{c,N,j})||_2^2 = ||\hat{x}_{ID,i} - \hat{K} \cdot \hat{L}_N(\hat{x}_{c,N,i})||_2^2 + ||\hat{x}_{ID,j} - \hat{K} \cdot \hat{L}_N(\hat{x}_{c,N,j})||_2^2,$$

(3.58)

where $\hat{L}_N(\cdot)$ is the inverse function of the estimated correction model $\hat{L}^{-1}_N(\cdot)$. Moreover, $\hat{x}_{c,N,i}$ and $\hat{x}_{c,N,j}$ are the estimated distortion-free projections after the Hartley-Sturm correction [75]. For the point correspondence $\hat{x}_{N,i} \leftrightarrow \hat{x}_{N,j}$ obtained from $\hat{x}_{N,i} = \hat{L}^{-1}_N(\hat{K}^{-1} \cdot \hat{x}_{ID,i})$ and $\hat{x}_{N,j} = \hat{L}^{-1}_N(\hat{K}^{-1} \cdot \hat{x}_{ID,j})$, the optimal estimate $\hat{F}$ is determined by minimizing the cost based on the distance function Eq. 3.56 or 3.57. Considering the inaccuracy in estimated parameters and the inaccuracy in determined point correspondence, there is in general $\hat{x}_{ID,i}^T \cdot \hat{F} \cdot \hat{x}_{ID,i} \neq 0$. With the help of the Hartley-Sturm correction, the corrected projections $\hat{x}_{c,N,i}$ and $\hat{x}_{c,N,j}$ are searched on the lines orthogonal to the corresponding epipolar lines $([\hat{e}_i]_x) \cdot \hat{x}_{N,i}$ and $([\hat{e}_j]_x) \cdot \hat{x}_{N,j}$, while these lines also go through the points $\hat{x}_{N,i}$ and $\hat{x}_{N,j}$, respectively. $\hat{x}_{c,N,i}$ and $\hat{x}_{c,N,j}$ are finally determined as those points minimizing $||\hat{x}_{N,i} - \hat{x}_{c,N,i}||_2^2 + ||\hat{x}_{N,j} - \hat{x}_{c,N,j}||_2^2$ subject to the epipolar constraint $(\hat{x}_{c,N,j})^T \cdot \hat{F} \cdot \hat{x}_{c,N,i} = 0$.

For the reader’s convenience and to provide a better overview, the entire algorithm for performing the proposed online correction of lens distortion is summarized in Algorithm 1 with step-by-step instructions. As a desirable property, the resulting estimation does not require any special initialization step for $\hat{K}$ since the suggested initial estimates are in practice close to the real values.

### 3.2.3 Online estimation of camera calibration matrix

Besides lens distortion/correction parameters, camera’s intrinsic parameters may also deviate from their original values during operation, mainly
Algorithm 1 Online correction of lens distortion

**Input:** point correspondences for each image pair of all \(D\) views.

**Output:** correction function \(\mathcal{L}^{-1}_N(\cdot)\) and affine transformation matrix \(K\).

1: Move the origin to the image center and normalize points so that their mean squared distance to the origin equals 1.
2: Initialize \(\hat{K}\) with \(\hat{a} = 1, \hat{s} = \hat{p} = \hat{q} = 0\).
3: Initialize \(\mathcal{L}^{-1}_N(\cdot)\) with zero distortion or using conventional method.
4: for \(N\) iterations do
5: for each view pair do
6: Recover \(\hat{x}_{N,i}, \hat{x}_{N,j}\) from \(\tilde{x}_{ID,i}, \tilde{x}_{ID,j}\) according to Eq.3.55.
7: Estimate \(\hat{F}\) from \(\hat{x}_{N,i}, \hat{x}_{N,j}\) for each image pair.
8: Obtain \(\hat{x}^c_{N,i}, \hat{x}^c_{N,j}\) using the Hartley-Sturm correction.
9: end for
10: Update \(\mathcal{L}^{-1}_N(\cdot), \hat{K}\) to minimize Eq.3.58 for all view pairs.
11: if global cost converged then
12: terminate.
13: end if
14: end for
15: Recover \(\mathcal{L}^{-1}_N(\cdot), K\) from \(\mathcal{L}^{-1}_N(\cdot), \hat{K}\).
16: Restore \(\mathcal{L}^{-1}_N(\cdot), K\) with respect to the transformation in step one.

due to mechanical vibrations, temperature fluctuations and manual faulty operations. After recovering the original geometric information in images as described above, it is possible to estimate the camera calibration matrix \(K\), which is essential for retrieving metric information from the visual measurements. More specifically, in the case of PCB analysis, the size as well as the height of components and PCBs can be estimated given known camera’s intrinsic parameters and using either structured illumination [76,77] or structure from multiple views [46]. Instead of using special images acquired from well-designed calibration patterns, there are some calibration methods relying on general constraints on image sequences, where the corresponding calibration processes do not require any images of specific calibration objects and are thus more flexible. Such calibration is referred to as self-calibration and preferred over off-line calibration since it can be fully automated during operation. In comparison to the classic self-calibration assuming image sequences of stationary scenes with fixed
spatial structures, a novel extension is presented in this subsection, which is capable of realizing global self-calibration in case of non-stationary scenes with rigid scene objects undergoing independent motions. More specifically, based on the classic self-calibration applied on each scene object individually, the proposed extension is aiming at answering two principal questions: which constraints can be combined across motions of rigid scene objects and how should these constraints be combined? It should be noted that $K$ is assumed remaining constant over all views for the analysis presented below.

### 3.2.3.1 Non-stationary scene

Without loss of generality, a scene consisting of two rigid objects $A$ and $B$ undergoing different motions is considered throughout this subsection if not stated otherwise. An image sequence with $D$ views of this non-stationary scene is available. For simplification, each object is considered in its body frame so that it stays stationary in the coordinate system corresponding to the body frame, while a virtual camera is taking images of this object. For scene objects moving with speeds far less than the speed of light, the real image sequence can be considered as the composition of the image sequences obtained from the virtual cameras, where the motion of each virtual camera is the fusion of the object and real camera motions. For a better understanding, a stationary and a non-stationary scene consisting of three objects are visualized in Figure 3.7. Since the virtual cameras are the observations of the real camera in the body frames of all scene objects, they share the same camera calibration matrix $K$. Only the camera’s extrinsic parameters $\{R_A, t'_A\}$ and $\{R_B, t'_B\}$ differ in general from each other. For the $i$-th view of the image sequence, the image $\tilde{x}_{i,i,A}$ of the world point $\tilde{x}_{w,i,A}$ in the body frame of object $A$ and the image $\tilde{x}_{i,i,B}$ of the world point $\tilde{x}_{w,i,B}$ in the body frame of object $B$ are written as

\[
\tilde{x}_{i,i,A} \approx K \cdot [R_i, A | t'_{i,A}] \cdot \tilde{x}_{w,i,A} = P_{i,A}^E \cdot \tilde{x}_{w,i,A}^E,
\]
\[
\tilde{x}_{i,i,B} \approx K \cdot [R_i, B | t'_{i,B}] \cdot \tilde{x}_{w,i,B} = P_{i,B}^E \cdot \tilde{x}_{w,i,B}^E,
\]  

(3.59)

where $P_{i,A}^E = K \cdot [R_i, A | t'_{i,A}]$ and $P_{i,B}^E = K \cdot [R_i, B | t'_{i,B}]$ are the Euclidean camera matrices of the two virtual cameras for the $i$-th view. For clarity, the superscript $E$ denotes the coordinates and the projections defined for
3.2. Geometric calibration

Figure 3.7: Stationary and non-stationary scenes. Translation velocities \((v_t, v_{t,1}, v_{t,2} \text{ and } v_{t,3})\) are visualized as green solid arrows, while rotation velocities \((\omega_r, \omega_{r,1}, \omega_{r,2} \text{ and } \omega_{r,3})\) are visualized as angular motions about the corresponding axes defined by red solid arrows. For the stationary scene, the spatial structure defined by the three objects is rigid. For the non-stationary scene, each of the three objects undergoes independent motion and the spatial structure defined by them is variable.

Euclidean reconstructions in projective space \(\mathbb{P}^3\) and satisfying Eq. 3.15. If the two points \(\tilde{x}_{W,i,A}^E\) and \(\tilde{x}_{W,i,B}^E\) are located on the surface of the two objects, their coordinates are constant in the body frames and independent of the current view. Regarding this, their coordinates are further denoted by \(\tilde{x}_{W,A}^E\) and \(\tilde{x}_{W,B}^E\).

Given the point correspondences across multiple views and using the factorization-based algorithm from Sturm and Triggs [78], it is possible to obtain the reconstructions \(\tilde{x}_{W,A}^P\) and \(\tilde{x}_{W,B}^P\) of the points \(\tilde{x}_{W,A}^E\) and \(\tilde{x}_{W,B}^E\), respectively. However, \(\tilde{x}_{W,A}^P\) and \(\tilde{x}_{W,B}^P\) are nothing more than algebraic reconstructions without geometric constraints in projective space \(\mathbb{P}^3\). Thus, ambiguities similar to those in Eq. 3.37 also exist in general for the projectively reconstructed point \(\tilde{x}_{W}^P\) in \(\mathbb{P}^3\):

\[
\tilde{x}_1 \cong P^E \cdot \tilde{x}_W^E = P^E \cdot (H_{EP})^{-1} \cdot H_{EP}^E \cdot \tilde{x}_W^E = P^P \cdot \tilde{x}_W^P, \tag{3.60}
\]

where \((H_{EP})^{-1}\) is a 4×4 rectifying homography transforming the obtained projective reconstruction \(\tilde{x}_W^P\) into the desired Euclidean reconstruction \(\tilde{x}_W^E\) with \(\tilde{x}_W^E \cong (H_{EP})^{-1} \cdot \tilde{x}_W^P\).

Regarding the hierarchy of the geometric invariances:

**collinearity** \(\rightarrow\) **parallelism** \(\rightarrow\) **perpendicularity** \(\rightarrow\) **rigidity**
Chapter 3. Imaging system characterization

under different groups of geometric transformations, the obtained recon-
structions and the correspondingly reconstructed geometries are divided
into the following strata:

**projective ← affine ← metric ← Euclidean.**

This relationship can also be mathematically formulated as

\[ \tilde{x}_1 \cong P^E \cdot \tilde{x}_W^E = P^E \cdot (H_{EM})^{-1} \cdot H_{EM} \cdot \tilde{x}_W^E = P^M \cdot \tilde{x}_W^M = P^M \cdot (H_{MA})^{-1} \cdot H_{MA} \cdot \tilde{x}_W^M = P^A \cdot \tilde{x}_W^A = P^A \cdot (H_{AP})^{-1} \cdot H_{AP} \cdot \tilde{x}_W^A = P^P \cdot \tilde{x}_W^P, \]  

(3.61)

where \( \tilde{x}_W^E, \tilde{x}_W^M, \tilde{x}_W^A \) and \( \tilde{x}_W^P \) denote the Euclidean, metric, affine and projective reconstructions, respectively. Moreover, the hierarchy of geometric transformations are defined by the 4×4 homographies \( H_{EM}, H_{MA} \) and \( H_{AP} \) with \( \tilde{x}_W^M \cong H_{EM} \cdot \tilde{x}_W^E, \tilde{x}_W^A \cong H_{MA} \cdot \tilde{x}_W^M, \tilde{x}_W^P \cong H_{AP} \cdot \tilde{x}_W^A \) and \( H_{EP} = H_{AP} \cdot H_{MA} \cdot H_{EM} \).

As suggested in [46], the obtained projective camera matrix \( P^P \) of the first view can be transformed into the canonical form with \( P^P_1 \cong I \cdot [I | 0] \). If the world coordinates in the body frame of each object are defined to be coincident with the camera coordinates in the first frame, there are \( R_1 = I \) and \( t'_1 = 0 \), which lead to \( P^E_1 = K \cdot [I | 0] \). The inverse rectifying homography \( H_{EP} \) must thus take the following form

\[ H_{EP} = \begin{bmatrix} K & 0 \\ p^T & c \end{bmatrix} \]  

(3.62)

so that \( P^E_1 \cong P^P_1 \cdot H_{EP} \), where \( p \) is an unknown 3×1 vector and \( c \) a free scale factor. Apparently, the camera calibration matrix \( K \) defines the transformation \( H_{EA} \) between the Euclidean and affine reconstructions, while \( p \) defines the transformation between the affine and projective reconstructions. The last transformation can also be considered as the transformation of the plane \( \tilde{\pi}_\infty \) at infinity, which retains the constant form \([0, 0, 0, 1]^T\) under Euclidean, metric and affine transformations. Let \( \tilde{\pi}_\infty^p \) denote the transformed plane at infinity after the projective reconstruction associated to \( H_{EP} \), there is

\[ \tilde{\pi}_\infty^p \cong (H_{EP})^{-1} \cdot \tilde{\pi}_\infty^p \cong [- (p^T \cdot K^{-1}), 1]^T := [q^T, 1]^T, \]  

(3.63)

where \( q = - (K^{-T} \cdot p) \) since \( (\tilde{\pi}_\infty^E)^T \cdot \tilde{v}_\infty^E = (\tilde{\pi}_\infty^P)^T \cdot H_{EP} \cdot \tilde{v}_\infty^E = 0 \) for any ideal point \( \tilde{v}_\infty^E \) lying on \( \tilde{\pi}_\infty^E \).
3.2. Geometric calibration

3.2.3.2 General constraints

As stated in Section 3.2.2.1, the fundamental matrix \( F_{ij} \) between two views \( i \) and \( j \) can be written as \( F_{ij} \sim ([\tilde{e}_j]_x) \cdot K \cdot R_j \cdot R_i^\dagger \cdot K^\dagger \). It is then straightforward to derive \( F_{ij} \cdot \omega^* \cdot F_{ij}^\dagger \sim ([\tilde{e}_j]_x) \cdot \omega^* \cdot ([\tilde{e}_j]_x) \) for the skew-symmetric matrix \( ([\tilde{e}_j]_x) \), where \( \omega^* = K \cdot K^\dagger \) is the dual image of the absolute conic \( \Omega_\infty \). This is the classic form of the Kruppa equations in [79,80] and induces an implicit constraint on \( K \). In [81], a more applicable form of the Kruppa equations was introduced, which relies on the singular value decomposition of the fundamental matrix \( F_{ij} \) of rank two:

\[
F_{ij} = \begin{bmatrix}
| & | & |
| & & |
| & & |
\end{bmatrix} \begin{bmatrix}
\sigma_1 \\
\sigma_2 \\
0
\end{bmatrix} \begin{bmatrix}
\begin{bmatrix}
\begin{bmatrix}
u_1^T \\
u_2^T \\
u_3^T
\end{bmatrix}
\end{bmatrix} \begin{bmatrix}
\begin{bmatrix}
\begin{bmatrix}
\begin{bmatrix}
V_T \\
V_T \\
V_T
\end{bmatrix}
\end{bmatrix}
\end{bmatrix}
\end{bmatrix} = (3.64)
\]

The last column vector \( u_3 \) of the unitary (here real orthogonal) matrix \( U \) spans the left null space of \( F_{ij} \) on the one hand with \( u_3^T \cdot F_{ij} = 0^T \); on the other hand, the epipole \( \tilde{e}_j \) is also the left null vector of \( F_{ij} \) with \( \tilde{e}_j^T \cdot F_{ij} = 0^T \). This yields the central equation \( \tilde{e}_j \sim u_3 \). Applying this equation to the constraint in the classic form, an explicit form of the Kruppa equations is obtained as

\[
\frac{u_2^T \cdot \omega^* \cdot u_2}{\sigma_1^2 \cdot v_1^T \cdot \omega^* \cdot v_1} = -\frac{u_1^T \cdot \omega^* \cdot u_2}{\sigma_1 \cdot \sigma_2 \cdot v_1^T \cdot \omega^* \cdot v_2} = -\frac{u_1^T \cdot \omega^* \cdot u_1}{\sigma_2^2 \cdot v_2^T \cdot \omega^* \cdot v_2}. \quad (3.65)
\]

Eq. 3.65 holds between two arbitrary views of object \( A \) as well as of object \( B \). However, as stated by Sturm in [82], the equations above are valid for conics generated from the projection of a 3D quadric. In other words, \( \omega^* \) as the projection of the degenerate quadric \( \Omega_\infty \) is not the unique solution for Eq. 3.65, which are in fact only weak constraints. This has been confirmed by Pollefeys and Van Gool with unsatisfactory calibration results using the Kruppa equations in [83].

Instead of the weakly constrained quadric-induced Kruppa equations, it is also possible to consider the constraint on the Euclidean and projective camera matrices \( P_E \) and \( P_P \). For the \( i \)-th view of the two objects \( A \) and \( B \), there are

\[
P_{i,A}^E = K \cdot [R_{i,A} | t_{i,A}'] \simeq P_{i,A}^P \cdot H_{A}^{EP} := [A_{i,A} | a_{i,A}] \cdot H_{A}^{EP},
\]

\[
P_{i,B}^E = K \cdot [R_{i,B} | t_{i,B}'] \simeq P_{i,B}^P \cdot H_{B}^{EP} := [A_{i,B} | a_{i,B}] \cdot H_{B}^{EP}. \quad (3.66)
\]
from Eq. 3.59 and 3.61, where \( H^E_A \) and \( H^E_B \) are the individual inverse rectifying homographies for the reconstructions of object \( A \) and \( B \), respectively. Regarding the special property of a rotation matrix \( R \) with \( R \cdot R^T = I \), it is reasonable to take a closer observation of the first three column vectors of the Euclidean camera matrix \( P^E_i \) in Eq. 3.66 with \( H^E \) substituted by its explicit form in Eq. 3.62:

\[
K \cdot R_i,A \simeq (A_i,A \cdot K + a_i,A \cdot p_A^T), \\
K \cdot R_i,B \simeq (A_i,B \cdot K + a_i,B \cdot p_B^T) \\
\implies \omega^* = K \cdot K^T \simeq (A_i,A - a_i,A \cdot q_A^T) \cdot \omega^* \cdot (A_i,A - a_i,A \cdot q_A^T)^T \\
\simeq (A_i,B - a_i,B \cdot q_B^T) \cdot \omega^* \cdot (A_i,B - a_i,B \cdot q_B^T)^T,
\]

where \( q_A = -(K^{-T} \cdot p_A) \) and \( q_B = -(K^{-T} \cdot p_B) \). This equation on \( \omega^* \), \( q_A \) and \( q_B \) is the extension to the absolute conic-induced constraint [46] for the case of non-stationary scenes.

In the last equation, the camera’s intrinsic parameters and the plane at infinity are encoded in \( \omega^* \) and \( q \) separately. To encode them in a more concise fashion, the absolute dual quadric \( \Omega^* \):

\[
\Omega^* = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix}
\]

was introduced in [84]. As a dual entity, \( \Omega^* \) is contravariant under a change-of-basis transformation \( \tilde{x}_W = H^E \cdot \tilde{x}_W^E \). More precisely, the transformed entity \( Q^* \) is written as

\[
Q^* := H^E \cdot \Omega^* \cdot (H^E)^T = \begin{bmatrix}
\omega^* & -\omega^* \cdot q \\
-q^T \cdot \omega^* & q^T \cdot \omega^* \cdot q
\end{bmatrix}.
\]

\( Q^* \) is of rank three with the null space spanned by \( \tilde{\pi}_\infty^p \) since \( (\tilde{\pi}_\infty^p)^T \cdot Q^* = Q^* \cdot \tilde{\pi}_\infty^p = 0 \). If its projection \( P^P \cdot Q^* \cdot (P^P)^T \) in images is considered for the \( i \)-th view of object \( A \) and \( B \), there are

\[
P^P_{i,A} \cdot Q^*_A \cdot (P^P_{i,A})^T = P^P_{i,A} \cdot H^E_A \cdot \Omega^* \cdot (H^E_A)^T \cdot (P^P_{i,A})^T \simeq \omega^*, \\
P^P_{i,B} \cdot Q^*_B \cdot (P^P_{i,B})^T = P^P_{i,B} \cdot H^E_B \cdot \Omega^* \cdot (H^E_B)^T \cdot (P^P_{i,B})^T \simeq \omega^*.
\]

So far, all derived equations require simultaneously solving unknown parameters of \( \omega^* \) and \( q \). To remove this limitation for a stratified parameter estimation with reduced complexity, the modulus (absolute value)
constraint-based methods are considered. The first practical modulus constraint for recovering the plane \( \tilde{\pi}_E \) at infinity and thereby achieving an affine reconstruction was introduced by Pollefeys in [83]. As demonstrated in Eq.3.16, a planar homography \( \mathcal{H}_E \) can be determined and defines the projection of ideal points lying on the plane \( \tilde{\pi}_E \) onto the view of a camera with the parameters \( \{ K, R, t \} \). For two views \( i \) and \( j \) of a camera with the parameters \( \{ K, R_i, t'_i \} \) and \( \{ K, R_i, t'_i \} \), respectively, a planar homography \( \mathcal{H}_{i,\infty}^{i,j,E} \) with

\[
\mathcal{H}_{i,\infty}^{j,E} \cong K \cdot R_j \cdot R_i^{-1} \cdot K^{-1}
\]  

(3.71)

is induced by the plane \( \tilde{\pi}_E \) and defines the transformation of the projected ideal points between these two views. Similarly, another planar homography \( \mathcal{H}_{i,\infty}^{j,i,P} \) can be derived via the plane \( \tilde{\pi}_P = [q^T, 1]^T \):

\[
P^P \cdot \tilde{x}_{w,\infty}^P \cong \tilde{x}_{i,j,\infty} \cong \mathcal{H}_{i,\infty}^{j,i,P} \cdot \tilde{x}_{i,i,\infty} \cong \mathcal{H}_{i,\infty}^{j,i,P} \cdot P^P \cdot \tilde{x}_{w,\infty}^P
\]

\[
\implies \mathcal{H}_{i,\infty}^{j,i,P} \cong (A_j - a_j \cdot q^T) \cdot (A_i - a_i \cdot q^T)^{-1}
\]

(3.72)

for any point \( \tilde{x}_{w,\infty}^P \) on the plane \( \tilde{\pi}_P \) and obeying \( (\tilde{\pi}_P)^T \cdot \tilde{x}_{w,\infty}^P = 0 \).

Since \( \mathcal{H}_{i,\infty}^{j,E} \) and \( \mathcal{H}_{i,\infty}^{j,i,P} \) define the same planar homography up to a non-zero scale factor between two views of points on the plane at infinity, there is

\[
K \cdot R_j \cdot R_i^{-1} \cdot K^{-1} \cong \mathcal{H}_{i,\infty}^{j,E} \cong \mathcal{H}_{i,\infty}^{j,i,P} \cong (A_j - a_j \cdot q^T) \cdot (A_i - a_i \cdot q^T)^{-1}.
\]

(3.73)

The 3×3 matrix \( \mathcal{H}_{i,\infty}^{j,E} \) is similar to the rotation matrix \( R_j \cdot R_i^{-1} \) up to a non-zero scale factor, whose eigenvalues are with the same modulus of one. Therefore, the eigenvalues of the matrix \( (A_j - a_j \cdot q^T) \cdot (A_i - a_i \cdot q^T)^{-1} \) must also have the same modulus, which may differ from one. Let \( \lambda \) generally denote eigenvalues, the characteristic polynomial of the above matrix is written as \( \det((A_j - a_j \cdot q^T) \cdot (A_i - a_i \cdot q^T)^{-1} - \lambda \cdot I) \) and is equal to zero. This equation can be interpreted as the determinant of the matrix \( ((A_j - a_j \cdot q^T) \cdot (A_i - a_i \cdot q^T)^{-1} \) equal to zero. As a result, the determinant of the matrix \( ((A_j - a_j \cdot q^T) \cdot (A_i - a_i \cdot q^T)^{-1} - \lambda \cdot (A_i - a_i \cdot q^T)) \) also equals zero. The last determinant can be expanded as \( \det((A_j - a_j \cdot q^T) - \lambda \cdot (A_i - a_i \cdot q^T)) = l_{3,ij} \cdot \lambda^3 + l_{2,ij} \cdot \lambda^2 + l_{1,ij} \cdot \lambda + l_{0,ij} \), where \( l_{3,ij}, l_{2,ij}, l_{1,ij}, \) and \( l_{0,ij} \) are, as shown in [83], linear polynomials of the entries in the vector \( q \). Under the constraint that all complex eigenvalues, if they exist at all, are pairwise
conjugate for the real matrix \( ((A_j - a_j \cdot q^T) - \lambda (A_i - a_i \cdot q^T)) \), a modulus constraint is obtained for each view pair and results in two equations for object \( A \) and \( B \):

\[
\begin{align*}
    l_{3,ij,A} & = l_{1,ij,A}^3 l_{0,ij,A} \\
    l_{3,ij,B} & = l_{1,ij,B}^3 l_{0,ij,B}.
\end{align*}
\] (3.74)

Once the planes \( \tilde{\pi}_{\infty,A} = [q_A^T, 1]^T \) and \( \tilde{\pi}_{\infty,B} = [q_B^T, 1]^T \) are determined, the dual image \( \omega^* \) of the absolute conic can be solved using the equations in Eq. 3.67 or 3.70.

Instead of investigating the homographies \( \mathcal{H}_{i,\infty}^{j,E} \) and \( \mathcal{H}_{i,\infty}^{j,P} \) induced via the plane \( \tilde{\pi}_{\infty} \), Ronda et al. [85] exploited the properties of horopter curves for deriving the constraint on \( \tilde{\pi}_{\infty} \). For arbitrary two views \( i \) and \( j \), there are a series of 3D points, denoted by \( \tilde{x}_{W,h_i} \), giving rise to the same projections in images: \( P_i^{E} \tilde{x}_{W,h_i} \simeq \tilde{x}_{E,i} = \tilde{x}_{1,h_i,j} \simeq P_j^{E} \tilde{x}_{W,h_i} \). The set of such points is defined as the horopter curve of the view pair \( i \) and \( j \). For the projective reconstruction \( \tilde{x}_{W,h} \) of a point \( \tilde{x}_{E,h_i} \) on the horopter curve, there is

\[
    P_i^{E} \tilde{x}_{W,h_i} \simeq P_i^{P} \tilde{x}_{W,h} = \theta_{ij} \cdot P_j^{P} \tilde{x}_{W,h} \simeq P_j^{E} \tilde{x}_{W,h},
\] (3.75)

where \( \theta_{ij} \) is a scale factor. The equation above can be rewritten as

\[
    (P_i^{P} - \theta_{ij} \cdot P_j^{P}) \cdot \tilde{x}_{W,h} := \begin{bmatrix} \tilde{\alpha}_i^T - \theta_{ij} \cdot \tilde{\alpha}_j^T \\ \tilde{\beta}_i^T - \theta_{ij} \cdot \tilde{\beta}_j^T \\ \tilde{\gamma}_i^T - \theta_{ij} \cdot \tilde{\gamma}_j^T \end{bmatrix} \cdot \tilde{x}_{W,h} := \begin{bmatrix} (\tilde{\pi}_{\alpha}^P)^T \\ (\tilde{\pi}_{\beta}^P)^T \\ (\tilde{\pi}_{\gamma}^P)^T \end{bmatrix} \cdot \tilde{x}_{W,h} = 0
\] (3.76)

with \( P_i^{P} := [\tilde{\alpha}_i \cdot \tilde{\beta}_i \cdot \tilde{\gamma}_i]^T \) and \( P_j^{P} := [\tilde{\alpha}_j \cdot \tilde{\beta}_j \cdot \tilde{\gamma}_j]^T \). The three homogeneous vectors \( (\tilde{\alpha}_i - \theta_{ij} \cdot \tilde{\alpha}_j), (\tilde{\beta}_i - \theta_{ij} \cdot \tilde{\beta}_j), (\tilde{\gamma}_i - \theta_{ij} \cdot \tilde{\gamma}_j) \) define three generally non-identical planes \( \tilde{\pi}_{\alpha}^P, \tilde{\pi}_{\beta}^P \) and \( \tilde{\pi}_{\gamma}^P \) touching the common point \( \tilde{x}_{W,h} \), respectively. According to [86], any plane \( \tilde{\pi}_{S}^P \) of the star through \( \tilde{x}_{W,h} \) is a linear combination of the three planes and therefore obeys

\[
    \det([\tilde{\pi}_{\alpha}^P | \tilde{\pi}_{\beta}^P | \tilde{\pi}_{\gamma}^P | \tilde{\pi}_{S}^P]) = (\tilde{x}_{W,h})^T \cdot \tilde{\pi}_{S}^P = 0.
\] (3.77)

Apparently, the coordinates of the point \( \tilde{x}_{W,h} \) are proportional to the coefficients of the four entries in \( \tilde{\pi}_{S}^P \) in the characteristic polynomial of the matrix \( [\tilde{\pi}_{\alpha}^P | \tilde{\pi}_{\beta}^P | \tilde{\pi}_{\gamma}^P | \tilde{\pi}_{S}^P] \). Let \( p_1(\theta_{ij}), p_2(\theta_{ij}), p_3(\theta_{ij}) \) and \( p_4(\theta_{ij}) \) denote
the corresponding cubic coefficient polynomials in the single variable $\theta_{ij}$, then
\[ \tilde{x}_{W,h}^E \cong [p_1(\theta_{ij}), p_2(\theta_{ij}), p_3(\theta_{ij}), p_4(\theta_{ij})]^T. \] (3.78)

For simplicity, the projective horopter curve between view $i$ and $j$ is denoted hereafter by $h_{ij}^P(\theta_{ij})$.

Similar to Eq. 3.76, a parametrized equation on $\tilde{x}_{W,h}^E$ is obtained as
\begin{align*}
(P_i^E - \eta \cdot \theta_{ij} \cdot P_j^E) \cdot \tilde{x}_{W,h}^E & = K \cdot [(R_i - \eta \cdot R_j) \cdot (t'_i - \eta \cdot t'_j)] \cdot \tilde{x}_{W,h}^E \\
& = 0,
\end{align*}
(3.79)

where $\eta$ is a non-zero scale factor arising from the scale ambiguity of homogeneous coordinates. For an intersection point $\tilde{x}_{W,h,\infty}^E = [(\tilde{x}_{W,h,\infty}^E)^T, 0]^T$ of the horopter curve with the plane at infinity, the equation above turns into
\begin{align*}
K \cdot (R_i - \eta \cdot \theta_{\infty,ij} \cdot R_j) \cdot \tilde{x}_{W,h,\infty}^E & = 0 \\
\implies (R_j^T \cdot R_i - \eta \cdot \theta_{\infty,ij} \cdot I) \cdot \tilde{x}_{W,h,\infty}^E & = 0,
\end{align*}
(3.80)

where $\tilde{x}_{W,h,\infty}^E$ and $\eta \cdot \theta_{\infty,ij}$ are the corresponding eigenvector and eigenvalue of the matrix $R_j^T \cdot R_i$, respectively. On the one hand, as a twisted cubic, the horopter curve $h_{ij}^P(\theta_{ij})$ has in general three intersection points with the plane at infinity and thus defines all eigenvectors and eigenvalues of the matrix $R_j^T \cdot R_i$; on the other hand, $R_j^T \cdot R_i$ is a rotation matrix, whose eigenvalues are with the same modulus of 1. Regarding these facts, modulus constraints on $\tilde{\pi}_{\infty,A}^P$ and $\tilde{\pi}_{\infty,B}^P$ for the projective reconstructions of object $A$ and $B$ are obtained as
\begin{align*}
|\theta_{\infty,ij,A,1}| & = |\theta_{\infty,ij,A,2}| = |\theta_{\infty,ij,A,3}| \\
|\theta_{\infty,ij,B,1}| & = |\theta_{\infty,ij,B,2}| = |\theta_{\infty,ij,B,3}|,
\end{align*}
(3.81)

where $\theta_{\infty,ij,A,1}$, $\theta_{\infty,ij,A,2}$, $\theta_{\infty,ij,A,3}$ and $\theta_{\infty,ij,B,1}$, $\theta_{\infty,ij,B,2}$, $\theta_{\infty,ij,B,3}$ define the intersection points of the projective horopter curves $h_{ij}^P(\theta_{ij})$ and $h_{ij}^P(\theta_{ij})$ with the planes $\tilde{\pi}_{\infty,A}^P$ and $\tilde{\pi}_{\infty,B}^P$, respectively. A special case occurs if the rotation angle $\varphi$ arising from $R_j^T \cdot R_i$ takes the value 0 or $\pi$. In this case, the three intersection points between $h_{ij}^P(\theta_{ij})$ and $\tilde{\pi}_{\infty}^P$ are non-unique. As a result, planes other than $\tilde{\pi}_{\infty}^P$ could be found and also satisfy Eq. 3.80 as well as Eq. 3.81 for three intersection points.
Chapter 3. Imaging system characterization

3.2.3.3 Approaches with linear initialization

After the analysis conducted above, constraints in diverse forms on the unknown parameters of $\omega^*$ and $\bar{\pi}_c^\infty$ for each of $M$ non-stationary scene objects are provided. However, a general problem in applying the obtained constraints is the existence of ambiguous solutions, and practical approaches employing these constraints for solving the sought parameters are still missing. For Eq. 3.67 and 3.70 relying on the absolute conic and the absolute dual quadric, quadratic instead of linear equations in entries of $\omega^*$ and $\Omega^*$ are obtained due to the equality up to an unknown scale factor between the left and the right side of the constraint equations. In the worst case, a total of $2^{(5+3\cdot M)}$ solutions for the $(5+3\cdot M)$ unknown parameters need to be considered, where the first five parameters are embedded in $\omega^*$ and the remaining $3\cdot M$ parameters are required for the affine reconstructions of the $M$ non-stationary objects in a scene. For the modulus constraint in Eq. 3.74, even after the plausibility investigation introduced in [87], there are still 21 possible solutions for recovering the affine reconstruction of a non-stationary scene object.

To deal with this problem, a common solution is to initialize the unknown parameters with some linear approximations, which is followed by a nonlinear optimization for refining the estimated values in an iterative fashion. As demonstrated in [88], a practical initialization can be achieved using the orthogonality constraint of the pixel elements, which results in linear equations of the unknown parameters. Recalling the definition of the camera calibration matrix $K$ in Eq. 3.3 and the definition of the dual image $\omega^*$ of the absolute conic, three linear equations for each view are obtained from Eq. 3.70:

\[
K = \begin{bmatrix} f & 0 & 0 \\ 0 & a \cdot f & 0 \\ 0 & 0 & 1 \end{bmatrix} \implies \omega^* = K \cdot K^T = \begin{bmatrix} \omega^*_{11} & 0 & 0 \\ 0 & \omega^*_{22} & 0 \\ 0 & 0 & \omega^*_{33} \end{bmatrix}
\]

\[
\bar{\alpha}^T \cdot Q^* \cdot \bar{\beta} = 0, \\
\bar{\alpha}^T \cdot Q^* \cdot \bar{\gamma} = 0, \\
\bar{\beta}^T \cdot Q^* \cdot \bar{\gamma} = 0
\]

for $P^p := [\bar{\alpha} | \bar{\beta} | \bar{\gamma}]^T$ under the assumptions $s = 0$ and $px_1 = py_1 = 0$. $s \neq 0$ in the case of orthogonal image axes is a reasonable approximation for the most commercial cameras and this constraint is even enforced if
3.2. Geometric calibration

the online lens correction in Section 3.2.2 has been applied in advance. Although the second assumption \( p x_i = p y_i = 0 \) does not hold in general, but starting with the initial estimates \( \hat{p}x_i = (\text{image width})/2 \) and \( \hat{p}y_i = (\text{image height})/2 \), which is followed by an iterative refinement, each camera matrix \( P^p \) is transformed into \( P^p \) with respect to the currently estimated principal point \( [\hat{p}x_i, \hat{p}y_i, 1]^T \):

\[
\begin{bmatrix}
1 & 0 & -\hat{p}x_i \\
0 & 1 & -\hat{p}y_i \\
0 & 0 & 1
\end{bmatrix} \cdot P^p \cdot \tilde{x}^p_w := P^p \cdot \tilde{x}^p_w
\]

so that the transformed homogeneous image coordinates \( \tilde{x}_i \) have the transformed principal point \( [(p x_i - \hat{p}x_i), (p y_i - \hat{p}y_i), 1]^T \) approximately at the origin \( [0, 0, 1]^T \) since \( (p x_i - \hat{p}x_i) \approx (p y_i - \hat{p}y_i) \approx 0 \). The symmetric 4×4 matrix \( Q^* \) has ten distinct entries, where one of them can be determined by setting it to a fixed value, for instance \( \omega^*_{33} = 1 \), to remove the scale ambiguity denoted by \( \approx \) due to homogeneous coordinates. A further constraint on the entries is \( \det (Q^*) = 0 \), which can, however, not be enforced using any linear equation. Thus, for a non-stationary scene with \( M \) objects undergoing independent motions, there must be \( D \) views obeying \( 3 \cdot M \cdot D \geq (5 + 4 \cdot M) \) for solving Eq. 3.82 in the \( M \) unknown \( Q^* \) matrices for the \( M \) projective reconstructions. To enforce \( \det (Q^*) = 0 \), the last singular value of \( Q^* \) is set to zero to project \( Q^* \) to rank three. The corresponding plane \( \tilde{\pi}^p_\infty \) at the infinity is obtained as the last right singular vector of \( Q^* \) since \( Q^* \cdot \tilde{\pi}^p_\infty \). To update the current estimation results, the residual error of the linear equations is considered as the consequence of the inaccurately estimated principal point, which is correspondingly corrected after each iteration. The iterative refinement of the estimates is repeated until the last singular values of all \( Q^* \) matrices are lower than a predefined threshold. In contrast to [88], a re-estimation of the projective reconstructions and camera matrices is no more required if the update of \( P^p \) is conducted according to Eq. 3.83 and using the current estimate \( [\hat{p}x_i, \hat{p}y_i, 1]^T \) of the principal point. A considerable drawback of this lin-
ear initialization is the more significant restriction on $M$ for satisfying
$3 \cdot M \cdot D \geq (5 + 4 \cdot M)$, while $5 \cdot M \cdot (D − 1) \geq (5 + 3 \cdot M)$ is the necessary re-
requirement for achieving direct solutions of the quadratic equations arising
from Eq. 3.67 and 3.70.

Starting with the obtained linear initialization, a nonlinear optimization
for refining the estimated values is conducted with respect to the con-
straints derived in Section 3.2.3.2. In fact, the constraint in Eq. 3.70
relying on the absolute dual quadric is only a more concise form of the
constraint in Eq. 3.67 relying on the absolute conic, which can be verified
by expanding the expressions in Eq. 3.70. Moreover, as proven in [85], the
homographies induced by the plane at infinity and horopter curves result
in equivalent constraints on the unknown parameters of $\tilde{\pi}_\infty^P$. Thus, to
avoid duplicate calibration, only the constraints induced by the absolute
dual quadric and by horopter curves are considered further.

An appropriate cost function defining the target to be minimized is indis-
pensable for achieving the desired optimization results. In the case of the
similarity ($\simeq$) constraint between two matrices $\omega^*$ and $P^P \cdot Q^* \cdot (P^P)^T$
in Eq. 3.70, the squared difference between the two sides of the equations af-
after a Frobenius norm-based normalization [83] is computed and the overall
squared difference is employed for guiding the optimization process:

$$
\hat{\Theta} = \arg\min_{\Theta} \sum_{m=1}^{M} \sum_{i=1}^{D} \frac{\|\omega^* - \frac{P_{i,m} \cdot Q^* \cdot (P_{i,m})^T}{\|P_{i,m} \cdot Q^* \cdot (P_{i,m})^T\|_F}\|_F^2}{2},
$$

(3.84)

where $\Theta$ denotes the set of all unknown parameters and the finally esti-
imated values in $\hat{\Theta}$ lead to the minimum cost over $D$ views of $M$
objects. The estimated camera calibration matrix $\hat{K}$ is obtained using the
Cholesky factorization [46] of $\hat{\omega}^*$ as $\omega^*$ is a positive definite symmetric
real matrix. If the provided point correspondences are with inadequate
accuracy, the positive-definiteness condition of $\hat{\omega}^*$ could be violated and
may result in unsuccessful calibration.

In the case of the unimodular constraint on the three intersection points
between the horopter curve $h^p_{ij}(\theta_{ij})$ and the plane $\tilde{\pi}_\infty^P$ at infinity
in Eq. 3.81, the optimization of $\tilde{\pi}_\infty^P$ for the projective reconstruction of each
object is defined as

$$
\tilde{\pi}_\infty^P, m = \arg\min_{\tilde{\pi}_\infty^P, m} \sum_{i=1}^{D} \sum_{j=1}^{D} \sum_{k=1}^{3} \sum_{l=1}^{3} (\frac{\theta_{\infty,ij,m,k}}{\theta_{\infty,ij,m,l}} - 1)^2,
$$

(3.85)
3.2. Geometric calibration

where $i \neq j$ and $k \neq l$. It should be emphasized that the employed cost function is symmetric since each pair of the three intersection points and each pair of the $D$ views are considered twice for covering all possible combinations. To update the $\theta_{ij}$ values with respect to the currently estimated $\hat{\Pi}_s^\infty$, a cubic equation in $\theta_{ij}$ arising from Eq. 3.76 and 3.77 with $\hat{\Pi}_s^p = \hat{\Pi}_s^\infty$ needs to be solved. This is achievable using the companion matrix eigenvalue-based approach [89]. If the estimated plane $\hat{\Pi}_s^\infty = [\hat{q}^\top, 1]^\top$ is substituted into Eq. 3.67, quadratic equations in entries of $\omega^*$ are obtained due to the similarity between the matrices on the two sides of the equations. As suggested in [83], the scale ambiguity between the two sides is removed if $\det (A - a \cdot \hat{q}^\top) = 1$ is enforced and the equations in entries of $\omega^*$ become therefore linear, which can be solved by minimizing the overall squared error.

Similar to the last step of the offline calibration, a bundle adjustment-based refinement according to Eq. 3.27 is also applicable for the estimated parameters. Assuming white Gaussian noise of the provided point correspondences, the ML estimate is achieved by minimizing the overall squared geometric distance between the observed and the reprojected image points.

3.2.3.4 Motion segmentation

A significant challenge in the online calibration for non-stationary scenes is to track point correspondences over all views of rigid objects undergoing independent motions. In other words, the movements of the feature points in images should be segmented into different sections corresponding to the objects, which is referred to as motion segmentation in the literature. With help of three consecutive images from a video, this motion segmentation issue emerging in PCB images is visualized in Figure 3.8. In consideration of the motion of PCBs and the motion of the camera over the conveyor belt, a practical combination of the Kanade–Lucas–Tomasi (KLT) tracking algorithm [90,91,92], the distance measure based on Linear Combination of Views (LCV) [93,94] and the spectral clustering algorithm [95,96] is employed for separating the motion of PCBs from the global motion arising from the displacement of the camera between consecutive image acquisitions. Alternative solutions using the agglomerative lossy compression [97] as well as the local subspace affinity [98] are not
Chapter 3. Imaging system characterization

Figure 3.8: Multiple motions in PCB images. In the three consecutive images from a video, the global vertical motion (top to down) caused by the displacement of the camera over the conveyor belt and the motion (left to right) of the PCB are illustrated.

preferred due to their high computational complexity and the resulting extremely long processing time observed in tests.

