Robert Viehauser

Efficient Methods for Detection of planar, untextured Objects in Indoor Environments

MASTER'S THESIS

to achieve the university degree of
Diplom-Ingenieur

Master's degree programme
Telematics

submitted to

Graz University of Technology

Supervisor

Univ.-Prof. DI. Dr. Horst Bischof
Institute for Computer Graphics and Vision

Graz, Austria, April 2018

To my two beloved girls,
Sidney & Sabine

It is the framework which changes with each new technology and not just the picture within the frame.

Marshall McLuhan (1911 - 1980)

ABSTRACT

In the recent century, Computer Vision and Image Understanding tremendously advanced from a scientific perspective, but also finally arrived in the midst of society due to implementations in digital cameras, mobile phones and, most recently, self-driving cars. Undeniably, Computer Vision will continue conquering a much larger spectrum of applications in everybody's life. One example is addressed in this thesis. We developed a vision-based algorithm for detecting wall-mounted electrical devices as part of an EU-founded, publicly accessible framework for digitalizing buildings that targets a widespread audience. In order to meet the requirements for this use-case, we carefully analyze the given data to derive the most promising solutions. In particular, the low number of training samples, the insufficient data quality and the plain, texture-less nature of the target objects form the most challenging aspects of the addressed detection task. We propose to tackle these problems by utilizing pre-calculated wall-texture images as input for a classical vision pipeline consisting of feature descriptors and a classifier. To best exploit the characteristic properties of the target objects, we introduce a novel, specially designed feature descriptor, but also apply well-established approaches for describing texture-less objects. The feature representation is evaluated by a random forest classifier for which we propose some modifications to better handle the highly imbalanced and general low amount of training data, as well as incorporating prior knowledge about the common mounting practice. The evaluation on our testdata attests the derived algorithm good detection results under various difficult circumstances. We also demonstrate the limits of the method.

Keywords. object detection, untextured targets, gradient features, random forest, imbalanced data, small training-set, practical application

KURZFASSUNG

Im Laufe des letzten Jahrzehnts wurde computergestützte Bilderkennung nicht nur von Wissenschaftlern technischer Disziplinen wahrgenommen und eingesetzt, sondern fand zunehmends auch den Weg in die Wahrnehmung der breiten Öffentlichkeit. Obwohl sich hier die Anwendungsgebiete von Bilderkennung noch hauptsächlich auf Digitalkameras, Mobiltelefone und aktuell selbstfahrende Fahrzeuge konzentriert, wird die computergestützte Bilderkennung ohne Zweifel auch andere, breitgefächerte Anwendungsgebiete erobern. Diese Diplomarbeit behandelt die Entwicklung eines Detektions-Algorithmus für Elektroinstallationen in Innenräumen als Teil eines EU-geförderten und öffentlich zugänglichen IT-Services zur Digitalisierung von bestehenden Gebäuden, welches einen hohen Nutzen für zahlreiche Berufsgruppen darstellt. Dieser Anwendungsfall bringt auch einige schwierige Anforderungen mit sich. Als ersten Schritt werden in dieser Arbeit die zur Verfügung gestellten Ausgangsdaten analysiert um daraus die vielversprechendsten Bilderkennungs-Methoden abzuleiten. Insbesondere die geringe Anzahl an Trainingsdaten, deren unzureichende Qualität und die flache, untexturierte Bauart der Zielobjekte stellen die größten Herausforderungen für den entwickelten Algorithmus dar. Als Lösungsansatz präsentieren wir die Verwendung von Texturbildern der zu untersuchenden Wände, welche aus den Daten des 3D Scanners vorausberechnet werden. Um die speziellen Eigenschaften der Eingangsdaten und der Zielobjekte zu nutzen, stellen wir eine neue eigens dafür entwickelte Bildrepräsentation vor, verwenden aber zugleich auch existierende und bereits erfolgreich eingesetzte Methoden. Um die Bildrepräsentationen einer bestimmten Kategorie zuzuordnen, verwenden wir einen Random Forest Klassifikator mit einigen Modifikationen um die ungleiche Verteilung und geringe Anzahl der Trainingsdaten zu kompensieren. Weiters präsentieren wir Methoden zur Verwendung des Vorwissens über die gebräuchlichen Einbaupraktiken der gesuchten Objekte. Auf den Testdaten erreicht der entwickelte Algorithmus gute Detektionsergebnisse unter verschiedenen und teilweise sehr schwierigen Bedingen, welche jedoch auch die Grenzen unserer Methoden aufzeigen.