Under the assumptions of constant pixel values between two views and constant movement in a local window $\mathcal{W}$, as illustrated in Figure 3.9, the KLT algorithm results in a feature tracking minimizing the image dissimilarity $\epsilon(x_i)$ in $\mathcal{W}(x_i)$ centered at $x_i$:

$$\epsilon(x_i) = \int \int_{\mathcal{W}(x_i)} (I_j(x_{i,\mathcal{W}} + v_i) - I_i(x_{i,\mathcal{W}}))^2 \cdot w(x_{i,\mathcal{W}} - x_i) \cdot dx_{i,\mathcal{W}}, \quad (3.86)$$

where $w(x_{i,\mathcal{W}} - x_i)$ is a weighting function and $x_{i,\mathcal{W}} \in \mathcal{W}(x_i)$. $I_i$ and $I_j$ denote the images of the $i$-th and the $j$-th view, respectively. Moreover, the centers of the local windows in the two images are $x_i$ and $(x_i + d_i)$. The displacement $v_i$ of any point $x_{i,\mathcal{W}}$ in the local window $\mathcal{W}(x_i)$ is modeled as an affine motion with $v_i = A_i \cdot (x_{i,\mathcal{W}} - x_i) + d_i$, where $A_i$ is a $2 \times 2$ transformation matrix. Consequently, there is $I_i(x_{i,\mathcal{W}}) = I_j(x_{i,\mathcal{W}} + v_i)$.

The minimum dissimilarity $\epsilon(x_i)$ is determined by partially differentiating Eq. 3.86 with respect to the unknown parameters in $A_i$ and $d_i$, and setting the results to zero. The obtained equation is simplified if the first-order Taylor approximation $I_j(x_{i,\mathcal{W}} + v_i) \approx I_j(x_{i,\mathcal{W}}) + (g_j(x_{i,\mathcal{W}}))^T \cdot v_i$ with
3.2. Geometric calibration

Figure 3.9: Affine motion field assumed for KLT tracking. \( x_i \) and \((x_i + d_i)\) are the corresponding feature positions in the two views \( i \) and \( j \). A point \( x_{1,W} \) in the local window \( \mathcal{W}(x_i) \) of \( x_i \) undergoes an affine transformation and is located at the position \( A_i \cdot (x_{1,W} - x_i) + (x_i + d_i) + x_i \) in the transformed local window \( T_i(\mathcal{W}(x_i)) \).

\[
g_j(x_{1,W}) = \frac{\partial I_j(x_{1,W})}{\partial x_{1,W}} \text{ is adopted for a small motion } d_i \text{ between two views:}
\]

\[
\int \int_{\mathcal{W}(x_i)} (I_j(x_{1,W}) - I_i(x_{1,W})) \cdot G_j(x_{1,W}, v_i) \cdot w(x_{1,W} - x_i) \cdot dx_{1,W} \
\approx \int \int_{\mathcal{W}(x_i)} G_j(x_{1,W}, v_i) \cdot (g_j(x_{1,W}))^T \cdot v_i \cdot w(x_{1,W} - x_i) \cdot dx_{1,W},
\]

(3.87)

where \( G_j(x_{1,W}, v_i) \) denotes the partial derivative matrix of \( I_j(x_{1,W} + v_i) \) at the point \( x_{1,W} \) with respect to \( A_i \) and \( d_i \). This equation can be solved in an iterative fashion and using the initial values \( \hat{A}_1 = I \) and \( \hat{d}_1 = 0 \). In each iteration, \( I_j(x_{1,W}) \) is replaced by \( I_j(x_{1,W} + \hat{v}_1) \) with the currently estimated pixel displacement \( \hat{v}_1 \). For achieving reliable tracking performance, image features with significant local variations in multiple orientations are preferred. Such feature points can be determined by considering the structure tensor \( S(I_j(x_i)) = \int \int g_j(x_{1,W}) \cdot (g_j(x_{1,W}))^T \cdot w(x_{1,W} - x_i) \cdot dx_{1,W} \) for the local window \( \mathcal{W}(x_i) \). Let \( \lambda_{i,1} \) and \( \lambda_{i,2} \) denote the two eigenvalues of \( S(I_j(x_i)) \), \( \min(\lambda_{i,1}, \lambda_{i,2}) > \lambda_{i,th} \) indicates an appropriate feature for
Figure 3.10: Tracked feature trajectories of multiple objects. Features are tracked across three views and the resulting trajectories arising from independent motions of the two objects are visualized in two different colors (green and red).

tracking, where $\lambda_{t,th}$ is a predefined threshold significant over the maximal noise level in images.

After tracking features on objects across multiple views, as illustrated in Figure 3.10, trajectories comprising sampled feature positions in images are obtained. To assign these trajectories to the corresponding objects, a similarity-based clustering could be a reasonable solution, where each cluster of trajectories represents the motions of feature points located on the same object. For a homogeneous point matrix $\tilde{X}_W = [\tilde{x}_{W,1}, \tilde{x}_{W,2}, \cdots, \tilde{x}_{W,N}] \in \mathbb{R}^{4 \times N}$ with its $n$-th column representing a homogeneous 3D world point $\tilde{x}_{W,n}$, the homogeneous image matrix $\tilde{X}_I = [\tilde{x}_{I,1}, \tilde{x}_{I,2}, \cdots, \tilde{x}_{I,N}] \in \mathbb{R}^{3 \times N}$ arising from a general camera matrix $P \in \mathbb{R}^{3 \times 4}$ obeys $\tilde{X}_I \simeq P \cdot \tilde{X}_W$ with its $n$-th column representing the homogeneous image $\tilde{x}_{I,n}$ of $\tilde{x}_{W,n}$. If an affine camera with the last row vector of $P$ equal to $[0, 0, 0, 1]$ is explicitly considered, the projection equation above is simplified and the non-homogeneous image matrix $X_I = [\tilde{x}_{I,1}, \tilde{x}_{I,2}, \cdots, \tilde{x}_{I,N}] \in \mathbb{R}^{2 \times N}$ is expressed as $X_I = P_A \cdot \tilde{X}_W$ with a $2 \times 4$ affine projection matrix $P_A$. This equation can be extended by vertically stacking the affine pro-
3.2. Geometric calibration

Projection matrices of \( D \) views into a joint projection matrix \( \mathbf{J}_p \):

\[
\begin{bmatrix}
x_{1,11} & x_{1,12} & \cdots & x_{1,1N} \\
x_{1,21} & x_{1,22} & \cdots & x_{1,2N} \\
\vdots & \vdots & \ddots & \vdots \\
x_{1,D1} & x_{1,D2} & \cdots & x_{1,DN}
\end{bmatrix}
= \begin{bmatrix}
X_{1,1} \\
X_{1,2} \\
\vdots \\
X_{1,D}
\end{bmatrix} = \begin{bmatrix}
P_{A,1} \\
P_{A,2} \\
\vdots \\
P_{A,D}
\end{bmatrix} \cdot \tilde{\mathbf{x}}_w,
\]

where \( \mathbf{X}_1 \) is extended into a matrix with vertically stacked non-homogeneous image matrices \( \mathbf{X}_{1,1}, \mathbf{X}_{1,2}, \cdots, \mathbf{X}_{1,D} \) of 3D points \( \tilde{\mathbf{x}}_{w,1}, \tilde{\mathbf{x}}_{w,2}, \cdots, \tilde{\mathbf{x}}_{w,N} \) from \( D \) views.

Since the non-homogeneous image matrix \( \mathbf{X}_1 \in \mathbb{R}^{(2\cdot D) \times N} \) is the product of the matrices \( \mathbf{J}_p \in \mathbb{R}^{(2\cdot D) \times 4} \) and \( \tilde{\mathbf{x}}_w \in \mathbb{R}^{4 \times N} \), the rank of \( \mathbf{X}_1 \) must obey

\[
\text{rank}(\mathbf{X}_1) \leq \min (\text{rank}(\mathbf{J}_p), \text{rank}(\tilde{\mathbf{x}}_w)) \leq \min (2\cdot D, 4, N).
\]

For adequate views and feature points in general, the maximal rank of \( \mathbf{X}_1 \) is bounded by four. Therefore, \( \mathbf{X}_1 \) with images \( [\mathbf{X}_{1,i}^\top, \mathbf{X}_{1,j}^\top]^\top \) of \( \tilde{\mathbf{x}}_w \) obtained from two distinct views \( i \) and \( j \) is sufficient for defining the subspace spanned by its row vectors in \( \mathbb{R}^{1 \times 4} \), where the distinctness between the two views should lead to \( \det ([\mathbf{P}_{A,i}^\top, \mathbf{P}_{A,j}^\top]^\top) \neq 0 \). Consequently, as stated in [94], the recovery of the 3D structure \( \tilde{\mathbf{x}}_w \) is quite straightforward:

\[
\tilde{\mathbf{x}}_w = \left( \begin{bmatrix} \mathbf{P}_{A,i} \\ \mathbf{P}_{A,j} \end{bmatrix} \right)^{-1} \cdot \begin{bmatrix} \mathbf{X}_{1,i} \\ \mathbf{X}_{1,j} \end{bmatrix}.
\]

Given the affine projection matrix \( \mathbf{P}_{A,k} \) of the third view \( k \), the projection \( \mathbf{X}_{1,k} \) of \( \tilde{\mathbf{x}}_w \) in this view can be synthesized using

\[
\mathbf{X}_{1,k} = \mathbf{Q}_k^\top \cdot \begin{bmatrix} \mathbf{1}^\top \\ \mathbf{X}_1 \end{bmatrix} = \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} & q_{15} \\
q_{21} & q_{22} & q_{23} & q_{24} & q_{25} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{1}^\top \\ \mathbf{X}_{1,i} \\ \mathbf{X}_{1,j} \end{bmatrix},
\]

where the matrix \( \mathbf{Q}_k \) defines the image \( \mathbf{X}_{1,k} \) of \( \tilde{\mathbf{x}}_w \) as a linear combination of images in \( \mathbf{X}_1 \). To also manage general cases of rigid transformations, \( \mathbf{X}_1 \) is extended with an additional row vector \( \mathbf{1}^\top \) [93], which leads to

\[
\mathbf{X}_{1,k} = \mathbf{Q}_k \cdot \begin{bmatrix} \mathbf{1}^\top \\ \mathbf{X}_1 \end{bmatrix} = \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} & q_{15} \\
q_{21} & q_{22} & q_{23} & q_{24} & q_{25} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{1}^\top \\ \mathbf{X}_{1,i} \\ \mathbf{X}_{1,j} \end{bmatrix}.
\]
To solve \( Q_k \), at least five point correspondences across the three views \( i, j \) and \( k \) are required. It should be noted that multiple solutions of \( Q_k \) exist in the case of degenerate motions, \textit{e.g.} pure translations, in-plane rotations, \textit{etc.}, since \( \text{rank}([1, X_i^T]) < 5 \).

Apparently, if \( X_i \) is known for some points on a rigid object and \( Q \) is known for all \( D \) views, the image trajectory \( t_i^t \) of an arbitrary point also located on this object can be synthesized according to Eq. 3.92 and by using its images in two distinct views with respect to the motion of the object. In contrast to [94] approximating the affinity matrix \( \mathcal{A} \) with the pairwise similarity between feature motions using \( \mathcal{A} \approx \mathcal{E} \cdot \mathcal{E}^T \) up to a scale factor, where \( \mathcal{E} \) denotes the matrix with the pairwise similarity between real and synthetic image trajectories, an affinity matrix \( \mathcal{A} \) with the pairwise similarity between distance vectors of trajectories is employed for exploiting the benefits of the self-tuning spectral clustering [96] without manual parameter tuning. For real and synthetic image trajectories \( t_i^t \) and \( \hat{t}_i^t \), the normalized distance value \( d^t \) is defined as \( d^t = \| t_i^t - \hat{t}_i^t \|_2/D \). For \( C \) groups of pre-selected image trajectories in \( C \) spatially neighboring regions, the distance value \( d^t \) is calculated between each image trajectory to be segmented and the generated synthetic trajectories using the \( C \) trajectory groups, which gives rise to the distance vector \( d^t \in \mathbb{R}^C \). A comparison between a tracked real image feature trajectory and two synthetic trajectories are visualized in Figure 3.11. Instead of the conventional similarity measure \( a_{nm} \) between two distance vectors \( d_n^t \) and \( d_n^t \) with \( a_{nm} = e^{-\|d_n^t - d_m^t\|^2/(2 \cdot \sigma^2)} \) [95], an adaptive similarity measure is used for computing \( \mathcal{A} \), where the entry \( a_{nm} \) of the \( n \)-th row and the \( m \)-th column is obtained as \( a_{nm} = e^{-\|d_n^t - d_m^t\|^2/(\sigma_n \cdot \sigma_m)} \) using the local scale factors \( \sigma_n \) and \( \sigma_m \). The local scale factor of the distance vector \( d^t \) is automatically determined through an analysis of the statistics in the neighborhood of \( d^t \). Subsequently, based on the eigenvector analysis of the normalized affinity matrix for a compact representation in the canonical form, the number of clusters in image trajectories is estimated in an iterative fashion, which reveals the number of objects undergoing independent motions. Moreover, the tracked image trajectories are also correspondingly clustered. The overall procedure of the presented motion segmentation algorithm is summarized in Algorithm 2.

Two assumptions have been made in the analysis of the motion segmentation so far. The first assumption is the applicability of the affine camera
3.2. Geometric calibration

Figure 3.11: Comparison between real and synthetic image feature trajectories. Real and synthetic image feature positions in all views are represented by dots in different colors.

matrix for each view. According to [46], the projection error induced by the approximation of a projective camera with an affine camera is proportional to the difference between the point depth $\rho$ of the considered 3D point and the average depth of feature points on the object. For PCBs and the background scene consisting of nearly planar structures, the resulting projection error is insignificant and thus negligible. The second assumption is the approximation of the affinity between tracked image feature trajectories with the similarity between the corresponding distance vectors. This is less accurate than the approximation $\mathbf{A} \approx \mathbf{E} \cdot \mathbf{E}^T$ employed in [94] since the latter affinity approximation is based on pairwise similarity, while the approximation employed in this thesis is based on pairwise dissimilarity. Nevertheless, if adequate synthetic trajectories are provided for building the distance vector $\mathbf{d}^t$, the affinity is well constrained by the distance values to the synthetic trajectories densely distributed in the subspace of possible image feature trajectories.

3.2.3.5 Degenerate structure and motion

For any online camera calibration employing the approaches described in Section 3.2.3, general structures and motions are required. Unfortunately, in the case of PCB images, these prerequisites are not necessarily satisfied,
Algorithm 2 Motion segmentation

Input: $D$ views of $M$ rigid objects undergoing independent motions.
Output: Segmented motions of $M$ rigid objects.

1: Track reliable features in images using the KLT tracker [92].
2: Randomly select $C$ distinct seed trajectories.
3: Build $C$ groups of local trajectories (between 5 and 7 for each group) arising in the spatial neighborhood of the $C$ seed trajectories.
4: for each of $C$ trajectory groups do
5: for each of $(D - 2)$ views: from 2 to $(D - 1)$ do
6: Solve $Q$ according to Eq. 3.92.
7: end for
8: end for
9: for each tracked image feature trajectory do
10: for each of $C$ trajectory groups do
11: Generate synthetic trajectory according to Eq. 3.92.
12: Compute distance between real and synthetic trajectories.
13: end for
14: Build distance vector $d^t$ using computed distance values.
15: end for
16: Determine local scale factors for all distance vectors.
17: Build affinity matrix $A$.
18: Segment all image feature trajectories using the affinity matrix $A$ and the self-tuning spectral clustering [96].

where degenerate structure and motion may also be involved. A special geometric characteristic of PCBs is the nearly planar structure in comparison to the object distance between the PCB being observed and the lens center. A typical configuration of the imaging system with some characteristic parameters is presented in Figure 3.12. The vertical distance between the PCB and the camera with a 16mm lens is 340mm, while the maximum height of the mounted PCB components does not exceed 35mm and is thus insignificant. Moreover, as illustrated in Figure 3.13, most features detected in images are located on PCBs, which therefore also obey the nearly coplanar constraint. In consideration of inaccuracy in measurements, this degenerate structure gives rise to difficulties in determining the fundamental matrices between view pairs. Let $\tilde{\pi}^e$ denote the
3.2. Geometric calibration

plane with the minimum distance to all feature points, there is a planar homography \( H \) approximately defining the transformation of points lying on \( \tilde{\pi}^E \) between two views \( i \) and \( j \). For a feature point close to the plane \( \tilde{\pi}^E \) and resulting in the image \( \tilde{x}_{1,i} \) in view \( i \), the corresponding image \( \tilde{x}_{1,j} \) in view \( j \) is approximated through

\[
\tilde{x}_{1,j} \approx \eta \cdot H \cdot \tilde{x}_{1,i}.
\]

The epipolar constraint in Eq. 3.28 is rewritten as

\[
\tilde{x}_{1,j}^T \cdot F_{ij} \cdot \tilde{x}_{1,i} \approx \eta \cdot \tilde{x}_{1,i}^T \cdot H^T \cdot F_{ij} \cdot \tilde{x}_{1,i} \approx 0.
\]

To enforce Eq. 3.93 for all feature points, it is sufficient to have the 3×3 matrix \( H^T \cdot F_{ij} \) being skew-symmetric, which, however, only provides six linear equations in the entries of \( F_{ij} \). As a result, the fundamental matrix \( F_{ij} \) cannot be solved without ambiguities even if the rank constraint \( \det(F_{ij}) = 0 \) is also considered. If an \( F_{ij} \) is still obtained from point correspondences, it is in general far from the actual fundamental matrix since inaccuracy in measurements substantially violates the planar constraint and has significant impact on the estimation results. Moreover, as depicted in Figure 3.12, coplanar features are rather a rough approximation and a planar homography \( H \) between two views \( i \) and \( j \) is therefore not a satisfactory solution either.

In addition to the degenerate structure, degenerate motion is also presented in image sequences of PCBs, which does not support the online
calibration methods introduced so far. Being transported on the conveyor belt, PCBs exhibit in general pure translations between consecutive image acquisitions. To assert if the observed motion is critical for conducting online camera calibration, the existence of those potential absolute conics must be investigated [99], whose images are different from $\omega$ and still remain constant across all views. A most straightforward analysis is to consider virtual conics on the plane $\pi_\infty^E$ at infinity, which are uniformly defined by a $3 \times 3$ diagonal matrix $C'$ but with different matrix entries. All points $x_\infty^E$ on $C'$ obey $(x_\infty^E)^T C' x_\infty^E = 0$ and none of them is real. As demonstrated in Eq. 3.17, the projection of $C'$ through a Euclidean camera matrix $P^E = [R | t']$ is $K^{-T} \cdot R \cdot C' \cdot R^T \cdot K^{-1}$. For pure translations with a constant rotation matrix $R$, the projections of all virtual conics on $x_\infty^E$ remains constant. This reveals the difficulty in conducting online calibration from projective reconstructions using PCB images.

Apart from point correspondences, geometric primitives can also provide necessary constraints for supporting online camera calibration even if only single images of scenes are available. This idea can be proven by reviewing Section 3.2.1.4, where vanishing points obtained from parallel lines result in effective constraints on $\omega$ and therefore also on $K$ as $\omega = K^{-T} \cdot K^{-1}$. Very frequently, man-made objects exhibit orthogonal structures predominantly defined by line segments. In case of two or more mutually orthog-
3.2. Geometric calibration

(a) PCB image

(b) detected geometric primitives

Figure 3.14: Detected geometric primitives for online camera calibration. Horizontal and vertical line segments detected in the original PCB image (left) are visualized in blue and red in the grayscale image (right), respectively. Green circles/ellipses are used to indicate the projections of circular contours detected with high confidence.

Figure 3.14: Detected geometric primitives for online camera calibration. Horizontal and vertical line segments detected in the original PCB image (left) are visualized in blue and red in the grayscale image (right), respectively. Green circles/ellipses are used to indicate the projections of circular contours detected with high confidence.

To estimate the camera calibration matrix $K$ under plausible assumptions [100, 101], e.g. $s = 0$, $a = 1$, $px_1 = \frac{\text{image width}}{2}$, $py_1 = \frac{\text{image height}}{2}$. A demonstrative online calibration using orthogonal line segments arising from buildings and lanes in city views has been presented in [101]. In PCB images, similar to the case of city views, components with special geometric structures are also available. As illustrated in Figure 3.14, significant line segments with two mutually orthogonal directions typically arising from slots, ICs and cooling systems, as well as circles typically arising from electrolytic capacitors and batteries can be detected in most PCB images. For achieving reliable calibration, only those entities detected with high confidence should be used, which is, in the case of PCBs, an appropriate restriction since adequate geometric primitives from densely mounted components are in general available in images.

For two groups of parallel line segments, the projections $\tilde{v}_{1,1}$ and $\tilde{v}_{1,2}$ of their vanishing points in 3D are determined as the intersection points of the fitted line models in the single view image. If the angular difference between the two directions of parallel lines in $\mathbb{E}^3$ is equal to $\pi/2$, Eq. 3.19
is consequently simplified into
\[(\tilde{v}_{1,1})^T \cdot \omega \cdot \tilde{v}_{1,2} = (\tilde{v}_{1,1})^T \cdot K^{-T} \cdot K^{-1} \cdot \tilde{v}_{1,2} = 0. \quad (3.94)\]

Besides the orthogonality constraint induced by line segments, an additional constraint is also available by regarding the projections of circular structures, e.g. batteries, electrolytic capacitors, screws, etc. Ideally, the image plane is parallel to the PCB plane and the shape of projections arising from circular structures only depends on \(K\). In practice, an arbitrary rotation between these two planes could exist and leads to an additional deformation of projections.

With an appropriate definition of the PCB and camera coordinate systems in Figure 3.15 (a), the \(z\)-axes of the two coordinate systems become collinear and their origins only differ in a vertical displacement orthogonal to the PCB plane. In order to align the \(x\)- and \(y\)-axes of the two coordinate systems into the same directions, axes \(x_C\) and \(y_C\) need to undergo an in-plane rotation about the axis \(z_C\) by the angle \(-\psi_1\), where "-" indicates the negative rotation direction (counter-clockwise). The projection of an electrolytic capacitor remains circular given the aspect ratio \(a = 1\) and becomes elliptical if \(a \neq 1\). With an appropriate definition of the PCB coordinate system in Figure 3.15 (b), the off-plane rotation axis of the camera coordinate system becomes parallel to \(y_W\). After a rotation by the angle \(-\psi_2\), an elliptical projection of the cylindrical electrolytic capacitor is obtained. An arbitrary rotation between the PCB and camera coordinate systems can be realized through an in-plane-off-plane-in-plane rotation sequence with the rotation angles \(-\psi_1\), \(-\psi_2\) and \(\psi_1\). If the off-line rotation angle \(\psi_2\) is close to zero, the projection of a circular object in the camera coordinate system becomes an ellipse (an enclosed conic section of a cone), where any degeneration (parabola, hyperbola, line or point) is excluded.

The conic projection of any circle considering in-plane and off-plane rotations in sequence can be modeled with three consecutive transformations defined by homographies \(H_{\text{in}}, H_{\text{off}}\) and \(H'_{\text{in}}\). Let \(C_{\text{PCB}}\) denote the ideal projection of an arbitrary circle located in a plane parallel to the PCB ground plane and thus also parallel to the ideal image plane, any point
3.2. Geometric calibration

$\mathbf{x}_{\text{C,PCB}}$ lying on $\mathbf{C}_{\text{PCB}}$ obeys

\[
0 = \mathbf{x}_{\text{C,PCB}}^T \mathbf{C}_{\text{PCB}} \mathbf{x}_{\text{C,PCB}} = \mathbf{x}_{\text{C,PCB}}^T \mathbf{H}_{\text{conic}}^T \mathbf{H}_{\text{conic}} \mathbf{C}_{\text{PCB}} \mathbf{H}_{\text{conic}}^{-1} \mathbf{H}_{\text{conic}} \mathbf{x}_{\text{C,PCB}},
\]

where $\mathbf{H}_{\text{conic}} \cong \mathbf{H}_{\text{in}}' \mathbf{H}_{\text{off}} \mathbf{H}_{\text{in}}$. Apparently, the conic projection of the circle $\mathbf{C}_{\text{PCB}}$ into the image plane is defined by $\mathbf{H}_{\text{conic}}^T \mathbf{C}_{\text{PCB}} \mathbf{H}_{\text{conic}}^{-1}$. For a known $\mathbf{K}$, the projection of $\mathbf{x}_{\text{C,PCB}}$ defined in the camera coordinate system can be recovered from Eq. 3.2 up to an unknown scale factor $1/\rho$. As a result, $\mathbf{K}$ can be canceled from $\mathbf{H}_{\text{in}}, \mathbf{H}_{\text{off}}$ and $\mathbf{H}_{\text{in}}'$. By placing the origin of the PCB coordinate system at the center of the camera, the three homographies are further simplified into $\mathbf{H}_{\text{in}} \cong \mathbf{R}_{\text{in}}, \mathbf{H}_{\text{off}} \cong \mathbf{R}_{\text{off}}$ and $\mathbf{H}_{\text{in}}' \cong \mathbf{R}_{\text{in}}^T$, where $\mathbf{R}_{\text{in}}$ and $\mathbf{R}_{\text{off}}$ are the in-plane and off-plane rotation matrices, respectively. For any detected conic projection of $\mathbf{C}_{\text{PCB}}$ in PCB images, $\mathbf{C}_{\text{PCB}}$ can be recovered if $\psi_1$ and $\psi_2$ are given.

Combining the orthogonality constraint in Eq. 3.94 and the eccentricity constraint on circles, an objective function is defined to guide an iterative optimization process for determining unknown parameters:

\[
\hat{\mathbf{K}} = \arg \min_{\omega, \psi_1, \psi_2} \frac{((\hat{\mathbf{v}}_{i,1})^T \omega \cdot \hat{\mathbf{v}}_{i,2})^2}{(\hat{\mathbf{v}}_{i,1})^T \omega \cdot \hat{\mathbf{v}}_{i,1} (\hat{\mathbf{v}}_{i,2})^T \omega \cdot \hat{\mathbf{v}}_{i,2}} + \frac{1}{L} \sum_{i=1}^{L} \left( \frac{\hat{r}_{i,a}^2 - \hat{r}_{i,b}^2}{\hat{r}_{i,a}^2 + \hat{r}_{i,b}^2} \right),
\]

Figure 3.15: Rotations between the PCB (denoted by W) and camera (denoted by C) coordinate systems.
where \( \hat{r}_{i,a} \) and \( \hat{r}_{i,b} \) are the major and minor axes of the conic recovered from the \( i \)-th detected ellipse, respectively. \( L \) denotes the total number of detected ellipses. Starting with an initial estimate of \( K \), rotation angles \( \psi_1 \) and \( \psi_2 \) are derived using the parallelism constraint between parallel line segments and the orthogonality constraint between two orthogonal directions recovered from vanishing points. The optimal estimations of \( \psi_1, \psi_2 \) and \( K \) with \( K^T K^{-1} = \omega \) are obtained if the overall residual error according to Eq. 3.96 is minimized.

### 3.3 Transverse CA (TCA) correction

Although achromats and apochromats can eliminate CA for certain wavelengths, CA may still be observed in images as the applied illumination often comprises significant color fractions other than the target colors of such complex lens systems. As illustrated in Figure 3.4, in comparison to axial chromatic aberration (ACA) resulting in uniform color fringes on the boundaries between regions of high contrast, transverse chromatic aberration (TCA) is more significant due to the asymmetric color distortion on the opposite sides of such boundaries, where the pattern of distorted colors depends on the position of the observed boundary in images. To tolerate the variation in illumination and to improve the quality of the intended information retrieval in PCB images, especially for text recognition and texture/boundary analysis, additional correction of TCA is required, while the artifacts arising from ACA are omitted due to their insignificance and the deconvolution-induced high computational complexity for refocusing images from different channels.

#### 3.3.1 Global TCA correction

To deal with TCA under global constraints, an offline approach based on calibration patterns has been developed and published in [3]. Benefiting from the well-defined control points in known patterns and the constraints derived from the imaging system model, more reliable performance can be achieved as compared to approaches relying on unsupervised feature extraction or fitting local correction models. A further favorable characteristic of the introduced correction is the straightforward integration into
standard calibration pipelines, which thus enables a more comprehensive camera calibration.

Instead of starting with some specific models [37, 102, 103], a generic imaging model for each wavelength (practically for each channel in a color image) defines the basis for conducting the desired TCA correction. Without explicit notation of the distortion center \( o_{ND} \) in Eq. 3.32, a more general expression for recovering distorted images is written as

\[
\hat{x}_1 = K \cdot x_N \cong K \cdot L_N^{-1}(K^{-1} \cdot \hat{x}_{1D}). \tag{3.97}
\]

Furthermore, without loss of generality, a simplified color image comprises only two channels: 1 and 2. Then, the objective of TCA correction is to map the pixels from the second channel to the corresponding positions in the first channel for a realigned color image without color distortion, where the first channel is referred to as the reference channel. For a given 3D point \( \hat{x}_w \), its normalized camera coordinates \( x_{N,1} \) and \( x_{N,2} \) of the two channels are obtained as

\[
x_{N,1} = (1/\rho_1) \cdot [R_1|t'_1] \cdot \hat{x}_w, \\
x_{N,2} = (1/\rho_2) \cdot [R_2|t'_2] \cdot \hat{x}_w, \tag{3.98}
\]

where \( \rho_1 = ([0, 0, 1]|R_1|t'_1) \cdot \hat{x}_w \) and \( \rho_2 = ([0, 0, 1]|R_2|t'_2) \cdot \hat{x}_w \) are the point depth values of \( \hat{x}_w \) in the camera coordinate systems of channel 1 and channel 2, respectively. Recalling \( \hat{x}_w = [x_w, y_w, z_w, 1]^T \) for homogeneous coordinates, the transformation between \( x_{N,1} \) and \( x_{N,2} \) is solved using the two equations above:

\[
x_{N,1} = (\rho_2/\rho_1) \cdot R_1 \cdot R_2^T \cdot x_{N,2} + (1/\rho_1) \cdot (t'_1 - R_1 \cdot R_2^T \cdot t'_2). \tag{3.99}
\]

This transformation can be simplified under an important parameter consideration \((1/\rho_1) \cdot (t'_1 - R_1 \cdot R_2^T \cdot t'_2) \approx 0\). For real imaging systems, this is indeed a plausible assumption since the translation between different channels is negligible with respect to the point depth: \( ||t'_1 - R_1 \cdot R_2^T \cdot t'_2||_2 \ll |\rho_1| \) with \( ||t'_1 - R_1 \cdot R_2^T \cdot t'_2||_2 \ll 1 \) and \( |\rho_1| \gg 1 \). Consequently, the normalized camera coordinates \( x_{N,1} \) and \( x_{N,2} \) are associated with each other through a planar homography \( H_{2 \rightarrow 1} = (\rho_2/\rho_1) \cdot R_1 \cdot R_2^T \) and obey \( x_{N,1} \times (H_{2 \rightarrow 1} \cdot x_{N,2}) \approx 0 \). The projection \( \hat{x}_{1,2} \) of the 3D point \( \hat{x}_w \) in the second channel is mapped to the image point \( \hat{x}_{1,1} \) in the first channel according to

\[
\hat{x}_{1,1} \approx K_1 \cdot H_{2 \rightarrow 1} \cdot L_{N,2}^{-1}(K_2^{-1} \cdot \hat{x}_{1,2}). \tag{3.100}
\]
If the distortion function $L_{N,2}(\cdot)$ is estimated in calibration instead of the correction function $L_{N,2}^{-1}(\cdot)$, the inverse form of the equation above:

$$\tilde{x}_{1,2} \approx K_2 \cdot L_{N,2}(H_{2\rightarrow1} \cdot K_1^{-1} \cdot \tilde{x}_{1,1})$$

(3.101)

is more convenient for mapping the pixels from channel 2 into channel 1, and the pixel value of $\tilde{x}_{1,1}$ is obtained through an interpolation using the neighboring pixels of the image point $\tilde{x}_{1,2}$. It should be emphasized that the approximation in Eq. 3.100 and 3.101 is valid only if the ratio $\rho_2/\rho_1$ between the two depth values is provided. To overcome this difficulty, it is thus necessary to re-scale $H_{2\rightarrow1} \cdot L_{N,2}^{-1}(K_2^{-1} \cdot \tilde{x}_{1,2})$ and $L_{N,2}(H_{2\rightarrow1}^{-1} \cdot K_1^{-1} \cdot \tilde{x}_{1,1})$ to enforce the last entry to be equal to one for both vectors.

In combination with standard calibration pipelines, the estimation of $H_{2\rightarrow1}$ is conducted after the determination of $K$ and $L_N(\cdot)$ for each channel. If $H_{2\rightarrow1}$ is only constrained to be a planar homography and the similarity to the rotation matrix $R_1 \cdot R_2^T$ is omitted, the estimation is straightforward and merely requires the point correspondences of control points (as illustrated in Figure 3.2.1.1) across calibration images. The optimal estimate in the sense of least squares is solved using the direct linear transformation algorithm [104]. To also incorporate the orthonormal constraint induced by $R_1 \cdot R_2^T$, the coordinate variation of control points between the camera coordinate systems of different channels is investigated. Considering the extrinsic parameters of the two channels in Eq. 3.98 and fixing the world coordinate system on the planar calibration pattern, a control point with the stationary world coordinates $\tilde{x}_W$ gives rise to the coordinates $\tilde{x}_{C,1}$ and $\tilde{x}_{C,2}$ defined with respect to the two channels. These coordinates can be further explicitly written as

$$\tilde{x}_{C,1} = \begin{bmatrix} R_1 & t'_1 \\ 0 & 1 \end{bmatrix} \cdot \tilde{x}_W := T_{W}^{C,1} \cdot \tilde{x}_W,$$

$$\tilde{x}_{C,2} = \begin{bmatrix} R_2 & t'_2 \\ 0 & 1 \end{bmatrix} \cdot \tilde{x}_W := T_{W}^{C,2} \cdot \tilde{x}_W.$$  

(3.102)

Apparently, there exists a linear transformation defined by $T_{C,2}^{C,1} \in \mathbb{R}^{4 \times 4}$ and obeys $\tilde{x}_{C,1} = T_{C,2}^{C,1} \cdot \tilde{x}_{C,2}$, where

$$T_{C,2}^{C,1} = T_W^{C,1} \cdot (T_W^{C,2})^{-1} = \begin{bmatrix} R_1 \cdot R_2^T & (t'_1 - R_1 \cdot R_2^T \cdot t'_2) \\ 0 & 1 \end{bmatrix}.$$  

(3.103)

Given the coordinates of control points in each channel and for all calibration images, the optimal estimate of $H_{2\rightarrow1} = R_1 \cdot R_2^T$ under full constraints
3.3. Transverse CA (TCA) correction

is also determined using the direct linear transformation algorithm. In consideration of the inaccuracy in positions of control points and the resulting violation of the orthonormal constraint on vectors of the rotation matrix $R_1 \cdot R_2^T$, a fine-tuned estimate obtained as $U \cdot I \cdot V^T = U \cdot V^T$ is recommended, where $U \cdot \Sigma \cdot V^T$ is the singular value decomposition of $R_1 \cdot R_2^T$ and the diagonal matrix $\Sigma$ is replaced by the identity matrix $I$.

3.3.2 Local TCA correction

As an alternative solution to the global approach introduced above, local color correction based on the detected violations of the color difference condition for normal edges [105] can also be used to improve the quality of PCB images. Since the modeling of CA does not rely on any offline calibration, an online correction can thus be accomplished. Besides TCA, this local analysis method can further deal with ACA since the total color distortion is captured and corrected in the local regions of high contrast.

CA is usually visible in narrow transition areas between low and high pixel values, where significant edges are also presented. Without color distortion, the value difference between two image channels is governed by the values of boundary pixels of the transition areas. However, in the case of inter-channel pixel shift introduced by CA, the value difference can be dramatically increased and exceed the extreme values defined by the corresponding boundary pixels, where the color difference condition for normal edges is violated. A better understanding of this problem can be achieved by considering the example presented in Figure 3.16. On the one hand, in the homogeneous regions of the cropped PCB image, the color difference between red and blue channels is hardly observable. In transition areas with high contrast, on the other hand, unexpected high difference values are observed, which identify the existence of CA at the corresponding pixel positions. After global and local corrections, CA is partially or totally removed and the sudden variation in difference values is well suppressed. In comparison to the image after local correction, the image corrected using global constraints maintains less details. This is the consequence of the bilinear interpolation adopted for mapping pixels onto the reference channel. Although more sophisticated interpolation methods may better maintain high-frequency image contents, but they
Figure 3.16: Global and local TCA corrections. Red dashed lines in the cropped PCB images indicate the image lines, for which the profiles of red, green and blue channels are plotted. For identifying the regions with CA, the value difference between red and blue channels is also presented.

provide in general poor efficiency and can introduce undesired artifacts into images, which make them less suited for the automated analysis of PCBs.
To eliminate CA using local information, all transition areas are first localized with the help of edge detection. Then the boundary pixels are found at the positions where the spatial variation of pixel values becomes insignificant or two adjacent transition areas are detected. Any intermediate edge/transition pixels between two boundary pixels and exceeding the color difference defined by these boundary pixels are modified to obey the color difference condition again. In comparison to [105], the local CA correction approach is augmented in this thesis with an automatic selection of the reference channel for determining edges with high confidence. This is essential for achieving reliable performance on PCB images since all kinds of color transitions may occur.

3.4 Evaluation

3.4.1 Feature matching

Since detecting distinct features and matching corresponding feature points are prerequisites for the subsequent online camera calibration, it is thus of great interest to investigate the performance of widely used feature detectors and descriptors. Although the benchmark from [106, 107] and the evaluation protocol according to [108] are adopted in this thesis, an essential modification is introduced to the standard performance assessment procedure: all test images undergo additional lens distortion. This modification simulates the very initial state of online camera calibration, where corresponding feature points need to be matched before conducting the desired correction of lens distortion and estimation of the camera calibration matrix $K$.

Aiming at a comprehensive performance assessment, most state-of-the-art feature detectors and descriptors are considered in this evaluation: MSER [109], SIFT [110], CSIFT [111], SURF [112], DAISY [113], ORB [114], BRISK [115], FREAK [116], KAZE [117] and AKAZE [108]. For realizing a side-by-size comparison, the average matching score is employed and quantifies the ratio of matched features vs. overall detected distinct features. The obtained feature matching results are plotted in Figure 3.17 with respect to the isotropic radial lens distortion defined by $k_1$ in Eq. 3.6. As expected, the overall matching performance decreases with increasing
Chapter 3. Imaging system characterization

Figure 3.17: Feature matching in images with lens distortion. Curves in the first plot are obtained on a dataset with varying view point angle, which results in varying perspective transformation, while a dataset with varying scaling and rotation about the axis perpendicular to the image plane is applied for the second plot.

lens distortion. Among all ten features, KAZE, MSER, SIFT and ORB lead to more reliable matching. Regarding the expensive explicit non-linear diffusion employed for KAZE, the area-based localization of MSER and the unstable performance of ORB, SIFT turns out to be in general the best suited detector and descriptor for feature matching in the presence of lens distortion. However, if there is no hard limit on time budget, KAZE becomes preferable.

3.4.2 Online correction of lens distortion

To provide the absolute ground truth required in assessing the performance of different undistortion models, synthetic image sequences with known point correspondences are generated using a real camera model and serve as the input data fed to the image correction process. This real camera model is acquired using the standard offline calibration approach described in Section 3.2.1 and the obtained model parameters are listed in Table 3.1 for a better overview. The inaccuracy in detected feature position in practice is simulated through additive white Gaussian noise with zero mean and the standard deviation $\sigma = 0.5$ pixel, which is a reasonable value for common feature detection algorithms with subpixel accuracy. In
3.4. Evaluation

<table>
<thead>
<tr>
<th>parameter</th>
<th>absolute value</th>
<th>normalized value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f$</td>
<td>4961 pixels</td>
<td>$f/f : 1.000$</td>
</tr>
<tr>
<td>$f' = f \cdot a \cdot \sin \alpha$</td>
<td>4978 pixels</td>
<td>$f'/f' : 1.000$</td>
</tr>
<tr>
<td>$px_1$</td>
<td>1248 pixels</td>
<td>$px_1/(w/2) : 1.020$</td>
</tr>
<tr>
<td>$py_1$</td>
<td>978 pixels</td>
<td>$py_1/(h/2) : 0.954$</td>
</tr>
<tr>
<td>$\alpha = \arccot(-s)$</td>
<td>89.5°</td>
<td>$\alpha/90° : 0.994$</td>
</tr>
<tr>
<td>$k_1$</td>
<td>-0.1854</td>
<td>-</td>
</tr>
<tr>
<td>$k_2$</td>
<td>0.2356</td>
<td>-</td>
</tr>
<tr>
<td>$k_3$</td>
<td>-0.0025</td>
<td>-</td>
</tr>
<tr>
<td>$p_1$</td>
<td>0.0005</td>
<td>-</td>
</tr>
<tr>
<td>$p_2$</td>
<td>0.0000</td>
<td>-</td>
</tr>
</tbody>
</table>

| Image size $w \times h$ | 2448 × 2050 pixels |

**Table 3.1**: The reference camera model for online calibration.

order to draw statistical conclusions, *Root-Mean-Square Error* (RMSE) of the recovered image coordinates $\hat{x}_i$ with respect to the ground-truth distortion-free image coordinates $x_i$ is computed over 50 independent trials (image sequences).

Due to the single-parameter correction function, DCM is supposed to be more stable than PCM using full parameters while exhibiting less accurate correction in case of increased high-order distortion fractions. To achieve a close observation of this hypothesized effect, two lens distortion configurations are employed, where $k_2$ is switched between zero and the original value 0.2356 to omit and introduce a significant high-order distortion fraction, respectively. Moreover, the suggested modification to PCM by setting $p_1 = p_0 = 0$ is also investigated and denoted by PCM*. The obtained RMSE of different image undistortion methods is depicted in Figure 3.18. If the parameters of the employed correction models are estimated across more than two views, the proposed PCM* leads in general to the best undistortion performance, which is closely followed by the proposed PCM. In case of less high-order distortion ($k_2 = 0$), satisfactory image correction is achievable using the conventional DCM and its performance is stabilized over different numbers of views. Similar to the case of PCM, the difference between $\hat{x}_i$ and $x_i$ is substantially reduced
Figure 3.18: Online correction of lens distortion. PCM and PCM* denote polynomial correction models with and without decentering parameters $\{p_1, p_2\}$, respectively.

if the proposed DCM is employed. For significant high-order distortion ($k_2 = 0.2356$), the observation of DCM is reversed: $\hat{x}_1$ drifts away from $x_1$ and the proposed DCM leads to less accurate image correction in comparison to the conventional method. This confirms the hypothesis made about DCM: the accuracy in image correction is traded off for the stability in parameter estimation. It should be emphasized that the proposed methods are not able to provide reasonable correction results if there are only two views available for determining model parameters. This is a direct consequence of the assumption that point correspondences are arbitrarily/evenly distributed over the entire image area, which is not always the case for only two views.

Using the estimated model parameters, distortion-free images can be recovered as illustrated in Figure 3.19. Apparently, epipolar curves become epipolar lines, on which corresponding feature points are located. However, in practice, some issues have been identified. Due to the less reliable feature matching in the case of lens distortion, it is very difficult to obtain necessary point correspondences across multiple views. This becomes
3.4. Evaluation

Figure 3.19: Distorted and undistorted images. In the top images with lens distortion, epipolar lines become epipolar curves, on which corresponding feature points are located. In the bottom images, straight lines are recovered and intersect at the corresponding epipoles. All point correspondences are marked with red rectangles.

worse on PCB images as there are often similar or equivalent structures available, e.g., slots, connectors, etc. PCB images further suffer from the nearly coplanar problem, which arises from features detected on the surface of PCBs. On the one hand, if a fundamental matrix is obtained from such point correspondences, inaccuracy in measurements will in general lead to inaccurate estimation results. On the other hand, a planar homography is also merely a rough approximation of the feature position transformation between two views and cannot be considered as a satisfactory solution.
Chapter 3. Imaging system characterization

3.4.3 Online estimation of camera calibration matrix

3.4.3.1 Non-stationary scene

Similar to the evaluation of online image undistortion, online estimation of the camera calibration matrix $K$ for non-stationary scenes is also quantitatively investigated with the help of synthetic image sequences, which are generated using the same camera model from Table 3.1. The considered non-stationary scene comprises three objects (cuboid, sphere and ellipsoid) with different geometries and each object undergoes independent motion in $\mathbb{E}^3$. Synthetic images of the scene are obtained by randomly placing the camera at diverse positions around the objects. Regarding the residual distortion, the standard deviation of Gaussian noise is increased to 1.0 pixel. Moreover, 1000 independent trials are used to draw statistical conclusions since the associated estimation is more efficient than in the case of image correction due to effective linear initialization.

In Figure 3.20, a comparison between online calibration approaches is presented. The absolute dual quadric-based calibration and the horopter curve-based calibration both employ the orthogonality constraint assuming $s = 0$ for achieving a linear initialization of intrinsic parameters embedded in $K$. Only the subsequent refinements of initial estimates differ in the employed objective function: the absolute dual quadric-based calibration is governed by Eq. 3.84 and the horopter curve-based calibration by Eq. 3.85. These two calibrations are realized using either the standard or the proposed methods. In the case of standard calibration approaches, three configurations of $K$ emerge from the motions of independent objects. For the convenience of comparison, the final estimate of $K$ is obtained as the mean value of all configurations. Using the normalization defined in Table 3.1, the RMS deviation of $\hat{p}_{x_1}$, $\hat{p}_{y_1}$, $f$, $\hat{f}'$ and $\hat{\alpha}$ from their ground truth values is best visualized. In comparison to standard approaches, the proposed extension leads to an overall improvement in the accuracy of estimated parameter values. This is especially important for the absolute dual quadric-based method, which is otherwise unable to provide comparable calibration performance. Furthermore, if the horopter curve-based calibration is combined with the proposed extension, quite promising results are achievable even using only six views.
3.4. Evaluation

Figure 3.20: Online estimation of $K$ for non-stationary scenes, where 1000 synthetic image sequences are used to draw statistical conclusions. The inaccuracy in point correspondences is modeled as Gaussian noise with the standard deviation $\sigma = 1.0$ pixel.

3.4.3.2 Motion segmentation

In PCB recycling, there exists a typical non-stationary scene: PCBs are moving along with the conveyor belt and the camera is moving perpen-
Figure 3.21: Motion segmentation results. Tracked feature trajectories are assigned to corresponding motions and marked with different colors (green vs. violet).

Dorically to the conveyor belt to enlarge the field of view while preserving the desired spatial resolution. To separate foreground PCBs from moving background, the motion segmentation introduced in Section 3.2.3.4 is applied on the acquired image sequence and distinguishes between these two different motions. The obtained segmentation results are visualized in Figure 3.21. With the help of the tracked feature trajectories in different colors, two distinct motions are recognizable. Subsequently, regarding features associated with corresponding motions, images can be divided into PCB and background sections. A more precise partitioning of images can be achieved through densely tracking pixels and assigning their trajectories to the segmented motions.

3.4.3.3 Degenerate structure and motion

An important assumption made in general online calibration is the existence of general motions, which provide sufficient constraints on the image $\omega$ of the absolute conic $\Omega_\infty$ and thus on the camera calibration matrix $K$ \cite{99}. This prerequisite is, however, violated in the case of PCB recycling as PCBs usually undergo linear translations and such critical motions give rise to ambiguous solutions of $\omega$. Another difficulty in applying general online calibration on PCB images arises from the nearly planar structure of PCBs, which leads to the ambiguity identified in Eq. 3.93. To overcome these difficulties, the geometry-based online calibration introduced in Sec-
3.4. Evaluation

Figure 3.22: Geometry-based online calibration. The normalized estimation error $\Delta \hat{f}$ of the focal length $f$ and $\Delta \hat{f}'$ of the focal length $f'$ are presented. Two independent estimation processes are initialized with $\hat{f}_0 = \hat{f}'_0 = 2448$ and $\hat{f}_0 = \hat{f}'_0 = 10000$, respectively.

In total ten different PCB images are used to test the performance of the geometry-based online calibration. For each image, the normalized estimation error is depicted in Figure 3.22. Since an iterative optimization process is involved in this calibration, different initial values of $\hat{f}$ and $\hat{f}'$ lead to different final estimates. For the initial values $\hat{f}_0 = \hat{f}'_0 = 2448$ closer to the real values $f = 4961$ and $f' = 4978$ in Table 3.1, the estimation results are in general more reliable. The overall estimation error varies primarily between 0.050 and 0.200, which is sufficient for a qualitative assertion of dimensions, but inadequate for achieving an accurate dimensional measurement. As confirmed by employing the state-of-the-art calibration from [101], this is due to the nearly parallel configuration between PCBs and the image plane, where parallel line segments of components are still nearly parallel in images and their intersection points cannot be accurately determined. As a result, this state-of-the-art calibration is unable to provide reasonable results, while the proposed calibration overcomes this issue by incorporating the additional geometric constraint induced from circles.
Figure 3.23: Complementary experimental setups for assessing the performance of global TCA correction. In the first experiment, a reflector lamp combined with five narrow-band color filters is used to result in different colors of the black-white checkerboard. By this means, the variation in feature color is simulated for the case of constant illumination. In the second experiment, sunlight and LEDs are used to give rise to the variation in illumination.

3.4.4 TCA correction

3.4.4.1 Global TCA correction

To assess the performance of global TCA correction, the consistency of imaging is considered. More specifically, for a 3D feature point, independent of its color and external illumination conditions, its projection in images remains consistent if TCA is perfectly corrected and the camera’s intrinsic as well as extrinsic parameters are fixed. In case of residual TCA, variations in the feature position can be observed and higher inconsistency indicates worse correction performance. Two complementary experiments are designed to investigate the performance of global TCA correction with respect to varying feature color and to varying illumination. The corresponding experimental setups are illustrated in Figure 3.23.

In the first experiment, all model parameters are determined under the direct illumination of a reflector lamp. To realize the desired variation in color of features, the lamp is placed behind a narrow-band color filter, whose central wavelength takes one of the following values: 450, 500, 550, 600 and 650nm. For each selected central wavelength, the observed checkerboard appears in a certain color. After localizing corners across
### Table 3.2: RMS deviation from the mean feature position using different global TCA correction methods.

<table>
<thead>
<tr>
<th>color filter array</th>
<th>resolution (pixels)</th>
<th>focal length (mm)</th>
<th>TCA correction method</th>
<th>RMS deviation (pixels)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$w$</td>
<td>$h$</td>
<td></td>
<td>setup I</td>
</tr>
<tr>
<td>Bayer RGGB</td>
<td>2448</td>
<td>2050</td>
<td>16</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[37]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$\mathcal{H}$-based</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$R$-based</td>
</tr>
<tr>
<td>Bayer RGGB</td>
<td>2448</td>
<td>2050</td>
<td>25</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[37]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$\mathcal{H}$-based</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$R$-based</td>
</tr>
<tr>
<td>Bayer RGGB</td>
<td>1280</td>
<td>960</td>
<td>25</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[37]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$\mathcal{H}$-based</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$R$-based</td>
</tr>
</tbody>
</table>

In all cases, the proposed methods, especially the rotation matrix-based correction, outperform the baseline. Considering the general applicability of the proposed methods arising from the generic formulation of lens distortion in Eq. 3.97, they are preferred over the baseline, which relies on a specific polynomial distortion model. It is noticeable that the variation in illumination has greater impact on the performance of TCA correction, in
comparison to the variation in feature color. According to this result, it is important to maintain constant illumination conditions to reduce TCA in images. Among the listed combinations between cameras and lens systems, the $2448 \times 2050$ camera and the $25\text{mm}$ lens system are selected for the best trade-off between spatial resolution and image quality.

### 3.4.4.2 Local TCA correction

In principle, the local TCA correction method described in Section 3.3.2 should also be able to provide similar performance as the global correction methods. Unfortunately, as confirmed through tests on PCB images, local TCA correction results in a significant decline in image quality instead of any improvement. This is mainly due to the unreliable detection of transition areas on highly complex surface of PCBs. However, if the complexity is limited, for example in cropped text regions, local TCA correction is able to substantially remove color distortion arising from CA. Bearing this in mind, the performance evaluation of TCA correction is adapted accordingly and the resulting image enhancement is investigated through the analysis of the color transition between different image regions.

In cropped local images without CA, edge pixel values between different regions are strongly correlated with the region colors. In case of CA, this correlation is weakened since additional colors are artificially introduced through color distortion. With the help of a Principal Component Analysis (PCA) on edge pixels, the significance of the correlation is quantified: the higher the deviation along the first principal component, the stronger the correlation between the edge pixel values and the region colors, which are ideally located on the axis defined by the first principal component. This idea is illustrated in Figure 3.24 for a comparison between no correction and using the original [105], as well as the modified local correction. Apparently, the original method cannot correctly deal with thin text objects with significant color distortion and gives rise to undesired artifacts lowering image quality. On the contrary, the modified correction method introduced in this thesis recovers the correlation between edge pixels and region colors as desired.

On a dataset consisting of 430 cropped local images similar to the demo image in Figure 3.24, the average energy distribution of edge pixel values
3.4. Evaluation

Figure 3.24: Visualization of local TCA correction. Edge pixels are visualized as blue dots in the RGB color space. The principal components of edge pixels are represented by the red, green and blue lines. At the top of each RGB space representation, the energy distribution along three principal components is listed.

is computed for the purpose of performance assessment. As illustrated in Table 3.3, more energy is concentrated along the first principal component if a local TCA correction is conducted. For a better comparison, the energy along each component is also normalized with respect to the corresponding value obtained on the original images. Following the normalized energy distribution, the modified local TCA correction outperforms the original method by a wide margin.

<table>
<thead>
<tr>
<th>local TCA correction method</th>
<th>energy distribution along three components</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>absolute</td>
</tr>
<tr>
<td></td>
<td>1st</td>
</tr>
<tr>
<td>none</td>
<td>0.911</td>
</tr>
<tr>
<td>original [105]</td>
<td>0.976</td>
</tr>
<tr>
<td>modified</td>
<td>0.998</td>
</tr>
</tbody>
</table>

Table 3.3: Energy distribution of edge pixel values along three principal components after local TCA correction.
3.5 Discussion

Although the offline camera calibration approach described in Section 3.2.1 is able to precisely characterize the employed imaging system, well-designed calibration objects, skilled labor and extra interruption of the regular image acquisition are necessary prerequisites for achieving the desired calibration, which, however, lead to a strong limitation of applying this offline approach in PCB recycling. Moreover, a continuous monitoring of the variation in parameter values and an online adjustment of the image analysis with respect to the parameter variation are hardly feasible. To alleviate these barriers and to realize a more flexible imaging system characterization, approaches for achieving online camera calibration have been investigated in this chapter. It should be emphasized that the corresponding considerations were first made for the general case and then specialized for the case of PCB recycling. As a result, the general applicability of these approaches is maximized.

Unlike common online calibration approaches assuming known parameters of lens distortion or correction models, the online calibration introduced in this thesis begins with online image correction for compensating lens distortion, where the most general constraint relying on the epipolar geometry between two views and generic correction models are considered, while special geometric primitives and specific correction models are usually utilized in state-of-art methods. With the help of the new formulation of correction models presented in Section 3.2.2.2, more desirable image correction performance is in general achievable. The only requirement for achieving the improved accuracy is an adequate number of views, where four is found sufficient for all cases. Through a subsequent in-depth review of most commonly used correction models, general guidance has been provided for selecting appropriate models and their configurations for diverse applications. PCM and DCM with proper parameter configurations can lead to a unique solution, while RCM suffers from ambiguities in model parameters and is thus unsuited for recovering distortion-free image coordinates only using the epipolar constraint. If reliability is of higher priority, especially in case of an inadequate number (e.g., two) of views, DCM is preferred. If the employed lens system exhibits high-order distortion, PCM results in far more accurate image correction. For most lens systems with limited high-order isotropic radial distortion, the conducted analysis suggested a modified parameter
configuration of PCM for more stable performance, which has also been confirmed through experiments.

As revealed through the evaluation of the matching performance obtained from widely used feature detectors and descriptors, significant degradation is correlated to increased lens distortion. This fact suggests continuously tracking features in image sequences instead of merely matching them across multiple views. By this means, a wide baseline (the line joining the two camera centers) between two views is avoided and more reliable feature matching can be obtained. Since the calibration process is primarily conducted in the background computation and a frequent update of model parameters is commonly not required, the resulting high computational complexity is thus affordable. In the overall image correction process, difficulties have been especially observed in this step, due to the ambiguity arising from features extracted from similar PCB components and the dominance of nearly coplanar features. For dealing with these problems, an extension by introducing additional constraints induced by straight lines is advisable, which, however, requires a reliable detection of line segments despite numerous distractions in distorted PCB images.

As a consequence of multiple motions available in image sequences of PCBs, the assumption of stationary scenes made for state-of-the-art online calibration methods is violated and multiple solutions of the camera calibration matrix $K$ are obtained from the projective reconstructions of objects undergoing independent motions. The introduced extensions to state-of-the-art methods lead to a unique solution with higher accuracy by answering two questions: which constraints can be combined across independent motions and how should these constraints be combined? However, a motion segmentation for general non-stationary scenes is still missing. A possible solution is to employ advanced object tracking techniques, which reliably separate foreground objects undergoing independent motions from stationary or moving background.

Another difficulty in applying general online calibration in PCB recycling arises from degenerate structures and motions of PCBs, even when they can be distinguished from background through a motion segmentation based on the affine camera model. Regarding characteristics of mounted components, an alternative calibration method relying on constraints induced from geometric primitives has been developed, which requires single images instead of image sequences. This flexible method is able to provide
Chapter 3. Imaging system characterization

calibration results for obtaining qualitative dimensional measurements. An accurate quantitative measurement is so far, however, not supported. The performance of the parameter estimation is mainly governed by the quality of detected ellipses and the accuracy of located vanishing points. Aiming at improved calibration results, higher spatial resolution should be combined with the optimized imaging configuration, where an adequate off-plane rotation is achieved and vanishing points of parallel line segments become close to the image center for a more accurate localization.

The global and local methods for correcting TCA can also be combined with offline and online camera calibration processes, respectively. The global TCA correction relies on generic lens distortion model and can thus be merged into diverse calibration methods. The merging is quite straightforward as the estimated camera parameters of each channel are fed to the appended TCA correction process. On the contrary, the local TCA correction with modifications to the original method leads to substantially enhanced quality of cropped images without considering the underlying imaging system. A blind and flexible image enhancement is therefore realized. Besides the quality evaluation conducted in this chapter, it is also of great interest to investigate if TCA correction necessarily leads to any improvement in the desired information retrieval on PCBs. On the one hand, as confirmed by qualitative evaluation, TCA gives rise to slightly improved localization performance in boundary and corner detection, which is relevant for the segmentation of specific components in Chapter 4. Moreover, this pre-processing step is also desired for achieving better text analysis performance, which will be quantitatively clarified in Chapter 6 with the help of the benefit observed in reading unconstrained text. On the other hand, no significant performance improvement can be observed in localizing and categorizing general components in Chapter 5 after applying TCA correction, which is plausible since no boundary or texture information is explicitly considered in analyzing such components. To assess the performance of different combinations between lens systems and cameras, the two suggested experiments with complementary setups should be considered and used to determine the optimal imaging system with least TCA for acquiring high-quality images.
Chapter 4

Assembly print-guided segmentation of SMDs

In PCB recycling, there is in general no prior knowledge available. Indeed, there exist some special PCB entities carrying auxiliary information, which can be utilized to assist in the desired information retrieval. Among such entities, assembly print has been explicitly considered in a previous publication [4] and employed to guide the automated segmentation of Surface-Mounted Devices (SMDs), especially for those of small sizes. In addition to [4], more details concerning the segmentation algorithm and the performance evaluation are presented in this chapter. As illustrated in Figure 4.1, assembly print emerges dominantly in the form of rectangular borders and in bright colors, e.g. white and yellow. With the help of assembly print, the placement of devices as well as the inspection of the realized circuits can be substantially simplified, for both manual and automated assembly processes. In comparison to SMDs, Through-Hold Components (THCs), e.g. electrolytic capacitors, slots and cooling systems, are usually oversized with respect to the underlying assembly print, which becomes partially or completely invisible. For this reason, a similar analysis of THC turns out to be unreliable and is therefore excluded from further consideration in this chapter.
4.1 Segmentation of small devices

For SMDs of small sizes, conventional segmentation methods based on homogeneity or similarity of pixels in local regions are unable to provide satisfactory analysis results, which is the consequence of their insignificant dimensions (typically between 0.3 and 3.2mm) with respect to THCs (typically of several centimeters) and the high visual inconsistency between different body parts. Most surface-mounted resistors, for instance, comprise two conductive ends for soldering and a coated middle body part with marked resistance value. As illustrated in Figure 4.2, these different body parts of the same devices exhibit distinct visual properties. The
4.1. Segmentation of small devices

Figure 4.2: Surface-mounted resistors.

two conductive ends of each resistor are electrically connected to traces on PCBs via solder pads and give rise to the two solder joints in white. The middle part of resistors is covered by the protective coat in dark colors. Moreover, bright labels are printed on the coat to indicate the resistance values of resistors. Together with assembly print in the form of white borders, difficulties arise if segmentation approaches relying on image partitioning are employed.

In the literature, a practical solution for localizing SMDs has been introduced [30]. Using a carefully designed illumination system, indicative color patterns emerge on solder joints of devices and guide the desired localization. Limited by the specific imaging condition, this method is only applicable for the AOI purpose during manufacturing, but not for recycling, since the color patterns observed on solder joints on waste PCBs become unpredictable due to erosion and soiling. A more generic detection of solder joints using dedicated thresholding has been suggested by Mar et al. [28], which, however, also suffers from the severe difficulty of numerous false alarms triggered by a great number of background distractions similar to solder joints. For achieving the desired segmentation of SMDs on waste PCBs, assembly print instead of solder joints is considered as the necessary contextual information guiding the localization process. Regarding the existence of assembly print on popular PCBs arising from mass production, this approach is expected to be reliable and be able to provide valuable analysis results for most PCBs in recycling.

4.1.1 Border candidates

The assembly print-guided segmentation of SMDs begins with detecting rectangular bounding borders. Ideally, the bright border lines can be
Chapter 4. Assembly print-guided segmentation of SMDs

separated from dark background without great effort. Practically, this separation turns out to be difficult due to imperfections of the imaging system, e.g. defocusing, CA, motion blur, etc., and inhomogeneous illumination. To reduce the impact of the image quality and to adapt the separation with respect to the local information, an effective border detection is realized by combining the adaptive thresholding algorithm using the integral image [118, 119] with the image sharpening method using unsharp masking [120]. After improving the local contrast and compensating the slow spatial variation of illumination with image sharpening, each image pixel is binarized using an adaptive threshold computed in its local neighborhood and is assigned to foreground or background regions. By this means, all border lines along with a great number of false alarms, e.g. text, solder pads, traces, etc., are obtained.

To further localize the sought assembly print in binarized foreground pixels, the geometric constraint on rectangles is exploited. As a more robust detection approach in comparison to the classic template matching for searching candidates of given shapes, a corner-based model fitting relying on the RANdom SAmple Consensus (RANSAC) robust estimation algorithm [121] is employed to localize the borders of SMDs, which is able to deal with the varying aspect ratio of rectangles and incomplete detection of border lines. Each rectangle is defined using the model parameters \( m = [c_x, c_y, w, h]^\top \), where \( c_x \) and \( c_y \) denote the position of the centroid, while \( w \) and \( h \) denote the width and the height of the rectangle, respectively. Given the four vertices \( v_{ul} \) (upper left), \( v_{ur} \) (upper right), \( v_{ll} \) (lower left) and \( v_{lr} \) (lower right) of a rectangle \( m \), the model parameters can be determined using any two diagonal vertices: \( v_{ul} \) and \( v_{lr} \) or \( v_{ur} \) and \( v_{ll} \), according to

\[
\begin{align*}
    m &= \begin{bmatrix}
        (v_{ul} + v_{lr})/2 \\
        -v_{ul} + v_{lr}
    \end{bmatrix} = \begin{bmatrix}
        1 & 0 & 0 & 0 \\
        0 & 1 & 0 & 0 \\
        0 & 0 & -1 & 0 \\
        0 & 0 & 0 & 1
    \end{bmatrix} \begin{bmatrix}
        (v_{ur} + v_{ll})/2 \\
        -v_{ur} + v_{ll}
    \end{bmatrix}.
\end{align*}
\] (4.1)

For detecting vertices, eight structuring elements are constructed and used to localize rectangle corners in images. As visualized in Figure 4.3, two groups of structuring elements are applied in consideration of normal and rounded corners. After scanning images with these structuring elements
4.1. Segmentation of small devices

Figure 4.3: Corner detection using structuring elements. Considering the existence of two different types of corners: normal and rounded, there are in total eight structuring elements used.

individually, locations of sought corners are obtained, which in turn define the positions of sought vertices.

In the model fitting stage, rectangles are estimated from neighboring diagonal vertices. However, multiple matches of the same corners may arise from thick border lines and lead consequently to ambiguous solutions of the model parameters. To deal with this problem, the sought rectangle models are fitted in the RANSAC paradigm. Given two groups $\mathcal{V}_{ul}$ and $\mathcal{V}_{lr}$ of potential vertices corresponding to two diagonal corners, two candidates $v_{ul,i}$ and $v_{lr,j}$ are randomly picked. Using the model parameters obtained from Eq. 4.1, foreground pixels belonging to the currently fitted rectangle can be determined and are denoted by $\mathcal{P}_f$. A natural measure for assessing the quality of the current model is the cardinality of the pixel set $\mathcal{P}_f$: $\#(\mathcal{P}_f)$, which can also be interpreted as the votes for the underlying model. In order to eliminate the dependence on the size of rectangles, this measure is further normalized with respect to the perimeter. The resulting quality measure $G$ is defined as $G = \#(\mathcal{P}_f)/(2\cdot\hat{w} + 2\cdot\hat{h})$ with $0 \leq G \leq 1$, where $\hat{w}$ and $\hat{h}$ are the estimated width and height values.

By repeating this process for a sufficient number of iterations, the fitted model with the highest quality value is selected as the best fit. Since two groups of fitted models are obtained from the two configurations of diagonal corners, an additional fusion step of the obtained models is thus indispensable. For two similar models $\mathbf{m}_1$ and $\mathbf{m}_2$ arising from different configurations, their merged quality value is

$$G = [G_1 (1 - G_1)] \cdot \begin{bmatrix} w_1 & w_2 \\ w_2 & w_3 \end{bmatrix} \cdot \begin{bmatrix} G_2 \\ (1 - G_2) \end{bmatrix}, \quad (4.2)$$
where $G_1$ and $G_2$ denote the measured quality of $m_1$ and $m_2$, respectively. $w_1$, $w_2$ and $w_3$ are weighting factors with $w_1 > w_2 > w_3 \geq 0$. After this quality fusion, candidates with both low quality values are suppressed. Moreover, the final model is selected from $m_1$ and $m_2$ with the best quality value.

### 4.1.2 Rotation estimation

So far, all PCBs are assumed to be aligned with the x and y axes of images, which is in practice not necessarily the case. Instead, there is often an in-plane rotation of PCBs about the optical axis, which leads to difficulties in detecting assembly print. To avoid this issue by rotating PCB images back to the ideal orientation and restoring the expected appearance of sought borders, an effective algorithm for estimating the rotation angle in an automated manner has been developed. This estimation algorithm provides an advanced solution to the Hough transform-based [122] rotation estimation in [123] and has some extensions for achieving more robust estimation performance in presence of disturbances.

The central idea of this rotation estimation algorithm is to find the dominant direction of component contours in PCB images. With the help of the Hough transform, line directions can be estimated even using spatially isolated edge pixels and the requirement for connected contour pixels is not necessary. In the 2D histogram over the line direction and the distance to the origin, reliable line models are detected as the extreme points with the local maximum votes, where the sought model parameters are read as their coordinates. To emphasize contours without changing their geometric characteristics, the Perona-Malik diffusion [124] is adopted in pre-processing, after which significant edges of components are preserved, while cluttered structures in background regions are sufficiently suppressed. Subsequently, from the edge pixels obtained through the Canny edge detection algorithm [125], votes for each line model in the 2D histogram are accumulated since each edge pixel expressed in polar coordinates $[\theta, r]^T$ gives rise to a sinusoidal curve:

$$d = r \cdot \cos (\theta - (\phi - \pi/2)) = r \cdot \sin (\phi - \theta), \quad (4.3)$$

where $[\phi, d]^T$ defines a point in the 2D histogram with the line direction $\phi \in [-\pi/2, \pi/2)$ and the distance $d$ to the origin. It is then straightforward to determine confident line models by localizing extreme points.
4.1. Segmentation of small devices

through a Non-Maximum Suppression (NMS) [126]. Unfortunately, as illustrated in Figure 4.1, some unexpected line structures, typically conducting traces, also result in line models with high confidence, which cannot be distinguished from the sought contour line segments. In the proposed rotation estimation, this problem is tackled by analyzing line models in bundles of same directions and exploiting the perpendicular contour segments of most PCB components. Let \( v(\phi, d) \) denote the votes (or the occurrence probability after normalization) for the line model \([\phi, d]^T\), the optimal estimate \( \hat{\phi}_{\text{PCB}} \) of the PCB orientation \( \phi_{\text{PCB}} \) is obtained by solving the following maximization problem:

\[
\hat{\phi}_{\text{PCB}} = \arg \max_\phi \sum_d (v(\phi, d) - \bar{v}(\phi))^2 \cdot \sum_d (v(\phi + \frac{\pi}{2}, d) - \bar{v}(\phi + \frac{\pi}{2}))^2, \quad (4.4)
\]

where \( \bar{v}(\phi) \) is the mean votes for line models of the direction \( \phi \) and remains constant for all directions. The target function is the product of two fractions arising from perpendicular directions, which implies the orthogonality constraint on contour segments of components. For each direction, the existence of confident line models is stressed by the squared deviation from the average votes \( \bar{v}(\phi) \) and a small value is obtained if no significant lines can be found for the current direction.

From a practical point of view, the rotation estimation described above may have a potential issue due to the discrete pixel position. A representative example for demonstrating artifacts related to this quantization is presented in Figure 4.4. Line segments not aligned to \( 0^\circ, \pm 45^\circ \) and \( 90^\circ \) still consist of small fractions of such directions. As a result, \( 0^\circ, \pm 45^\circ \) and \( 90^\circ \) could become trivial solutions for Eq. 4.4 in case of dense contours. To avoid this potential issue, it is a good practice to virtually rotate PCB images over some special angles, e.g. \( 22.5^\circ, 45^\circ \), and find the optimal estimate \( \hat{\phi}_{\text{PCB}} \) different from the trivial solutions.

4.1.3 Hierarchical assessment

As illustrated in Figure 4.5 (b), not only true devices, but also many false candidates are obtained after the border detection described in Section 4.1.1. These false alarms are arising from diverse non-border objects and unnecessarily printed borders without mounted devices. Regarding
Chapter 4. Assembly print-guided segmentation of SMDs

Figure 4.4: Artifacts in rotation estimation using edge pixels. Left: rotated PCB image. Right: cropped image with detected edge pixels in black. Although the actual rotation angle is $15^\circ$, all detected edge pixels can be decomposed into line segments of the directions $0^\circ$, $\pm 45^\circ$ and $90^\circ$.

their different origins, a hierarchical assessment is applied to the obtained detection results for achieving the desired segmentation of small devices.

The intended cancellation of false alarms begins with the validation of border candidates, where only printed bounding borders should be maintained for further analysis. In Figure 4.5 (b), the most noticeable disturbances are those yellow rectangles with totally or partially missing border lines and corners. Consequently, such candidates lead to a low quality value $G$ according to Eq. 4.2. Thus, a straightforward method for removing the associated false alarms is to conduct a binary classification with respect to an appropriately selected threshold $T_G$ of $G$. It should be noted that $T_G$ must effectively reject false detection results while retaining the true borders. In other words, a wide margin between assembly print and some artifacts must be presented. To verify this prerequisite, an in-depth investigation of the threshold $T_G$ is provided in Section 4.3. Besides the quality measure $G$, shape and texture of small devices are also employed to further improve the accuracy of detection results. On the one hand, considering the standards of SMD packages and the spatial resolution available in PCB images, possible shapes defined by the width and height values of bounding borders in the form of rectangles can be determined and used to validate detected border candidates. On the other hand, classic texture descriptors, e.g. Local Binary Pattern (LBP) [127] and Haralick features [128] from Gray-Level Co-occurrence Matrices (GLCMs) [129],
4.1. Segmentation of small devices

Figure 4.5: Hierarchical cancellation of false candidates in the segmentation of small devices. True candidates are marked with green rectangles, while false candidates with yellow rectangles.

can be combined with machine learning methods, e.g. $k$ Nearest Neighbors ($k$-NN) [130], Support Vector Machine (SVM) [131] and Random Forest (RF) [132], for distinguishing true borders from non-border objects. In case of multiple detections of the same objects, NMS is applied as an additional validation criterion to enforce the non-overlapping constraint of components. For demonstrating the effectiveness of the suggested valida-
tion steps, the intermediate segmentation results of Figure 4.5 (a) with all validated bounding borders are illustrated in Figure 4.5 (c).

Although most false candidates are eliminated through the validation above, there are some unexpected empty borders, as marked with yellow rectangles in Figure 4.5 (c), still remaining in the current segmentation results. These real but unnecessary borders are not occupied by any devices and therefore exhibit the background color of PCBs. This is a distinctive and reliable difference from printed borders with mounted devices, which can be exploited for the purpose of improving segmentation performance. To this end, the Ridge-based Analysis of Distributions (RAD) [133] for color segmentation is employed and distinguishes PCB background from components. In consideration of spatially distributed segments of the same semantic objects, the segmentation analysis is conducted in color spaces, or more specifically, on histograms of pixel values. In addition to clustering samples in color spaces, RAD also investigates the connectivity between neighboring clusters by tracking ridges of color distributions, where connected ridges are assumed from the same/similar semantic objects. Subsequently, the background color is determined as the dominant cluster with similar hue values in the Hue-Saturation-Value (HSV) [134,135] color space. The obtained final segmentation of small devices is visualized in Figure 4.5 (d). As a well-known issue of color segmentation, the quality of discrimination is subject to the intra-class variance. If the original color distributions are altered due to uneven illumination or reflectance, as well as soiling, the segmentation results might become less reliable, which is demonstrated by the remaining false alarm at the top of Figure 4.5 (d).

4.2 Segmentation of ICs

As stated through Figure 4.1, assembly print is far from a reliable indicator for localizing ICs and thus cannot be employed as guidance for the segmentation purpose. Regarding the general characteristics of ICs, their homogeneous surface areas and standardized shapes are considerable features for facilitating the desired segmentation. Since partitioning images into homogeneous regions is quite straightforward and the validation of
4.2. Segmentation of ICs

Figure 4.6: Size validation of IC candidates. An IC candidate visualized as the white foreground in (a) is considered as valid only if the length of valid skeleton fractions in (d) exceeds a predefined threshold.

required shapes is better applicable on localized candidates, these two features are sequentially applied in the following segmentation pipeline.

In general, significant transition areas exist between ICs and background or surrounding components. As a natural consequence of the background color estimation conducted in the last validation step of Section 4.1.3, the candidates of mounted ICs are simultaneously obtained since the PCB image is partitioned into subregions of homogeneous pixel values through RAD. After this candidate generation relying on the first homogeneity feature, the second feature associated with the shape constraint is considered for determining the segments of real ICs. Besides the standard shape validation considering region properties, e.g., bounding box, solidity, major and minor axes, etc., with respect to the shapes defined by the package standards of ICs, an additional size validation is applied for achieving more confident segmentation results. Given an IC candidate visualized as the white foreground in the Black-White (BW) image $I_{bw}$ of Figure 4.6 (a), the Euclidean distance value $d(x)$ for any pixel position $x$ via Distance Transform (DT) [136] is computed as follows:

$$d(x) = \arg\min_{x_i \in B(I_{bw})} \|x - x_i\|_2,$$  \hspace{1cm} (4.5)

where $B(I)$ denotes all background pixels of $I_{bw}$ in black. The resulting DT image is illustrated in Figure 4.6 (b) with brighter pixels representing larger distance values to background. To extract the relevant information for the intended size validation, the skeleton of this candidate is also constructed using an efficient thinning operation [32] and marked with red pixels in Figure 4.6 (c). To conduct the size validation against the minimum dimension $l_{min}$ of ICs, only those skeleton fractions with distance
values higher than $l_{\text{min}}$ are counted as valid, which are marked with green pixels in Figure 4.6 (d). If the total length of valid skeleton fractions exceeds a predefined threshold, the corresponding candidate is validated. In practice, $l_{\text{min}}$ is selected to be slightly lower than the half of the minimum short side length of ICs and the validation threshold is determined according to the minimum long side length.

### 4.3 Evaluation

#### 4.3.1 Rotation estimation

The rotation estimation in Section 4.1.2 is an essential pre-processing step for achieving reliable segmentation results. Aiming at a consistent and objective evaluation of its performance, 14 PCBs were attached to a turntable, which rotated over 27 predefined orientations between $0^\circ$ and $360^\circ$ for each PCB. The corresponding boxplots indexed from 1 to 14 for the measured estimation error are presented in Figure 4.7. Apparently, there exist some systematic biases in these error values, which are arising from the inconsistency between the orientations of the turntable and PCBs. Thus, an appropriate error analysis was conducted on the distribution of the estimation error regarding the median error value. For most cases, the deviation is limited to $\pm 1.5^\circ$, which validates the quality of
4.3. Evaluation

the proposal rotation estimation. Moreover, even the absolute estimation error remains predominately below $8^\circ$ and verifies an adequate correction for the assembly print-guided segmentation of SMDs. The proposed estimation algorithm has also been evaluated on challenging images exhibiting diverse disturbances, e.g. non-fronto-parallel view, low resolution and non-dominant PCB structures. Even on those images, robust estimation performance, as demonstrated in Figure 4.8, has been observed. By rotating all PCB images back to the desired orientation, the structuring elements in Figure 4.3 are able to detect corners of most printed borders. This substantially improves the number of segmented SMDs.

4.3.2 Segmentation

4.3.2.1 Data and performance measures

Due to the insignificant size of small devices in comparison to general PCB components and the required auxiliary information in the form of assembly print, a general dataset “PCB General” introduced in Section A is unsuited for assessing the performance of the proposed SMD segmentation. To provide the desired reference data in conducting a quantitative evaluation, an additional dataset “PCB SMDs” consisting of 32 images has been created, where potential influencing factors are also included for a more detailed performance analysis. For the reader’s convenience, a summary of the dataset “PCB SMDs” with statistical information is presented in Table 4.1.
Table 4.1: Statistics of the dataset “PCB SMDs”. \#(\cdot) denotes the cardinality operator on sets.

<table>
<thead>
<tr>
<th>category</th>
<th>illumination</th>
<th>resolution (pixels/mm)</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>lateral</td>
<td>dome</td>
<td></td>
</tr>
<tr>
<td>#(images)</td>
<td>11</td>
<td>21</td>
<td>16</td>
</tr>
<tr>
<td>#(empty borders)</td>
<td>257</td>
<td>626</td>
<td>463</td>
</tr>
<tr>
<td>#(small devices)</td>
<td>1128</td>
<td>2096</td>
<td>1716</td>
</tr>
<tr>
<td>#(ICs)</td>
<td>38</td>
<td>90</td>
<td>84</td>
</tr>
</tbody>
</table>

Chapter 4. Assembly print-guided segmentation of SMDs

Usually, image analysis algorithms are sensitive to the variations of lighting conditions. To characterize the segmentation performance under varying illumination, two widely used lighting constructions, e.g. lateral and dome, are considered. As illustrated in Figure 4.9, dome construction results in more homogeneous illumination in contrast to the lateral variant, while the latter gives better contrast on certain objects. It is thus of great interest to known if any variant is superior to the other and should be preferred in practice. Another external factor with potentially significant influence on segmentation results is the spatial resolution of images. A carefully tuned trade-off is often required: adequate resolution for the access of detailed information and limited resolution for the computational efficiency. To this end, the 32 PCB images have been acquired with different spatial resolutions, ranging from 13 pixels/mm to 24 pixels/mm.

Besides a suitable dataset, appropriate performance measures are also essential for retrieving relevant information from segmentation results. Given the ground truth data with all SMDs (in total 3325) in the 32 images, it is possible to determine the sets of True Positive ($TP$), False Positive ($FP$) and False Negative ($FN$) samples for any segmentation. If $N$ hypotheses are made for the sought SMDs and $M$ of them are found real in the ground truth, there are in fact $M$ true positive and $(N - M)$ false positive samples. The remaining $(3325 - M)$ unmatched SMDs in the ground truth become false negative samples. In comparison to classic two-class problems, no True Negative ($TN$) samples can be determined for segmentation tasks since true background objects are not explicitly defined. Using $TP$, $FP$ and $FN$, three performance measures, e.g. precision, recall and F-score, are computed as follows for achieving a quanti-
4.3. Evaluation

(a) lateral illumination  (b) dome illumination

Figure 4.9: SMDs in images under different illuminations.

tative assessment of the SMD segmentation:

\[
\begin{align*}
\text{precision} & = \frac{\#(TP)}{\#(TP) + \#(FP)}, \\
\text{recall} & = \frac{\#(TP)}{\#(TP) + \#(FN)}, \\
\text{F-score} & = \frac{(1 + \beta^2) \cdot \text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}},
\end{align*}
\]

(4.6)

where \( \beta \) is a positive real scale factor defining the relative importance of recall with respect to precision. The first two measures quantify the sensitivity and the reliability of the segmentation on the involved dataset, respectively, while the F-score provides a comprehensive quality measurement through weighted average of precision and recall. If not stated otherwise, \( \beta \) is set to one throughout this thesis so that the F\(_1\)-score with balanced weights between precision and recall is in fact obtained.

4.3.2.2 Results

As stated in Section 4.1.3, the quality threshold \( T_G \) needs to be determined experimentally. To avoid possible overfitting, only those values
are to be selected, near which no strong variations of the segmentation performance are observed. In other words, the performance curves of precision, recall and F-score should exhibit slow variations at the selected $T_G$. In Figure 4.10, recall is well stabilized for the interval $0 \leq T_G \leq 0.6$ and decreases significantly after $T_G = 0.6$. A plausible selection of $T_G$ is thus the interval $[0, 0.6]$. Since the performance curves of precision and F-score are smooth and vary slowly at this interval, 0.6 turns out to be the optimal value of $T_G$. However, 0.6 is also a critical value due to the immediate decrease in recall for values $> 0.6$ and should be avoided in general. Applying a tolerance of 0.05 for maintaining a safety distance from the critical point 0.6, $T_G = 0.55$ is finally selected for validating border candidates with respect to their quality measurement $G$.

For the comparison purpose, alternative methods need also be considered. Thresholding-based detection of solder joints and intensity/color-based segmentation of general objects as two additional solutions for generating SMD candidates before the hierarchical assessment have also been tested. However, they are not able to provide reliable segmentation results as the assembly print-guided method, which is proven especially effective in the case of SMD segmentation.

In the pipeline of the proposed SMD segmentation, there exists a machine learning-based validation of border candidates. This leads to difficulties in conducting the desired performance assessment as no additional data are available. To overcome this problem, the leave-one-out cross-validation method has been adopted, where the classifier was trained using all but the current test image. The values of single measures precision, recall,
4.3. Evaluation

and F-score were computed on each test image individually. The final global performance measurements were obtained by averaging them over all images, which are in fact macro performance measurements.

To determine the best combination between features and classifiers for the optimal discrimination, a comprehensive analysis is provided. It should be noted that empty borders are also considered as true candidates in this step. After leave-one-out cross-validations using all possible combinations, no significant difference (less than 1%) between Haralick and LBP features could be observed, no matter which classifier has been used. Due to the lower dimension of Haralick features and the associated efficiency, they are preferred over LBP features in practice. In comparison to SVMs and RFs, k-NN classifiers resulted in slightly lower precision and recall, but the differences were limited to 2%. SVMs and RFs provided comparably good results (precision and recall close to 1), once linear SVMs have been used. Radial kernel functions have led to a slight decrease (less than 1%) in the classification performance of SVMs, both in precision and recall. Regarding the better computational efficiency observed for SVMs, they were employed in the pipeline to produce the final segmentation results.

In the final evaluation, images in the dataset “PCB SMDs” were categorized according to different criteria. To investigate the influence of lighting conditions, two subsets acquired under lateral and dome illuminations were evaluated individually, where the cross-validations were only performed on the corresponding images. Using the boundary value 17 pixels/mm, the images were also divided into two subsets with low and high spatial resolutions, respectively. With the help of these two subsets, an investigation of the second influencing factor, spatial resolution, became possible. Moreover, a global test using all images has been conducted and provided an overall performance assessment. All evaluation results are illustrated in Figure 4.11.

Apparently, better segmentation results are obtained for small devices than for ICs. This is a consequence of two reasons. On the one hand, auxiliary information in the form of assembly print is available for guiding the localization of small devices, while candidates of ICs are generated using image partitioning method based on color information. In comparison to the hierarchical assessment of small devices, on the other hand, fewer constraints have been considered in the validation stage of
Figure 4.11: Segmentation results on the dataset “PCB SMDs”, where 0.5 is selected as the threshold of the confidence score to distinguish between true and false candidates in the final validation step. The segmentation performance is evaluated according to two individual criteria: illumination and resolution.

the IC segmentation. These two facts are responsible for the lower values in recall and precision in Figure 4.11 (b). However, the segmentation performance in Figure 4.11 (b) exhibits lower variations in contrast to Figure 4.11 (a), which indicate higher robustness under varying image conditions. This is a plausible consequence of the region-based analysis applied on ICs, which is less sensitive to the variations of lighting conditions and spatial resolution. Among the results obtained for small devices, significant differences are observed between the two illumination types. A higher value in recall is observed if dome illumination is employed, while a higher value in precision arises from lateral illumination. Interestingly, only slight improvement can be observed for images of higher spatial res-
oluition. But this improvement is available for both precision and recall. As expected, the overall segmentation performance is different from the averaged performance of different imaging configurations since the images used for cross-validations are different. The final values of F-score for the segmentation of small devices and ICs are obtained as 0.827 and 0.740, respectively.

4.4 Discussion

A great challenge in the segmentation of SMDs for PCB recycling is the lack of prior knowledge, which is, on the contrary, usually available for AOI tasks and can substantially reduce the complexity of the underlying image analysis. Other factors further increase the uncertainty of the components to be analyzed are aging, erosion, soiling, etc., through which the characteristics of SMDs may be altered from their original status. For instance, the essential properties of solder surfaces for localizing SMDs using specific lighting conditions [30] are in general not presented in the case of recycling, mainly due to aging and soiling. To address these problems, an assembly-guided SMD segmentation has been introduced in this chapter and it is believed to be the first feasible solution with quantitative performance assessment found in the literature.