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

The text document uploaded to TUGRAZonline is identical to the presented master's thesis dissertation.

Place	Date	Signature

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Masterarbeit identisch.

Ort	Datum	Unterschrift

ACKNOWLEDGMENTS

First and foremost I would like to thank Prof. Bischof. His inspiring lectures made me deciding to start the Master program in Computer Vision and provided me the opportunity and workplace for researching this Master's thesis at the facilities of the Institute for Computer Graphics and Vision. On top of that, he preserved this opportunity and gave me the necessary support for completing my study during the time I needed for rehabilitation from a serious road accident, and the followed birth of my beloved daughter which made me move 250km away from Graz.

I am also deeply grateful for my wonderful fiancée Sabine who was supporting me finishing this thesis with all she had despite the exhausting challenges of raising a new life.

Finally, I want to thank all the great people I met and was working with during my study, in line with all the experiences I was privileged to made in Graz.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Contents

In the recent decades of modern society, computers changed and affected almost every part of our working routines by enabling us to process tasks faster, more efficient and more reliable through automatization. The utilization of computers for increasing work effectiveness is still rapidly growing due to the expanding, more and more specializing fields in computer science - with no end in sight. One of the fastest growing research areas of the last few decades is Computer Vision, which aims at enabling computers to interpret visual information by mimicking the visual sense of humans, developing a vast amount of new application fields for computer systems. This Master's Thesis addresses the development and implementation of an automatic visual object detection system within a challenging practical application context.

Recent research came up with many object detection approaches on any kind of data, each showing different strengths and weaknesses regarding various application-dependent aspects like appearance of the physical target objects, properties of the data, computational costs and other characteristics. Thus, analyzing the target objects along with the application context yielding a well-defined problem description is a basic requirement in order to determine the best performing method for a practical use case.

In this first chapter we will outline the approached application, followed by a detailed discussion of the data and its acquisition procedure. Afterwards, we analyze the characteristics of the actual target objects along with their environment. All technically relevant aspects are then consolidated in order to derive the key requirements for our visual object

detection algorithm. Finally, this chapter is concluded by a summary, followed by a short outline of the thesis.

1.1 Application context

1.1.1 The DURAARK-Project

Figure 1.1: The official logo of the DURAARK project (taken from 1)

The application this thesis is addressing was derived in the course of a Europe-wide research project named DURAARK, the abbreviation of '**Dura**ble **Ar**chitectural **K**nowledge'. [1] It is funded by the European Commission within the 7^{th} Framework Programme, aiming at developing tools for the architecture community performing computer aided analysis and preservation of existing buildings and built structures. The project is organized in functional work packages, collectively pursuing multiple objectives:

- Enrich existing construction plans with 'as-build' data, enabling difference and evolution over time analysis
- Gaining additional semantic information about buildings, for instance connections of rooms and floors
- Longterm digital preservation of highly detailed 3D models

Thus the resulting tools are beneficial for various stakeholders like architects, engineers, construction companies, building owners, lawyers, researchers, public administrators, cultural heritage institutions etc., operating nationally or throughout Europe. Research, development and implementation of the DURAARK framework is cooperatively conducted by several consortium members, as depicted in Figure 1.2.

The approach presented in this thesis contributes to the DURAARK work package 'Recognition of Architecturally Meaningful Structures and Shapes', aiming at semantically enriching building models. A subcomponent named 'Shape grammars for almost invisible objects' is therefore supposed to estimate the location of electrical power lines inside walls. Necessary input for these shape grammar models are the locations of wall-mounted

[1] Official DURAARK Website - http://www.duraark.eu

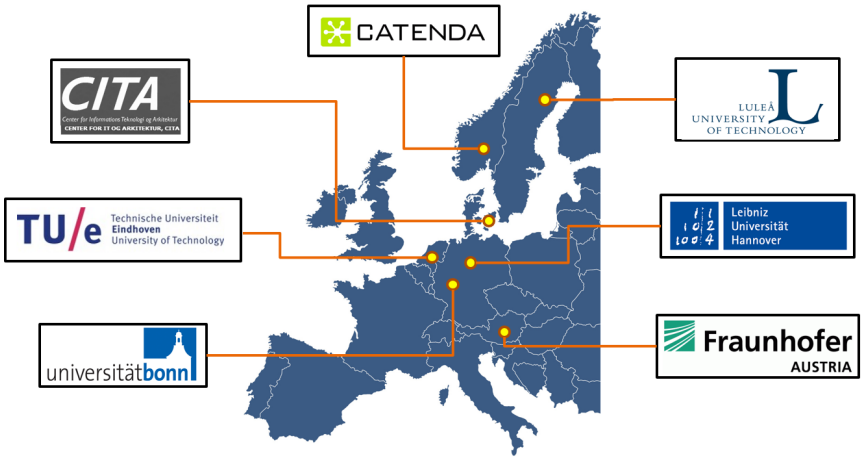


Figure 1.2: DURAARK contributors all over Europe. This map is taken from 1, where also more detailed information about all components and the ongoing development progress can be found.

power sockets and light switches, which should be automatically detected without the need for user interaction. More precisely, also the correct type of electrical device needs to be determined, since an existing light switch that is detected as power outlet, or vice versa, can lead to different results of the estimated power lines. This thesis approaches the automatic detection and categorization of these objects by machine vision methods. [2] Additionally, the developed algorithm has to feature adding and updating object categories to preserve general usability and maintainability.

1.1.2 Data acquisition and properties

All developed tools of the DURAARK framework are designed to operate on collectively shared input data that is retrieved from an onetime acquisition procedure. This convention is supposed to minimize the operators effort and maximize the benefit of the DURAARK tools for all stakeholders. In practice, the data is captured by a stationary, commercial laser scanner placed at different locations inside a building and consists of typically one colored 3D point cloud per room and an associated panoramic image of the indoor scene, which are generated by a proprietary scanner software. Obviously, the detection algorithm we address in this thesis should also operate on the same data standards defined for the DURAARK framework. Figure 1.3 illustrates an acquired point cloud and panoramic image of a sample room that fulfills the requirements for the DURAARK tools.

Basically, for data acquisition any arbitrary scanner that is capable of retrieving a complete 3D point cloud and a panoramic image of a room can be used. Nevertheless,

[2] Note that the development of the associated DURAARK component is assigned to Fraunhofer Austria. Thus, some denoted approaches beyond image recognition that are mentioned in this thesis were developed in cooperation, as they form necessary actions for the component integration.

a)

b)

Figure 1.3: Data samples of a sample room acquired by a *Faro Focus 3D* LiDAR-scanner for the DURAARK framework. Figure a) depicts a snippet of the colored point cloud. This scanner type acquires the range data by measuring the traveling time of an emitted laser beam that gets reflected by the scenery. Typical phenomenons in LiDAR-scans like hidden regions or hallucinated points induced by occlusion, transparency, multi-path reflections or depth discontinuities are clearly visible. Figure b) shows the corresponding panoramic image that is additionally acquired by the scanner software by stitching single shots from an embedded camera. Since the purpose of this camera is mainly to retrieve the color information for the 3D points, it operates on fixed zoom level and its native resolution is rather low. Also the dynamic range of the camera is insufficiently adapted to capture objects near the light-flooded windows.

in order to estimate the expectable density of the data, we shortly cover some relevant specifications of the *Faro Focus 3D* scanner, since this model was used for capturing the reference data.