To provide a deeper insight into the suggested segmentation, a comprehensive performance analysis has been conducted. As demonstrated through the evaluation of diverse combinations between features and classifiers, reliable classification has been permanently observed, regardless of the employed feature and classifier. This indicates a good separability between printed bounding borders and disturbances. Starting from this desirable property, an efficient alternative solution relying on detection of SMDs becomes considerable. Similar to face detection using a cascade of weak classifiers [118], candidates originating from Section 4.1.1 or some generic proposal generation algorithms, e.g. [137, 138], are to be validated through a series of Haar-like features, which are extracted at very low computational complexity using integral images [118]. However, to ensure the desired classification performance in the case of cascaded features, far more extensive training samples are required.
From the evaluation results presented in Figure 4.11 (a), several observations can be made. If highly reliable segmentation results are the objective, lateral illumination is the right choice. Otherwise, the dome variant is preferred for finding more devices while bearing lower precision. Moreover, as observed in a preliminary study, the segmentation performance in the latter case can be further improved, especially in terms of reliability, by employing more distinctive features and advanced machine learning methods, e.g., the Bag-of-Visual-Words (BoVW) model [139] or Deep Learning (DL) [140]. As verified through the comparison between results on image sets of low and high spatial resolutions, the obtained segmentation is robust to the variation of spatial resolution. This favorable property is very valuable for practical applications since PCB images can be acquired with low spatial resolution for less computational complexity while maintaining the desired segmentation performance.

The resulting F-score is much lower on ICs than on small devices, which is the consequence of the color-based image partitioning and the weak geometric constraints applied for identifying ICs in images. To address this problem and also to achieve a more comprehensive information retrieval on PCBs, advanced approaches exploiting general 2D information and facilitating an analysis of general components will be presented in the next chapter.
In the last chapter, a dedicated segmentation aiming at SMDs as a subset of PCB components has been introduced, while a comprehensive analysis of general components on PCBs is still missing. Apart from the special auxiliary information, e.g. assembly print, which could be absent on PCBs manufactured in small batches or with low density of components, more general information is to be considered. Since the most common and realistic configuration of industrial imaging is acquiring the 2D projection of PCBs using a single camera, analysis approaches presented in this chapter focus on single 2D images with color information. For a better understanding of the intended 2D analysis of components, a typical PCB image with the desired information retrieval is shown in Figure 5.1. All components of significant sizes are localized via bounding boxes in the form of rectangles, which need to be determined via an object localization approach designed for PCBs. Moreover, the localized components are assigned to the corresponding categories (slot, IC, capacitor, transistor, etc.) by the labels visualized on the top of bounding boxes, which requires an appropriate categorization relying on machine learning.
Chapter 5. General 2D information-based analysis of components

Figure 5.1: Objectives of the general 2D information-based component analysis. All components localized via bounding boxes (rectangles with yellow edges) are automatically assigned to the corresponding categories (labels in magenta).

Regarding the two-stage information retrieval defined above, the remainder of this chapter is organized as follows: after reviewing the state of the art in object localization, classification and detection, approaches developed for localizing and categorizing general component on PCBs are introduced in Section 5.2 and 5.3, respectively. Especially, in comparison to the previous publication [7], a novel end-to-end solution with significantly improved localization performance is proposed. To conduct a quantitative and comprehensive performance assessment of these approaches, results obtained on the dataset “PCB General” are evaluated using a set of appropriately defined measures in Section 5.4, where the achievement of the desired information retrieval is also confirmed. In the last section of this
chapter, a detailed discussion of the presented 2D analysis is provided and leads to some general conclusions.

5.1 State of the art

Object localization and classification, or jointly object detection, are primary tasks in computer vision. A more complex image understanding at semantic level, e.g. scene analysis and interpretation, is usually built upon them. Depending on the employed paradigm, the desired object analysis in images can be realized through two successive stages comprising localization and classification or conducted in a unified detection framework. It should be emphasized that obtaining region proposals (bounding boxes) of the sought objects is the goal of object detection, rather than generating pixel-wise segmented masks. This is a sufficient and more realistic solution for PCB recycling.

In the context of object localization, exhaustive searches via sliding window and image partitioning relying on some prior knowledge are most commonly used approaches. In an exhaustive search, dense detection windows covering possible sizes and aspect ratios of objects are sliding over the entire image to discover diverse objects at all spatial positions and scales. Subsequently, a scoring of the cropped local images is performed and only a small fraction of them is maintained as candidates for the sought objects. In the literature, a great number of scoring functions have been proposed, which are typically derived from the quality of edges [138], from the objectness [141] and from the output of pre-trained linear classifiers [142]. However, sliding window approaches suffer from two major drawbacks: high computational complexity due to the exhaustive search and extremely low precision of detection due to the large number of generated region proposals even after the scoring. To deal with these problems, search algorithms [118,143] with higher efficiency and advanced classification methods are required. There are also alternative approaches using pixel grouping for localizing objects of interest which can in general avoid these undesired side effects. Instead of seeking candidates in a huge search space, region proposals are automatically localized as the optimal image partitioning boundaries are determined by minimizing the value of an objective function with respect to the prior
knowledge, \textit{e.g.} low variation between neighboring pixels of the same objects \cite{144}, similar pixel values within objects \cite{145,146}, low curvature of object boundaries \cite{147} and given initial seeds \cite{148,149}. For a better efficiency, grouping is applied not on single pixels, but on superpixels after an over-segmentation step \cite{150}, where the complexity of solving the optimal partitioning is substantially reduced. While pixel-grouping approaches provide reasonably good localization performance in images of simple scenes, their application on natural images or images of complex scenes is strongly limited since complicated prior information required for reliable partitioning is often absent. Moreover, it is difficult to explicitly integrate such information into the optimization framework.

Usually, there exist a great number of false alarms along with real objects in the localization results. A necessary post-processing step is thus to remove them before assigning the region proposals to the predefined categories of objects. This cancellation of false proposals can also be combined with the subsequent categorization by introducing additional categories for non-relevant image regions. Methods relying on machine learning are primarily adopted for this classification task. In a cascaded paradigm, representative features, \textit{e.g.} Histograms of Oriented Gradients (HOG) \cite{142}, the Bag-of-Visual-Words (BoVW) \cite{139}, \textit{etc.}, are first extracted in region proposals and then fed to a pre-trained classifier, \textit{e.g.} \textit{k}-Nearest-Neighbours (\textit{k}-NN) classifier \cite{130}, Decision Tree (DTree) \cite{151}, Random Forest (RF) \cite{132}, Support Vector Machine (SVM) \cite{131}, Artificial Neural Network (ANN) \cite{152}, \textit{etc.}, for determining the associated classes or categories of these regions. In a compact paradigm, feature extraction and training classifiers are conducted simultaneously on extensive region samples in a training stage. By this means, handcrafted features are omitted, while more representative and reliable features can be obtained if sufficient training data are provided. Furthermore, in comparison to the conditional optimization of classifiers for the predefined features, a global optimization of the overall classification performance is achieved. Classification methods based on \textit{Deep Learning} (DL) \cite{140} commonly fall into this category and demonstrate in general superior performance than stepwise variants with explicitly defined feature extraction.

Benefiting from the recent progress in machine learning and the emerging of extraordinarily large scale datasets \cite{153,154,155}, object detection without intermediate processing stages is becoming more realistic. In such unified systems \cite{156,157,158}, original images are used as input and class-
specific locations of sought objects are generated as output. By discarding expensive searches and explicit incorporation of prior information, object detection turns out to be much more efficient and straightforward. However, due to the far from sufficient size of available PCB datasets and the inflexible architecture of current systems for compact object detection, the presented 2D analysis of PCB components is still realized in a cascaded fashion, where localization and categorization of sought objects are conducted in two successive stages.

5.2 Localization

Based on state-of-the-art segmentation and classification methods, a combinatorial localization approach driven by the diversification strategy has been developed and published in [7], which is the first proposed solution for a general analysis of components in PCB recycling. Later, after a thorough review of this approach and further development, a more compact analysis pipeline combining a novel, robust region proposing and a DL-based validation in the regression-classification scheme has also been proposed, where more satisfying results can be obtained while using a more straightforward analysis. This compact localization approach has been partially published in an electronic preprint [9]. For a better representation of this chronological development, these two approaches are introduced successively in this section.

5.2.1 Combinatory approach

As illustrated in Figure 5.2, the combinatorial localization [7] consists of four major steps: scale-space generation, pixel/superpixel-level feature extraction, image partitioning and validation of generated region proposals. The first three steps jointly propose region candidates of sought objects, where the risk of missing PCB components is minimized by adopting the diversification strategy. In the final validation step, predefined distinctive features (e.g. HOG and BoVW features) are first extracted from region proposals and then used to cancel false alarms with the help of pre-trained classifiers.
Figure 5.2: Workflow of the combinatory localization approach. The first three steps: scale-space generation, feature extraction and image partitioning, jointly propose region candidates of sought PCB components. The obtained candidates are subsequently validated with the help of distinctive features and pre-trained classifiers.

5.2.1.1 Diversification-driven proposal generation

PCB components of distinct electronic characteristics and emerging from different manufacturing processes exhibit a great diversity in their appearance, e.g. color, size, shape, texture, etc. Their assembly on PCBs is also highly unpredictable and depends on the intended circuit implementation as well as marginal conditions, e.g. spatial constraints, electromagnetic compatibility, etc. All of these facts lead to a great challenge for state-of-the-art localization algorithms: on the one hand, varying size and aspect ratio over wide ranges require a hardly manageable search space for a slid-
5.2. Localization

window-based analysis, while, on the other hand, highly complicated prior information must be explicitly considered in solving the optimal image partitioning boundaries if a pixel-grouping analysis is preferred.

There exists a similar challenge in analyzing natural scene images, where a high variability of sought objects is also present. To address this problem, approaches [138,159,160,161,162] were proposed and led to substantial progresses in detecting objects in such images. Among them, Selective Search [137] provided a remarkable boost to state-of-the-art performance of proposing region candidates, where the diversification strategy plays a key role for a sufficient localization of sought objects in complex scenes. To deal with the problem of insufficient localization arising from approaches only using single parameter settings, scales and image cues (visual information embedded in brightness, color, texture, shape, etc.) [163], complementary results from diverse parameter settings (including different color spaces), scales and merging criteria are combined in Selective Search for maximizing the probability of discovering sought objects. Inspired by this idea, a diversification-driven generation of comprehensive region proposals is also considered in the combinatory approach for localizing PCB components. However, regarding the fact that PCBs are usually more complex than natural scenes in size and in density of sought objects, as well as in distracting background objects, a modified diversification strategy is applied here. Rather than a bottom-up merging of initial region proposals, which results in an unmanageable amount of region candidates on PCB images, a diversification in scales, image cues and segmentation algorithms is employed for exploiting diverse prior and scale information while keeping the overall proposed candidates in a reasonable amount.

Although automatic scale selection [164] has shown its relevance in feature detection [110,112], it is difficult to define an appropriate objective function for determining the optimal scale for representing the underlying PCB components. Thus, a scale-space representation [165,166] of PCB images is required for the successive feature extraction in the localization pipeline. Gaussian scale space is commonly the first choice for the purpose of multi-scale representation of images. But an essential drawback of this straightforward method is the constant kernels applied for smoothing, where relevant information, e.g. edge and texture, may also be smoothed away. By contrast, an alternative representation in nonlinear scale space (NSS) suppresses irrelevant details while preserving or even enhancing significant variations at all scales, where the Gaussian smoothing is re-
placed by a nonlinear filtering. Guided by different objective functions, Anisotropic Diffusion [124] (in fact, it uses an isotropic diffusion function), Total Variation Flow [167], Edge Enhancing Flow [168] and Mean shift [146] lead to different representations in NSS. Moreover, based on the Fast Explicit Diffusion (FED) scheme [169,170] performing explicit diffusion with varying time steps instead of a constant step size, a fast approximation of nonlinear diffusion is possible [108]. An overview of images arising from these scale spaces is presented in Figure 5.3. Apparently, Gaussian smoothing and anisotropic diffusion are incapable of efficiently suppressing distracting details while sufficiently preserving significant structures. In comparison to edge enhancing flow and Mean Shift, total variation flow generates more homogeneous images by weakening region boundaries between different objects and smoothing away small objects. Regarding these facts and the computational efficiency, edge enhancing FED and Mean Shift are thus suitable methods for a scale-space representation of PCB images. Indeed, a combined scale-space representation using both methods is unnecessary. The scale space arising from Mean Shift is generated through a mode analysis at different scales, where pixels of the same mode (or spatially neighboring pixels of similar modes) are merged into the same region. By this means, pixels of similar values are grouped correspondingly and reasonable boundaries for partitioning images are automatically obtained. To avoid a duplicate analysis, Mean Shift is therefore shifted into the later stage “image partitioning” of the localization pipeline, which implicitly realizes a simultaneous generation of the associated scale space.

An essential step for the success of Selective Search is exploiting diverse image cues embedded in different color spaces and texture. This idea is also adopted in the diversification-driven proposal generation for localizing diverse PCB components. More specifically, all commonly used color spaces and image features for grouping pixels/superpixels are considered in the first place. Inappropriate and redundant image cues are filtered out later through a comprehensive evaluation with respect to their resulting localization performance. For a compact presentation, all involved color spaces and image features are listed below:

**color space** gray value, RGB, CIE L*a*b*, CIE L*u*v*, HSV [134,135];

**texture** features extracted using cut-off windows [171,172].
5.2. Localization

To further extend the diversity of region proposals, multiple state-of-the-art algorithms providing complementary segmentation performance are combined in the image partitioning stage and group pixels into regions according to different image cues and prior information.

In the context of color-based image segmentation, *Active Contours without Edges* (Active Contours) [145], *Graph Cuts* [144, 173, 174, 175], *Efficient Graph* [176] and Mean Shift [146] are well-established algorithms. Applying Active Contours on an image with a given initial segmentation, boundaries separating different regions are continuously evolved in an iterative manner and this optimization process is terminated if the best compromise between the homogeneity within partitioned regions, region
sizes and the overall boundary length is achieved. In case of given labeling of some seed points, Graph Cuts algorithm models the labeling of remaining pixels as the max flow/min cut search in a graph, where image pixels are vertices of the graph. Usually, the cost of removing connections between pixels is positively correlated with their similarity or, in other words, decreases for increasing discontinuity in pixel values. Graph Cuts can be further extended with a regularization of cutting surfaces in the sense of minimum geodesic contours and an approximation of metric surfaces is realizable using the graph construction introduced in [177]. If no initial seed points are provided, cluster centers of pixels in a selected color space can be utilized as seed points for resulting in a reasonable image partitioning. Relying on a similar concept of using graphs for representing images, the image segmentation problem can also be solved using Efficient Graph, where pixels are hierarchically merged into regions instead of a global splitting of images as in Graph Cuts. Differentiating from the above algorithms aiming at a direction solution of the optimal region boundaries, Mean Shift solves the image partitioning problem in an indirect way and conducts the kernel density estimation-based mode detection over the whole image. After localizing the local maxima of a density function defined by an appropriate kernel (for instance Epanechnikov [178] or truncated Gaussian kernel) and a bandwidth matrix for the pixel position space as well as for the feature space, pixels converging to the same modes are grouped together.

In images of complex scenes, where the homogeneity assumption of colors within the same objects does not hold any more, texture is often used as an alternative and reliable image cue for facilitating the desired segmentation. Compression-based Texture Merging (CTM) and its further development Texture and Boundary Encoding-based Segmentation (TBES) are typical algorithms exploiting texture information for splitting images into reasonable regions. By interpreting image segmentation as the search for the optimal encoding of textural information and boundaries with the shorted coding length, an automated segmentation relying on accurately estimating the true entropy in images is realized. Employing the approximate rate-distortion function [179] of the Gaussian, the texture information in all segmented subregions of an image is represented by a mixture of Gaussians and the total textural coding length is subject to the selected distortion level as well as to the mean vectors and the covariance matrices of underlying feature vectors extracted from the subregions.
In addition to the generic segmentation algorithms reviewed above, a specific method for PCB images and generating extra region boundaries through a background estimation is developed. In the context of component localization, background refers to the regions of PCBs without any functional devices. Such regions comprise typically substrates applied for mechanical support, flame retardancy and electronic isolation, where similar colors are commonly observed on same PCBs. Since background regions build natural boundaries between components, plausible region proposals can thus be generated if these regions are reliably segmented in images. Nevertheless, this task is different from the color estimation conducted in Section 4.1.3 and requires higher precision to avoid shifting any sought components into background. Therefore, the segmentation method RAD used for SMD segmentation is unsuited since the ridge-based analysis of pixel value distributions tends to merge components with background. Instead of investigating the similarity between color distributions, the proposed background estimation groups pixels into single clusters in an appropriate color space using the *Fast Mean Shift* algorithm, which exploits the similarity between background regions in color and retains sought components in foreground. After approximating the real color distribution with data points on a regular grid, the mean shift vector is computed for each grid cell and quantized into integer indexes of grid cells. By this means, all convergence cells of data points are determined in one go and the repeated local mode seeking with high computational complexity is omitted. Considering fragmented data points arising from approximated mean shift, an additional merging step is performed and investigates the connectivity between neighboring clusters in the color space. Finally, to suppress insignificant colors, all clusters with their centers closer than a small distance value are further merged. Although PCB images are split into subregions of similar colors according to the clustered pixels, a classification of the background regions is still missing. In comparison to well-designed components, background regions usually exhibit irregular contours, cluttered gradients, great spatial extent and dominance in area. With the help of machine learning, a strong classifier is trained and applied for distinguishing background from components using these features. The high confidence provided by this classifier for rejecting a non-background candidate is essential for maintaining PCB components in foreground and leading to an improved image partitioning.
5.2.1.2 Region validation

Using the boundaries generated through diverse image partitioning methods, a comprehensive set of region proposals covering PCB components as well as a great number of false alarms is provided. To achieve a precise localization of sought objects while excluding unexpected disturbances, a further validation step is required. Since feature extraction is more convenient on rectangular image crops and the final detection results are to be presented in the form of bounding boxes, bounding boxes of partitioned image subregions are considered for the purpose of region validation.

From the perspective of classification, region validation defines a binary classification problem, where false candidates should be distinguished from the true proposals of sought objects. In contrast to the validation step in Section 4.1.3 for SMD segmentation, a significantly higher variability of region candidates is present for the analysis of general components. Consequently, aiming at a reliable cancellation of background objects, more distinctive features instead of classic LBP or Haralick features need to be considered. After reviewing the latest development in feature extraction for general object recognition, HOG and BoVW features are found capable of providing the desired discrimination performance for categorizing diverse objects from a large number of classes. Regarding the comparable complexity associated with the analysis of PCB components, these two features are adopted in this combinatory localization approach for supporting the employed classifiers in validating generated region proposals.

As illustrated in Figure 5.4, the HOG feature relies on the orientation and magnitude of local gradients extracted in cells of a spatial grid, which splits an image into several rectangular subregions of the same size. While local shapes are captured by cells, global arrangement of local parts is embedded in the combination of cells over the whole image, which leads to a more semantic representation of objects and provides additional margins even if objects from different classes consist of similar local structures. To also deal with the local variations in illumination and contrast, blocks as an intermediate geometric layer between cells and the whole image have been introduced. The histograms of oriented gradients from cells are first block-wise normalized and then concatenated to form the entire feature
5.2. Localization

Figure 5.4: HOG features [180] (in the bottom line) of PCB components (in the top line). Line segments are used to visualize the orientation bins in each cell, where darker line segments indicate stronger gradients. The clusters of line segments are located in the corresponding grid cells.

vector. For the PCB components in Figure 5.4, HOG results in reasonable representations (VLFeat [180] implementation) in the feature space.

Rather than representing the extracted structures in an HOG manner, the BoVW model employs an additional encoding step for extracting compact and representative features even from a great number of primary features. A visual representation of the BoVW model and the corresponding feature is provided in Figure 5.5 for a better understanding. Primary feature descriptors are densely extracted at detected keypoints or on a spatial grid. A great set of these descriptors is collected over extensive training images and divided into $V$ clusters in the primary feature space, which represent the $V$ visual words of the target vocabulary. With the help of an appropriate model for describing the partitioning of the feature space through these visual words, e.g. cluster centroids and distance values for a hard partitioning or a Gaussian mixture model for a soft partitioning, an arbitrary point in the feature space can be assigned to the corresponding visual word or visual words and thus be encoded. Given dense primary features extracted from a test image, a histogram of their assignments to the $V$ visual words is generated through encoding and used as the final feature obtained from the BoVW model. Alternatively, as in the case of Fisher Vector (FV) [181, 182], statistical difference between extracted primary features and the learned visual words can also be used as the
feature for the classification purpose. For further incorporating spatial information into the BoVW model, images can be divided into subregions, in which the feature extraction described above is performed individually.

Besides extracting distinctive features, selecting suited classifiers and determining their optimal configurations are also essential factors for achieving the desired classification. In general classification tasks, SVM classifiers have demonstrated superior performance \cite{131, 183, 184, 185, 186} due to their learned optimal separating hyperplane in the sense of maximum margin \cite{187} and the applied kernel trick for creating nonlinear class boundaries \cite{188}. If they are combined with HOG and BoVW features for computer vision tasks, very promising classification performance can be obtained \cite{142, 189}. Regarding their proven discrimination power, SVMs are utilized for the region validation step. However, concerning the limited size of the dataset available for training, other stable classifiers suited for small datasets also need to be considered and should replace SVMs in case of overfitting. Relying on the Bootstrap Aggregating or Bagging (Bagging) \cite{190} algorithm and the random subspace method \cite{191}, RFs are designed to increased accuracy and decreased variance in comparison to single DTrees. More specifically, samples in a training set are uniformly resampled with replacement to form a couple of new training sets of a certain size. Weakly correlated DTrees are then fitted on these training sets and their outputs are averaged to form the final classification results. To further reduce the correlation between these DTrees, only a random subset of features (or a random subset of components in feature vectors) is considered for determining the best split at each decision node.
By this means, RFs become insensitive to noise in training data and in general resistant to overfitting. Therefore, classifiers in the form of RFs and trained on the small dataset "PCB General" are still able to provide stable classification performance.

It is well known that the model parameters of classifiers have a significant impact on the resulting classification rate. Thus, to ensure that the employed SVM and RF classifiers work properly, their parameters need to be carefully tuned. In addition to an exhaustive search over sparse grid points in the parameter space, extra optimization is adopted for conducting parallel parameter tuning and for providing a reference point to assess the quality of the configuration arising from the grid search. Regarding the two dominant integer parameters in the model of RFs: the number of DTrees and the dimension of the subspaces considered for decision splits, the genetic algorithm [192] is used to find additional parameter configurations for RF classifiers on a bounded region and goes through the same number of trials (chromosomes) as in the grid search. In case of SVM classifiers, the two dominant parameters: the kernel scale and the coefficient for penalizing false labeling of samples, take unbounded continuous values. To achieve an appropriate parameter optimization, the Nelder-Mead simplex algorithm [193] is adopted and conducts a derivative-free search based on the parameters obtained through the genetic algorithm. It should be emphasized that the optimal model parameters are determined merely using the training data in each training-test round of a cross-validation. To improve the reliability of the parameter optimization, training data are further divided into $k$ folds and the optimal parameter configuration is determined through minimizing the overall training-test cost of another cross-validation on these $k$-fold data.

### 5.2.2 Compact approach

Although the combinatory approach described above provides a practical solution for localizing PCB components, the involved diversification in the proposal generation and the cascaded region validation turned out to be very complicated and result in a low scalability for an increasing amount of images. Moreover, the localization performance depends predominantly on the quality of region proposals arising from image partitioning, which suffers from difficulties in case of complex components. Bearing these facts
in mind, a generic segmentation method and a regression-classification scheme utilizing the power of DL are developed. They realize consequently a more compact workflow as illustrated in Figure 5.6, which results in more straightforward proposal generation and region validation, as well as improved quality of the obtained region candidates.

5.2.2.1 Local variance-based segmentation

Homogeneity is important for visual perception and has been widely used in image analysis and computer vision applications. Currently, most ap-
proaches exploit homogeneity in an implicit form using local gradient information (e.g. edge detection [125]) or considering variations in pixel values (e.g. maximally stable extremal regions [109]). In the field of text recognition, the local standard deviation is often used as the advanced measure of homogeneity and makes an adaptive thresholding of pixels [194, 195, 196, 197] possible. As informative image cues, other variants of the homogeneity measure have also been applied for image segmentation purpose [198, 199]. Inspired by these applications of homogeneity, especially by the standard deviation-based measure, a similar homogeneity analysis conducted in local regions of multiple scales is proposed for better exploiting the underlying image contents and is referred to as the multi-scale Local-Variance (LV) analysis in this thesis.

To clarify the advantages of the proposed LV analysis, a close view of its mathematical definition is provided. Given an image $I$ and the pixel $I(x_i)$ at the position $x_i$, the pixel value $I_{LV}(x_i)$ at the same position in the corresponding LV image $I_{LV}$ is determined in a local window $W(x_i)$ centered at $x_i$ and according to one of the following equations:

$$I_{LV}(x_i) = \left( \sum_{x_{1,i} \in W(x_i)} \frac{\|I(x_{1,i}) - \overline{I(W(x_i))}\|_2^2 \cdot w(x_{1,i} - x_i)}{\sum w(x_{1,i} - x_i)} \right)^{\frac{1}{2}}, \quad (5.1)$$

$$I'_{LV}(x_i) = \left( \sum_{x_{1,i} \in W(x_i)} \frac{\|I(x_{1,i}) - \overline{I(W(x_i), x_{1,i})}\|_2^2 \cdot w(x_{1,i} - x_i)}{\sum w(x_{1,i} - x_i)} \right)^{\frac{1}{2}}, \quad (5.2)$$

where

$$\overline{I(W(x_i))} = \frac{\sum_{x_{1,i} \in W(x_i)} I(x_{1,i}) \cdot w(x_{1,i} - x_i)}{\sum w(x_{1,i} - x_i)}, \quad (5.3)$$

$$\overline{I(W(x_{1,i}))} = \frac{\sum_{x_{1,j} \in W(x_{1,i})} I(x_{1,j}) \cdot w(x_{1,j} - x_{1,i})}{\sum w(x_{1,j} - x_{1,i})}, \quad (5.4)$$

and

$$w(x_{1,i} - x_i) = \begin{cases} \frac{1}{2 \sigma^2 \pi \cdot 2 \sigma^2} e^{-\frac{\|x_{1,i} - x_i\|^2}{2 \sigma^2}} & \text{for } x_{1,i} \in W(x_i), \\ 0 & \text{otherwise}. \end{cases} \quad (5.5)$$

These two alternative definitions model LV using the weighted variations of pixel values in a local window $W(x_i)$. In addition to the first definition
Chapter 5. General 2D information-based analysis of components

According to Eq. 5.1, the second definition in Eq. 5.2 involves a further smoothing of LV through the adaptive mean value \( \overline{I}(W(x_i)) \) for each pixel \( x_{i} \) in \( W(x_i) \), which avoids the bias of the mean value \( I(W(x_i)) \) towards the central pixel value \( I(x_i) \). Theoretically, instead of using the root of the weighted sum of \( L_2 \)-norm variance values \( \| I(x_{i}) - \overline{I}(W(x_i)) \|_2^2 \), LV may also be computed as the weighted sum of \( L_1 \)-norm variance values \( \| I(x_{i}) - \overline{I}(W(x_i)) \|_1 \). However, the computation of the \( L_1 \)-norm variance values leads to an implementation issue for the first definition of LV: the value of each pixel in \( I \) must be subtracted by a great number of neighboring pixel values and the number is proportional to the size of \( W \). This difficulty can be overcome if Eq. 5.1 based on the \( L_2 \)-norm variance values is employed and further transformed into the following expression:

\[
I_{LV}(x_i) = (\sum_{x_{i} \in W(x_i)} \frac{\| I(x_{i}) \|_2^2 \cdot w(x_{i} - x)}{\sum w(x_{i} - x)} - \| \overline{I}(W(x_i)) \|_2^2)^{\frac{1}{2}} . \tag{5.6}
\]

Since the mean value \( \overline{I}(W(x_i)) \) is decoupled from the weighted sum of \( \| I(x_{i}) \|_2^2 \), the computation of \( I_{LV}(x_i) \) becomes straightforward.

On the one hand, LV analysis gives rise to a strong response in proximity to and at boundaries between different regions, which is similar to conducting edge detection. On the other hand, in case of point-wise disturbances or single lines, local regions are spatially smoothed, which is similar to conducting image enhancement. Moreover, significant response with low spatial variability is obtained on strongly textured or cluttered regions. This leads in general to favorable segmentation results. For some special tasks relying on structure analysis, for instance feature-based detection of symmetry axes, LV analysis is also a preferable method for enhancing relevant information while suppressing disturbances. As illustrated in the top line of Figure 5.7, more SIFT features are detected on the LV image and capture the structure of the street light. After applying a state-of-the-art detection algorithm [200], most symmetry axes are localized. This is a direct consequence of the windowed analysis of LV. Besides enhanced contours of different parts, extra structure features are also introduced. For instance, lines give rise to distinct response in comparison to corners and irregular structures (e.g. different parts joining in the same local window). In the bottom line of Figure 5.7, distracting background features are smoothed away through LV analysis, while foreground objects are emphasized and more dense features of interest are available for
5.2. Localization

![Figure 5.7: SIFT features (red crosses) and axes of symmetry (line segments) detected in original (the first column, available from [201]) and local-variance (the fourth column) images.](image)

detecting symmetric structures. By this means, false alarms arising from background disturbances, for instance the second axis (marked with a blue line segment) in the third image of the bottom line, can be avoided. Original images used in Figure 5.7 are available from a public dataset [201].

Along with the advantages mentioned above, LV analysis also leads to some undesirable artifacts in nature. A major drawback of a raw LV image is the bold transition regions from high to low response, and vice versa, which are the results of the local window-based analysis and can be observed in Figure 5.8 (b). To deal with this problem, certain post-processing steps must be considered. The most straightforward solution for narrowing the transition regions is to incorporate the position of edges and use it to strengthen or weaken LV response, respectively. To this end, the *Laplacian of Gaussian* (LoG) response of the image $I$ defined as

$$I_{\text{LoG}} = I \ast \nabla_{x_1}^2 G(\sigma)$$

(5.7)

with

$$\nabla_{x_1}^2 G(\sigma) = \frac{\partial^2 G(\sigma)}{\partial x_1^2} + \frac{\partial^2 G(\sigma)}{\partial y_1^2}$$

(5.8)

is used to modify the LV image $I_{LV}$. The pixel value $I_{LV}(x_1)$ at position $x_1$ is multiplied with the coefficient $C_{\text{LoG}}(x_1)$:

$$C_{\text{LoG}}(x_1) = 1 - \|I_{\text{LoG}}(x_1)\|_2,$$

(5.9)
Chapter 5. General 2D information-based analysis of components

Figure 5.8: The raw LV image undergoes further post-processing and multi-scale filtering for achieving enhanced image quality.

where $G(\sigma)$ is a truncated zero-mean Gaussian kernel with the variance $\sigma^2$ and $\|I_{\text{LoG}}\|_2$ is defined on a normalized value range with $0 \leq \|I_{\text{LoG}}(x_1)\|_2 \leq 1$ for any $x_1$. Additionally, a further reduction of the transition regions is achievable with the help of the edge enhancing FED [108]. Although the modified LV image has better emphasized boundaries, extremely high response values arising from strongly textured or cluttered regions may govern the value range of $I_{\text{LV}}$. In such cases, an uneven distribution of response values is usually presented, which could lead to difficulties in subsequent image processing, e.g. edge detection or segmentation. Fortunately, these extreme values can be suppressed if the LV image undergoes a simple logarithmic transformation $T_{\log}(x_1) = \ln(1 + I_{\text{LV}}(x_1) \cdot (e - 1))$ for $I_{\text{LV}}$ with response values normalized between 0 and 1.

Following the idea of capturing features of diverse spatial extent in a scale space [164, 165, 166], multi-scale LV images are generated using local windows of different sizes and extract homogeneity information at all relevant scales. For a selected scale $s$, a centered Gaussian kernel with the variance $\sigma^2 = (2^s)^2$ is truncated on a local window $\mathcal{W}$ of the size $(6 \cdot 2^s + 1) \times (6 \cdot 2^s + 1)$ and used for conducting LV analysis at each pixel position. To preserve homogeneous regions detected at different scales, they are cumulatively merged into the LV images at lower scales. Starting from the coarsest scale $N$, the corresponding LV image $I_{\text{LV},N}$ is obtained using Eq. 5.6 or 5.2 and undergoes post-processing steps. After further conducting LV analysis at the next lower scale $(N - 1)$, the corresponding $I_{\text{LV},N-1}$ is obtained by merging the temporary LV image with $I_{\text{LV},N}$ and only retaining the minimum LV response value for each pixel position: $I_{\text{LV},N-1}(x_1) = \min(I_{\text{LV, temporary}}(x_1), I_{\text{LV},N}(x_1))$. This multi-scale filtering
is repeated for each scale to maintain all homogeneous regions emerging at higher scales. For the reader’s convenience, the algorithm for generating multi-scale LV images is summarized in Algorithm 3.

**Algorithm 3** Generating multi-scale LV images

**Input:** image $I$, the top level $N$.

**Output:** local variance images $I_{LV,0}, I_{LV,1}, \cdots, I_{LV,N}$ over scales 0 to $N$.

1: Set current scale parameter $\sigma = 2^N$.
2: Get weighting factors $w$ on local window $W$ using truncated Gaussian function and the scale parameter $\sigma$.
3: Compute LV image $I_{LV,t}$ from $I$ using Eq. 5.6 or 5.2.
4: Perform post-processing on $I_{LV,t}$.
5: Set $I_{LV,N} = I_{LV,t}$.
6: **for** $s = (N - 1)$ to 0 **do**
7: Set current scale parameter $\sigma = 2^s$.
8: Update $w$ on new local window $W$ determined by $\sigma$.
9: Compute LV image $I_{LV,t}$ from $I$ using Eq. 5.6 or 5.2.
10: Perform post-processing on $I_{LV,t}$.
11: **for** each pixel position $x_1$ in $I$ **do**
12: Set $I_{LV,s}(x_1) = \min(I_{LV,t}(x_1), I_{LV,s+1}(x_1))$.
13: **end for**
14: **end for**

The multi-scale LV analysis can be combined with state-of-the-art image segmentation algorithms and used as a pre-processing step for improving segmentation performance. As demonstrated in Figure 5.9 (c), boundaries of PCB components are adequately strengthened in LV images and reasonable segmentation results can thus be expected even merely using an appropriate thresholding algorithm, which is more efficient than other image partitioning techniques. Otsu [202] and *Mean Shift* clustering [146] are the two most favorite adaptive thresholding algorithms. However, regarding the high variability of PCB images and the resulting unknown distribution of LV response, Otsu is less suitable since it suffers from a severe difficulty of requiring the predefined number of clusters. *Mean Shift* clustering is therefore adopted for the purpose of splitting PCB images into region candidates using a global binarization of edge pixels, where the
logarithmic transformation of LV images significantly improves the reliability of the local mode seeking through the equalization of the overall distribution of LV response.

A well-known issue associated with the edge detection-based image segmentation is the fragmentation of object boundaries, which leads to unclosed contours and decreases the segmentation performance. A practical solution to deal with this problem is first splitting images into superpixels and then linking fragmented edge pixels using boundaries of generated superpixels. Among a wide range of superpixel algorithms [150], *Watershed* [203] provides a straightforward image partitioning determined by ridges of pixel values, which can be used as materials for linking contours of components in LV images. To filter out irrelevant edge candidates, all regions of ridges are localized and used as the mask for maintaining valid superpixel boundaries, while regions of valleys are excluded. In general, ridge detection is nontrivial and requires complex analysis [204, 205, 206]. Nevertheless, detecting regions of ridges can be achieved with the help of the second-order directional derivative $D^2_{v_1}(I_{LV} * G(\sigma))$ of $I_{LV}$:

$$D^2_{v_1}(I_{LV} * G(\sigma)) = D_{v_1}(D_{v_1}(I_{LV} * G(\sigma)))$$

$$= D_{v_1}(I_{LV} * \left(\frac{\partial G(\sigma)}{\partial x_1}\cdot \cos (\theta_v) + \frac{\partial G(\sigma)}{\partial y_1}\cdot \sin (\theta_v)\right))$$

$$= I_{LV} * \left(\frac{\partial^2 G(\sigma)}{\partial x_1^2}\cdot \cos^2 (\theta_v) + \frac{\partial^2 G(\sigma)}{\partial y_1^2}\cdot \sin^2 (\theta_v)\right)$$

$$+ 2 \cdot \frac{\partial^2 G(\sigma)}{\partial x_1 \cdot \partial y_1} \cdot \sin (\theta_v) \cdot \cos (\theta_v)$$

$$\approx \frac{1}{(I_{LV} * \frac{\partial G(\sigma)}{\partial x_1})^2 + (I_{LV} * \frac{\partial G(\sigma)}{\partial y_1})^2}$$

$$\cdot ((I_{LV} * \frac{\partial^2 G(\sigma)}{\partial x_1^2}) \cdot (I_{LV} * \frac{\partial G(\sigma)}{\partial x_1})^2$$

$$+ 2 \cdot (I_{LV} * \frac{\partial G(\sigma)}{\partial x_1} \cdot \frac{\partial G(\sigma)}{\partial y_1}) \cdot (I_{LV} * \frac{\partial G(\sigma)}{\partial x_1})$$

$$+ (I_{LV} * \frac{\partial^2 G(\sigma)}{\partial y_1^2}) \cdot (I_{LV} * \frac{\partial G(\sigma)}{\partial y_1})^2),$$

where the operator $\cdot$ between two images denotes the pixel-wise multiplication. Moreover, $[\cos (\theta_v) \sin (\theta_v)]^T$ defines the direction vector $v_1$ at
each pixel of the gradient image $\nabla x I_{LV}$ and

$$\cos (\theta_v) \approx \frac{I_{LV} \ast \frac{\partial G(\sigma)}{\partial x_1}}{\sqrt{(I_{LV} \ast \frac{\partial G(\sigma)}{\partial x_1})^2 + (I_{LV} \ast \frac{\partial G(\sigma)}{\partial y_1})^2}},$$

$$\sin (\theta_v) \approx \frac{I_{LV} \ast \frac{\partial G(\sigma)}{\partial y_1}}{\sqrt{(I_{LV} \ast \frac{\partial G(\sigma)}{\partial x_1})^2 + (I_{LV} \ast \frac{\partial G(\sigma)}{\partial y_1})^2}}. \tag{5.11}$$

Any pixel $x_i$ resulting in $D^2 v_1 (I_{LV} \ast G(\sigma))(x_i) < 0$ is considered as lying in a ridge region. In fact, the denominator of Eq. 5.10 is greater than or equal to zero. Thus, only the numerator is required for determining the sign of $D^2 v_1 (I_{LV} \ast G(\sigma))$. A further simplification of Eq. 5.10 is possible if only the dominant horizontal and vertical directions of component boundaries are considered after correcting the PCB orientation according to Section 4.1.2. In this case, the sign of the LoG response on $I_{LV}$ is employed for detecting regions of ridges.

Edge pixel candidates obtained through thresholding build in general narrow transition regions and emerge in the form other than true boundaries typically of single-pixel width. This is problematic for general segmentation tasks, but sufficient for the purpose of localizing PCB components since only comprehensive region proposals of sought objects are required, where adequate overlapping areas between proposals and sought objects, instead of precise segmentation, are necessary. For a rapid image splitting, boundaries of transition regions as well as valid superpixel boundaries are used and a thinning operation on the binary image with edge pixel candidates is thus omitted. The resulting contour pixels of a demo PCB are visualized in Figure 5.9 (d). Apparently, edge transition regions of the LV image in Figure 5.9 (c) give rise to narrow holes along the true boundaries, which, however, do not have great impact on the localization performance. As illustrated in Figure 5.9 (b), adequate high-quality proposals are obtained through the LV-based image segmentation.

5.2.2.2 Bounding-box regression

As natural consequences of the multi-scale segmentation, two issues can be observed in Figure 5.9 (b): most proposals are not exactly covering
Chapter 5. General 2D information-based analysis of components

(a) original image  
(b) all positive proposals

(c) local variance at scale 2  
(d) boundaries at scale 2

Figure 5.9: Positive proposals arising from the LV-based image segmentation. The color shift of the bounding boxes from green to red indicates a decline in the localization accuracy.

the underlying components and many components are repeatedly localized across different scales, which leads to a great number of redundant proposals. These problems exist widely in computer vision applications if candidate localization and object recognition are realized in two successive steps. So far, region regression of bounding boxes followed by a Non-Maximum Suppression (NMS) of proposals has been considered as the most practical solution and demonstrated satisfactory performance in general object detection [162] as well as in text spotting [207]. Noticeably, reliable bounding-box regression has been always achieved with the help of DL, especially in the form of CNNs, which is expected to provide a
5.2. Localization

(a) IoU = 0.547  (b) IoU = 0.736  (c) IoU = 0.814  (d) IoU = 0.959

**Figure 5.10:** Iterative bounding-box regression from left to right. The predicted bounding box in yellow solid lines converges to the ground truth in green dashed lines with increased number of iterations.

significant boost to the quality of the analysis on PCB components. A brief depiction of the bounding-box regression and the resulting performance improvement are best visualized with the help of the images in Figure 5.10, where IoU denotes the Intersection-over-Union score.

The most straightforward implementation of bounding-box regression for improving localization accuracy is adopting features arising from intermediate layers of CNNs and feeding them into a linear model for predicting the position and size of the underlying object in the current bounding box. Since only those region proposals containing sought objects are of interest and undergo a categorization/classification step for object recognition purpose later, an additional training of CNN models is usually unnecessary. Pre-trained classification models from the same dataset or from datasets of the same domain can be directly used for extracting the relevant features [162]. If the generation of adequate high-quality proposals is infeasible and an accurate localization is best achievable through region regression of low-quality proposals, more specific CNNs models with adapted architectures become essential, which are explicitly trained on appropriate datasets. However, all of these applied CNNs are closely related to the architectures primarily designed for solving classification problems. How to incorporate layers in CNNs and how to utilize extracted deep features in the sense of region regression are still open questions.

After a close review of the existing architectures and regarding the special properties of DL, a novel architecture arising from a straightforward but very effective extension to state-of-the-art CNNs is proposed for signifi-
cantly improving region regression performance. In consideration of the fact that the deep features and their response maps from lower levels are reused in the final prediction stage, the resulting new CNN is referred to as *Convolutional Neural Network With Recycled Deep Features* (CNN-WRDF) in this thesis. A preliminary description of CNN-WRDF has been previously published in an electronic preprint [9].

To better understand the idea behind a CNN-WRDF, some fundamentals of CNNs are introduced first. In the case of BoVW, as described in Section 5.2.1.2 and visualized in Figure 5.5, manually engineered features are extracted at dense spatial sampling positions and globally pooled to build feature vectors fed into classifiers. Nevertheless, two severe limitations exist in such a classification paradigm: less generalizable learning due to expertise and heuristic-driven feature design, and omitted spatial information. To solve these problems in a unified framework, the primary form of CNNs had been proposed in [208] with the name *Neocognitron*. Rather than applying handcrafted features through transforming raw data (e.g. speech signals, images, videos) using a manually defined representation mechanism, the neural networks are self-organized and automatically learn the most distinctive features from extensive samples shown during training. Another important improvement in comparison to conventional machine-learning models is the convolution-based multi-layer/multi-module architecture. The convolution operation does not only lead to the most dense sampling of features, but also, in combination with the multi-layer structure, to a translation-invariant recognition of the underlying stimuli patterns, while the spatial arrangement of hierarchical features in this bottom-top structure is further maintained. This idea is best clarified with the help of Figure 5.11. For each *Convolutional* (Conv) layer, a nonlinear activation always follows the convolution operation.