This particular LiDAR-scanner emits the laser beam via a horizontal and vertical rotating deflection unit. A distance measurement is triggered 40.960 times per rotation, or at fixed angular steps of 0.009° for each axis. A proprietary noise reduction algorithm reduces the ranging error to $\pm 2mm$, but simultaneously decreasing the number of points by a factor up to 16. Thus, the scanner achieves a theoretical number of over $100M$ points per scan. Due to self occlusion and redundancy filters, the effective resolution of the resulting point cloud is about $25M$ points for a single scan. This is an excellent value for a modern 3D scanner, nevertheless, the number of 3D points are limited. In varying geometric setups, this limitation may become relevant for the detection of small objects. Due to the rotation-based capturing mechanism of almost every LiDAR-scanner, the density of measurement points sensing a cuboid room is not homogeneous. The expected scan resolution of a planar wall segment significantly decreases with the relative distance and angle between the scanner and the walls surface. As light switches, but especially power sockets are often mounted close to room corners where the measurement density is supposed to reach its minimum, a more detailed analysis of the lowest expectable data resolution becomes necessary. For example, in a room of $5m \times 6m$ and a height of $3m$, a center-placed *Faro Focus 3D* scanner achieves a spatial resolution of only about 7 to 8 points per cm^2 at the least dense sampled wall regions, which is about one point each $3mm$. In a scenario similar to the test room shown in Figure 1.3 where the scanner is positioned off-center, the lowest achievable resolution further drops to 3 to 4 points per cm^2. The calculation and simulated scenarios are described in Appendix A in more detail.

Beside the 3D point cloud, also a panoramic image of the room is generated by the scanner. The necessary 2D data is captured by a build-in RGB camera with a native resolution of 2 megapixels that takes about 85 separate images during a 360° scan. The proprietary scanner software warps and stitches those images internally, yielding the registered panoramic image. Although the single images can be exported from the raw data format, it is difficult to utilize them as an additional information source since even after a request to the scanner manufacturer the capture poses of the individual images are not accessible. Anyway, similarly to the 3D point cloud, the level of detail conveyed by the 2D data strongly varies with the scan position relative to the corresponding wall segment.

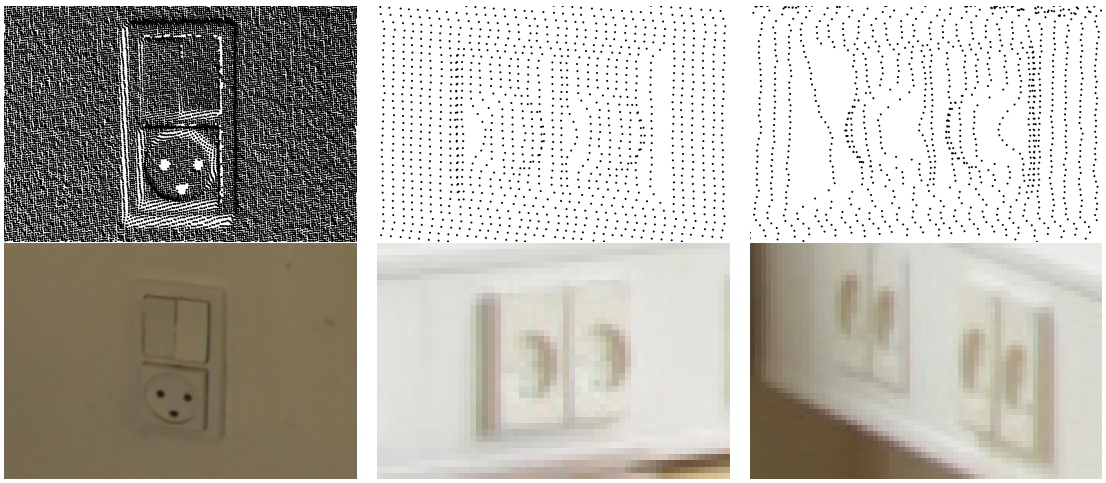
Practically, off-centered acquisition setups can quickly become the norm since stakeholders using the DURAARK tools will most likely tend to select the scanning positions as economical as possible to fulfill their main intention, which is digitally modeling a buildings structure. In other words, the scanner will be placed where the majority of the walls can be observed at once, in some cases simultaneously decreasing the effective resolution of the acquired 3D point cloud and the panoramic image significantly. Figure 1.4 shows an illustrative sample of a scanned room, where the chosen scanner position clearly supports recovering the rooms full structure, while causing a highly inhomogeneous and tremen-

dously low data resolution at some wall segments. Beside the rooms architecture, also existing furniture may drive a stakeholder to choose a suboptimal scanning position in respect to sampling density, as happened in the example of Figure 1.3. Figure 1.5 presents data snippets of different target objects from Figure 1.3, pointing out the effects of the chosen scanner position and limited data resolution on the actual data.