A key aspect for achieving successes with CNNs is the selection of an adequate “depth” for neural networks. In general, responses to elementary features in the form of edges and texture emerge in the feature maps extracted at low levels of CNNs. At higher levels close to the top and with reduced spatial resolution, composed features representing motifs and discriminative parts become dominant. The most indicative features for identifying semantic objects are presented at the top level and usually only maintain very coarse spatial information. With the help of reasoning
5.2. Localization

Figure 5.11: Design of CNNs. After multi-layer convolution using the weights shared across neurons of the same layers, the receptive field of the neuron in the top layer (level IV) covers the whole input data. Thus, independent of the position of the sought stimuli pattern in raw data, a corresponding firing of the top neuron is always observed. Since neurons in each layer have a limited “field of view” on the data from its bottom layer, the spatial arrangement of hierarchical features is also maintained. Moreover, small variations in the relative feature position at lower levels are well compensated through the spatial pooling at higher levels and less model parameters are required as the data dimensions are reduced through the simultaneous spatial down-sampling. A further reduction of model parameters can be achieved if greater convolution kernels in (a) are replaced by smaller kernels in (b). For instance, to retain the same “field of view” as in the case of $5 \times 5$ convolution kernel with 25 parameters, two $3 \times 3$ convolution kernels resulting in $3 \times 3 \times 2 = 18$ parameters can be applied and give rise to an additional intermediate layer. For a more clear depiction, all bias parameters are omitted in the diagrams.

layers, e.g. Fully Connected (FC) layers, stacked on the top of Conv layers, classification or regression results can be typically obtained.

As stated in [140], only in combination with feasible training methods, especially with stochastic gradient descent-based back-propagation [209], CNNs can provide an end-to-end solution for solving practical problems. In fact, the back-propagation procedure is straightforward when applied for training feed-forward neural networks. The gradient of the cost function (defining the training objective) with respect to the weights between or in layers can be easily derived using the chain product of derivatives computed from single layers above the corresponding layer and the deriva-
tive of the next layer’s input with respect to the weights. Benefiting from the latest development in parallel computing using Graphics Processing Units (GPU), from the simple but effective Rectified Linear Units (ReLUs) [140] for generating non-linear neuron activation, as well as from the Dropout regularization in training and from the virtual augmentation of sample data variations [210], very deep CNNs have been enabled to bring remarkable improvement in accomplishing many computer vision tasks [158, 162, 210, 211, 212].

In state-of-art CNN architectures for the purpose of region regression, only the topmost feature maps from the embedded bottom-top and fine-coarse feature hierarchy are directly forwarded to the reasoning layers, while the feature maps from lower levels with more accurate spatial information are dismissed. Although those high-level semantic features lead to reliable prediction results, a precise localization is however sacrificed due to two reasons: localization accuracy traded off for reliable feature detection at the top level despite of small variations in the relative feature position at lower levels and the strongly down-sampled feature maps. To better predict the position of the underlying objects in local regions, state-of-the-art CNNs need to explicitly learn a huge number of high-level appearance features encoding the corresponding position information of objects, which are in general non-centered, partially missing, rotated or in a combined form of these factors in cropped images. Even when the required learning is supported by very complex architectures and highly comprehensive training data, the trained models also have difficulties in generalizing region regression on test images not exactly aligned to training samples since variations in appearance are only sampled by the learned parameters in the top Conv layer, where a linear relationship between the resulting responses in the top feature maps and the sought object position is hardly available. Regarding these facts, a reuse of the deep features and their response maps from lower levels becomes desirable.

To utilize these additional feature maps with embedded spatial information without significantly altering the well-established architecture for effective feature extraction, feature responses obtained in the topmost layers of all levels are simultaneously forwarded to the first FC layer. This gives rise to the novel architecture CNN-WRDF recycling deep features and their response maps from low levels, which is a generic extension to state-of-the-art CNNs. This combination also has its origin in the nature perception mechanism: low-level features, e.g. edges and discriminative
5.2. Localization

Figure 5.12: CNN architectures for bounding-box regression. Note that the only feed-forward connection marked with a red arrow from the top layer in the state-of-the-art CNN is replaced by parallel feed-forward identity shortcuts marked with green arrows from the top layers of all levels.

Parts, are commonly required for determining the accurate position, orientation and dimensions of a recognized object. A comparison between a state-of-the-art region-regression CNN and its straightforward extension in the sense of CNN-WRDF is presented in Figure 5.13. In the reasoning layers of a CNN-WRDF, low-level features are correspondingly activated by high-level features for characterizing spatial appearance of the underlying object with a higher resolution. Moreover, combinations of low-level features, e.g. corners of a rectangular PCB component, often result in linear models for predicting the sought bounding box and therefore lead to more generalized region regression.
The realized architecture in Figure 5.12 is similar to some advanced variants of CNNs [212, 213, 214, 215]. However, there are indeed essential differences in the core focuses of these network architectures. In Residual CNNs (Residual CNNs) [212, 213], repeated identity shortcuts between Conv layers are used to forward input data of Conv layer blocks to their output and build, together with the feature maps at the top of the current block, the input to the next layer. By this means, layer blocks are forced to learn residual representations of the expected input to deeper layers. For very deep neural networks aiming at gaining extra performance boost through significantly increased depth, the associated training becomes more feasible since the forwarded input data provide a reference for the parameter learning, which addresses the degradation problem of deep CNNs in comparison to shallower networks. As extended Fully Convolutional Networks (FCNs) [214], U-Net CNNs (U-Net CNNs) [215] for the image segmentation purpose have a more intrinsic connection to CNN-WRDF. To achieve a pixel-wise classification of image elements, it is necessary to generate a full-resolution output map from the most abstracted context information in the top Conv layer of conventional CNNs. Rather than directly working on the top coarse feature maps with inadequate spatial information, more reasonable segmentation performance can be expected if labels of image pixels are determined by combining the coarse context information with the fine low-level representations, which is quite similar to the basic idea of CNN-WRDF. In U-Net CNNs, feature maps extracted at each level of the bottom half of networks are forwarded to the symmetric position in the top half, where they are concatenated to the up-sampled output maps from the next abstracted level along the feature dimension (channels). Subsequently, the corresponding output maps of this level are generated through several successive convolution operations on these combined feature maps. As a result of this down-up-down sampling sequence, full-resolution maps are obtained at the bottom and at the top of networks, while the middle feature maps are with the most coarse spatial resolution. The major difference of CNN-WRDF in comparison to FCNs and U-Net CNNs is the reuse of low-level features as auxiliary information for better localization, while in segmentation high-level representations are auxiliary information for improving the classification rate of pixels associated with low-level feature maps.

To realize an adequate depth for neural networks running on mainstream GPU devices with limited memory capacities, considerations need to be
made in determining the size of cropped images at the bottom of region-regression CNNs. Since dense fine structures exist on PCB components, region proposals of the size $128 \times 128$ are used for bounding-box regression on PCB images and the widely used CNN bottom image size $227 \times 227$ established in [210] or $224 \times 224$ in [211] for general classification purpose is not adopted. Instead of sparse spatial sampling using great convolutional kernels of the size $7 \times 7$ after the input layer, fine features are densely sampled using small convolutional kernels of the size $3 \times 3$. Regarding the desired “field of view” of neurons in the topmost Conv layer, CNNs for bounding-box regression are organized in four levels with four down-sampling layers, so that the entire region proposal is in the receptive fields of those neurons. A demonstrative CNN-WRDF with forward identity shortcuts is depicted in Figure 5.13 for a better understanding. As suggested in [216] with the concept of Network in Network (NIN), each $3 \times 3$ Conv layer is followed by a $1 \times 1$ Conv layer to build a micro network, in which the $1 \times 1$ convolution merely realizes an inter-feature (or inter-channel) permutation in the third dimension of intermediate data and strengthens the ability to approximate arbitrary nonlinear feature functions in combination with the ReLUs. Another important consideration in designing CNNs is the increased number of feature maps (or channels) along with increasing levels. This relies on the observation of limited fundamental features and the dramatically increased diversity of high-level representations. To reflect this fact in CNNs, feature numbers at the third and fourth levels are substantially enlarged to capture the high variability of semantic context extracted through top Conv layers.

Besides the basic architecture illustrated in Figure 5.13, further variants of CNN-WRDF with appropriate extensions are also investigated for improving the performance of bounding-box regression. A first practical extension can be realized by appending additional $1 \times 1$ convolutional and ReLUs layers to each micro network to increase the nonlinearity of features. Another network variant arises as the multi-purpose-learning problem of the recycled features is tackled. In the basic architectures, features from lower levels are employed for both constructing more complex representations at higher levels and describing detailed spatial appearance of the underlying object. There could be a conflict between these two different objectives and compromised features might be learned due to the competition in back-propagation. To enable a decoupling of
Figure 5.13: A simple example of CNN-WRDF. Each convolutional layer (denoted by conv) is followed by rectified linear units. At some layers, sparse sampling or pooling is performed according to the given stride (step size) $X$, which is denoted by $S_X$ in the figure above. The normalized 2D coordinates of the top left and bottom right corners of the predicted bounding box are collected at the top of the network. The red rectangle is used to visualize the predicted bounding box of the transistor.

These two objectives in parameter learning, additional Conv layers can be inserted in the forward shortcuts from low-level feature maps to the first FC layer. This extension leads to new localization features exhibiting a weaker correlation with those features forwarded to higher Conv layers.
5.2. Localization

Considering the decreased reliability of features arising from lower levels, additional regularization can also be integrated into the shortcut paths. In contrast to semantic context, low-level features are sensitive to small variations in images, e.g., illumination, contrast, noise, orientation, occlusion, etc. Consequently, if regression decisions made in FC layers mainly rely on the existence of such unstable features, a decline in the quality of the predicted bounding boxes might be observed. Thus, towards a more reliable prediction, the dropout strategy should be applied, where a random portion of feature responses are set to zeros in each training iteration. The positive impact resulting from this regularization is two-fold: the importance of stable semantic representations is increased and strongly redundant instead of single or few low-level features are employed for activating corresponding neurons in the first FC layer, which are less sensitive to disturbances. Constrained by the reasoning FC layers, the input region proposals must be warped to a fixed size of $128 \times 128$. This image transformation can significantly change the original aspect ratios of objects and give rise to additional ambiguities between distorted appearances. An economic solution is to introduce the original region size into networks and feed it to one of the FC layers. Through this slight feature augmentation, conditioned decisions regarding the original shape information become possible and result in more accurate predictions. It should be emphasized that these additional extensions make CNN-WRDF more differentiable from FCNs and U-Net CNNs.

One of the most important aspects in machine learning is the definition of an appropriate objective function, which governs the success of the backpropagation procedure and the quality of the trained model. Considering the limited size of the training data and the strongly unbalanced instance numbers of different classes (as listed in Table A.2), a class-agnostic bounding-box regression is conducted on PCB images. More specifically, the region-regression CNN generates a unique prediction $b_p$ of the true bounding box $b_t$ for the object found in the current region proposal $b$, instead of generating $K$ predictions for all $K$ possible classes of components. Therefore, the very natural objective function would provide a quantitative measure of the misalignment between $b_p$ and $b_t$, which results in an increased loss value as these two bounding boxes are less aligned. A plausible realization of the desired objective function is the mean squared Euclidean distance between corresponding vertices of bounding boxes. Let $[x_{ul}, y_{ul}]^T$ and $[x_{lr}, y_{lr}]^T$ denote the upper-left and lower-right vertices
of $b$, respectively, the region proposal $b$ is mathematically expressed as $[x_{ul}, y_{ul}, x_{lr}, y_{lr}]^T$. After applying the same model for $b_p$ and $b_t$ with $b_p = [x_{ul,p}, y_{ul,p}, x_{lr,p}, y_{lr,p}]^T$ and $b_t = [x_{ul,t}, y_{ul,t}, x_{lr,t}, y_{lr,t}]^T$, the loss value of the objective function is obtained as the mean squared Euclidean distance between the two upper-left as well as between the two lower-right vertices of $b_p$ and $b_t$. However, from a practical point of view, this absolute loss value could lead to inconveniences in training since a wide value range arises from the absolute vertex distance. To fit the distribution of the loss value closer to the $\chi^2$ distribution comprising independent squared standard normal variables with zero mean and unit variance, for which practical training techniques and parameter configurations are well-established, $b_p$ and $b_t$ must undergo a uniform value normalization as a pre-processing step. Let $\tilde{b}_p$ and $\tilde{b}_t$ denote the predicted and true bounding boxes after normalization, respectively, their model parameters $[\tilde{x}_{ul,p}, \tilde{y}_{ul,p}, \tilde{x}_{lr,p}, \tilde{y}_{lr,p}]^T$ and $[\tilde{x}_{ul,t}, \tilde{y}_{ul,t}, \tilde{x}_{lr,t}, \tilde{y}_{lr,t}]^T$ are obtained using the following equations:

$$
\tilde{x}_{ul/lr,p/t} = \frac{2 \cdot x_{ul/lr,p/t} - (x_{ul} + x_{lr})}{2 \cdot (x_{lr} - x_{ul})}, \\
\tilde{y}_{ul/lr,p/t} = \frac{2 \cdot y_{ul/lr,p/t} - (y_{ul} + y_{lr})}{2 \cdot (y_{lr} - y_{ul})}.
$$

(5.12)

The normalized loss value $E_{loss}$ of the objective function is thus

$$
E_{loss} = \frac{1}{2 \cdot N} \sum_{n=1}^{N} \| \tilde{b}_{n,p} - \tilde{b}_{n,t} \|_2^2,
$$

(5.13)

where $n$ denotes the $n$-th of in total $N$ samples used in a learning iteration. Once $\tilde{b}_p$ has been predicted through the region-regression CNN, the absolute bounding box $b_p$ can be restored according to

$$
x_{ul/lr,p} = \tilde{x}_{ul/lr,p} \cdot (x_{lr} - x_{ul}) + \frac{x_{ul} + x_{lr}}{2}, \\
y_{ul/lr,p} = \tilde{y}_{ul/lr,p} \cdot (y_{lr} - y_{ul}) + \frac{y_{ul} + y_{lr}}{2}.
$$

(5.14)

Even when applying the adjusted architecture of CNNs and using the normalized objective function, the desired bounding-box regression may still fail. The reason can be best clarified with the help of the image in
5.2. Localization

(a) aug. size 0  (b) aug. size 1  (c) aug. size 2  (d) aug. size 4

**Figure 5.14**: Augmentation of a region proposal. The augmentation size is normalized with respect to the area of the original region proposal.

Figure 5.14 (a). For such a region proposal without visible peripheral regions of the central component, it is difficult to determine if it is an object of interest and to what extent the corresponding bounding box should be expanded. As demonstrated in Figure 5.14 (b) and (c), these ambiguities are removed as soon as this region proposal has been appropriately augmented. If the proposal is over augmented as Figure 5.14 (d), the desired region regression becomes less feasible as neighboring components also emerge in the image and may prevent the bounding box from converging onto the central component. Selecting an appropriate augmentation size is thus necessary and needs to be investigated later in evaluation.

### 5.2.2.3 Region merging and validation

A dramatic side effect of the diversification-driven (Section 5.2.1.1) and the multi-level LV analysis-based (Section 5.2.2.1) proposal generation are multiply localized PCB components. As a result, redundant proposals of the same objects are presented. This undesired effect cannot be well suppressed in the case of the combinatory localization approach in Section 5.2.1, regarding the strongly differentiated proposals emerging from diverse algorithms, scales and parameter settings, as well as the conventional features and classifiers with limited discrimination power. On the contrary, this issue can be solved in an elegant fashion using the framework of the compact approach. After bounding-box regression, as demonstrated in Figure 5.10, proposals of the same objects converge to very similar bounding boxes, among which only slight differentiation is observed. Consequently, these spatially strongly overlapping bounding boxes can be merged through an NMS operation with respect to an ap-
appropriately defined quality measure of region proposals. If the regression model is merely trained using positive samples, region proposals exhibiting adequate overlap with ground-truth objects are expected to move to and remain on the underlying components, while the remaining negative proposals will probably concentrate on certain subregions of background. Therefore, transitional proposals between sought objects and false alarms can be significantly reduced and margins between positive and negative proposals are correspondingly enlarged. This has very positive impact on the subsequent validation of regressed region proposals, where false alarms are rejected with the help of a pre-trained binary classifier.

In case of given ground-truth bounding boxes, the preferred quality measure for guiding NMS-based region merging is the Intersection-over-Union (IoU) score. For a region proposal $b_p$ and the underlying true bounding box $b_t$, the corresponding IoU score is defined as

$$\text{IoU} = \frac{b_p \cap b_t}{b_p \cup b_t},$$

where $\cap$ denotes the area of overlap between two regions and $\cup$ denotes the area of union. For two predicted bounding boxes with significant overlap, the one with lower IoU score is removed, while the other survives as the merged region proposal. In practical localization without ground-truth data, a synthesized IoU score, i.e., the Post-regression-overlapping-Pre-regression (PoP) score [9], can be used in NMS. Formally, the PoP score is expressed as

$$\text{PoP} = \frac{b_p \cap b}{b_p \cup b},$$

where $b$ and $b_p$ denote the region proposals before and after bounding-box regression, respectively. It is assumed that region proposals close to sought objects converge to ground-truth bounding boxes and the resulting better overlap with sought objects improves the localization accuracy after region regression. Therefore, it is a plausible assertion that higher PoP score indicates a better original proposal $b$ and thus a more accurately predicted bounding box $b_p$. Moreover, if region proposals are insufficient for localizing the underlying components or located completely on background, significant changes can usually be observed between bounding boxes before and after region regression since the predictions diverge from the original proposals and seek more reasonable regions. Through an
iterative bounding-box regression, these changes become more traceable and the PoP score can even be used for effectively rejecting false alarms.

Although bounding-box regression leads to a reduced diversity of the remaining region proposals, it is still non-trivial to select appropriate features and classifiers for distinguishing components from disturbances, which arise from highly complex background of PCB images. Considering the recent successes of DL in complicated classification tasks [140,210] and the efficiency of CNNs, a binary CNN classifier is trained for conducting the final validation step and only retaining the true region proposals for the subsequent component categorization.

5.3 Categorization

In the categorization step of the overall detection, objects localized through the validated bounding boxes are to be classified into corresponding categories of PCB components. After bounding-box regression and cancellation of false alarms, the remaining objects exhibit, on contrary to the original region proposals, only limited variations in shape, color and texture, while the variation in size is compensated by warping cropped images to a fixed size. Regarding the reduced complexity of the categorization task and the resulting decline in number of distinct training samples, conventional features and classifiers may become superior to DL. To investigate the performance of diverse classification methods and to determine the optimal solution for PCB components, a comprehensive evaluation has been conducted in a Master’s thesis [217]. To provide a structured view of this performance investigation, all involved features, fundamental classifiers and advanced learning methods are listed in Appendix C.

5.4 Evaluation

Due to the different objectives of localization and categorization, these two steps in analyzing general PCB components have been evaluated on corresponding datasets using individual performance measures. To provide a better overview of the obtained evaluation results, these two consecutive evaluation procedures follow a sequential presentation in this section.
5.4.1 Localization

5.4.1.1 Data and performance measures

As described in Section 5.2, the combinatory and compact localization approaches strongly rely on machine-learning methods, where extensive training data are required. However, the available dataset “PCB General” is of a limited size and inadequate for training the desired models for bounding-box regression and proposal validation. To deal with this problem, all available training data have been first augmented using three complementary strategies, which can effectively result in substantially enlarged datasets of sufficient differentiation.

Although the sought components in images of the dataset remain the same, different algorithms for generating region proposals result generally in different candidates. This diversity arising from multiple proposing algorithms can be utilized and realizes an effective training data augmentation. A further data augmentation strategy relies on the variation in the region size inspired by the consideration depicted in Figure 5.14. Regarding the symmetric structures commonly observed on PCB components, another data augmentation through geometric transformations is also applied. A detailed illustration of the employed geometric transformations is presented in Figure 5.15. This augmentation is able to generalize regression and validation models since possible variations in the orientation of components are introduced into the training processes.

Following the widely adopted protocols [153, 155] for assessing the performance of object detection, the IoU score is employed for quantifying
5.4. Evaluation

the quality of localized regions proposals. Moreover, true proposals are distinguished from false alarms using the threshold $\text{IoU} = 0.5$. Consequently, the performance measures precision, recall and F-score defined in Eq. 4.6 can be applied on the binarized region proposals. Similar to the measure $\text{Area under the Receiver-Operating-Characteristic Curve (AUC)}$ for a global assessment of the discrimination power for distinguishing between positive and negative samples, the $\text{Average Precision (AP)}$ measure is applied for assessing localization results, where its value is computed as the area under the $\text{Precision-Recall Curve (PRC)}$. This measure is better suited in the sense of detection: only sought objects are of interest and the true negative proposals are not uniquely defined for employed proposing methods. A further consideration about the multiply localized PCB components emerging from the diversification-driven proposal generation is necessary since the common one-to-one matching between proposals and ground-truth objects omits redundant true region proposals. Bearing this in mind, additional measures $\text{N-to-one precision (N-precision)}$ and $\text{N-to-one F-score (N-F-score)}$ are introduced, where the matching between $N$ true proposals and a single ground-truth object is allowed. Apparently, these measures give a better view of the quality of the overall proposals arising from diverse sources, e.g. multiple pixel-grouping algorithms, multiple scales, etc.

5.4.1.2 Combinatory approach

To determine the best combination between multiple algorithms, scales and parameter settings for the diversification-driven proposal generation, comparisons between variations in each of the four aspects have been conducted at first. Originally, there were in total six pixel-grouping algorithms considered: Active Contours, Graph Cuts, Efficient Graph, Mean Shift, CTM and TBES. An optional analysis algorithm for background estimation is also available and is expected to result in improved segmentation results. Through a preliminary qualitative evaluation, it has been confirmed that Active Contours, CTM and TBES are far from suited for the proposal generation in PCB images due to their extremely high computational burden and difficulties in simultaneously segmenting small and huge components from background. Therefore, only Graph Cuts, Efficient Graph and Mean Shift as well as the background estimation algorithm are available for further investigation. A joint evaluation of these four algo-
Chapter 5. General 2D information-based analysis of components

<table>
<thead>
<tr>
<th>algorithm</th>
<th>Graph Cuts</th>
<th>Efficient Graph</th>
<th>Mean Shift</th>
</tr>
</thead>
<tbody>
<tr>
<td>background est.</td>
<td>−</td>
<td>+</td>
<td>−</td>
</tr>
<tr>
<td>$N$-precision</td>
<td>0.076</td>
<td>0.178</td>
<td>0.069</td>
</tr>
<tr>
<td>recall</td>
<td>0.441</td>
<td>0.620</td>
<td>0.454</td>
</tr>
</tbody>
</table>

Table 5.1: Performance of single pixel-grouping algorithms. Symbols + and − indicate the pixel grouping algorithms with and without background estimation, respectively.

The first investigation of the diversification strategy is conducted with respect to the configurations of scale and parameter setting. As listed in Table 5.2, single and multiple configurations are applied for these two factors alternatively. Instead of $N$-precision, the average number of proposals per image is considered and quantifies the resulting computational burden for the proposal validation step. From the viewpoint of efficiency, Graph Cuts using single scale and multiple parameter settings is preferred due to the lower number of proposals. If a more comprehensive localization is summarized in Table 5.1 for a better overview. Apparently, the additional background estimation leads to an overall improvement of localization results. It is thus considered as a necessary pre-processing step for all following evaluation procedures of the combinatorial approach.

<table>
<thead>
<tr>
<th>algorithm</th>
<th>configurations</th>
<th>recall</th>
<th>#(proposals) per image</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graph Cuts</td>
<td>single scale</td>
<td>0.711</td>
<td>159</td>
</tr>
<tr>
<td></td>
<td>parameters</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Efficient Graph</td>
<td>single scale</td>
<td>0.673</td>
<td>156</td>
</tr>
<tr>
<td></td>
<td>parameters</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean Shift</td>
<td>single scale</td>
<td>0.871</td>
<td>649</td>
</tr>
<tr>
<td></td>
<td>parameters</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>parameter scales</td>
<td>0.887</td>
<td>660</td>
</tr>
</tbody>
</table>

Table 5.2: Performance of single pixel-grouping algorithms using different configurations of scale and parameter setting.
5.4. Evaluation

<table>
<thead>
<tr>
<th>algorithm</th>
<th>combination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graph Cuts</td>
<td>×</td>
</tr>
<tr>
<td>Efficient Graph</td>
<td>×</td>
</tr>
<tr>
<td>Mean Shift</td>
<td>×</td>
</tr>
<tr>
<td>recall</td>
<td>0.739</td>
</tr>
</tbody>
</table>

Table 5.3: Performance of combined pixel-grouping algorithms using multiple scales and parameter settings.

For the continued investigation of the diversification strategy, single pixel-grouping algorithms using multiple scales and parameter settings are combined with each other. This evaluation also reveals the power of diverse pixel-grouping algorithms since all possible proposals are exploited through the comprehensive combinations between scales and parameter settings. It is noticeable that no significant improvement can be observed in Table 5.3 after further increasing the diversity in configuration and algorithm. As the combination with the best trade-off between efficiency and proposal generation, Efficient Graph using single parameter setting and multiple scales in Table 5.2 is further employed for the following evaluation of region validation. There are in total 28,731 region proposals to be validated, which comprise 3,890 positive and 24,841 negative samples. After an appropriate data augmentation and a subsequent NMS operation on false alarms, 61,608 region proposals consisting of 27,296 positive and 34,312 negative samples are available for training the corresponding region validation classifiers.

As stated in Section 5.2.1.2, HOG and BoVW features are combined with RF and SVM classifiers for the purpose of distinguishing between true and false region proposals. Although FV features may lead to better classification performance than BoVW features [182], they are however unsuited for the associated training processes due to the resulting high dimensional feature vectors and the huge data size, as well as the long processing time. Therefore, they are excluded from this validation step and only adopted for the purpose of categorizing PCB components after
localizing objects of interest, where the number of test samples is substantially reduced and the computational burden becomes more affordable. To isolate the training and test processes, all 31 images in the dataset “PCB General” are divided into five subsets of similar sizes (six or seven images in each subset), on which a fold-wise cross-validation is performed: each subset is used for testing the classifier trained on the data formed by the other four subsets and the average classification performance is considered as the final performance measure. Regarding the objective of object localization, \( N \)-F-score is used as the quality measure of the trained models and governs the optimization of classifiers, where a high recall value is enforced by setting \( \beta \) in Eq. 4.6 to 2 during training and to 1 during test. If not stated otherwise, \( N \)-precision, recall and \( N \)-F-score are always evaluated on the original region proposals for computational convenience, where recall = 1 means the correct classification of all true region proposals in the localization data. To characterize the significance of the obtained classification results with respect to random guessing, Correct Classification Rate (CCR) is also provided and it should take the value 0.5 for binary classification results arising from the random guessing based on the uniform distribution:

\[
CCR = \frac{\#(\{x|\hat{y}(x) = y(x)\})}{\sum_{c=1}^{C} \#(\{x|\hat{y}(x) = c\})}, \tag{5.17}
\]

where \( \hat{y}(x) \) and \( y(x) \) denote the predicted and the ground-truth class label of the test sample represented by the feature vector \( x \), respectively. Moreover, the class label \( c \) denotes the \( c \)-th of all \( C \) classes.

As a result of the dense grid structure for extracting HOG features and the large vocabulary size of BoVW models for achieving the desired discrimination power, the obtained representations of region proposals are high-dimensional feature vectors and lead to difficulties in training validation models. To reduce the dimensions of feature data while preserving the most information, Principal Component Analysis (PCA) is applied which transforms the feature data into a low-dimensional space spanned by their principal components representing the major variability. Different numbers of principal components have been evaluated individually. 40 as the lower boundary for reasonable classification performance and 200 as the upper boundary for significant performance improvement are selected for demonstrating the influence of dimensionality reduction.
In Table 5.4, the quality of RF classifiers trained using different features and optimization options is summarized. In comparison to HOG features, BoVW features provide a significant performance boost for all models arising from different optimization processes. For both feature groups, the performance variation is limited as the dimension of the feature vectors varies. This confirms the appropriate application of PCA for dimensionality reduction. In training classification models, superior performance can in general be expected if the model optimization is conducted on separated training and test datasets, due to the less biased evaluation results with respect to the training samples. However, this effect is not observed in the case of RF classifiers, no matter if the models during optimization are validated on separated test data or on the same training data. This in fact reflects the most favorable property of RFs: they are resistant to overfitting, which is a consequence of using the Bagging algorithm and the random subspace method in ensemble learning. To determine the two most important model parameters, the number of DTrees and the dimension of the subspaces considered for decision splits, a parameter optimization is realized with the help of a grid search on coarse samples or the fine-tuning genetic algorithm (GA). It is interesting to note that both algorithms lead to very similar classification results. Thus, a rough parameter tuning using the grid search is sufficient for optimizing RFs, which is in general less time consuming than a fine-tuning optimization.

Similar to the case of RF classifiers, the region validation performance of SVM classifiers is also evaluated with respect to different features and optimization options. An overview of the evaluation results is presented in Table 5.5. It should be emphasized that all SVM classifiers are using radial kernel functions as linear SVMs were found uncompetitive compared with nonlinear variants through a preliminary test. Again, BoVW features lead to substantially improved classification performance in comparison to HOG features. Although $N$-F-score becomes less stable if the dimension of feature vectors varies, the overall performance variations still remain insignificant if the cross-validation is involved. The most noticeable difference to Table 5.4 is the great impact of the cross-validation-based training on the quality of trained SVM models. If training and validation data are the same during model optimization, the resulting classification results are often far from reasonable. Thus, cross-validation procedures are essential for optimizing SVM classifiers properly and can avoid possible overfitting. In case of a clean separation between training and validation data, a rough
grid search and a fine parameter tuning lead to comparable region validation results, where slightly superior performance is observed for models optimized through a grid search.

Since each true region proposal in localization data is considered as independent from each other, the obtained recall and N-F-score values might be different from the values obtained on ground-truth components in PCB images, where each component is counted exactly only once. Bearing this in mind, N-precision, recall and N-F-score are computed again on the ground truth of the dataset “PCB General”. For the best region validation classifier using the feature BoVW-200 and the SVM model optimized through a grid search combined with cross-validation procedures, the corresponding values of the three quality measures become N-precision = 0.775, recall = 0.720 and N-F-score = 0.746. Finally, to summarize the combinatory approach, the overall localization performance after different stages is depicted in Figure 5.16 and the incremental improvement is best visualized. To suppress redundant region proposals arising from multiply localized PCB components, an NMS operation is performed on all overlapping proposals with an IoU score more than 0.5. The final one-to-one matching performance is obtained as precision = 0.626, recall = 0.658 and F-score = 0.637. Apparently, the combinatory approach is able to support a binary decision on the existence of sought objects according to N-to-one matching and a quantitative analysis based on one-to-one matching is, however, less realistic.

5.4.1.3 Compact approach

As described in Section 5.2.2, the compact localization approach starts with the LV-based segmentation, which generates comprehensive region candidates of PCB components. To assess the performance of the proposed segmentation algorithm, it is compared to state-of-the-art proposal generation algorithms in terms of N-precision, recall and # (proposals) per image, where the last two measures are more important since recall determines the sensitivity of the localization to underlying objects and # (proposals) per image is associated with computational burden, while N-precision is dependent on the subsequent processing steps, viz. region
<table>
<thead>
<tr>
<th>option</th>
<th>HOG-40</th>
<th>HOG-200</th>
</tr>
</thead>
<tbody>
<tr>
<td>CV training</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>grid search</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>GA optimization</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>mean N-precision</td>
<td>0.493</td>
<td>0.454</td>
</tr>
<tr>
<td>mean recall</td>
<td>0.698</td>
<td>0.715</td>
</tr>
<tr>
<td>mean N-F-score</td>
<td>0.577</td>
<td>0.554</td>
</tr>
<tr>
<td>mean CCR</td>
<td>0.863</td>
<td>0.846</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>option</th>
<th>BoVW-40</th>
<th>BoVW-200</th>
</tr>
</thead>
<tbody>
<tr>
<td>CV training</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>grid search</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>GA optimization</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>mean N-precision</td>
<td>0.615</td>
<td>0.616</td>
</tr>
<tr>
<td>mean recall</td>
<td>0.803</td>
<td>0.800</td>
</tr>
<tr>
<td>mean N-F-score</td>
<td>0.696</td>
<td>0.696</td>
</tr>
<tr>
<td>mean CCR</td>
<td>0.907</td>
<td>0.907</td>
</tr>
</tbody>
</table>

**Table 5.4**: Performance of RF classifiers in region validation. Feature vectors of different dimensions (40 or 200) are differentiated from each other. The optimal parameters of RFs are determined through either a grid search or an optimization process using the genetic algorithm (GA), where the quality of the currently trained models are optionally asserted through an additional cross-validation on the training data.
TABLE 5.3: Performance of SVM classifiers in region validation. Feature vectors of different dimensions (40 or 200) are differentiated from each other. The optimal parameters of SVMs are determined through either a grid search or an optimization process using the genetic algorithm (GA) combined with the Nelder-Mead simplex algorithm, where the quality of the currently trained model is optionally asserted through an additional cross-validation on the training data.

<table>
<thead>
<tr>
<th></th>
<th>mean CCR</th>
<th>mean N-P-score</th>
<th>mean recall</th>
<th>mean N-F-score</th>
<th>mean CCR</th>
<th>mean N-P-score</th>
<th>mean recall</th>
<th>mean N-F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>option HOG-40</td>
<td>0.965</td>
<td>0.774</td>
<td>0.232</td>
<td>0.723</td>
<td>0.968</td>
<td>0.777</td>
<td>0.231</td>
<td>0.723</td>
</tr>
<tr>
<td>option BoVW-40</td>
<td>0.896</td>
<td>0.747</td>
<td>0.124</td>
<td>0.662</td>
<td>0.898</td>
<td>0.748</td>
<td>0.125</td>
<td>0.663</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>mean CCR</th>
<th>mean N-P-score</th>
<th>mean recall</th>
<th>mean N-F-score</th>
<th>mean CCR</th>
<th>mean N-P-score</th>
<th>mean recall</th>
<th>mean N-F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>option HOG-200</td>
<td>0.965</td>
<td>0.774</td>
<td>0.232</td>
<td>0.723</td>
<td>0.968</td>
<td>0.777</td>
<td>0.231</td>
<td>0.723</td>
</tr>
<tr>
<td>option BoVW-200</td>
<td>0.896</td>
<td>0.747</td>
<td>0.124</td>
<td>0.662</td>
<td>0.898</td>
<td>0.748</td>
<td>0.125</td>
<td>0.663</td>
</tr>
</tbody>
</table>
5.4. Evaluation

Regression, merging and validation, and can be substantially improved. All results are summarized in Table 5.6 for a better overview.

First of all, LV analysis using Eq. 5.6 leads to more comprehensive region proposals than using Eq. 5.4. This implies that an additional smoothing of LV is unnecessary and homogeneity information is best captured by comparing pixels in a local window with the pixel at the central position. Among the four state-of-art algorithms, Selective Search is able to localize the most components with a huge number of candidates, while Efficient Graph results in the least number of candidates with good localization performance. BING and Edge Boxes are unable to provide reasonable ratios between recall and #(proposals) per image. To gain a deeper insight into the potential of the proposed LV analysis, it is combined with different segmentation algorithms for achieving the desired proposal generation. If the proposed LV-based segmentation introduced in Section 5.2.2.1 is applied, promising results are obtained. Moreover, this segmentation algorithm does not rely on any parameters to be tuned and is thus an out-of-the-box solution for localization tasks. In the case of Efficient Graph and Selective Search-based segmentation, most components are localized using only single color space, while the original algorithms require information from multiple color spaces for achieving comparable results. This further confirms the reliable localization performance arising from LV analysis and the reduced dependence on parameter tuning. For further analysis, region candidates are obtained by applying the adaptive thresholding on

![Figure 5.16](image-url): Overall localization performance of the combinatory approach with incremental improvement after different steps.
Chapter 5. General 2D information-based analysis of components

<table>
<thead>
<tr>
<th>algorithm</th>
<th>color space</th>
<th>N-precision</th>
<th>recall</th>
<th>#(proposals)per image</th>
</tr>
</thead>
<tbody>
<tr>
<td>BING</td>
<td>single</td>
<td>0.005</td>
<td>0.248</td>
<td>3 074</td>
</tr>
<tr>
<td>Edge Boxes</td>
<td>single</td>
<td>0.033</td>
<td>0.878</td>
<td>9 600</td>
</tr>
<tr>
<td>Efficient Graph</td>
<td>single</td>
<td>0.110</td>
<td>0.835</td>
<td>1 399</td>
</tr>
<tr>
<td></td>
<td>multiple</td>
<td>0.055</td>
<td>0.962</td>
<td>5 136</td>
</tr>
<tr>
<td>Selective Search</td>
<td>single</td>
<td>0.104</td>
<td>0.870</td>
<td>2 931</td>
</tr>
<tr>
<td></td>
<td>multiple</td>
<td>0.068</td>
<td>0.990</td>
<td>12 464</td>
</tr>
<tr>
<td>segmentation based on LV (Eq. 5.4)</td>
<td>single</td>
<td>0.047</td>
<td>0.836</td>
<td>2 283</td>
</tr>
<tr>
<td></td>
<td>multiple</td>
<td>0.042</td>
<td>0.928</td>
<td>4 096</td>
</tr>
<tr>
<td>segmentation based on LV (Eq. 5.6)</td>
<td>single</td>
<td>0.053</td>
<td>0.897</td>
<td>2 617</td>
</tr>
<tr>
<td></td>
<td>multiple</td>
<td>0.047</td>
<td>0.951</td>
<td>4 587</td>
</tr>
<tr>
<td>LV (Eq. 5.6) + Efficient Graph</td>
<td>single</td>
<td>0.045</td>
<td>0.964</td>
<td>5 367</td>
</tr>
<tr>
<td></td>
<td>multiple</td>
<td>0.035</td>
<td>0.987</td>
<td>10 589</td>
</tr>
<tr>
<td>LV (Eq. 5.6) + Selective Search</td>
<td>single</td>
<td>0.079</td>
<td>0.959</td>
<td>3 990</td>
</tr>
<tr>
<td></td>
<td>multiple</td>
<td>0.072</td>
<td>0.984</td>
<td>9 428</td>
</tr>
</tbody>
</table>

Table 5.6: Comparison between proposal generation algorithms.

LV images using single color space. This configuration provides less biased evaluation results since no parameter tuning is adopted.

In the following evaluation procedures, images from the dataset “PCB General” are divided into two sets: 25 images for training and the remaining six images for test. Due to the time-consuming data handling and the extremely long training time of CNN models, a cross-validation is discarded. To provide a fair comparison to the combinatory approach, its region validation performance is evaluated again using this training-test data configuration. To train models for bounding-box regression, all region candidates with a IoU exceeding 0.5 with respect to an arbitrary ground-truth object are considered. After an appropriate data augmentation, there are in total 358 704 training samples available. To reduce the variation in data values and to enable the general training rules (e.g. learning rate decay, weight decay for the regularization purpose, etc.), all training samples need to undergo the data standardization, where the standardized pixel values at the same position have zero mean and are
Figure 5.17: Test loss of different CNN architectures for bounding-box regression. A comparison between state-of-the-art CNNs and the proposed CNN-WRDF is depicted in (a). A further comparison for investigating the performance of different model extensions to CNN-WRDF is presented in (b).

Evaluation results of different bounding-box regression models are illustrated in Figure 5.17. The first comparison is between state-of-the-art CNNs and the proposed CNNs-WRDF sharing the same major architecture, as well as two basic model extensions: “s” and “d” denote additional size information and randomly dropping out recycled features, respectively. In Figure 5.17 (a), the most straightforward conclusion is that the normalized to have their standard deviation equal to one. It is worth mentioning that the zero-mean pixel values at each position are realized by subtracting their global mean value and the resulting global mean image could introduce additional shape information since diverse PCB components have different features localized at different positions in images.
proposed CNN architecture with recycled deep features is able to significantly reduce test loss, as represented by the squared Euclidean distance between the vertices of predicted and ground-truth bounding boxes, and thus leads to improved localization accuracy. It is also interesting to see that the additional size information (denoted as “CNN + s” and “CNN-WRDF + ds”) further boosts the region regression performance. Furthermore, as predicted in Section 5.2.2.2, disabling a random fraction of the recycled features (denoted as “CNN-WRDF + ds”) in each training iteration enforces the use of redundant information and subsequently increases the quality of the regressed bounding boxes. Nevertheless, if additional feature drop out is applied in the standard CNN model (denoted as “CNN + ds”), the overall regression performance decreases substantially.

A further comparison depicted in Figure 5.17 (b) is conducted between different extensions to the fundamental CNN-WRDF to better exploit the potential of this novel architecture. Besides “s” and “d”, the additional symbol “2” denotes the extension of micro networks into the original form of NIN [216] with two successive $1 \times 1$ convolutional + ReLUs layer blocks, while the other symbol “D” indicates CNNs-WRDF with the recycled features decoupled from those features forwarded to deeper convolutional layers. As confirmed through the evaluation results, the enhanced nonlinearity of extracted representations through the extended micro networks (denoted as “CNN-WRDF + 2d”) alone does not improve the performance of region regression models. Only in combination with the decoupling of features (denoted as “CNN-WRDF + 2Dd”), the advantage of additional $1 \times 1$ convolutional + ReLUs layers is observed and comparable results to “CNN-WRDF + ds” are achievable even without additional size information. As a reference model, the 50-layer ResNet [212] is trained with an additional FC reasoning layer. However, the corresponding test loss does not converge below 0.04 and no benefit is thus obtained using much deeper networks. Aiming at a fair comparison, the first FC reasoning layer of the state-of-the-art CNN architecture “CNN + s” is also augmented with additional neurons to compensate the additional connections in CNNs-WRDF arising from the recycled feature maps. The obtained new CNN has in total 443,588,608 connections in comparison to the original CNN with 392,384,608 connections before the augmentation, while the proposed “CNN-WRDF + ds” has in total 427,853,824 connections. However, simply increasing the complexity of the network does not lead to better regression performance and the resulting test loss is very
5.4. Evaluation

similar to the case without additional neurons. This further confirms the advantage of the proposed CNN architecture. According to the comprehensive comparison presented above, the two networks “CNN-WRDF + ds” and “CNN-WRDF + 2Dd” are found superior to other candidates and should be adopted for improving the localization accuracy. Due to its smaller model size, “CNN-WRDF + ds” is considered as the reference model for further evaluation.

After conducting bounding-box regression on training data using different region augmentation sizes and different iteration numbers, 0.3 and 5 are found as the best values for augmenting region proposals and for obtaining converged bounding boxes, respectively. Larger augmentation size leads to unstable regression performance and more iterations do not provide significant further improvement. To demonstrate the power of the proposed bounding-box regression, comparisons between original and predicted region proposals using different models are depicted in Figure 5.18. For each ground-truth component, its best IoU score is computed on the original (green bars) and on the predicted (blue bars) region proposals, respectively. For any IoU score on the horizontal axis, all ground-truth components with their best IoU score below this value are summed up and the quantity of these ground-truth components is represented by the height of the bar plotted at the corresponding position. For a better comparison, all quantity values are normalized with respect to the total number of ground-truth components and fall between 0.0 and 1.0. With the help of the widely adopted threshold IoU = 0.5 for determining valid proposals [153,155], it is clarified that the recall value after bounding-box regression becomes very close to 1.000 (1.000 − 0.025 = 0.975 for the state-of-the-art CNN and 1.000 − 0.008 = 0.992 for the proposed CNN-WRDF) even when the original recall value was about 1.000 − 0.103 = 0.897. Superior regression performance is observed for the proposed CNN-WRDF architecture and more than 40% of the sought objects have their best IoU score over 0.9, while this value drops to less than 15% if the state-of-the-art CNN architecture is adopted.

Before feeding the predicted bounding boxes to the final validation CNN, most false alarms and redundant proposals can be canceled using the PoP score combined with an NMS operation, where the PoP score is utilized as the quality measure in merging spatially overlapping region proposals. More specifically, all bounding boxes with a low PoP score are removed
Figure 5.18: Improvement of the best IoU score through bounding-box regression. In each of the above plots, ground-truth objects are cumulatively summed up over their best IoU score from 0.0 to 1.0. The dashed vertical line in red marks the widely adopted threshold IoU = 0.5 for determining valid proposals [153, 155]. The upper dashed horizontal line in red marks the number of ground-truth objects for which there is no valid region proposal found in the original bounding boxes. The lower red line marks the corresponding value for the predicted bounding boxes after region regression.

and all bounding boxes with significant overlap are merged according to their PoP scores. To preserve the high recall value while substantially reducing the number of candidates to be validated, PoP = 0.85 and IoU = 0.85.
0.6 are found as the best values on training data for canceling false alarms and merging redundant bounding boxes.

For the purpose of region validation, the state-of-the-art and the proposed CNN architecture are considered again. After applying the corresponding classification models in validating remaining region proposals, the final localization results are obtained. A comprehensive evaluation of the compact localization approach is achieved by investigating diverse combinations between proposal generation algorithms, bounding-box regression models and region validation models. For a clear overview of the comparison regarding the overall localization performance, some representative combinations are selected for the comparative depiction in Figure 5.19. The localization results obtained using the combinatory approach are also visualized in these figures and are considered as the baseline solution. A quantitative assessment of the obtained localization is achieved with the help of the AP (average precision) and F-score.

As demonstrated in the precision-recall plots, the compact approach results generally in higher precision and recall values than the combinatory approach and thus gives rise to higher AP and F-score values. Moreover, the proposed architecture CNN-WRDF is superior to the state-of-the-art CNN architecture in region regression, but not in region validation. This effect is well explained through the mechanism of the proposed CNN architecture: the forwarded feature maps from lower levels should contribute to the improvement in localization accuracy, while the discrimination power of classification CNNs mainly depends on the quality of the top semantic feature maps. As a reasonable consequence, no significant variation can be observed in the region validation performance if the same major architecture is shared between the employed CNN and CNN-WRDF classifiers. Besides precision-recall curves, F-score of different combinations is also plotted over the underlying threshold of the confidence score, which is used to distinguish true region proposals from false alarms in inference. Apparently, the margin between true and false region proposals is enlarged in the case of the compact approach and the localization performance becomes less sensitive to the selected threshold of the confidence score. A straightforward comparison between different combinations is presented in the last plot of Figure 5.19, where the AP and the maximum F-score are employed as the performance measures. With AP = 0.704 and the maximum F-score = 0.715, the combinatory approach is unable
to provide comparable localization results to the compact approach. The highest AP value is achieved by combining the LV-based thresholding with the CNN-WRDF-based bounding-box regression as well as region validation, where $AP = 0.915$ and the maximum F-score $= 0.882$. If higher and more stable F-score is required, the state-of-the-art CNN architecture should be adopted for the region validation purpose, where $AP = 0.903$ and the maximum F-score $= 0.895$. Regarding the least number of pro-

Figure 5.19: Overall localization performance of the compact approach.
posals provided by Efficient Graph in Table 5.6 and the resulting good localization performance (AP = 0.877 and the maximum F-score = 0.872) using the compact approach, the alternative combination relying on Efficient Graph and CNN-WRDF is considerable if better efficiency is required. Although LV + Selective Search and LV + Efficient Graph lead to higher recall values in Table 5.6, the simultaneously increased number of proposals, however, triggers more false alarms on background regions and thus decreases the overall localization performance. A visualization of the final localization results is provided in Figure 5.20, where the remaining region proposals in a demo PCB image are marked with yellow rectangles. A wide margin between sought components and disturbances can be confirmed through the confidence score assigned to each proposal.

5.4.2 Categorization

5.4.2.1 Data and performance measures

As stated in Table A.2, there are in total 1224 samples of PCB components available in the dataset “PCB General”. Due to the bounding-box regression in localization, region proposals arising from different algorithms, scales and parameter configurations become very similar and exhibit limited variability in shape, size, orientation, background, etc. As a result, training classifiers for the intended categorization becomes hardly feasible. To overcome this difficulty, all ground-truth samples have to undergo a data augmentation process, which gives rise to a huge number of new component images and thus supports the training of categorization classifiers. Considering the characteristics of region proposals fed into the categorization process, strategies other than those in the case of localization are employed to generate synthetic samples. With the help of the ground-truth masks, all components are precisely separated from the background in the original images and can be blended into any new background images, which are either synthetic or real. Additional rotation, scaling and noise are then applied on the obtained synthetic images to introduce further realistic variations. Some demo training samples originating from the same ground-truth component are depicted in Figure 5.21 for a better understanding. To avoid overfitting and to estimate the performance of categorization classifiers in real application, each training
Chapter 5. General 2D information-based analysis of components

Figure 5.20: Localization results on a demo PCB. Localized objects are marked with their bounding boxes (in yellow) and a score falling between 0 and 1 is assigned to each object for quantifying the confidence of Localization, where higher values for more confident results. Most PCB components are accurately localized with high confidence, while false alarms are associated with very low confidence values. A wide margin is thus provided for separating sought objects from disturbances.

The procedure is based on a cross-validation, where data are divided into ten subsets according to the associated images and all tests are conducted only on the original samples obtained through the localization process.

Usually, CCR is employed for assessing the quality of classification results and quantifies the probability of correctly classifying objects into their corresponding categories. This performance measure is best suited for the case of balanced classes, where all classes are comparably significant.
in the considered data and none of them is dominant. In the case of “PCB General”, however, components from different categories are highly unbalanced. For instance, there are in general many capacitors and, on the contrary, only occasionally LEDs. As a result, the overall CCR is primarily determined by the dominant classes and the obtained quality assessment is strongly biased. To deal with this problem and aiming at a less biased quality assessment, mean F-score (m-F-score) based on mean precision (m-precision) and mean recall (m-recall) is used to quantify the performance of categorization classifiers:

\[
\begin{align*}
\text{m-precision} &= \frac{1}{C} \cdot \sum_{c=1}^{C} \text{precision}(c), \\
\text{m-recall} &= \frac{1}{C} \cdot \sum_{c=1}^{C} \text{recall}(c),
\end{align*}
\]

(5.18)

where precision\((c)\) and recall\((c)\) denote the precision and recall values of the \(c\)-th class, respectively.

5.4.2.2 Results

Since those ensemble methods described in Section 5.3 are used to boost the performance of fundamental classifiers, an investigation of the combinations between features and fundamental classifiers is thus conducted at first and establishes the necessary basis for the further evaluation of more sophisticated classifiers. In other words, only those feature-classifier com-
Combinations resulting in reasonable categorization performance are considered for the ensemble methods. Moreover, in comparison to localization, there is a significantly reduced sample variability in categorization, which consequently results in lower computational complexity in training categorization classifiers. Therefore, it becomes feasible to conduct a more comprehensive evaluation of features and classifiers.

The investigation of fundamental feature-classifier combinations starts with Haralick features, which provide satisfactory classification results in the case of SMDs. However, their categorization performance on general PCB components turns out to be far from applicable and the best m-F-score value is only 0.385. In consideration of this, they are unsuited for the intended categorization and thus excluded from further evaluation. In the case of HOG features, to maximize their discrimination power, a parameter tuning is first conducted and determines the optimal feature configuration. Since $k$-NN classifiers with $k = 1$ are non-parametric and deterministic, they are employed as the reference classifiers in tuning HOG parameters. It is worth mentioning that the distance metric used for $k$-NN models has great impact on their classification performance. Therefore, there is in fact a co-optimization of HOG parameters and the distance metric. By exploring this combined parameter space, the $\chi^2$ distance combined with the cell size of 15 and the block size of $2 \times 2$ is found to result in the optimal categorization, while the other three distance metrics “Hellinger”, “Euclidean” and “city block” give rise to a decline in the quality of the obtained classification results, especially for increased number of nearest neighbors. This parameter configuration is subsequently applied for training corresponding DTree and SVM classifiers, where these classifiers also undergo a further parameter optimization process. The performance of fundamental feature-classifier combinations based on HOG features is summarized in Table 5.7 for an overview. For all training-test data settings, an SVM classifier is superior to the other two alternatives. Moreover, by introducing additional rotation of components into training and test data, a global decline in performance is observed. If additional noise is introduced, $k$-NN and SVM classifiers lead to a more precise categorization, while DTree classifiers do not benefit from the augmented variability of training samples.

Regarding the similarity between BoVW and FV features, a joint analysis is conducted on these two feature types. The primary features (e.g. SIFT and SURF) used in BoVW and FV models are typically histograms of local
5.4. Evaluation

<table>
<thead>
<tr>
<th>feature</th>
<th>classifier</th>
<th>m-F-score</th>
<th>rotation</th>
<th>noise</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>standard</td>
<td>rotation</td>
<td>noise</td>
</tr>
<tr>
<td>HOG</td>
<td>k-NN</td>
<td>0.645</td>
<td>−0.128</td>
<td>+0.014</td>
</tr>
<tr>
<td></td>
<td>DTree</td>
<td>0.517</td>
<td>−0.103</td>
<td>−0.061</td>
</tr>
<tr>
<td></td>
<td>SVM</td>
<td>0.694</td>
<td>−0.132</td>
<td>+0.018</td>
</tr>
<tr>
<td>BoVW</td>
<td>k-NN</td>
<td>0.722</td>
<td>−0.032</td>
<td>−0.032</td>
</tr>
<tr>
<td></td>
<td>DTree</td>
<td>0.678</td>
<td>−0.090</td>
<td>−0.031</td>
</tr>
<tr>
<td></td>
<td>SVM</td>
<td>0.845</td>
<td>−0.079</td>
<td>−0.100</td>
</tr>
<tr>
<td>FV</td>
<td>k-NN</td>
<td>0.759</td>
<td>−0.043</td>
<td>−0.073</td>
</tr>
<tr>
<td></td>
<td>DTree</td>
<td>0.617</td>
<td>−0.068</td>
<td>−0.175</td>
</tr>
<tr>
<td></td>
<td>SVM</td>
<td>0.855</td>
<td>−0.073</td>
<td>−0.098</td>
</tr>
</tbody>
</table>

Table 5.7: Comparison between fundamental feature-classifier combinations. In the standard training-test data setting, no additional rotation and noise are introduced. For the settings with additional rotation and noise, the variation of m-F-score with respect to the standard setting is given for a better comparison.

gradients. To measure the difference (or affinity) between two histograms, χ² and Hellinger are more suited than Euclidean distance. In practice, as suggested in [218], the Hellinger kernel is preferable over the χ² test since the associated computational complexity is much lower. If the Hellinger measure is adopted, the predicted performance boost is observed for both BoVW and FV features, especially in combination with k-NN and SVM classifiers. To optimize the performance of BoVW and FV models, the size of their visual vocabularies is fine-tuned: more than 1000 words are necessary for achieving the desired categorization with BoVW models, while 20 words are adequate for stabilizing the performance of FV models. For a side-by-side comparison, the final classification results using BoVW and FV features are also listed in Table 5.7. It is interesting to see that SVM classifiers are generally superior to the others and DTree classifiers are ranked last in all cases. In comparison to HOG features, BoVW and FV features result in much better classification results. As a desirable consequence of global feature pooling, they are also able to provide more stable performance in case of additional rotation of components.
Relying on the determined optimal configurations of features and fundamental classifiers, ensemble methods are employed to boost the classification performance. Regarding the characteristics of the three fundamental classifiers and of the considered four ensemble methods, following advanced feature-classifier combinations are investigated: \( k \)-NN + random subspace and DTree + Bagging/RF/AdaBoost. Each ensemble-based classifier is further combined with all three feature types. The resulting categorization of these advanced combinations is summarized in Table 5.8. Despite the improvement of \( k \)-NN and DTree classifiers through ensemble learning, SVM classifiers remain the best for the standard training-test data setting. Among the listed advanced feature-classifier combinations, Bagging and RF in general lead to better categorization results. Moreover, the random subspace method, Bagging and RF are able to stabilize the classification performance if rotation and noise are presented. On the contrary, SVM classifiers exhibit generally higher sensitivity to variations in data and become thus less reliable.

During training DL classifiers for this categorization task, the DBN architecture from [219] and a simplified architecture of the region validation CNN in Section 5.2.2.3 are adopted. The input images are warped to \( 40 \times 40 \times 3 \) for the DBN and to \( 64 \times 64 \times 3 \) for the CNN to reduce overall model parameters, which is essential for training DL classifiers on small datasets. The best categorization is achieved using the CNN classifier combined with noisy training data and the resulting m-F-score = 0.606 is incomparable to the results listed in Table 5.8. This is mainly due to the insufficient size of training data and significant overfitting is observed through a comparison between training and inference results.

## 5.5 Discussion

Although the combinatory localization approach provided for the first time a feasible solution to achieve the desired 2D analysis of general PCB components, it still suffers from several performance issues. As stated in Section 5.4.1.2 and through the comparison between \( N \)-precision and precision, as well as between \( N \)-F-score and F-score, there exist many redundant region proposals arising from the same objects of interest. This is a direct consequence of the diversification-driven algorithm for region...
### Table 5.8: Comparison between advanced feature-classifier combinations using different ensemble methods. In the standard training-test data setting, no additional rotation and noise are introduced. For the settings with additional rotation and noise, the variation of m-F-score with respect to the standard setting is given for a better comparison.

<table>
<thead>
<tr>
<th>feature</th>
<th>classifier</th>
<th>ensemble method</th>
<th>m-F-score</th>
<th>standard</th>
<th>rotation</th>
<th>noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOG</td>
<td>k-NN</td>
<td>/</td>
<td>0.645</td>
<td>-0.128</td>
<td>+0.014</td>
<td></td>
</tr>
<tr>
<td></td>
<td>subspace</td>
<td></td>
<td>0.664</td>
<td>-0.122</td>
<td>+0.021</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DTree</td>
<td>/</td>
<td>0.517</td>
<td>-0.103</td>
<td>-0.061</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bagging</td>
<td>0.671</td>
<td>-0.117</td>
<td>-0.028</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>RF</td>
<td>0.657</td>
<td>-0.100</td>
<td>-0.010</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>AdaBoost</td>
<td>0.620</td>
<td>-0.153</td>
<td>-0.072</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SVM</td>
<td>/</td>
<td>0.694</td>
<td>-0.132</td>
<td>+0.018</td>
<td></td>
</tr>
<tr>
<td>BoVW</td>
<td>k-NN</td>
<td>/</td>
<td>0.722</td>
<td>-0.032</td>
<td>-0.032</td>
<td></td>
</tr>
<tr>
<td></td>
<td>subspace</td>
<td></td>
<td>0.753</td>
<td>-0.002</td>
<td>-0.013</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DTree</td>
<td>/</td>
<td>0.678</td>
<td>-0.090</td>
<td>-0.031</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bagging</td>
<td>0.755</td>
<td>-0.021</td>
<td>-0.015</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>RF</td>
<td>0.752</td>
<td>-0.004</td>
<td>-0.002</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>AdaBoost</td>
<td>0.753</td>
<td>-0.077</td>
<td>-0.028</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SVM</td>
<td>/</td>
<td>0.845</td>
<td>-0.079</td>
<td>-0.100</td>
<td></td>
</tr>
<tr>
<td>FV</td>
<td>k-NN</td>
<td>/</td>
<td>0.759</td>
<td>-0.043</td>
<td>-0.073</td>
<td></td>
</tr>
<tr>
<td></td>
<td>subspace</td>
<td></td>
<td>0.742</td>
<td>-0.027</td>
<td>+0.005</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DTree</td>
<td>/</td>
<td>0.617</td>
<td>-0.068</td>
<td>-0.175</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bagging</td>
<td>0.786</td>
<td>-0.042</td>
<td>-0.019</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>RF</td>
<td>0.788</td>
<td>-0.055</td>
<td>-0.020</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>AdaBoost</td>
<td>0.757</td>
<td>-0.063</td>
<td>-0.065</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SVM</td>
<td>/</td>
<td>0.855</td>
<td>-0.073</td>
<td>-0.098</td>
<td></td>
</tr>
</tbody>
</table>

Proposal generation and additional merging operation is thus necessary. The state-of-the-art approach to deal with this problem is suppressing non-maximum overlapping region proposals with respect to their confidence score. However, to obtain a reliable confidence score for supporting the intended NMS, region validation classifiers need to undergo carefully designed training procedures, for instance hard negative mining [137,159]. This tricky and time-consuming adaption of classification models results
in difficulties in extending the established localization for new component groups and requires huge datasets for avoiding possible overfitting. Besides repeated localization of same components, another unsolved problem is low-quality proposals of sought objects, which is due to the extremely high variability of PCBs and the consequent difficulty in directly generating accurate region proposals. This has negative impact on both region validation and component categorization since the image content of test samples may strongly deviate from the reference component models. A potential issue of the combinatorial approach is the heavy dependence on color information, especially the background estimation employed for improving the performance of region generation. If the substrates of PCBs have similar colors, typically black, as components, any color-based analysis becomes less reliable and leads to inaccurate localization results.

Benefiting from the LV-based thresholding and the CNN-WRDF-based bounding-box regression, the compact approach can in contrast solve the localization problem in a more intuitive and elegant fashion. Relying on a homogeneity analysis, LV is able to emphasize sought objects from background and enables non-parametric region generation using an adaptive thresholding. The localization accuracy is subsequently improved through bounding-box regression, which also merges redundant proposals by evolving them into similar regions accurately bounding the underlying components. With the help of a further CNN classifier, false alarms are reliably suppressed and the desired localization is obtained. This approach is merely data-driven without tedious parameter tuning and can thus be extended for further component groups or other localization tasks without great effort. As demonstrated on the dataset “PCB General”, the compact localization is realizable using datasets of limited size, which makes it more accessible for general applications.

Relying on the acquired knowledge through the comprehensive evaluation of methods for region generation, for region validation, as well as for component categorization, this 2D analysis of PCB components can be reconfigured to meet diverse requirements in practice. Depending on the intended trade-off between efficiency and quality, Efficient Graph and LV-based thresholding should be selected accordingly to reduce the number of proposals or to localize more components. If a fast adaption for new components is desired and the provided training data are inadequate, RF is a preferable classifier due to its insensitivity to parameter variations and resistance to overfitting. If only the size of available datasets is
strongly limited, SVM should be considered for better classification performance and their optimal model parameters are determined through a time-consuming optimization process. Given adequate training data, the establishment of the desired localization is quite straightforward and the suggested configuration of the compact approach is to be adopted.

As demonstrated with the detection of symmetry axes in Section 5.2.2.1, the proposed LV analysis is rather a general method for better exploiting image content. It can typically be used as pre-processing for segmentation and feature detection tasks, which emphasizes relevant information while suppressing distractions. For localization-involved tasks, the novel CNN architecture with recycled deep features is able to significantly boost the power of CNN models. By modeling text recognition as position-dependent object classification, CNN-WRDF is also employed for improving the performance of text recognition on PCBs, which will be described in the next chapter.
Chapter 5. General 2D information-based analysis of components
Chapter 6

Text analysis

Due to the highly dynamic development in the manufacturing of electronic devices and the extremely high variability in shape, size, color and texture of PCB components, the material composition of PCBs deduced from the categorization results suffers in general from reliability issues. Indeed, this problem can be solved in consideration of the text/label information associated with components and used for the identification purpose. If the provided text information is retrieved and utilized in the intended material analysis based on a comprehensive data bank covering diverse components, the reliability of the asserted material composition can be substantially improved. By also retrieving the text information associated with PCBs, the material analysis becomes more straightforward in case of an additional data bank provided for understanding uniformly defined character strings and acquiring prior information of previously studied sample PCBs. For better understanding the relevance of the text analysis in PCB recycling, some typical text information associated with components and PCBs are demonstrated in Figure 6.1. The special character string “2002/95/EC ROHS COMPLIANT” indicates for example the restricted existence of lead, mercury, cadmium, hexavalent chromium, polybrominated biphenyls and polybrominated diphenyl ethers according to the Directive 2002/95/EC of the European Parliament and of the Council [220].
Chapter 6. Text analysis

Figure 6.1: Relevant text information on PCBs. Green rectangles are used to highlight the text information associated with components, while the text information associated with PCBs is marked with red rectangles.

To identify the challenges imposed by the text analysis on PCBs to state-of-the-art techniques for Optical-Character-Recognition (OCR) applications, this chapter starts with a comprehensive review of so far established OCR methods and approaches. Regarding the conducted survey, the remaining sections of this chapter tightly focus on practical solutions for addressing the identified challenges, including essential pre-processing operations published in [5,6] and adjusted text recognition. To confirm the significance of the proposed modifications and extensions, a detailed comparison between state-of-the-art and the introduced OCR systems is also provided. At the end of this chapter, an in-depth discussion on obtained OCR results is conducted and gives an insight into the further improvement of the text analysis on PCBs.

6.1 State of the art

Text remains one of the most researched objects in computer vision, due to its wide application for highly efficient and unambiguous information retrieval. From earlier digitization of documents and automatic text translation to advanced data mining in natural scene images and videos, text analysis has gained special attention, especially with the recent breakthroughs in machine learning and object detection. As a result, novel applications, for example automatic information retrieval for assisting blind and visually impaired users, steadily emerge from further development in state-of-the-art OCR systems. For a better organized survey on OCR
methods and approaches, they are reviewed in this thesis according to the complexity of the intended application scenarios, rather than according to the underlying analysis: pixel-wise binarization, feature-based region segmentation, sliding-window detection, heuristic-based determination of text lines, machine learning-based proposal validation and text recognition, template matching-based text recognition, for which a clear categorization is non-trivial.

Typical application scenarios for OCR systems with increasing complexity are depicted in Figure 6.2, where images from lower rows represent more challenging scenarios. The top two images reflect the most basic analysis on black-white documents. With the help of an appropriate binarization, background is separated from the foreground text objects. The classification between foreground and background pixels can be achieved using either a global threshold obtained from Otsu’s method [202] or adaptive local thresholds relying on the mean pixel value and the standard deviation in a predefined local neighborhood [194,195,196,197]. An ensemble classifier combining complementary binarization methods may also be applied for improving the overall classification performance. However, in case of extremely uneven illumination, both global and local thresholding can hardly provide satisfactory results if the parameters of adaptive binarization methods are not adjusted accordingly. To eliminate the requirement on additional parameter tuning, morphological closing operation is applied for estimating the uneven background illumination [221], which is subsequently removed from the original image for a reliable binarization. Occasionally, there exist randomly distributed local disturbances, as in the case of the first image in Figure 6.2, and the associated binarization problem cannot be well addressed using the methods described above. As suggested in [222], discrete curvelet transform is suited for suppressing both low-frequency background distractions and high-frequency noise in such cases. The desirable edge-preserving property maintains text objects and thus leads to a better separation between text and background.

In the second image of Figure 6.2, an additional skew correction of text lines is necessary, which rotates the text running direction to be aligned with the desired orientation (horizontal or vertical). Considering the rich characters and text lines available in documents, the skew angle of text lines can be estimated using diverse statistical analysis. One of the most commonly used methods for the purpose of skew correction is the anal-
Chapter 6. Text analysis

Figure 6.2: Different application scenarios for OCR systems.

ysis of projection profiles. Pixels of a binarized document image can be summed up for each horizontal line to generate a 1D histogram of the number of text pixels along the horizontal direction. The accumulated squared value or difference between adjacent lines is employed as the measure to assert the quality of skew angle candidates, where the maximum value indicates the optimal estimate. In consideration of ef-
ficiency and accuracy, different strategies for searching the angle of the maximum quality value have been suggested [223, 224]. As demonstrated in [225], the classical Hough transform of binarized document images are similar to stacked projection profiles along all directions between 0 and 180°. The skew angle is then determined using the same quality measure, where the 1D histogram along each single direction is investigated individually. A more straightforward estimation is realized through localizing the point in the 2D histogram, in whose neighborhood the maximum number of votes is obtained [226]. To reduce computational complexity, 2D histograms can alternatively be generated only for centroids of connected components by approximating each character through adjacent blocks [227]. To deal with the case of raw images without binarization, sub-band images combined with projection profiles [228] or the interline cross-correlation [229] are recommended solutions. If there is no clearly defined spacing between text lines, typically in documents with complex layout, connections between nearest neighbors can be utilized to retrieve the sought skew angle [230, 231].

The binarized and skew-corrected document images are subsequently fed into OCR engines, which extract the present text information through recognizing single characters. Although commercial products usually provide superior recognition performance, open source engines are in general more attractive for the research purpose, due to their technical transparency. Among them, Tesseract-ocr [232], Cuneiform-linux [233], GOCR [234], OCRAD [235] are most applied and can be used as off-the-shelf components in OCR systems. As stated in a previous publication [5], Tesseract-ocr is preferable over other engines due to its better performance, wider application and transparent implementation [236].

Besides in documents, text also widely exists in scenes of daily life. As depicted in the second and third rows of Figure 6.2, text information can be typically found on signboards, streets, cars, etc. in natural scenes. Due to the associated high variability in text and in background, the desired information retrieval in such images becomes less straightforward than in document images. For those natural scene images with centered text objects and clean background as well as foreground, for example the images in the second row of Figure 6.2, satisfactory OCR performance can still be obtained using more sophisticated binarization methods. Following the approach introduced in [237], which combines the adaptive thresholding [195] with a background estimation, candidate characters are
first localized as connected components in images. With respect to some heuristic criteria based on observed character and inter-character properties, false alarms are subsequently removed to produce a binary image only with extracted text objects. For unknown text color, this analysis is performed on the original as well as on the complementary image and the binarized image with more hypothesized text objects is selected as the final output. To overcome the difficulty of a direct binarization on natural scene images, the stroke width transform was introduced in [238] for identifying candidate text pixels. With the help of edge pixels localized using Canny detector [125], stroke width values are assigned to corresponding image pixels by tracing rays from edge pixels and along their gradient directions, where the stroke width for pixels on an arbitrary ray is defined as the distance between the start point and the intersection point with another edge pixel exhibiting a nearly opposite gradient direction. Neighboring pixels with similar stroke width values are grouped together to form single characters and false alarms are canceled using some heuristic criteria again. Further using inter-character properties, text lines and single words are recovered from characters. For retrieving text information, the obtained binary images undergo a text recognition procedure, which is also supported by conventional OCR engines.

In more challenging images illustrated in the third row of Figure 6.2, text objects are not necessarily centered and even might be blurred. Moreover, characters of foreground text might not be clearly separated from each other and there are significant distractions arising from background. Apparently, binarization-based approaches become less suited in such cases and more general OCR approaches are required. Spotting text objects in images can be considered as a special case of general object detection tasks. As stated in Chapter 5, exhaustive search and feature-based image partitioning/segmentation are two possible proposal generation approaches for localizing objects in complex scenes. The obtained proposals undergo further validation, refinement and classification steps for achieving the desired recognition of objects. This workflow is also adopted by state-of-the-art OCR systems and leads to promising results on natural scene images. In [239], a text salience map is generated through asserting the text confidence of local images cropped using sliding windows of different sizes. After pixel-wise binarization, text line segmentation and word localization, hypotheses of words for a given lexicon are scored using the character probability maps for any cropped local word image.
The hypothesis with the best score is considered as the final recognition result. During text spotting, CNN classifiers play a central role and generate the essential confidence/probability maps. The same authors also investigated the performance of a more general object detector, viz. Edge Boxes [138] relying on a sliding-window search [207]. Through a series of post-processing steps comprising pre-validation, bounding-box regression, post-validation and recognition, text information can be retrieved in images of a wide spectrum of natural scenes. As an alternative solution, specific features and the resulting local regions can also be used to generate text proposals. Due to the commonly available homogeneity of colors within single characters, MSER [109] is considered as suitable region detector for the purpose of text spotting. In [240], MSER is combined with characterness cues to localize text in wild, while a more recent research [241] demonstrated impressive results on most public datasets by applying modified Selective Search on primary detection results arising from MSER proposals. For generating more comprehensive text proposals, both sliding widow and feature-based approaches have been employed in [242] and realized as a hybrid proposal generation algorithm. It is worth mentioning that reliable text spotting results of the desired OCR systems heavily rely on dictionaries, lexicons and language models (character-level and word-level \(n\)-grams).

In comparison to the application scenarios reviewed so far, the desired retrieval of text information on PCBs imposes several new challenges. As can be observed in the bottom images of Figure 6.2, the orientation of text is unconstrained and the usual assumption of nearly horizontal text lines becomes invalid. Regarding the highly variable contrast between printed text and background, as well as the great diversity in text color, font and size, both global and local binarization methods are unable to provide a clear separation between foreground and background. Also due to the highly variable size of text, any sliding window-based proposal generation becomes infeasible as the resulting search space expands dramatically. In consideration of the nearly random character strings on diverse components and PCBs, there arises another important issue: dictionaries, lexicons and language models can hardly be defined. Without this useful constraint, it is less reliable to distinguish between text objects and dense background distractions using state-of-the-art OCR engines. Bearing these challenges in mind, modifications and extensions to state-of-the-art OCR systems become necessary. More specifically, region
proposals obtained from general text localization approaches and from components localized in Chapter 5 need to first undergo pre-processing steps, which recover the desired text orientation and extract single text lines or words. The resulting local text images are then fed to an OCR engine appropriately configured for recognizing text on PCBs. Since the generation of primary text proposals remains unaltered, the following two sections merely deal with the essential pre-processing steps and the correct configuration of the required OCR engine.

6.2 Pre-processing

Conventional skew correction and text line segmentation methods usually rely on statistical properties of text, which implicitly requires a great number of characters and text lines. This requirement is, however, not fulfilled on PCBs as there are typically only a few characters available in cropped local images for the purpose of identifying components. Even worse, this problem is often correlated with varying contrast, color, font, size, layout and orientation as illustrated in Figure 6.3. To realize the intended pre-processing, the only possible solution is to explicitly consider some inter-character properties, which consequently requires accurate localization of characters.
6.2. Pre-processing

6.2.1 Binarization

The most straightforward solution for localizing characters is performing image binarization using global or local thresholding methods. Due to the challenging characteristics of PCB images depicted in Figure 6.3, state-of-the-art thresholding methods are in general unable to provide satisfactory performance. Aiming at a reliable localization of characters in such images, a novel binarization approach with advanced local thresholding method has been developed [5] and is described in this subsection.

The overall binarization begins with some image enhancing steps, which essentially suppress noise and improve the contrast between foreground and background. The most suited processing for such purposes is nonlinear diffusion [124,168]. It removes undesired disturbances while preserving or enhancing edges. Especially on PCB images, it practically improves the homogeneity between foreground pixels, as well as between background pixels. Moreover, the boundaries between text and background are enhanced for appropriate diffusion parameters, which leads to improved contrast and therefore better binarization performance. Occasionally, the contrast between foreground and background still remains inadequate even after nonlinear diffusion. This case is identified through an analysis of the mean absolute variation of pixel values with respect to the global mean pixel value, where the obtained contrast measure is below a predefined threshold. To further improve the contrast, all image pixels undergo an exponential transformation, which expands the narrow value range of foreground or background correspondingly.

Due to their ability to be adapted for local illumination and contrast, local thresholding methods can be applied on the enhanced images for localizing text objects. Taking the text-specific binarization from Wolf [196] for example, the threshold for identifying text pixels is governed by the mean value and standard deviation in local windows, and is adapted to local contrast using a parameter $k$. However, the local window size and $k$ need to be determined experimentally as an adaptive parameter selection is missing in the literature. Consequently, the resulting thresholding is inappropriate for PCB images due to varying font size. To overcome this issue, the local window size is determined through a stroke width estimation, see Figure 6.4. A very similar analysis method has been parallely developed and applied for detecting scene text in [240,243]. For any char-
Chapter 6. Text analysis

Figure 6.4: Stroke width estimation. Only the distance between skeleton pixels and background is considered. Using the dominant distance value 4, the stroke width is determined as $(4 \times 2 - 1) = 7$.

(а) character image (b) distance map (c) skeleton

(d) stroke width histogram

acter (Figure 6.4 (a)), the distance (Figure 6.4 (b)) between its skeleton pixels (Figure 6.4 (c)) and background is computed, and all distance values are summarized in a histogram (Figure 6.4 (d)). Finally, the stroke width is determined as the doubled dominant distance value minus one and the optimal local window size is selected as the doubled stroke width, which results in the highest standard deviation value for ideal text (black) and background (white). Benefiting from the enhanced image quality, the $k$ parameter does not require any adaption and is set to a constant value, typically 0.1. From a practical point of view, there remains an unsolved problem: how to achieve the initial binarization for estimating the stroke width. This problem is solved if Otsu’s non-parametric global thresholding method [202] is employed for the initialization purpose and the stroke width is not estimated for single characters but globally. Optionally, the stroke width estimation and local thresholding can be organized in an iterative manner for a more precise binarization.
A common limitation of binarization methods is their desired application on grayscale images, which discards additional color information if available. This could lead to degraded performance on PCBs since intensity information becomes less reliable if uneven illumination is present. To utilize color information in localizing text objects, all image pixels undergo a linear transformation after an initial binarization, where pixel values are projected along the direction vector defined by the cluster centroids of foreground and background pixels. By this means, a new grayscale image is obtained, whose pixel values give rise to the maximum inter-class distance, and text can thus be better separated from background.

Unlike in documents, text on PCBs can appear in diverse colors and the important assumption of dark text on bright background becomes improper. A general solution for identifying the text color relies on the comparison between binarization results obtained on original and complementary images. Usually, complicated decision rules based on heuristics are involved in this comparison [237]. However, such rules turn out to be too specific for the considered scenes and a great set of parameters must also be tuned accordingly. To make the estimation of the text color more generic and to avoid any parameter tuning, a novel and efficient method is proposed. Regarding the fact that text objects consist of thin line structures, while background fills up gaps between characters and also fills up the huge border area around text, a simple criterion for determining the text color is designed as

\[
\sum_{p_i \in \Omega_T} d(p_i, C_T) < \sum_{p_i \in \Omega_B} d(p_i, C_B), \quad (6.1)
\]

where \(\Omega_T\) and \(\Omega_B\) denote the area of text and of background, respectively. Given the boundary pixels \(C_T\) of \(\Omega_T\) and the boundary pixels \(C_B\) of \(\Omega_B\), \(d(p_i, C_T)\) and \(d(p_i, C_B)\) denote the shortest distance between a pixel \(p_i\) and the boundary pixels \(C_T\) as well as \(C_B\). This inequality must hold for a correct estimate of the text color.

### 6.2.2 Skew correction and text line segmentation

Although the binarization approach introduced above leads in general to accurate binarization results without any parameter tuning, some difficult cases could exist and give rise to a less reliable localization of text objects.
Chapter 6. Text analysis

As a result, the inter-character connection-based analysis [230,231] cannot provide a reliable estimate of the skew angle since many non-character distractions are also present in binarized images. Regarding this issue, a novel but more reliable algorithm relying on intrinsic properties of printed text has been developed [6] for estimating the skew angle and extracting single text lines, which especially eliminates the requirement of additional parameter tuning for general OCR applications.

For better reading experience and for better distinguishing between different text lines, adequate spacing is always inserted between single characters, words and lines in printed text. Moreover, the spacing between lines turns out to be wider than between single characters. To capture these intrinsic properties of printed text, subregions defined by single characters and the spacing in their direct neighborhood can be considered. As illustrated in Figure 6.5 (b) and (e), the spacing pixels between single characters are assigned to the nearest characters according to the skeleton of the spacing areas. The layout and the skew direction are roughly embedded in the shape of the resulting subregions. Aiming at strengthening the correlation between the shape of subregions and the skew direction, a subsequent refinement of subregions is indispensable. The first modification relies on filling up characters along the $y$ and the $x$-axis. This is then followed by a further skeleton transform of spacing pixels, where pixels after the first skeleton transform connected to the image border are excluded. Another modification is the bounding of subregions, which removes the exterior pixels with respect to the median bounding box. Subregions after the described refinement are presented in Figure 6.5 (c) and (f) for demonstrating their strengthened correlation with the skew direction, as well as with the layout.

The skew direction is determined through the PCA analysis of the overall covariance matrix, which is generated as the weighted sum of covariance matrices computed for single subregions. Moreover, the relevance of single subregions depends on the similarity between the bounding boxes of subregions and the median bounding box. For a more accurate estimation, the entire process can be repeated until the equilibrium is reached. It should be emphasized that subregions lead to less reliable skew estimation in case of slanted fonts. To deal with this problem and also to provide a reasonable initial estimate, additional skew angle estimates arising from the Euclidean Minimum Spanning Tree (EMST) method [231] and from
6.2. Pre-processing

(a) original BW image  (b) subregions  (c) modified subregions

(d) corrected BW image  (e) subregions  (f) modified subregions

**Figure 6.5:** Skew correction using subregions. Results for the original and the skew-corrected image are depicted in the first and the second row, respectively. The skew direction is embedded in the shape of subregions.

the PCA analysis on entire images as well as on single characters are also involved. The final optimal estimate is obtained as the angle close to the subregion-based estimate and resulting in the minimum length of the binarized horizontal projection profile.

Since there are often multiple text lines available in cropped local images, single text lines need to be extracted and fed into OCR engines for achieving the desired text recognition. Using the layout information embedded in subregions, the intended text line segmentation can be realized without great effort. As illustrated in Figure 6.6, centroids of subregions are first extracted in the skew-corrected text image and undergo a 1D Gaussian blur along the $y$-axis. This smoothing is essential for dealing with centroids deviating from the central lines and the size of the Gaussian kernel is related to the median bounding box of subregions. Consequently, the number and the positions of text lines are determined through a Mean Shift clustering applied on the horizontal projection profile of the blurred centroids. In Figure 6.6 (d), red solid lines mark the $y$-position of the detected text lines, while green dashed lines indicate their boundaries.
The segmented centroids are then used for fitting the corresponding central lines and a refined skew correction is conducted with respect to the fitted lines. By assigning single characters to the corresponding central lines, single text lines are thus segmented. If required, this segmentation results can also be used to partition the original text image into sections of text lines. It is noticeable that there still exist non-text objects in Figure 6.6 (f). This is a result of the generic text analysis so far. To remove such disturbances, reliable OCR engines must be provided.

6.3 Text recognition

Benefiting from the recent development in machine learning, especially in deep learning, breakthroughs have also been made in text analysis and lead to very promising OCR performance, for example [207,242,244]. Nevertheless, such OCR systems commonly combine character-level (single
6.3. Text recognition

character or character strings) CNN-based OCR engines with dictionaries, lexicons, language models and joint appearance models of \( n \)-grams, which can be implemented with both CNNs and Recurrent Neural Networks (RNNs [140]). In the case of PCB images with nearly random character strings, there arises a natural question: are these combinations suited for the text recognition on PCBs? There exists a further question concerning the design of CNNs presented in Section 5.2.2: is state-of-the-art CNN structure appropriate for reading character strings in images?

To answer the first question, a closer consideration of the uncertainty associated with the text on PCBs is necessary. Depending on the manufacturers, arbitrary character strings can be used for the identification purpose and there is no general constraint available on such text, which therefore denies the application of dictionaries, lexicons and language models. Considering all possible characters at each position of unconstrained strings, it turns out to be hardly possible to train any joint appearance model based on \( n \)-grams: even when \( n \) is set to a small value of 4: there are in total \( 68^4 = 21381376 \) possible combinations for a character set including 52 letters (lowercase and uppercase), ten digits, five special symbols ("/", "-", ":", ",", "&", " ") and one empty space holder ("ø"). If a great number of fonts are to be covered, an explicit training of such models exploiting the joint appearance of a consecutive chain of characters becomes impractical. In the context of the second question, reading character strings is equivalent to recognizing spatially arranged objects of different classes. Therefore, state-of-the-art CNN structure can provide reliable assertion of the existence of single characters, but will lead to less accurate localization performance and thus less reliable retrieval of the spatial arrangement of characters. In consideration of the essential spatial information, the novel CNN-WRDF proposed in this thesis are preferred over state-of-the-art CNNs and provide superior OCR performance.

The most challenging problem to be addressed in training a CNN-based OCR engine is the demand of an extraordinarily huge set of training samples. Although there are some public datasets [245,246,247] available for OCR applications, they are limited either in size or in the applicability for recognizing text on PCBs. To support the training of the desired OCR engine, a convenient tool for generating synthetic training samples has been developed in a Master’s thesis [248] and implemented using Python [249]. This tool is similar to the tool introduced in [245], but with better flexibility and additional functionality for assessing the performance of OCR.
systems, which can also be utilized to rapidly train OCR systems for specific applications through an iterative training-assessing procedure. By defining a “receipt” in the YAML [250] format, a sample generation pipeline consisting of modularized processing units is configured without great effort. For generating realistic sample images, a comprehensive simulation of the image formation process is presented. A typical pipeline begins with the generation of binary text images of varying font, size and string. After disturbing single text pixels and distorting entire images with respect to the given projective transformation (random rotation, shearing, anisotropic scaling and perspective distortion), text samples are colorized and blended into background images. By applying additional noise and image compression, final synthetic images are obtained.

To facilitate the realization of OCR engines in the form of CNN-WRDF, modifications should be made to the networks described in Section 5.2.2. Regarding the increased relevance of spatial information, the forwarded connections are associated with more sophisticated subnetworks, which do not only extract more distinctive features, but also lead to better decoupling between the features for the classification and for the localization purpose. At the top of networks, the multinomial logistic regression loss for the ML estimation is employed as the objective function and the corresponding loss value $E_{\text{loss}}$ for any training sample is computed across all positions in the character string of the maximum length $L$:

$$E_{\text{loss}} = \frac{1}{L} \sum_{l=1}^{L} \left( \ln \left( \sum_{\hat{c}=1}^{C} e^{\hat{y}(\hat{c}, l)} \right) - \ln(e^{\hat{y}(c, l)}) \right), \quad (6.2)$$

where $e^{\hat{y}(\hat{c}, l)}$ of the network output $\hat{y}(\hat{c}, l)$ denotes the estimated unnormalized probability of the case that the character at the position $l$ is from the $\hat{c}$-th of all $C$ classes, while $e^{\hat{y}(c, l)}$ is for the ground-truth class $c$ at the current position $l$. For unconstrained strings, characters at $L$ positions are assumed independent and the joint probability of all $L$ characters is thus decomposed into the product of probabilities for single characters.
6.4 Evaluation

6.4.1 Data and performance measures

To gain a deeper insight into the OCR performance on PCBs, the localization and the subsequent information retrieval are investigated individually, instead of an overall assessment of their combined performance. For evaluating the performance of localization approaches, in total 31 images with 6,836 manually marked bounding boxes of character strings are provided. In the further evaluation of the text reading, the 430 cropped local text images employed for the evaluation of the local TCA correction performance in Section 3.4.4.2 are considered again. In consideration of an unbiased evaluation with uniformly distributed text and background colors, this dataset is augmented with the corresponding complementary images, which finally results in 860 local text images. Moreover, 712 images contain only one single text line, while the other 148 images contain multiple text lines.

The performance measures for assessing the quality of the proposals obtained through localizing text objects in PCB images remain the same as in the case of localizing PCB components in Chapter 5: precision, recall and F-score, where IoU > 0.5 must be fulfilled for any valid matching between proposals and ground-truth bounding boxes. For the evaluation of the text reading performance, the F-score arising from precision and recall is employed again, where the matching quality between the retrieved and ground-truth character strings is quantified with respect to the edit distance defined by Levenshtein [251]. For the retrieved string “oi234678g” and the ground-truth string “0123456789”, the corresponding edit distance is 4: “o” → “0”, “i” → “1”, insert “5” between “4” and “6”, “g” → “9”. Consequently, precision of the OCR result is $(10 - 4)/9 = 0.667$ and recall is $(10 - 4)/10 = 0.600$, where the numerator remains the difference between the maximum length of the two strings and the edit distance, while the denominator takes the length of the retrieved string or of the ground-truth string, respectively.