Figure 1.4: A sample room scanned in the course of the DURAARK project. The scanner was placed above the dark spot. Due to the chosen scanning position, the wall surfaces and thus the room structure is captured at once. However, the chosen scan position is unfavorable in terms of data density on long-sided wall segments. This image was taken from [33].

1.2 Target objects

Considering their physical appearance, wall-mounted light switches and power outlets generally consist of rather plain, untextured, monochromatic surfaces. They are typically designed to be unobtrusive and assimilate with the walls surface and their most distinctive visual information are edges caused by the objects silhouette and functional indentations. Although separate instances of target objects serve the same purpose, their appearance strongly varies as a result of manufacturers offering many different designs to fulfill all possible customer requirements regarding color, shape and functionality. Furthermore, different national standards for electrical installations and miscellaneous jointly framed switch-socket combinations additionally increase the design-induced object variability. For illustration, Figure 1.6 shows different designs of similar instances of a widely used mounting setup. However, common practice limits the expectable rotation of a mounting frame relative to the wall to 90° steps. The most common visual characteristics of our target objects are therefore strictly horizontal or vertical orientation of edges, accompanied by a rather fixed physical size and a symmetric shape. As defined by the application, the environment in which the electrical devices should be detected are typically full-furnished rooms.



a) b) c)

Figure 1.5: Snippets of the point cloud and the corresponding parts of the panoramic image. For better visibility, the acquired 3D points are shown as black dots on white background. Figure a) shows a target object that is represented in a high level of detail, since the scanner was positioned right in front of the electrical device at a distance of about 1 meter. Other parts of the room are scanned less dense due to higher relative distances and acquisition angles, as shown in b) and c). Whereas the power sockets in b) were scanned from a higher distance compared to c), the outlets in c) were captured from a more skewed view point, which results in a diverging horizontal and vertical resolution and an overall lower number of points. Therefore, the most distinctive parts, the object gradients, are only barely captured in b) and c). For this visualization, the image parts were taken from an HDR panoramic image, which was captured by a separate DSLR camera. Its alignment and size were fitted to the original panoramic image, that was generated by the proprietary scanner software.

a) b) c) d) e)

Figure 1.6: Target objects of the same type and arrangement, but different designs. Figure a) shows a Danish socket model without the commonly known cylindrical notch, whereas b) to e) show the in German speaking countries more established Schuko plugs. Regardless of the type of an electrical device, there exist a vast of different models and designs.

When analyzing the visual properties of target instances with respect to data characteristics, we can obtain that the object classes show a high *intra*-class variability, which denotes the variances within a single object category. Further, the expected environmental scene is also characterized by a high variety. Since all these possible non-target objects can be collectively represented by a background (or 'negative') class, also this class is characterized by a high *intra*-class variability. When observing possible environmental objects at the targets scale, they often show similar structures and edges compared to the actual target objects. This occurring similarities between environmental structures and targets cause a low, so-called *inter*-class variability, which describes the discriminability between different data classes. Furthermore, the rather similar appearance between different categories of electrical installation devices (e.g. power outlets and light switches) confirms a low *inter*-class variability also for object type, or 'positive' classes.

Both properties, a high *intra*-class variability combined with a low *inter*-class variability over all positive and negative classes, unavoidably evoke several situations where target objects of different classes and/or their environment are hard to distinguish. This characterizes a challenging task for data separation in general and so for the automatic detection algorithm we are addressing, already when considering the bare appearance of the target objects.