In training CNN-based OCR engines, 2,675,265 samples are used and another 297,196 samples serve for the purpose of validating the reading performance of OCR engines during training, where 478 fonts are involved in the text generation. Some synthetic text images for training and for vali-
Chapter 6. Text analysis

Figure 6.7: Random samples of synthetic text images.

dation are randomly selected and presented in Figure 6.7. Through random color permutation and projective transformation, 1675 background images are obtained from 113 original PCB images without printed text. To avoid possible overfitting arising from the limited variability of PCB images, another 971 natural scene images from [154] are applied as additional background images in the generation of synthetic samples.

6.4.2 Localizing text objects

Through a comprehensive review and considering the desirable text spotting performance on natural scene images, four state-of-the-art approaches are selected for localizing text objects on PCBs: WDCT relying on the windowed discrete cosine transform [252], SWT exploiting the stroke width of neighboring pixels [238], Selective Search + MSER [241] and Edge Boxes employing this general object detector [207]. Another important criterion in selecting appropriate approaches is that they should generate complementary proposals. For all considered approaches, their missing source code and models (for regression, validation and text recognition) are reimplemented and retrained, respectively, where the public ICDAR datasets [246, 253] are involved in training text-specific models. Before applying the Edge Boxes-based approach [207], the text spotting result on the SVT dataset [254] has been reproduced and compared with the original result to confirm the properness of the reimplementation. The obtained F-score is 0.452, which is comparable to the original performance 0.530. The decent in the OCR performance is probably due to the missing aggregate channel feature detector [255], for which no adequate details are provided. Any further confirmation is not considered as the original implementation of Selective Search + MSER [241] is publicly available.
and no appropriate comparison can be conducted for WDCT [252], as well as for SWT [238]. Moreover, inspired by the stroke width estimation introduced in Section 6.2.1, the stroke width in [238] can also be determined through the distance transform instead of the expensive ray tracing analysis, which gives rise to an alternative approach DT.

In Figure 6.8, the resulting recall vs. the average number of initial proposals per image is plotted. Since there is no confidence score assigned to proposals arising from WDCT, SWT, DT and Selective Search + MSER, they result in single points in the plot, whereas Edge Boxes leads to a rising curve with decreasing threshold of the confidence score and rising average number of proposals. By combining all results, the blue curve begins with recall = 0.442 and the average number of initial proposals is equal to 662. At the end of the curve, all proposals from Edge Boxes are also merged into the combined results. However, for an average of 188 969 candidates per image, the maximum recall remains at 0.891. After a close investigation of localization results, labels on ICs are often missing in the localization results. This is problematic in PCB recycling since most valuable elements are related to these components.

Following the text spotting pipeline defined in [207], all initial proposals undergo successive processing steps including initial validation, subsequent refinement and final validation. According to the confidence score
predicted by the final validation classifier, the Precision-Recall Curve (PRC) of the obtained localization results is plotted in Figure 6.9. A dramatic decent in the maximum recall from 0.891 to the final value 0.499 is observed. This is mainly due to the refinement of proposals through bounding-box regression, which reduces the maximum recall from 0.817 after the initial validation to 0.565 before the final validation. Correspondingly, the average number of proposals is also reduced from 34 584 to 5 851. In the final localization results, there are on average 2 821 proposals per image remaining for the subsequent pre-processing and text recognition. As a direct consequence of the low maximum recall value and the large number of proposals, the Average Precision (AP) of the localization results in an extremely low value of 0.114. Recalling the missing labels of ICs, an essential further development of state-of-the-art text spotting approaches is thus required for achieving reasonable localization performance on PCBs.

6.4.3 Retrieving text information

For a comprehensive evaluation of the proposed pre-processing of local text images in Section 6.2, combinations between diverse binarization methods, skew correction and text line segmentation algorithms, as well as OCR engines are considered. The resulting case-insensitive F-score is
summarized in Table 6.1 for a better overview. To demonstrate the significance of the proposed text color estimation, state-of-the-art binarization methods are also extended with this additional analysis. As stated in Section 3.5, it is of great interest to investigate if TCA correction can lead to the desired improvement of the intended analysis of PCBs. To this end, the proposed pre-processing is evaluated with and without the local TCA correction for its better flexibility, respectively. There are in total five algorithms for estimating the skew angle and extracting single text lines involved: no correction (original), EMST-based algorithm (EMST) [231], connected components-based algorithm (CC) [5], the proposed skew correction (SC) and the proposed skew correction followed by single line extraction (SCSLE). The off-the-shelf OCR engine Tesseract-ocr discarding dictionaries and the trained CNN-WRDF model (hereafter referred to as CNN-WRDF) are employed for reading text information in pre-processed images. Tesseract-ocr is applied on binary images as it cannot provide reasonable performance on original images even after skew correction and text line segmentation. On the contrary, CNN-WRDF is applied on original images for avoiding any loss of information through unreliable binarization. Furthermore, to increase the variation in the skew angle, all binarized images are rotated by $0^\circ$, $15^\circ$, $45^\circ$ and $90^\circ$ before undergoing the subsequent skew correction. The mean F-score on all 860 images with four rotation angles is used for quantifying the quality of the obtained OCR results.

The overall best performance with F-score $= 0.804$ is obtained for the proposed pre-processing combined with the local TCA correction and CNN-WRDF. This is followed by the combination only differing in the omitted local TCA correction, for which F-score $= 0.789$ with a moderate quality decrease of 0.015 is obtained. If Tesseract-ocr is adopted, the best two combinations remain the same and a wide margin between their performance and the performance based on state-of-the-art binarization methods is observed. For pre-processing approaches resulting in reasonable OCR performance, the proposed skew correction steadily leads to improved text reading results and the additional segmentation of text lines gives rise to another significant performance boost (more than 0.1). In comparison to the connected components-based algorithm, the EMST-based algorithm often provides better text reading results, even when single text lines are not separated from each other. This is a direct consequence of the great number of parameters employed in the connected
components-based algorithm and lead to less generalizable performance since these parameters must be carefully tuned. This generalization issue can be commonly observed for connected components-based analysis and is considered as its general drawback. With the help of the proposed text color estimation, the performance of state-of-the-art binarization methods is substantially improved. Especially, despite their unreliable binarization performance, the proposed skew correction and text line segmentation algorithm often leads to very promising OCR results if combined with CNN-WRDF. In other words, this combination is able to essentially bridge the gap between different binarization methods. As a result, there exists great flexibility in adapting OCR systems for diverse applications, while the OCR performance is stabilized.

It is noticeable that the text reading results relying on the binarization method [194] is independent of the text color estimation. Through a close investigation, it is found that thin line structures are obtained on both original and complementary images. This consequently leads to undetermined text color and the same results are thus obtained. This further reveals the fact that the text color estimation is applicable only when distinct binarization results arise from original and complementary images. In comparison to the evaluation results presented in [6], different OCR performance is obtained in this thesis, which is due to the difference between the adopted evaluation protocols. In [6], the final results were considered as the optimal estimate between upright and $180^\circ$-rotated images since the confidence score of the retrieved character strings was missing. In this thesis, with the help of the derived confidence score, the asserted optimal estimate is considered as the final output.

In the evaluation of OCR engines, Tesseract-ocr, CNN-WRDF (trained on approximately 2.7 million synthetic images with random character strings), the original CharNet [244] (trained on eight million synthetic images with words from a 90K dictionary) and an adapted CharNet (with comparable architecture to CNN-WRDF and trained on the same unconstrained text samples) are considered. The proposed binarization method combined with the local TCA correction is used to pre-process images before conducting skew correction and text line segmentation. All CNN-based OCR engines are applied on both original and binarized images, while the application of Tesseract-ocr on original images is discarded due to performance issues. For a more in-depth understanding of the pre-
<table>
<thead>
<tr>
<th>OCR engine</th>
<th>skew + text line</th>
<th>original binarization method</th>
<th>extended binarization method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[202]</td>
<td>[194]</td>
<td>[195]</td>
</tr>
<tr>
<td>original</td>
<td>0.118</td>
<td>0.096</td>
<td>0.109</td>
</tr>
<tr>
<td>EMST</td>
<td>0.168</td>
<td>0.179</td>
<td>0.178</td>
</tr>
<tr>
<td>CC</td>
<td>0.224</td>
<td>0.235</td>
<td>0.245</td>
</tr>
<tr>
<td>SC</td>
<td>0.192</td>
<td>0.207</td>
<td>0.208</td>
</tr>
<tr>
<td>SCSLE</td>
<td>0.220</td>
<td>0.292</td>
<td>0.257</td>
</tr>
<tr>
<td>original</td>
<td>0.296</td>
<td>0.296</td>
<td>0.296</td>
</tr>
<tr>
<td>EMST</td>
<td>0.422</td>
<td>0.432</td>
<td>0.418</td>
</tr>
<tr>
<td>CC</td>
<td>0.408</td>
<td>0.393</td>
<td>0.387</td>
</tr>
<tr>
<td>SC</td>
<td>0.323</td>
<td>0.542</td>
<td>0.341</td>
</tr>
<tr>
<td>SCSLE</td>
<td>0.397</td>
<td>0.568</td>
<td>0.417</td>
</tr>
<tr>
<td>original</td>
<td>0.127</td>
<td>0.096</td>
<td>0.136</td>
</tr>
<tr>
<td>EMST</td>
<td>0.302</td>
<td>0.179</td>
<td>0.327</td>
</tr>
<tr>
<td>CC</td>
<td>0.345</td>
<td>0.235</td>
<td>0.405</td>
</tr>
<tr>
<td>SC</td>
<td>0.326</td>
<td>0.207</td>
<td>0.368</td>
</tr>
<tr>
<td>SCSLE</td>
<td>0.419</td>
<td>0.292</td>
<td>0.479</td>
</tr>
<tr>
<td>original</td>
<td>0.296</td>
<td>0.296</td>
<td>0.296</td>
</tr>
<tr>
<td>EMST</td>
<td>0.531</td>
<td>0.432</td>
<td>0.523</td>
</tr>
<tr>
<td>CC</td>
<td>0.555</td>
<td>0.392</td>
<td>0.520</td>
</tr>
<tr>
<td>SC</td>
<td>0.593</td>
<td>0.542</td>
<td>0.594</td>
</tr>
<tr>
<td>SCSLE</td>
<td><strong>0.742</strong></td>
<td>0.568</td>
<td><strong>0.757</strong></td>
</tr>
</tbody>
</table>

Table 6.1: Resulting case-insensitive F-score of diverse pre-processing approaches. Tesseract-ocr is applied on binarized images, while CNN-WRDF on original images. State-of-the-art binarization methods are extended with the proposed text color estimation, while the proposed binarization is extended with the local TCA correction.
sented OCR performance, the corresponding evaluation is conducted on test images either with only one single text line or with multiple text lines individually.

The overall case-sensitive and case-insensitive F-score on single-line test image is depicted in the plots of Figure 6.10, where results arising from the five algorithms for estimating the skew angle and extracting single text lines are distinguished from each other. Since the original CharNet is trained on case-insensitive samples, a substantial variation in its performance is observed. Among the remaining OCR engines, Tesseract-ocr exhibits the least variability between the two plots, while the others suffer from a slight decent in the OCR quality if uppercase letters are explicitly distinguished from lowercase. In consideration of the overall performance, there is in general: CNN-WRDF $>$ adapted CharNet $>$ Tesseract-ocr $>$ original CharNet. The limited performance of the original CharNet has its root in the limited variability of the constrained training character strings coming from the 90K dictionary. After the appropriate adaption, the CNN-based CharNet provides better text reading results than Tesseract-ocr if applied on original images. In the case of connected components-based analysis with a removal of non-character objects, Tesseract-ocr is superior to the adapted CharNet. These two observations together reveal the fact that CNN-based models are more stable to background distractions, while Tesseract-ocr better tolerates the variation in shape of binarized characters. This is a direct consequence of the binary sample-based shape analysis in training Tesseract-ocr. Benefiting from the novel architecture for better localization performance, CNN-WRDF is steadily superior to CharNet in all cases and the best performance is obtained by applying CNN-WRDF on original images after the proposed skew correction and text line segmentation: F-score = 0.797 for case-sensitive and F-score = 0.820 for case-insensitive.

As depicted in the plots of Figure 6.11, a great decrease in the overall OCR performance is obtained, especially for those algorithms (original, EMST and SC) without additional text line segmentation. More surprisingly, Tesseract-ocr becomes superior to CNN-WRDF and CharNet in such cases. This effect is due to the native layout analysis embedded in Tesseract-ocr, which leads to better text reading results by slicing images into single lines. However, if the proposed full algorithm is applied, a much better layout analysis than the native variant can be achieved, with F-score = 0.467 for SC + Tesseract-ocr increased to F-score = 0.659.
for SCSLE + Tesseract-ocr. In comparison to single-line test images, the best OCR performance on images with multiple text lines is obtained by applying CNN-WRDF on binarized images after the proposed skew correction and text line segmentation: F-score = 0.705 for case-sensitive and F-score = 0.733 for case-insensitive. Moreover, the difference between CNN-WRDF and the adapted CharNet becomes less significant.

Besides the global evaluation, the OCR performance is also individually evaluated for each of the four additional rotation angles applied on the original 860 local text images: $0^\circ$, $15^\circ$, $45^\circ$ and $90^\circ$. In Figure 6.12 with the results obtained by applying Tesseract-ocr as well as CharNet on binarized single-line test images and applying CNN-WRDF on original images, stable performance is observed across different angles and skew correction algorithms. Better results are in general available for the two angles $0^\circ$ and $90^\circ$, while $15^\circ$ and $45^\circ$ suffer from less accurate skew angle estimation. This reflects the fact that text in most original test images without additional rotation has either a horizontal or a vertical running direction. In Figure 6.13 with the results obtained on test images with multiple text lines, the proposed full skew correction and text line segmentation algorithm is essentially superior to alternative algorithms. Relying on the binarized images with less ambiguity, the original CharNet gives rise to better OCR performance in the case of connected components-based analysis than CNN-WRDF applied on original images, where low quality images emerge from the less reliable line segmentation results.

To also investigate the generalizability of the CNN-based OCR engines, they are applied for the word recognition task of the second challenge in the ICDAR 2015 robust reading competition [246], where 1,095 words cropped from natural scene images and with nearly horizontal running direction are to be recognized. Apparently, the original CharNet trained on the 90K dictionary and more samples (eight millions vs. 2.7 millions) is advantageous since the character strings presented in natural scenes are commonly from dictionaries and thus constrained. As summarized in Table 6.2, CNN-WRDF provides comparable results to the original CharNet, whereas the adapted CharNet leads to far less reliable recognition results. It should be emphasized that, in comparison to earlier commercial OCR
Figure 6.10: Text recognition results on images with one single text line. The listed OCR engines (represented by bars in different colors) are applied on images undergoing different skew corrections: original, EMST, CC, SC and SCSLE. Since OCR is performed either with or without binarization, “BW” is used to denote those OCR engines applied on binarized images.

206
Figure 6.11: Text recognition results on images with multiple text lines. The listed OCR engines (represented by bars in different colors) are applied on images undergoing different skew corrections: original, EMST, CC, SC and SCSLE. Since OCR is performed either with or without binarization, “BW” is used to denote those OCR engines applied on binarized images.
Figure 6.12: Text recognition results on rotated binary images with one single text line. Three OCR engines (Tesseract-ocr, CharNet and CNN-WRDF) are combined with different skew correction algorithms: original, EMST, CC, SC and SCSLE. The applied rotation angles of binary images are 0°, 15°, 45° and 90°, which are represented by bars in different colors.
Figure 6.13: Text recognition results on rotated binary images with multiple text lines. Three OCR engines (Tesseract-ocr, CharNet and CNN-WRDF) are combined with different skew correction algorithms: original, EMST, CC, SC and SCSLE. The applied rotation angles of binary images are 0°, 15°, 45° and 90°, which are represented by bars in different colors.
Chapter 6. Text analysis

<table>
<thead>
<tr>
<th>OCR engine</th>
<th>case-sensitive</th>
<th>case-insensitive</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>normalized edit distance</td>
<td>correct words</td>
</tr>
<tr>
<td>original CharNet</td>
<td>622.6</td>
<td>19.36%</td>
</tr>
<tr>
<td>adapted CharNet</td>
<td>280.3</td>
<td>50.96%</td>
</tr>
<tr>
<td>CNN-WRDF</td>
<td>203.6</td>
<td>68.04%</td>
</tr>
<tr>
<td>CNN-WRDF + skew correction</td>
<td>238.8</td>
<td>66.12%</td>
</tr>
<tr>
<td></td>
<td>normalized edit distance</td>
<td>correct words</td>
</tr>
<tr>
<td></td>
<td>194.3</td>
<td>70.87%</td>
</tr>
<tr>
<td></td>
<td>258.5</td>
<td>52.69%</td>
</tr>
<tr>
<td></td>
<td>188.9</td>
<td>69.22%</td>
</tr>
<tr>
<td></td>
<td>226.1</td>
<td>67.12%</td>
</tr>
</tbody>
</table>

Table 6.2: Constrained word recognition performance in the ICDAR 2015 robust reading competition.

Engines, superior performance can be obtained using CNN-based models trained on pure synthetic data. It is also interesting to see that the proposed skew correction is able to stabilize the OCR performance even given unconstrained skew direction.

6.5 Discussion

Although state-of-the-art text spotting approaches are able to provide desirable performance on natural scene images available from public OCR datasets, unsatisfactory localization results are, however, observed on PCBs. As demonstrated in Section 6.4.2, a huge number of bounding boxes must be localized in the initial spotting stage through an exhaustive proposal generation so that most presented text objects can be captured. Due to the large amount of proposals, the following analysis becomes hardly affordable for the intended information retrieval in PCB recycling. Even after the multi-stage validation and refinement, the number of remaining proposals is still substantially higher than the number of actual text objects and gives rise to great difficulties in reading correct text information while rejecting false candidates by applying OCR engines.
To address this generalization issue of state-of-the-art text spotting approaches for complex scenes, more text-specific detectors need to be developed to replace general or less specific detectors such as Edge Boxes, MSER. A patch-based classification combined with the coarse-to-fine localization [258,259] can be typically used for this purpose. Nevertheless, such approaches suffer from high computational complexity arising from the exhaustive search over possible scales, aspect ratios and rotation angles. Considering the well-established image analysis paradigm in the frequency domain through e.g. Fourier transform and wavelet transform, distinct features can be efficiently extracted over scales, aspect ratios and orientations, which are less sensitive to the illumination condition and the image quality. The applicability of this idea has been confirmed in [252], where features extracted through the windowed discrete cosine transform were combined with heuristic-based decision rules for detecting text in images. With the help of more advanced analysis methods in the frequency domain and exploiting the power of deep learning, the desired general text spotting approach is thus realizable. Aiming at a closed form for adapting OCR systems for new tasks, the introduced tool [248] for generating synthetic training samples should be combined with the configuration according to the intended application, which then correspondingly places random text sample in background images for the training purpose. As a result, OCR systems can be conveniently adapted or extended for new applications through modifying the configuration file of the data generation tool and retraining deep learning models.

In consideration of the limited performance of state-of-the-art text spotting approaches and the resulting high computational complexity, it is a more practical solution to focus on text present on localized components. This analysis is able to extract the most relevant text information on PCBs since the composition of valuable elements depends dominantly on the mounted components, especially ICs. As confirmed through the evaluation on local text images, the related information retrieval turns out to be more reliable using the pre-processing methods and the OCR engine introduced in this thesis. Moreover, the proposed pre-processing is applicable for general OCR tasks since it does not make any specific assumptions or require additional parameter tuning. Its only limitation is the requirement on uniform skew angle in images. For multiple skew angles of text, a clustering of the direction vectors can be used to assign subregions to the corresponding skew directions, which is followed by the
Chapter 6. Text analysis

same skew correction and text line segmentation for characters associated with single skew angles. The proposed CNN-WRDF model for reading text in cropped images is better generalizable compared to those models trained on constrained text strings. For new applications, an adaption of the employed dictionaries or lexicons is sufficient, while the expensive retaining of the CNN-WRDF model is unnecessary. This also results in a more convenient realization of global performance improvement since the CNN-WRDF model can be centrally modified through more advanced architectures and more extensive training data. This modified model is then redistributed into diverse OCR systems without resulting in additional modifications for specific applications.
Chapter 7

Conclusion

7.1 Summary

Regarding the huge quantities of valuable materials and hazardous substances contained in waste PCBs, an appropriate recycling of them is highly recommended, for both economic and environmental reasons. Nevertheless, due to the high complexity and variability in their material composition, state-of-the-art industrial recycling systems commonly suffer from an unstable recycling performance resulting in a suboptimal eco-efficiency. This fact prevents a wide application of such recycling systems for a comprehensive recovery of valuable materials and a good control of hazardous substances, as the high investment of industrial infrastructure becomes less affordable in the absence of optimal eco-efficiency.

The best solution for maximizing the overall eco-efficiency is a selective and dynamic recycling of PCBs, where fractions containing different materials are sent to the correspondingly configured processing routes. In the most simplified case, targets are automatically classified and separated according to the quantity of recoverable gold. They are sent to different processing routes for recovering gold or other materials, where the maximized eco-efficiency is achieved. To facilitate the desired recycling described above, a continuous and automated retrieval of the material composition is necessary. As the best suited method, an image-based
analysis of PCBs and the subsequent retrieval of relevant material information from a comprehensive data bank are gaining attention. However, due to the technical challenges imposed by the high complexity and variability of PCBs, a practical solution for realizing the intended analysis was not available before this thesis.

Through the research work conducted in this thesis, a realistic PCB analysis is presented for the first time, where the overall complexity and diversity are substantially reduced by performing the analysis at the level of single components instead of entire PCBs. This establishes the necessary technical support for realizing indirect retrieval of the sought material composition. Moreover, a systematic investigation of all relevant aspects for achieving the desired analysis is performed, which covers the characterization of the employed imaging system, the localization and categorization of single components, as well as the extraction of text information. By this means, achievements, limitations and potentials are identified for each aspect, which are essential for the further development of the image-based PCBs analysis.

The very first step of the overall analysis is to recover the spatial and color information in images. This is accomplished by the characterization of the employed imaging system. Rather than an offline characterization, online calibration approaches are preferred in this thesis for their better flexibility. Starting with general cases, a novel reformulation of state-of-the-art correction models for dealing with lens distortion in images is proposed and leads to more accurate results. This is then followed by the camera self-calibration, where the camera’s intrinsic parameters are estimated. In consideration of independent motions of objects in general image sequences, state-of-the-art methods and approaches assuming rigid scenes are accordingly extended. However, degenerate motions and structures are commonly observed in PCB recycling, for which the application of general self-calibration approaches is infeasible. To deal with this special case, a single image-based calibration approach is developed, which exploits the geometric constraints available on PCBs. After geometric calibration, color information is restored through the correction of chromatic aberration in images. Both global and local correction methods are considered for this purpose. Especially, the proposed global correction of transverse chromatic aberration has wide applicability due to its generic formulation of lens distortion and can be integrated into standard camera calibration workflows without great effort.
As demonstrated through the quantitative evaluation in Figure 3.22, only a subset of the camera’s intrinsic parameters can be estimated using the geometric constraint-based online calibration. Depending on the initial estimates and the quality of the detected lines as well as ellipses in images, the normalized estimation error varies primarily between 5 and 20%. With reasonable assumptions of the remaining intrinsic parameters, e.g. square pixels and the principal point at the image center, the online calibration results can be used to derive a qualitative assertion of spatial information in PCB recycling, while an accurate assertion is only achievable using offline approaches combined with well-designed calibration patterns. As confirmed through the comparison results in Table 3.2, 3.3 and 6.1, the correction of chromatic aberration leads to better consistency between different color channels and improves image quality, which consequently gives rise to improved image analysis performance.

To realize the intended analysis of single components, approaches based on both specific and general information are proposed. Moreover, for avoiding additional difficulty arising from unknown PCB orientation, an effective and reliable approach is designed to rotate PCBs back to a horizontal or vertical orientation. In the case of surface-mounted devices, assembly print and color information are employed for facilitating the desired segmentation of small devices and ICs, respectively. Towards a general analysis of PCB components, more advanced localization approaches are proposed. Besides a combinatory localization approach relying on the appropriate combination of a wide spectrum of proposal generation algorithms, an alternative approach resulting in a more compact realization is also presented, which exploits the underlying image content using a novel local variance-based analysis and localizes components through a non-parametric thresholding. All obtained proposals are refined and validated through CNNs, where a novel network architecture is proposed to provide superior localization performance. Finally, all localized objects are categorized into different component groups with the help of appropriate features and classifiers. For achieving the optimal classification performance at different stages, a comprehensive evaluation of features and classifiers is also performed.

Through the extensive evaluation results presented in chapters 4 and 5, satisfactory localization performance with F-score > 0.800 is confirmed for the surface-mounted small devices using assembly print and for general components using the compact approach. Especially, the obtained
localization performance is resistant to variations in imaging conditions, which is an important prerequisite for practical applications. According to the comprehensive evaluation results of combinations between different features and classifiers in Table 5.8, promising categorization performance with F-score $> 0.850$ is also confirmed.

Aiming at providing additional information of target PCBs and components, text analysis for reading relevant label information is also conducted. In comparison to text spotting in whole images, text recognition in local images, for instance in the cropped images of localized components, is of greater interest as the presented label information is related to the underlying components and can be used for identification. To address the challenges arising from varying color, font, size, orientation and layout of text, novel pre-processing methods are proposed, including adaptive thresholding of text objects, reliable skew correction and extracting single text lines. A convolutional neural network realized in the proposed architecture for better localization performance is employed for recognizing single characters in the pre-processed local images.

According to the evaluation results in Figure 6.9, state-of-the-art text spotting approaches are unable to extract relevant text information as in the case of natural scene images, which is due to the high complexity of PCBs. If the intended text analysis is focused on local images, promising OCR performance with F-score $> 0.800$ is observed in Table 6.1. In comparison to state-of-the-art text recognition, the obtained performance boost essentially relies on the proposed pre-processing methods, the novel network architecture and the correction of chromatic aberration.

As an overall conclusion, general PCB components are best localized through the compact approach relying on local variance-based analysis and bounding-box regression. The assembly print-guided segmentation is optional for surface-mounted small devices. All localized components are subsequently categorized using predefined features and classifiers. Additional information for identifying components is retrieved by reading the label information in the corresponding local images. Depending on the desired flexibility and accuracy, online or offline approaches for recovering spatial and color information in images should be considered. A very desirable property of this implementation is the resulting good extensibility of the overall analysis. Due to the highly dynamic production of PCB components, new recycling targets steadily emerge in recycling pro-
cesses. To adapt the overall analysis for the newly emerging targets, only a few data-driven models need to be retrained, which is straightforward for given adequate training data. Explicit parameter tuning is in general unnecessary as the remaining proposal generation and text analysis are either not affected or featured with automated parameter adaption as required.

In consideration of the associated high complexity and variability in size, shape, color and texture, PCBs and the mounted components generally give rise to a very challenging application scenario in computer vision. To address this problem, generic and adaptive analysis has been considered throughout the entire development process. As a result, most of the obtained methods, algorithms and approaches are not limited to the specific application on PCBs, but are also applicable in more generic tasks.

7.2 Outlook

Although a realistic and systematic analysis of waste PCBs is proposed and implemented for the first time in this thesis, some further aspects can be considered for improving the overall performance and achieving a more comprehensive information retrieval.

The realized analysis merely utilizes 2D information in RGB images. If additional depth information of the corresponding pixels is also provided, more accurate and efficient component analysis can be expected. This has been partially demonstrated in a previous publication [260], where a compact sensor for acquiring RGB-D data was applied. Despite the inadequate spatial resolution and the limited accuracy of depth measurements, some significant components exhibiting non-uniform colors and irregular shapes were efficiently localized even using straightforward analysis methods. Nevertheless, an efficient acquisition of accurate and dense depth information for supporting a comprehensive analysis of all components, especially those of tiny sizes, is very challenging.

An alternative method for localizing components exhibiting non-uniform colors and irregular shapes is the analysis of symmetric structures, which are predominantly found on PCB components. As investigated in a Master’s thesis [261], a feature-based symmetry analysis can overcome the difficulties associated with region-based localization approaches since the de-
tection of features is not affected by the color inhomogeneity or the shape
irregularity. However, state-of-the-art symmetry analysis approaches are
unable to provide reliable detection results on PCBs due to the high com-
plexity. More advanced approaches are thus required for localizing PCB
components with the help of a reliable symmetry analysis.

Besides single components, component clusters for accomplishing specific
functionality can also be considered in the analysis of PCBs. They exhibit
in general limited variability and often indicate the existence of specific
elements or materials. For utilizing them in the desired information re-
trieval, further research work on the detection of such component clusters
and on the associated material information is necessary.

It should be emphasized that the presented PCB analysis is intended as
a proof-of-concept investigation and serves as the essential basis for the
realization of an industrially mature information retrieval system. For
accomplishing this final objective, some open questions must be answered.

The desired material analysis relies on matching detected PCB compo-
nents to reference samples saved in a comprehensive data bank with their
material composition, where the correspondence between test and refer-
ence samples is to be determined with respect to their category assign-
ments, appearance and related label information. However, the effective-
ness of such indicative information in matching objects of interest to the
great number of entities in the data bank is unknown. Moreover, the op-
timal organization of visual information for achieving reliable matching
performance is also a subject of ongoing research.

In comparison to the laboratory implementation of the overall analysis, a
couple of practical issues need to be addressed in realizing the correspond-
ing industrial implementation. A typical obstacle in such laboratory-to-
industry transfer is the demand for the real-time capability. Nevertheless,
this demand is not uniformly defined and varies according to the specific
time constraint of the intended application. In the case of PCB recycling,
the time available for conducting the image-based information retrieval
depends on the size of PCBs, the throughput of recycling routes and the
construction of recycling systems. To get a better control of such issues
emerging in practical applications, a close collaboration between experts
for image analysis, recycling and machine construction must be estab-
lished.
Appendix A

Tool for generating reference data

Since the intended image-based information retrieval should be realized in an automated manner, appropriate validations and evaluations based on available reference data for demonstrating the feasibility of the developed analysis methods and for assessing the quality of the obtained analysis results are indispensable. Typically for a supervised investigation of segmentation methods, the obtained segmentation results, *e.g.* foreground and background region proposals, should be compared with the manually labelled ground truth regions and evaluated using application-specific performance measures [262, 263], where the ground truth, *e.g.* the reference data, plays a key role. Also for many data-driven analysis and optimization approaches, for instance classification and fine tuning of parameters, reference data are a most essential prerequisite. With these considerations, it becomes necessary to figure out, how we can practically get access to the desired reference data.

Generally, there are two possibilities to obtain reference data: using known ground truth or performing a manual annotation. The former method is widely applied for well-controlled problems, where the data forming process and the objects of interest are known. Typically, validations of industrial quality control systems can benefit from this method by using data from inspection targets with known defects. In contrast, the latter
method does not rely on any additional knowledge about data and thus leads to a more generic application. However, the cost paid for the wider applicability is the much higher complexity. Often, the annotation process is very tedious since each object of interest must be precisely labelled in a huge amount of data.

As stated in Chapter 2, the major objectives of the proposed information retrieval are to localize and to categorize the presented PCB components, as well as to read text information on PCBs. To this end, positions of sought objects along with relevant meta information, e.g. category of components, recognized text, etc., are expected in the provided reference data, which should be manually annotated in the obtained images since prior knowledge of the target PCBs is in general not available. To simplify this tedious annotation process and to increase the overall productivity, a novel interactive segmentation tool with high usability and adequate flexibility has been developed in two consecutive Master’s theses [264,265] for assisting users in generating reference data.

To address the motivations for developing a new segmentation tool, a review of available tools for the same or similar purposes is given as the first part of this chapter. Subsequently, by tackling the problems presented by standard tools, the principal concepts and the detailed implementation of the novel segmentation tool are presented. For investigating the usability and the flexibility of the developed tool, a qualitative evaluation is conducted on PCB images and demonstrates the desirable properties of this tool. Further discussion is also given at the end of this chapter and clarifies the relevance of the developed tool for general applications.

A.1 State of the art

Aiming at providing the desired functional features in different application domains, dozens of tools are available for annotating the ground truth in images. Regarding the associated workflows of these tools, they can be grouped into three major categories: image-editing programs, dedicated segmentation tools and dataflow-based frameworks.

Image-editing programs are primarily developed to provide users with the opportunities to manipulate images even without knowledge about image
processing and programming. Amongst such programs, *Adobe Photoshop* [266] and the *GNU Image Manipulation Program* (GIMP) [267] are most commonly used due to their excellent usability and wide applicability. Often there are numerous functions integrated in image-editing programs for selecting certain regions in images for further manipulation. Typically, such functions are the rectangle and ellipse selection, the *Magic Wand*, the *Lasso* (free selection) and the *Intelligent Scissors* [268, 269, 270]. They are especially suited for generating the desired reference data by explicitly defining the corresponding masks of the ground-truth objects in images. A further quite desirable feature is the flexible management of the obtained masks, which is realized by employing the layer mechanism, where the masks are individually or jointly saved in user-defined layers. Thus, the ground-truth objects can be arbitrarily grouped to achieve a flexible labelling of the presented multiple classes of objects. In contrast to their excellent usability, adequate flexibility for customizing the labelling process is often missing. If any desired selection function is not provided, the only solution is to write your own plug-in adjusted to the special structure predetermined by the vendors. Defining a sophisticated pipeline consisting of a series of consecutive operations is hardly possible, which is, however, essential for dramatically reducing the complexity of the labelling work. Furthermore, the functionality for integrating meta information into the reference data is usually not provided.

Besides the image-editing programs suited for the general purpose, there are also dedicated tools for annotating reference objects in specific image data, for instance labelling cells in slides for biomedical analysis. Specialized tools, such as Ilastik [271], ITK-SNAP [272] and MITK [273], are covered by this category. Due to the fact that the objects of interest for such applications exhibit less variability, fixed annotation procedures with the optimized work efficiency are employed in such tools. As a trade-off between efficiency and flexibility, modifications and extensions to the established workflows can only be realized with extensive efforts. Further common weaknesses of the dedicated segmentation tools are moderate interactive functions and poor management of the generated reference data.

To deal with a wide spectrum of image analysis tasks, *MeVisLab* [274] and *MiToBo* [275] offer a large number of fundamental image processing functions. They also employ the dataflow [276] concept for rapidly prototyping application-specific solutions. By selecting appropriate functions as nodes and connecting them to form a processing pipeline, image data
flow through all nodes in the pipeline and the analysis results are obtained at the pipeline output. A highly flexible interactive segmentation tool can be implemented in this manner if interactive selection and labelling operations are also provided as fundamental functions. Since this is rather a general scheme for solving image processing problems, the usability of interactive operations and the management of reference data are far from satisfying.

A.2 A better segmentation tool for PCBs

Good usability and segmentation performance, flexible data management, as well as good accessibility and extensibility are necessary key features of an appropriate segmentation tool for achieving convenient annotations on PCB images exhibiting high complexity and variability. However, as stated in the review of standard segmentation tools, none of them can provide all of these features. This raises therefore demand for developing A Better Segmentation Tool (ABeST).

A.2.1 Concepts

After the identification of need, the next step of software development is to clarify the requirements on the software. Considering a typical workflow for labelling reference objects in PCB images, the user first marks some pixels or regions on the objects of interest as well as on the background. The user input is utilized as seed points and fed to an appropriately defined segmentation pipeline, which automatically generates segmentation results. In case of unsatisfying segmentation results, convenient modification functions are employed for assisting in manual correction. Especially, modifications to the intermediate results of the pipeline are possible. By assigning corresponding labels to segmented objects in images, they can be arbitrarily grouped to yield the desired categorization. Moreover, meta information can be attached to or removed from any segments effortlessly. Regarding the intended annotation procedure, essential requirements on ABeST are summarized as follows:

1. practical and convenient functions for selecting seed pixels/regions, selecting regions of interest and modifying segmentation results;
2. flexible pipeline generation by graphically organizing functional units using drag-and-drop, as well as reuse of predefined pipelines;

3. individual user interfaces of functional units for setting parameters;

4. straightforward import of existing segmentation/analysis functions for generating additional functional units, which are implemented in diverse programming languages;

5. interrupting the employed pipeline before arbitrary functional units and manipulating the intermediate or final results;

6. attaching arbitrary meta information to segmented objects and convenient data management, e.g. modification, search, etc.;

7. resuming a saved labelling process.

To meet the listed requirements, some entities with special properties are designed for realizing the intended functionality of ABeST.

**Interactive tool** For obtaining good usability similarly to image-editing programs, the same or equivalent pixel/region selection tools are integrated. Further convenient tools are also provided for selecting the regions of interest or for modifying the obtained segmentation results.

**Plug-in** The key for dramatically reducing the complexity of the labelling work is exploiting the power of automated analysis functions, which are included as plug-ins into the proposed segmentation tool. Any implementations of such functions can be imported as plug-ins using the provided wrapping tool and individual user interfaces are automatically generated for manual configuration. Each plug-in is visualized as a functional unit and supports the drag-and-drop operation. Breakpoints can be set to functional units to halt the automated analysis.

**Pipeline** By graphically connecting functional units the user defines customized pipelines for improving the productivity or assisting in difficult annotation tasks. The generated pipelines can be saved and loaded again for reuse.
Appendix A. Tool for generating reference data

**Layer**
Inspired by the layer mechanism of the image-editing programs, layers are employed for managing single or grouped objects. Each layer carries the masks of the grouped objects, where groups of single objects are also possible. For flexible data management, layers can be further individually modified, decomposed or merged. Additional meta information is embedded in the layers.

**Project**
The labelling process of each individual image is considered as an independent project. After saving a project including all parameters and the currently defined pipeline along with intermediate results, the user can resume the labelling process at any time.

### A.2.2 Implementation

The very first question in the implementation is: which programming language is best suited for developing the new segmentation tool? Since ABeST is based on interactive operations, the programming of *Graphical User Interfaces (GUIs)* must be well supported. Moreover, the programming must be well accessible for most programmers and a wide range of image processing functions should be provided for the selected programming language. Python™ [249] is found to be able to satisfy these requirements by using Qt [277] via the Python binding PyQt [278], using OpenCV [56] and Scipy [279]. The resulting application software with all major components is illustrated in Figure A.1.

Using the menu bar, the user can load a new image for the annotation purpose or resume any saved labelling process. For the objects to be segmented, an appropriate segmentation pipeline is to be defined in the pipeline window, where essential functional units for realizing the desired segmentation can be found in the plug-in window and are visualized as blocks for graphical organization. Useful functions, *e.g.* Watershed [203], GrabCut [149], Lazy Snapping [148], GrowCut [280], are available for simplifying the tedious annotation task. Further functional units with customized segmentation functions and implemented using C/C++, Python or Matlab® [57] can be easily imported using the provided wrapper utility. Following the instructions in the template file, the input and output
A.2. A better segmentation tool for PCBs

Figure A.1: An overview of the developed ABeST. All major components are identified with numbers: 1. menu bar; 2. interactive tools; 3. plug-ins; 4. layers; 5. pipeline definition; 6. global segmentation results; 7. temporal segmentation results; 8. status bar.

data, as well as the programming language are defined in the XML [281] format for automatically wrapping the customer function. In the window of interactive tools, practical functions for positioning and region selection/modification are provided. With them the user can conveniently visualize the region of interest in the middle of the screen and manually define some seed pixels/regions as the input to the generated segmentation pipeline. All segmented objects are overlaid on the original image and illustrated in the window of global segmentation results. The visualization color, label and meta information of objects are assigned to the corresponding entries in the window of layers for a convenient management. For investigating the intermediate segmentation results, an additional window is used to visualize the currently segmented objects. The temporary results are integrated into the global segmentation results after the modification and the confirmation made by the user. Valuable information for assisting in labelling is presented in the status bar at the bottom of the main window. For achieving the desired flexible data management, all data are organized in the HDF5 [282] format and an external tool for managing the saved data is also available.
Appendix A. Tool for generating reference data

Table A.1: A qualitative comparison between segmentation tools. The most important four aspects listed in the first row of this table are considered to assess the applicability of these tools. Their performance in each aspect is quantified into three levels: high (+), medium (◦) and low (−).

Table A.1: A qualitative comparison between segmentation tools. The most important four aspects listed in the first row of this table are considered to assess the applicability of these tools. Their performance in each aspect is quantified into three levels: high (+), medium (◦) and low (−).

### A.3 Evaluation and application

To assess the quality of ABeST vs. other segmentation tools, an appropriate evaluation is to be conducted. Regarding the intended application for labelling objects in PCB images, the most important four aspects, i.e., usability, segmentation performance, data management and accessibility/extensibility, have been considered for achieving a qualitative comparison. Aiming at a detailed analysis, there are three levels defined for describing the performance in each aspect: high, medium and low. Evaluation results obtained for ABeST and for commonly used tools are presented in Table A.1 for the reader’s convenience.

As stated in Section A.1, Photoshop and GIMP provide the best usability. Similarly, ABeST exhibits the top performance in this aspect thanks to the convenient interactive tools. To segment diverse components in images, sophisticated and reconfigurable segmentation pipelines are necessary. This requirement is well satisfied only using MeVisLab, MiToBo and ABeST, which employ the dataflow concept and provide fundamental image processing functions for realizing the desired segmentation. However, none of standard tools enables the desired data management, while with ABeST the user can generate and modify data in a very flexible manner. Moreover, as a non-commercial product and being implemented using the
A.3. Evaluation and application

<table>
<thead>
<tr>
<th>component</th>
<th>number of samples</th>
<th>size (in pixels)</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>minimum</td>
<td>maximum</td>
<td>mean</td>
<td></td>
</tr>
<tr>
<td>battery</td>
<td>9</td>
<td>65 083</td>
<td>91 066</td>
<td>83 292</td>
<td></td>
</tr>
<tr>
<td>capacitor</td>
<td>363</td>
<td>1 024</td>
<td>25 990</td>
<td>11 075</td>
<td></td>
</tr>
<tr>
<td>cooling element</td>
<td>22</td>
<td>184 292</td>
<td>1 198 036</td>
<td>399 066</td>
<td></td>
</tr>
<tr>
<td>connector</td>
<td>38</td>
<td>17 877</td>
<td>292 454</td>
<td>98 216</td>
<td></td>
</tr>
<tr>
<td>CPU</td>
<td>7</td>
<td>516 148</td>
<td>1 470 192</td>
<td>777 736</td>
<td></td>
</tr>
<tr>
<td>diode</td>
<td>11</td>
<td>1 735</td>
<td>16 034</td>
<td>8 034</td>
<td></td>
</tr>
<tr>
<td>IC</td>
<td>203</td>
<td>2 402</td>
<td>254 150</td>
<td>23 258</td>
<td></td>
</tr>
<tr>
<td>inductor</td>
<td>45</td>
<td>10 622</td>
<td>41 573</td>
<td>23 553</td>
<td></td>
</tr>
<tr>
<td>LED</td>
<td>6</td>
<td>1 711</td>
<td>6 586</td>
<td>5 105</td>
<td></td>
</tr>
<tr>
<td>oscillator</td>
<td>33</td>
<td>4 970</td>
<td>31 800</td>
<td>10 860</td>
<td></td>
</tr>
<tr>
<td>slot</td>
<td>318</td>
<td>2 666</td>
<td>304 959</td>
<td>69 650</td>
<td></td>
</tr>
<tr>
<td>transistor</td>
<td>139</td>
<td>1 536</td>
<td>39 992</td>
<td>11 325</td>
<td></td>
</tr>
<tr>
<td>other</td>
<td>30</td>
<td>2 601</td>
<td>670 069</td>
<td>60 360</td>
<td></td>
</tr>
<tr>
<td>total</td>
<td>1 224</td>
<td>1 024</td>
<td>1 470 192</td>
<td>44 541</td>
<td></td>
</tr>
</tbody>
</table>

Table A.2: Statistics of the dataset “PCB General”.