1.3 Deriving the Computer Vision Problem

Based on all aspects of the practical problem, the technical requirements of the automatic visual object detection algorithm can be derived. First of all we notice, that since the shape grammar models of the associated DURAARK tool also require information about the type of electrical device, we are facing a multi-class detection problem. Thus, our automatic object detection algorithm has to be capable of detecting and, additionally, identify the correct object category. Furthermore, the corresponding DURAARK module should offer the possibility to add additional training examples and classes. Therefore, (re-)training and detection should be executed as short as possible in order to keep a certain level of usability. However, since training and detection can be done off-line on a provided server machine, resources and computation time are not strictly bound to specific requirements.

1.3.1 Handcrafted vs. Automatically Generated Features

Usually, visual data in its original form has a tremendously high degree of freedom, thus, it can transport a vast of informational content, as expressed by a famous idiom *'A picture is worth a thousand words'*. However, for an automatic recognition system to operate computationally feasible, the data has to be reduced in dimensionality by some mechanism while preserving the essential informations. Even when considering the recently trending convolutional neural networks (CNNs) for automatized pattern recognition [34, 47, 50],

dimensionality reduction is intrinsically performed by forcing the network to convey the input information by only a relative few amount of neurons in inner layers. Anyway, the required properties that assess visual information as essential are most often task-dependent. In computer vision, the common terminology of compressed data representing semantically meaningful information about the scene is *a feature*. In conventional computer vision approaches, algorithms that extract features from input data are mostly handcrafted and do not require an extra training procedure. Thus, in contrast to methods which are able to learn features automatically (like CNNs), training decision algorithms on handcrafted feature descriptors usually requires much less data examples in order to reach a sufficient level of performance. Considering our application, the amount of available data examples showing electrical installation devices is very limited, simply due to the relatively high effort that is necessary for capturing those data. Therefore, in the huge field of computer vision algorithms, we focus on classical object detection and recognition methods that are based on a pipeline consisting of carefully selected feature descriptors evaluated by a well designed machine learning classifier.

1.3.2 Exploring the 3D Data

As already outlined above, simulations (see Appendix A) and actual sample scans clearly indicate that the expectable, effective sampling resolution of the target objects is rather low. Especially for 3D points, low data quality may become very critical for the detection of relatively small and undistinctive structures. Therefore, we shortly cover existing computer vision approaches on 3D object description and detection in order to evaluate the quality of the available 3D data for our problem.

In the past years, free-form object detection and recognition on 3D data has been an important research field in computer vision. Basically, we can observe three trends in the recent development of detection algorithms on 3D data. The first pioneering kind of methods are based on point-wise, local 3D surface description and matching [1, 11, 30, 49], following the same intention as highly successful interest point descriptors for 2D images, like the SIFT [36] or SURF [2] descriptor. In general, all local 3D point representations are generated by evaluating geometric measures of the narrow surface region around the normal vector of a certain point. Whereas *Point Signatures* [11] record a one dimensional distance profile derived from the intersections of an imaginary sphere, *Spin Images* [30] project the 3D surface information to a discretized 2D image by accumulating the surface coordinates mapped to a cylindrical basis. Another point descriptor called *Point Fingerprint* [49] is generated by projecting centered geodesic circles of different radii onto the feature points tangent plane. Whereas [11, 30, 49] operate on a fixed, predefined size of their support regions, [1] additionally utilized the information of the natural scale of each point by searching for the maximum response over a discrete scale space.

The next generation of 3D object recognition approaches exploit the overall structure of target objects already in the matching phase by incorporating the spatial information of

two or more local surface cues, introducing an additional amount of discrimination power and robustness to clutter and occlusions [17, 19, 40]. In the early COSMOS framework [17] shapes are matched by modeling free-form objects as a combination of stereotypical shape primitives. A more modern approach utilizes vertex pairs to construct the bases for surface-descriptive 3D tensors which store intersecting areas of the local 3D surface [40]. The approach of [19] pre-calculates point-pair features from every possible pair constellation of a training object, which can be recognized within the query data by a fast Hough voting scheme.