...scripting language Python, ABeST is well accessible for users and programmers. With the provided user-friendly wrapper utility, ABeST can be easily extended by integrating further image analysis functions.

Finally, ABeST was applied to assist users in interactively labelling diverse PCB components. In 31 images, all significant components are segmented, where their image pixels have been correspondingly labelled. Specifications, e.g. class/category, text, color, shape, are saved in the HDF5 format and can be retrieved for the annotated objects. A total of 1 224 components with their size varying between 1 024 and 1 470 192 pixels are obtained. For the reader’s convenience, a statistical overview of the segmented objects is presented in Table A.2. To distinguish this dataset from others, it is referred to as “PCB General” in this thesis.

227
Appendix A. Tool for generating reference data

A.4 Discussion

Although ABeST was originally developed for solving the PCB annotation problem, due to its favorable features listed in Table A.1 it is also preferable in general annotation tasks. Interestingly, ABeST can be further applied for rapidly prototyping segmentation algorithms since sophisticated pipelines are generated via a couple of drag-and-drop operations and it is straightforward to import additional image analysis functions implemented in different programming languages.

Occasionally, images may contain a great number of objects which are difficult to segment using available image analysing functions and interactive tools. This is still a challenging scenario for ABeST and usually results in extremely tedious labelling work. A possible solution is to combine classification with online learning techniques [283]. By exploiting the manual feedback given to the current segmentation results, classifiers are incrementally trained to improve the correct classification rate of pixels/patches from different objects. After several iterations with a few manually labelled samples, the desired segmentation can be achieved.

Good classification results usually require huge datasets with hundreds of thousands of samples from different classes. Such datasets are accessible only if a large number of workers take part in the annotation task. To alleviate possible hardware and software barriers related to using the segmentation tool on different computers, it is a practical solution to access the labelling process over internet. Users interact with the tool via a web application, while the image processing and the data management are conducted on some remote servers.
Appendix B

Correction of lens distortion

B.1 Polynomial model using \{k_1\}

B.1.1 Proof of \(h_T^3 = [0, 0, \eta]\)

According to Eq. 3.42, there is equivalently

\[
\mathcal{H} \cdot \mathbf{x}_N = \mathcal{H} \cdot \begin{bmatrix} \mathbf{x}_{ND} \cdot (1 + k_1 \cdot \mathbf{r}^2) \\ \mathbf{y}_{ND} \cdot (1 + k_1 \cdot \mathbf{r}^2) \\ 1 \end{bmatrix} = \eta \cdot \begin{bmatrix} \mathbf{\hat{x}}_{ND} \cdot (1 + \hat{k}_1 \cdot \mathbf{\hat{r}}^2) \\ \mathbf{\hat{y}}_{ND} \cdot (1 + \hat{k}_1 \cdot \mathbf{\hat{r}}^2) \\ 1 \end{bmatrix} = \eta \cdot \mathbf{\hat{x}}_N, \quad (B.1)
\]

where \(\mathcal{H} = [\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3]^T\) with row vectors \(\{\mathbf{h}_1^T, \mathbf{h}_2^T, \mathbf{h}_3^T\}\) is a 3×3 invertible matrix and the non-zero scale factor \(\eta\) is equal to the inner product of the two vectors \(\mathbf{h}_3 = [\mathbf{h}_{31}, \mathbf{h}_{32}, \mathbf{h}_{33}]^T\) and \(\mathbf{x}_N = [\mathbf{x}_N, \mathbf{y}_N, 1]^T\):

\[
\eta = \mathbf{h}_3^T \cdot \mathbf{x}_N = \mathbf{h}_{31} \cdot \mathbf{x}_N + \mathbf{h}_{32} \cdot \mathbf{y}_N + \mathbf{h}_{33}. \quad (B.2)
\]

Substituting the transformation matrix \(\mathcal{T}\) From Eq. 3.41 into Eq. 3.40, the x and y coordinates of the point \(\mathbf{\hat{x}}_{ND} = [\mathbf{\hat{x}}_{ND}, \mathbf{\hat{y}}_{ND}, 1]^T\) are linear combinations of \(\{\mathbf{x}_{ND}, \mathbf{y}_{ND}, 1\}\):

\[
\begin{align*}
\mathbf{\hat{x}}_{ND} &= \mathbf{x}_{ND} + t_s \cdot \mathbf{y}_{ND} + t_p, \\
\mathbf{\hat{y}}_{ND} &= t_a \cdot \mathbf{y}_{ND} + t_q.
\end{align*} \quad (B.3)
\]
Appendix B. Correction of lens distortion

Since \( r^2 = x_{ND}^2 + y_{ND}^2 \) and \( \hat{r}^2 = \hat{x}_{ND}^2 + \hat{y}_{ND}^2 \), the x and y coordinates of the points \( \mathbf{x}_N = [x_N, y_N, 1]^T \) and \( \hat{\mathbf{x}}_N = [\hat{x}_N, \hat{y}_N, 1]^T \) are written as

\[
\begin{align*}
\mathbf{x}_N &= x_{ND} + k_1 x_{ND}^2 + k_1 x_{ND} y_{ND}, \\
y_N &= y_{ND} + k_1 x_{ND}^2 + k_1 y_{ND}, \\
\hat{\mathbf{x}}_N &= (x_{ND} + t_s y_{ND} + t_p) + \hat{k}_1 (x_{ND} + t_s y_{ND} + t_p)^3 + \\
&\quad \hat{k}_1 (x_{ND} + t_s y_{ND} + t_p) (t_a y_{ND} + t_q)^2, \\
\hat{y}_N &= (t_a y_{ND} + t_q) + \hat{k}_1 (x_{ND} + t_s y_{ND} + t_p)^2 (t_a y_{ND} + t_q) + \\
&\quad \hat{k}_1 (t_a y_{ND} + t_q)^3. \\
\end{align*}
\]

(B.4)

Besides Eq. B.2, two further independent equations can also be derived from Eq. B.1:

\[
\begin{align*}
\mathbf{h}_1^T \cdot \mathbf{x}_N &= h_{11} x_N + h_{12} y_N + h_{13} = \eta \hat{x}_N, \\
\mathbf{h}_2^T \cdot \mathbf{x}_N &= h_{21} x_N + h_{22} y_N + h_{23} = \eta \hat{y}_N. \quad \text{(B.5)}
\end{align*}
\]

Regarding Eq. B.4 and B.2, the two equations above can be expanded into

\[
\begin{align*}
&h_{11} k_1 x_{ND}^3 + h_{12} k_1 x_{ND}^2 y_{ND} + \\
&h_{11} k_1 x_{ND} y_{ND}^2 + h_{12} k_1 y_{ND}^3 + \\
&h_{11} x_{ND} + h_{12} y_{ND} + h_{13} = \\
&(h_{31} k_1 k_1) x_{ND}^6 + (h_{32} k_1 k_1 + 3 h_{31} k_1 k_1 t_s) x_{ND}^5 y_{ND} + \\
&\sum_{i=2}^{6} (a_{(6,6-i)} x_{ND}^{(6-i)} y_{ND}^i) + \sum_{i=0}^{5} \sum_{j=0}^{i} (a_{(i,i-j)} x_{ND}^{(i-j)} y_{ND}^j)
\end{align*}
\]

and

\[
\begin{align*}
&h_{21} k_1 x_{ND}^3 + h_{22} k_1 x_{ND}^2 y_{ND} + \\
&h_{21} x_{ND} y_{ND}^2 + h_{22} y_{ND}^3 + \\
&h_{21} x_{ND} + h_{22} y_{ND} + h_{23} = \\
&\sum_{i=0}^{6} \sum_{j=0}^{i} (b_{(i,i-j)} x_{ND}^{(i-j)} y_{ND}^j),
\end{align*}
\]

(B.6)

and

(B.7)

where \( a_{(i,i-j)} \) and \( b_{(i,i-j)} \) are coefficients determined by \{h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32}, h_{33}, k_1, k_1, t_s, t_a, t_p, t_q \}. For distorted images, there is 230
\( k_1 \neq 0 \). However, \( \hat{k}_1 \neq 0 \) and \( \hat{k}_1 = 0 \) should be considered individually.

For the case of \( \hat{k}_1 \neq 0 \), the right side of Eq. B.6 contains the terms \( x_{ND}^6 \) and \( x_{ND}^5 \cdot y_{ND} \), which do not exist on the left side of the equation. It is thus straightforward to see that \( (h_{31} \cdot k_1 \cdot \hat{k}_1) \cdot x_{ND}^6 \) and \( 3 \cdot h_{31} \cdot k_1 \cdot \hat{k}_1 \cdot t_s \cdot x_{ND}^5 \cdot y_{ND} \) must vanish with \( h_{31} = h_{32} = 0 \) since \( x_{ND} \) and \( y_{ND} \) can take arbitrary values. Regarding Eq. B.2, there is \( h_3^T = [0, 0, \eta] \).

If \( \hat{k}_1 = 0 \), Eq. B.6 is simplified to

\[
\begin{align*}
&h_{11} \cdot k_1 \cdot x_{ND}^3 + h_{12} \cdot k_1 \cdot x_{ND}^2 \cdot y_{ND} + \\
&h_{11} \cdot k_1 \cdot x_{ND} \cdot y_{ND} + h_{12} \cdot k_1 \cdot y_{ND} + h_{13} = \\
&(h_{31} \cdot k_1) \cdot x_{ND}^4 + (h_{32} \cdot k_1 + h_{31} \cdot k_1 \cdot t_s) \cdot x_{ND}^3 \cdot y_{ND} + \\
&(h_{31} \cdot k_1 + h_{32} \cdot k_1 \cdot t_s) \cdot x_{ND}^2 \cdot y_{ND}^2 + (h_{32} \cdot k_1 + h_{31} \cdot k_1 \cdot t_s) \cdot x_{ND} \cdot y_{ND}^3 + \\
&h_{32} \cdot k_1 \cdot t_s \cdot y_{ND}^3 + \\
&h_{31} \cdot k_1 \cdot t_p \cdot x_{ND}^3 + h_{32} \cdot k_1 \cdot t_p \cdot x_{ND} \cdot y_{ND} + \\
&h_{31} \cdot k_1 \cdot t_p \cdot x_{ND} \cdot y_{ND} + h_{32} \cdot k_1 \cdot t_p \cdot y_{ND}^3 + \\
&h_{31} \cdot x_{ND}^2 + (h_{32} + h_{31} \cdot t_s) \cdot t_p \cdot x_{ND} \cdot y_{ND} + h_{32} \cdot t_s \cdot y_{ND}^2 + \\
&(h_{33} + h_{31} \cdot t_p) \cdot x_{ND} + (h_{32} \cdot t_p + h_{33} \cdot t_s) \cdot y_{ND} + h_{33} \cdot t_p.
\end{align*}
\]  

(B.8)

Again, the terms \( x_{ND}^4 \) and \( x_{ND}^3 \cdot y_{ND} \) on the right side of the equation must vanish, which leads to \( h_3^T = [0, 0, \eta] \).

### B.1.2 Simplified equations for \( h_3^T = [0, 0, 1] \)

As proven in the last subsection, the last row \( h_3^T \) of the matrix \( \mathcal{H} \) is of the form \( h_3^T = [0, 0, \eta] \). For computational convenience, \( \mathcal{H} \) is scaled by the factor \( (1/\eta) \) so that \( h_3^T = [h_{31}, h_{32}, h_{33}] = [0, 0, 1] \). The obtained values
Given the transformation matrix $T$ (defined in Eq. 3.41) between the coordinates $\mathbf{z}_{ND}$ and $\hat{\mathbf{z}}_{ND}$ with $\hat{\mathbf{z}}_{ND} = T \cdot \mathbf{z}_{ND}$, there is also a corresponding transformation defined by a $10 \times 10$ invertible matrix $B_N$ between the lifted coordinates $\mathbf{x}_{ND}$ and $\hat{\mathbf{x}}_{ND}$: $\hat{\mathbf{x}}_{ND} = \hat{B}_N \cdot \mathbf{x}_{ND}$ following their definitions in Eq. 3.48. After replacing $\hat{\mathbf{x}}_{ND}$ and $\hat{\mathbf{y}}_{ND}$ in the coordinate vector $\hat{\mathbf{z}}_{ND} = [\hat{\mathbf{z}}_{ND}, \hat{\mathbf{y}}_{ND}, 1]^T$ with linear combinations of $\{\mathbf{z}_{ND}, \mathbf{y}_{ND}, 1\}$ according to $\hat{\mathbf{z}}_{ND} = T \cdot \mathbf{z}_{ND}$, the coordinate transformation matrix $B_N$ is obtained as

$$B_N = [b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{10}]^T,$$  \hspace{1cm} (B.11)
where
\begin{align*}
\textbf{b}_1^r &= [1, 3 \cdot t_s, 3 \cdot t_s^2, t_s^3, 3 \cdot t_p, 3 \cdot t_s \cdot t_p, 3 \cdot t_s^2 \cdot t_p, 3 \cdot t_p^2, 3 \cdot t_s \cdot t_p^2, t_p^3], \\
\textbf{b}_2^r &= [0, t_a, 2 \cdot t_a \cdot t_s, t_a \cdot t_s^2, t_q, 2 \cdot (t_a \cdot t_p + t_s \cdot t_q), \\
& \quad (2 \cdot t_a \cdot t_s \cdot t_p + t_s^2 \cdot t_q), 2 \cdot t_p \cdot t_q, (t_a \cdot t_p^2 + 2 \cdot t_s \cdot t_p \cdot t_q), t_p^2 \cdot t_q], \\
\textbf{b}_3^r &= [0, 0, t_a^2, t_a \cdot t_s, 0, 2 \cdot t_a \cdot t_q, (2 \cdot t_a \cdot t_s \cdot t_q + t_a \cdot t_p), t_q^2, \\
& \quad (2 \cdot t_a \cdot t_p \cdot t_q + t_s \cdot t_q^2), t_p \cdot t_q^2], \\
\textbf{b}_4^r &= [0, 0, 0, t_a^3, 0, 0, 3 \cdot t_a^2 \cdot t_q, 0, 3 \cdot t_a \cdot t_q^2, t_q^3], \\
\textbf{b}_5^r &= [0, 0, 0, 0, 1, 2 \cdot t_s, t_s^2, 2 \cdot t_p, 2 \cdot t_s \cdot t_p, t_p^2], \\
\textbf{b}_6^r &= [0, 0, 0, 0, 0, t_a, t_a \cdot t_s, t_q, (t_a \cdot t_p + t_s \cdot t_q), t_p \cdot t_q], \\
\textbf{b}_7^r &= [0, 0, 0, 0, 0, 0, t_a^2, 0, 2 \cdot t_a \cdot t_q, t_q^2], \\
\textbf{b}_8^r &= [0, 0, 0, 0, 0, 0, 0, 1, t_s, t_p], \\
\textbf{b}_9^r &= [0, 0, 0, 0, 0, 0, 0, 0, t_a, t_q], \\
\textbf{b}_{10}^r &= [0, 0, 0, 0, 0, 0, 0, 0, 0, 1].
\end{align*}

B.2.2 Ambiguities in estimated parameters

For the first two row vectors \( \textbf{h}_1^r \) and \( \textbf{h}_2^r \) in \( \mathcal{H} \), as well as for the transformation matrix \( \mathbf{T} \) and for the estimated inverse distortion function \( \hat{\mathbf{L}}^{-1}_a(\cdot) \), sufficient equations in the unknown parameters \( \{h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, t_a, t_s, t_p, t_q, \hat{k}_1, \hat{p}_1, \hat{p}_2\} \) can be obtained by substituting Eq. 3.49, B.11 and
B.12 into Eq. 3.50:

\[ h_{11} \cdot k_1 = \hat{k}_1, \]
\[ h_{21} \cdot k_1 = 0, \]
\[ h_{12} \cdot k_1 = 3 \cdot \hat{k}_1 \cdot t_s, \]
\[ h_{22} \cdot k_1 = \hat{k}_1 \cdot t_a, \]
\[ h_{11} \cdot k_1 = \hat{k}_1 \cdot (t_a^2 + 3 \cdot t_s^2), \]
\[ h_{21} \cdot k_1 = 2 \cdot \hat{k}_1 \cdot t_a \cdot t_s, \]
\[ h_{12} \cdot k_1 = \hat{k}_1 \cdot (t_a^2 \cdot t_s + t_s^3), \]
\[ h_{22} \cdot k_1 = \hat{k}_1 \cdot (t_s^3 + t_a \cdot t_s^2), \]
\[ 3 \cdot h_{11} \cdot p_1 + h_{12} \cdot p_2 = 3 \cdot \hat{k}_1 \cdot t_p + 3 \cdot \hat{p}_1, \]
\[ 3 \cdot h_{21} \cdot p_1 + h_{22} \cdot p_2 = \hat{k}_1 \cdot t_q + \hat{p}_2, \]
\[ h_{11} \cdot p_2 + h_{12} \cdot p_1 = \hat{k}_1 \cdot t_a \cdot t_q + t_a \cdot \hat{p}_2, \]
\[ h_{21} \cdot p_2 + h_{22} \cdot p_1 = \hat{k}_1 \cdot t_a \cdot t_p + t_a \cdot \hat{p}_1, \]
\[ h_{11} \cdot p_1 + 3 \cdot h_{12} \cdot p_2 = \hat{k}_1 \cdot t_a^2 \cdot t_p + t_a^2 \cdot \hat{p}_1, \]
\[ h_{21} \cdot p_1 + 3 \cdot h_{22} \cdot p_2 = 3 \cdot (\hat{k}_1 \cdot t_a^2 \cdot t_q + t_a^2 \cdot \hat{p}_2), \]
\[ h_{11} = \hat{k}_1 \cdot (3 \cdot t_p^2 + t_q^2) + 6 \cdot t_p \cdot \hat{p}_1 + 2 \cdot t_q \cdot \hat{p}_2 + 1, \]
\[ h_{21} = 2 \cdot \hat{k}_1 \cdot t_p \cdot t_q + 2 \cdot t_p \cdot \hat{p}_2 + 2 \cdot t_q \cdot \hat{p}_1, \]
\[ h_{12} = 2 \cdot t_a \cdot (\hat{k}_1 \cdot t_p \cdot t_q + t_q \cdot \hat{p}_1 + t_p \cdot \hat{p}_2), \]
\[ h_{22} = t_a \cdot (\hat{k}_1 \cdot t_p^2 + 3 \cdot \hat{k}_1 \cdot t_q^2 + 2 \cdot t_p \cdot \hat{p}_1 + 6 \cdot t_q \cdot \hat{p}_2 + 1), \]
\[ h_{13} = \hat{k}_1 \cdot t_p \cdot (t_p^2 + t_q^2) + \hat{p}_1 \cdot (3 \cdot t_p^2 + t_q^2) + t_p \cdot (2 \cdot t_q \cdot \hat{p}_2 + 1), \]
\[ h_{23} = \hat{k}_1 \cdot t_q \cdot (t_p^2 + t_q^2) + t_q \cdot (2 \cdot t_p \cdot \hat{p}_1 + 1) + \hat{p}_2 \cdot (t_p^2 + 3 \cdot t_q^2). \]

Regarding \( k_1 \neq 0 \) for distorted images and \( \det(\mathcal{H}) \neq 0 \) for the homography \( \mathcal{H} \), it is straightforward to solve \( \{ h_{12}, h_{21}, t_a, t_s \} \) with \( t_a = \pm 1 \) and \( h_{12} = h_{21} = t_s = 0 \). However, there exist ambiguities in \( \{ h_{11}, h_{13}, h_{22}, h_{23}, t_p, t_q, \hat{k}_1, \hat{p}_1, \hat{p}_2 \} \) even after discarding the four solutions with complex values. Especially, the parameters \( \{ t_p, t_q, \hat{k}_1, \hat{p}_1, \hat{p}_2 \} \) can take one set of the
following values:
\[
\begin{align*}
\{ & 0, 0, k_1, p_1, -p_2 \}, \\
\{ & 2p_1/k_1, 2p_2/k_1, k_1, -p_1, -p_2 \}, \\
\{ & 0, 0, k_1, p_1, p_2 \}, \\
\{ & 2p_1/k_1, -2p_2/k_1, k_1, -p_1, p_2 \}.
\end{align*}
\] (B.14)

B.3 Polynomial model using \( \{k_1, k_2, k_3, p_1, p_2\} \)

B.3.1 Transformed model for lifted coordinates

By introducing the additional parameters \( \{k_2, k_3\} \), the transformed correction model \( \mathbf{M}_N \) with \( \mathbf{x}_N = \mathbf{M}_N \mathbf{X}_{ND} \) for the lifted coordinates \( \mathbf{X}_{ND} \):

\[
\mathbf{X}_{ND} = \begin{bmatrix}
x_{ND}^7, & x_{ND}^6 y_{ND}, & x_{ND}^5 y_{ND}^2, & x_{ND}^4 y_{ND}^3, & x_{ND}^3 y_{ND}^4, \\
x_{ND}^2 y_{ND}^5, & x_{ND}^1 y_{ND}^6, & y_{ND}^7, & x_{ND}^2, & x_{ND}^2 y_{ND}^3, \\
x_{ND}^4 y_{ND}^2, & x_{ND}^3 y_{ND}^3, & x_{ND}^2 y_{ND}^4, & x_{ND}^3 y_{ND}^5, & y_{ND}^6, \\
x_{ND}^5, & x_{ND}^4 y_{ND}, & x_{ND}^3 y_{ND}^2, & x_{ND}^4 y_{ND}^3, & x_{ND}^5 y_{ND}^4, \\
y_{ND}^5, & x_{ND}^4, & x_{ND}^3 y_{ND}, & x_{ND}^2 y_{ND}^2, & x_{ND}^3 y_{ND}^3, \\
y_{ND}^4, & x_{ND}^3, & x_{ND}^2 y_{ND}, & x_{ND}^2 y_{ND}^2, & y_{ND}^3, \\
x_{ND}^2, & x_{ND} y_{ND}, & y_{ND}^2, & x_{ND}, & y_{ND}, \\
x_{ND}, & y_{ND}, & 1
\end{bmatrix}^T
\] (B.15)

is written as \( \mathbf{M}_N^T = [\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3] \), where \( \mathbf{m}_1^T, \mathbf{m}_2^T \) and \( \mathbf{m}_3^T \in \mathbb{R}^{1 \times 36} \) are the row vectors of \( \mathbf{M}_N \) with

\[
\mathbf{m}_1^T = [k_3, 0, 3k_3, 0, 3k_3, 0, k_3, 0, 0, \ldots, 0, \\
k_2, 0, 2k_2, 0, k_2, 0, 0, 0, 0, \ldots, 0, \\
0, k_1, 0, k_1, 0, 3p_1, 2p_2, p_1, 1, 0, 0],
\]

\[
\mathbf{m}_2^T = [0, k_3, 0, 3k_3, 0, 3k_3, 0, k_3, 0, \ldots, 0, \\
0, k_2, 0, 2k_2, 0, k_2, 0, 0, 0, \ldots, 0, \\
0, 0, k_1, 0, k_1, p_2, 2p_1, 3p_2, 0, 1, 0],
\]

\[
\mathbf{m}_3^T = [0, 0, 0, 0, 0, 0, 0, 0, 0, \ldots, 0, \\
0, 0, 0, 0, 0, 0, 0, 0, 0, \ldots, 0, \\
0, 0, 0, 0, 0, 0, 0, 0, 0, 1].
\]
Appendix B. Correction of lens distortion

Apparently, the estimated lifted coordinates \( \hat{\chi}_{ND} \) and the estimated model \( \hat{M}_N \) are in the same forms as Eq. B.15 and B.16, respectively.

**B.3.2 Constraint on \( x^i \cdot y^j \) for \( (i + j) = 6 \)**

Since the entries \( \{x^i_{ND} \cdot y^j_{ND} | (i + j) = 6\} \) in \( \chi_{ND} \) are not included in the correction model \( L^{-1}_N(\cdot) \), they must vanish on the right side of Eq. 3.52. Considering the special form of \( \hat{M}_N \) and the coordinate transformation \( [\hat{x}_{ND}, \hat{y}_{ND}, 1]^T = \mathcal{T} \cdot [x_{ND}, y_{ND}, 1]^T \), only the entries \( \{\hat{x}^i_{ND} \cdot \hat{y}^j_{ND} | (i + j) = 7\} \) in \( \hat{\chi}_{ND} \) need to be investigated for ensuring the constraint stated above. According to Section B.3.1, the coordinate transformation matrix \( B_N \) in Eq. 3.51 is of the size 36×36 and satisfies \( \hat{\chi}_{ND} = B_N \cdot \chi_{ND} \). If only the part related to the entries \( \{\hat{x}^i_{ND} \cdot \hat{y}^j_{ND} | (i + j) = 7\} \) in \( \hat{\chi}_{ND} \) is considered on the right side of Eq. 3.51, the right side is reduced to \( \hat{M}_{N,7} \cdot B_{N,7} \cdot \chi_{ND} \), where

\[
\hat{M}_{N,7} = \begin{bmatrix}
\hat{k}_3 & 0 & 3 \cdot \hat{k}_3 & 0 & 3 \cdot \hat{k}_3 & 0 & \hat{k}_3 & 0 \\
0 & \hat{k}_3 & 0 & 3 \cdot \hat{k}_3 & 0 & 3 \cdot \hat{k}_3 & 0 & \hat{k}_3 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

(B.17)

and \( B_{N,7} = [b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8]^T \) with \( b_1^T, \cdots, b_8^T \) denoting the first eight row vectors of \( B_N \). If the vanishing constraint on terms related to the entries \( \{x^i_{ND} \cdot y^j_{ND} | (i + j) = 6\} \) in \( \chi_{ND} \) is considered additionally, the remaining part \( \hat{M}_{N,7} \cdot B_{N,7} \cdot \chi_{ND} \) is further simplified and results in the following equation:

\[
\hat{M}_{N,7} \cdot B_{N,7,6} \cdot \chi_{ND,6} = 0,
\]

(B.18)
where $B_{N,7,6}$ consists of the 9-th to 15-th columns of $B_{N,7}$ with

$$B_{N,7,6} =$$

<table>
<thead>
<tr>
<th>$7 t_p$</th>
<th>$42 t_s t_p$</th>
<th>$105 t_s^2 t_p$</th>
<th>$140 t_s^3 t_p$</th>
<th>$105 t_s^4 t_p$</th>
<th>$42 t_s^5 t_p$</th>
<th>$7 t_s^6 t_p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_q$</td>
<td>$6(t_a t_p$</td>
<td>$15 t_s t_p$</td>
<td>$20 t_s^2 t_p$</td>
<td>$15 t_s^3 t_p$</td>
<td>$6 t_s^4 t_p$</td>
<td>$t_s^5 t_p$</td>
</tr>
<tr>
<td></td>
<td>$+t_s t_q)$</td>
<td>$(2 t_a t_p$</td>
<td>$(3 t_a t_p$</td>
<td>$(4 t_a t_p$</td>
<td>$(5 t_a t_p$</td>
<td>$(6 t_a t_p$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$+t_s t_q)$</td>
<td>$+t_s t_q)$</td>
<td>$+t_s t_q)$</td>
<td>$+t_s t_q)$</td>
<td>$+t_s t_q)$</td>
</tr>
<tr>
<td>0</td>
<td>$2 t_a t_q$</td>
<td>$5 t_a$</td>
<td>$20 t_a$</td>
<td>$10 t_a t_s^2$</td>
<td>$10 t_a t_s^3$</td>
<td>$t_a t_s t_q$</td>
</tr>
<tr>
<td></td>
<td>$(t_a t_p$</td>
<td>$(2 t_a t_p$</td>
<td>$(3 t_a t_p$</td>
<td>$(4 t_a t_p$</td>
<td>$(5 t_a t_p$</td>
<td>$(5 t_a t_p$</td>
</tr>
<tr>
<td></td>
<td>$+t_s t_q)$</td>
<td>$+t_s t_q)$</td>
<td>$+t_s t_q)$</td>
<td>$+t_s t_q)$</td>
<td>$+t_s t_q)$</td>
<td>$+t_s t_q)$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>$3 t_a^2 t_q$</td>
<td>$4 t_a^2 t_q$</td>
<td>$6 t_a^2 t_s$</td>
<td>$12 t_a^2 t_s^2$</td>
<td>$t_a^3 t_s^2$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$(t_a t_p$</td>
<td>$(2 t_a t_p$</td>
<td>$(2 t_a t_p$</td>
<td>$(2 t_a t_p$</td>
<td>$(4 t_a t_p$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$+3 t_s t_q)$</td>
<td>$+3 t_s t_q)$</td>
<td>$+3 t_s t_q)$</td>
<td>$+3 t_s t_q)$</td>
<td>$+3 t_s t_q)$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$4 t_a^3 t_q$</td>
<td>$3 t_a^3 t_q$</td>
<td>$6 t_a^3 t_s$</td>
<td>$t_a^4 t_s$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$(t_a t_p$</td>
<td>$(2 t_a t_p$</td>
<td>$(3 t_a t_p$</td>
<td>$(5 t_a t_p$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$+4 t_s t_q)$</td>
<td>$+2 t_s t_q)$</td>
<td>$+3 t_s t_q)$</td>
<td>$+4 t_s t_q)$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$5 t_a^4 t_q$</td>
<td>$2 t_a^4 t_q$</td>
<td>$t_a^5 t_s$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$(t_a t_p$</td>
<td>$(2 t_a t_p$</td>
<td>$(2 t_a t_p$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$+5 t_s t_q)$</td>
<td>$+5 t_s t_q)$</td>
<td>$+5 t_s t_q)$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$6 t_a^5 t_q$</td>
<td>$t_a^6 t_s$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$(t_a t_p$</td>
<td>$(6 t_s t_q)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$+6 t_s t_q)$</td>
<td>$+6 t_s t_q)$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$7 t_s^6 t_q$</td>
</tr>
</tbody>
</table>

and $\mathbf{x}_{N,6} = [x_{ND}^6, x_{ND}^5, y_{ND}, x_{ND}^4, y_{ND}^2, x_{ND}^3, y_{ND}^3, x_{ND}^2, y_{ND}^4, y_{ND}^5, y_{ND}^6]^T$. To satisfy Eq. B.18 for $\mathbf{x}_{N,6}$ taking arbitrary values, there must be $\hat{M}_{N,7} B_{N,7,6} = 0$. It is not difficult to see that there exist multiple parameter settings valid for this equation. Regarding the products of the first two row vectors in $\hat{M}_{N,7}$ with the first column vector in $B_{N,7,6}$, two equations are obtained:

$$7 \hat{k}_3 t_p = 0,$$

$$\hat{k}_3 t_q = 0.$$  (B.20)
Appendix B. Correction of lens distortion

If \( \hat{k}_3 \neq 0 \), the two equations above are true only for \( t_p = t_q = 0 \). Consequently, the coordinate transformation matrix \( B_N \) turns into

\[
B_N = \begin{bmatrix}
B_{N,7,7} & 0 & 0 & 0 & 0 \\
0 & B_{N,6,6} & 0 & 0 & 0 \\
0 & 0 & B_{N,5,5} & 0 & 0 \\
0 & 0 & 0 & B_{N,4,4} & 0 \\
0 & 0 & 0 & 0 & B_{N,3210,3210}
\end{bmatrix}, \tag{B.21}
\]

where the submatrices \( B_{N,7,7} \in \mathbb{R}^{8 \times 8}, B_{N,6,6} \in \mathbb{R}^{7 \times 7}, B_{N,5,5} \in \mathbb{R}^{6 \times 6}, B_{N,4,4} \in \mathbb{R}^{5 \times 5} \) and \( B_{N,3210,3210} \in \mathbb{R}^{10 \times 10} \) are with the coefficients corresponding to the entries \( \{x^i_{\text{ND}}\cdot y^j_{\text{ND}} \mid (i+j) = 7\}, \{x^i_{\text{ND}}\cdot y^j_{\text{ND}} \mid (i+j) = 6\}, \{x^i_{\text{ND}}\cdot y^j_{\text{ND}} \mid (i+j) = 5\}, \{x^i_{\text{ND}}\cdot y^j_{\text{ND}} \mid (i+j) = 4\} \) and \( \{x^i_{\text{ND}}\cdot y^j_{\text{ND}} \mid (i+j) \in \{0,1,2,3\} \} \) in \( \chi_{\text{ND}} \), respectively. The correction model \( \hat{M}_N \) can also be rewritten as \( \hat{M}_N = [\hat{M}_{N,7}, 0_{3 \times 7}, \hat{M}_{N,5}, 0_{3 \times 5}, \hat{M}_{N,3210}] \), where

\[
\hat{M}_{N,5} = \begin{bmatrix}
\hat{k}_2 & 0 & 2\hat{k}_2 & 0 & \hat{k}_2 & 0 \\
0 & \hat{k}_2 & 0 & 2\hat{k}_2 & 0 & \hat{k}_2 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}, \tag{B.22}
\]

and

\[
\hat{M}_{N,3210} = \begin{bmatrix}
\hat{k}_1 & 0 & \hat{k}_1 & 0 & 3\hat{p}_1 & 2\hat{p}_2 & \hat{p}_1 & 1 & 0 & 0 \\
0 & \hat{k}_1 & 0 & \hat{k}_1 & \hat{p}_2 & 2\hat{p}_1 & 3\hat{p}_2 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}. \tag{B.23}
\]

Thus, Eq. 3.51 results in three independent equations

\[
\mathcal{H} \cdot \hat{M}_{N,7} = \hat{M}_{N,7} \cdot B_{N,7,7},
\]

\[
\mathcal{H} \cdot \hat{M}_{N,5} = \hat{M}_{N,5} \cdot B_{N,5,5}, \tag{B.24}
\]

\[
\mathcal{H} \cdot \hat{M}_{N,3210} = \hat{M}_{N,3210} \cdot B_{N,3210,3210}
\]

for \( x_{\text{ND}} \) and \( y_{\text{ND}} \) being arbitrary. Interestingly, the last equation above is also obtained for the PCM with extended parameters \( \{k_1, p_1, p_2\} \). In other words, the same solutions with \( t_p = t_q = 0 \) are obtained:

\[
\{t_a, t_s, t_p, t_q, \hat{k}_1, \hat{p}_1, \hat{p}_2\} = \{\pm 1, 0, 0, 0, k_1, p_1, \pm p_2\}. \tag{B.25}
\]
Moreover, the two further correction parameters $\hat{k}_2$ and $\hat{k}_3$ are solved as $\hat{k}_2 = k_2$ and $\hat{k}_3 = k_3$, respectively.

However, if $\hat{k}_3 = 0$ or practically $\hat{k}_3 \approx 0$, which also implies $k_3 = 0$ or $k_3 \approx 0$, things turn out differently: $t_p$ and $t_q$ are not constrained any more according to Eq. B.18. It becomes even worse if $k_2$ is also equal or close to zero. In practice, due to the inaccuracy in determined point correspondences, equations like those in Eq. B.18 do not exactly hold. $t_p$ and $t_q$ can thus even take huge values without resulting in significant residual on the right hand of these equations. To avoid uncontrolled drift of $t_p$ and $t_q$ from their real values, dismissing the decentering components related to $\{p_1,p_2\}$ is necessary. By this means, the correction model is simplified to the basic form only using $k_1$, given $k_2 = k_3 = 0$. The deviation from the original full-parameter model can be well compensated by adapting all other parameters, especially $a$ and $s$ in $K$, through an appropriate optimization.

### B.4 Parameter constraints for division model

For DCM, Eq. 3.39 is rewritten as

$$
\mathcal{H} \cdot \begin{bmatrix} \frac{x_{\text{ND}}}{1 + \kappa \cdot r^2} \\ \frac{y_{\text{ND}}}{1 + \kappa \cdot r^2} \\ 1 \end{bmatrix} \approx \begin{bmatrix} \frac{\hat{x}_{\text{ND}}}{1 + \hat{k} \cdot \hat{r}^2} \\ \frac{\hat{y}_{\text{ND}}}{1 + \hat{k} \cdot \hat{r}^2} \\ 1 \end{bmatrix}
$$

(B.26)

$$
\iff \quad \mathcal{H} \cdot \begin{bmatrix} x_{\text{ND}} \cdot (1 + \hat{k} \cdot \hat{r}^2) \\ y_{\text{ND}} \cdot (1 + \hat{k} \cdot \hat{r}^2) \\ (1 + \kappa \cdot r^2)(1 + \hat{k} \cdot \hat{r}^2) \end{bmatrix} = \begin{bmatrix} \hat{x}_{\text{ND}} \\ \hat{y}_{\text{ND}} \\ 1 + \hat{k} \cdot \hat{r}^2 \end{bmatrix} \cdot \mathbf{h}_3^T \cdot \begin{bmatrix} x_{\text{ND}} \\ y_{\text{ND}} \\ 1 + \kappa \cdot r^2 \end{bmatrix}
$$

where $r^2 = x_{\text{ND}}^2 + y_{\text{ND}}^2$, $\hat{r}^2 = \hat{x}_{\text{ND}}^2 + \hat{y}_{\text{ND}}^2$ and $\mathcal{H} = [\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3]^T$. Using the linear coordinate transformation $[\hat{x}_{\text{ND}}, \hat{y}_{\text{ND}}, 1]^T = \mathcal{T} [x_{\text{ND}}, y_{\text{ND}}, 1]^T$, $\hat{x}_{\text{ND}}$ and $\hat{y}_{\text{ND}}$ are eliminated in the equation above. Apparently, for $\kappa \neq 0$ in a general case, the homography $\mathcal{H}$ must take the form

$$
\mathcal{H} = \begin{bmatrix} h_{11} & 0 & 0 \\ 0 & h_{22} & 0 \\ h_{31} & h_{32} & h_{33} \end{bmatrix}
$$

(B.27)
since the equality in Eq. B.26 must hold for arbitrary values of $x_{ND}$ and $y_{ND}$. Eq. B.26 is thus simplified and leads to

$$
\begin{bmatrix}
    h_{11}x_{ND}(1 + \hat{\kappa} \cdot \hat{r}^2) \\
    h_{22}y_{ND}(1 + \hat{\kappa} \cdot \hat{r}^2)
\end{bmatrix}
= \begin{bmatrix}
    \hat{x}_{ND} \\
    \hat{y}_{ND}
\end{bmatrix}
\cdot
\begin{bmatrix}
    h_{31} \\
    h_{32}
\end{bmatrix}
\begin{bmatrix}
    x_{ND} \\
    y_{ND} \\
    1 + \hat{\kappa} \cdot \hat{r}^2
\end{bmatrix}
$$

(B.28)

where $M_{N,L}$ and $M_{N,R}$ are the coefficient matrices for the entries in the lifted coordinate vector $\chi_{ND}$. After solving each unknown parameter in Eq. B.28, the same results as in the case of PCM only using the single parameter $k_1$ are obtained and the desired image correction is accomplished.
Appendix C

Features and classifiers

C.1 Features

**Haralick** Based on GLCMs computed for adjacent pixels along certain directions, statistical features [128], e.g. contrast, correlation, homogeneity, entropy, etc., are employed for describing the presented texture in images.

**HOG** As introduced in Section 5.2.1.2, Histograms of Oriented Gradients (HOG) [142] features consist of histograms quantifying the occurrences of oriented gradients in dense spatial cells and thus capture the local shape context of objects as well as the global arrangement.

**BoVW** In a Bag-of-Visual-Words (BoVW) [139] model, primary features are densely sampled (pooled) over the entire image or in the subdivided spatial cells. The underlying object is represented by the histogram or histograms quantifying the occurrences of visual words associated with the extracted primary features.

**FV** Similar to BoVW models, the Fisher Vector (FV) [181] of an image is also generated with the help of a visual vocabulary described typically by a Gaussian mixture model. Instead of

241
counting the occurrences of visual words, statistical difference between the extracted primary features and visual words is utilized for representing the image.

### C.2 Fundamental classifiers

**k-NN** Instead of explicitly defining class boundaries in feature space, a *k*-Nearest-Neighbors (*k*-NN) [130] classifier relies on the local majority labels of training samples. More specifically, the nearest *k* training samples of the test point in the feature space are considered and the class label with the majority votes is assigned to the test point. As a result, the number *k* of the nearest neighbors and the distance measure in the feature space turn out to be the only parameters to be adjusted.

**DTree** Starting from the root node and landing in the leaf nodes for different classes, test samples travel down a Decision Tree (DTree) [151] while being successively divided into subsets at each decision node regarding the optimal splitting feature or the optimal splitting feature component in feature vectors. Through tuning the maximal depth of DTrees and the minimal sizes for splitting and leafs, the trade-off between training error and overfitting can be adjusted.

**SVM** As introduced in Section 5.2.1.2, a Support Vector Machine (SVM) [131] seeks the optimal separating hyperplane in the feature space for maximizing the classification margin, where a subset of the training samples become support vectors, which are close to and thus defining the sought hyperplane. With the help of the kernel trick for defining distance measures in a high-dimensional feature space without any explicit transformation, nonlinear class boundaries for better classification performance can be localized in the original feature space through the linear separating hyperplanes determined implicitly in the high-dimensional feature space. In general, the kernel parameter and the penalty coefficient of false labeling are to be selected with caution.
C.3 Ensemble methods

**AdaBoost** Based on a set of weak classifiers, each of which performs slightly better than random guessing, a strong classifier with high precision can be created using the Boosting algorithm for ensemble learning. As a practical boosting method, *Adaptive Boosting* (AdaBoost) [284] sequentially adds retrained weak classifiers to the boosted classifier and generates the final prediction as a linear combination of these weak classifiers. Moreover, the importance (probability) of single training instances is dynamically adapted: increased in case of wrong classification using the currently added weak classifier and decreased otherwise.

**Bagging** Another practical ensemble learning algorithm is Bagging [190] (Bootstrap Aggregating). For improving stability and reducing variance as well as overfitting, less correlated classifiers are trained on a series of equal-sized new training sets sampled from the original training data, where samples in each training set are obtained through a uniform sampling with replacement. The final prediction is formed as the output with the majority votes.

**Random Subspace** Instead of using Bagging to reduce the correlation between single DTrees in ensemble learning, random subsets of features or random subsets of components in feature vectors [191] can be used to determine the best splits at decision nodes. By this means, the dominance of strong features is avoided and adequately randomized DTrees are thus constructed.

**RF** Combining Bagging and random feature subspaces in training a forest of DTrees, a stable classifier in the form of Random Forest (RF) [132] is obtained and provides desirable resistance to overfitting.
C.4 Deep learning

**DBN** A probabilistic generative model relying on Deep Belief Network (DBN) [219] comprises multiple layers of hidden units and is realized typically in the form of stacked Restricted Boltzmann Machines (RBMs), where the weight matrix between any two consecutive layers of the DBN is determined through training a RBM: the bottom visible input layer and the top hidden layer. By this means, the probability distribution of binary latent variables in the hidden layers is learned. In inference (test), the predicted label is obtained if its probability is the global maximum or the corresponding free energy is the minimum.

**CNN** As introduced in Section 5.2.2.2, Convolutional Neural Networks (CNNs) [140] usually consist of bottom Conv layers and top reasoning layers, for instance FC layers. After extracting the underlying stimuli patterns through successive convolutional operations and nonlinear transformations, semantic representations at the top Conv layer are forwarded to reasoning layers for generating prediction results.
Bibliography


Bibliography


266