The latest trend in 3D object detection algorithms advanced the incorporation of multi-modal data [18, 26], since modern acquisition hardware usually retrieve not only the 3D structure, but additional intensity or color information from the scene. The basic intention of those multi-modal detection algorithms is to combine the strength and compensate the weaknesses of different data modalities. [18] extends the point-pair features of [19] by aggregating both, intensity and depth values, to extract the more descriptive geometric edges of objects. In contrast, a more generic approach [25, 26] utilizes a large set of pre-calculated multi-modal templates, which in turn consist of the most discriminant features of the individual modalities.

Although recent research brought up many successful approaches on object detection and recognition in 3D data, applying them straight-forward to our problem is not supposed to work equally superior. To perform 3D object detection by matching local 3D surface representations, in general a high degree of repeatability, unambiguity and robustness to rigid transformations has to be achieved. These properties strongly depend on the distinctiveness of the target objects and a robust acquisition of the surface information. Considering the simple, flat and symmetric shape of our target objects, the error-proneness to self-similar structures when matching local interest point descriptors is nearly impossible to compensate by post-filtering the matching results. Furthermore, the low-resolution data due to unfavorable view-points inhibits a repeatable and accurate measurement of the surface and its normal vectors, which are mandatory for the calculation of local surface descriptors. Matching low-resolution data would require more relaxing tolerance parameters, which result in an additionally increasing probability of false matches.

Approaches modeling the global appearance of targets via a few amount of distinct local cues combined with their spatial configuration are therefore better suited for our small, unobtrusive objects. However, these methods are very sensible to even slight shape variations, since their trained object models are mostly designed to match the exact learned 3D structure within the scene. This lack of generalization capability makes these methods very hard to utilize for detecting a group of highly varying shapes. A high training effort of providing models of all possible targets, or loosened matching criteria would be required, however, which in consequence would introduce a higher possibility to falsely respond to all other planar non-object structures. Nevertheless, the recognition performance of global model descriptors also significantly decrease with the low data resolution. These

restrictions also apply to approaches processing multi-modal information. Further, the expected improvement of recognition performance by exploiting multi-modal data is also very limited for our problem due to the weak distinctiveness of the target objects in every modality.

We conclude that, in contrast to tasks whereat prominent objects with an unique and distinct 3D structure should be detected, for the wall-embedded and planar targets we are interested in, the available 3D data is neither valuable nor detailed enough to perform reliable object detection and recognition directly on that data. However, the acquired 3D data contains other very useful cues we can exploit.

1.3.3 Wall Texture Images and 2D Feature Requirements

Whereas the 3D information does not feature enough details for directly detecting the target objects, the room structure and thus the alignment and size of single wall segments can be reliably recovered. Thus, we pursue an approach that uses the knowledge about the room geometry in combination with the available panoramic image to generate separate rectified texture images of each wall segment. Performing object detection on such texture images benefits from a significant reduction of the complexity of our object detection problem because of two main reasons. First, due to the known physical dimensions of the room and the almost constant size of our wall-mounted target objects, a detection on texture images can be performed on a fixed scale representation by utilizing a simple sliding window approach. And second, whereas the prominent planarity of our target objects is in general an unfavorable property for almost every 3D recognition technique, on a texture image that is aligned to the major plane of flat objects otherwise challenging view-dependent distortions are rectified to a more invariant appearance. Furthermore, this approach also serves a more convenient usability for stakeholders when using the final software, since adding additional object classes or instances of their own interests requires only providing rectified images. This generally involves much less effort compared to providing new 3D data of unseen objects to retrain the algorithm. Beside the generation of texture images, a known room geometry can be also utilized to exclude certain areas from object search that naturally do not contain electrical wiring, like doors or windows, and positive matches can be verified by additional constraints like common mounting heights of specific object categories.

The texture generation algorithm was developed in cooperation and implemented by Fraunhofer Austria [33], as the room geometry is also a mandatory input for their power line estimation. In order to extract the wall segments from the 3D data they designed a semi-automatic approach involving several existing software tools. Given the wall configuration of a room, a texture image is created by back-projecting the panoramic image from the surface of a sphere onto the corresponding wall plane. In their work, they also describe the registration of an additionally captured HDR panoramic image to the original

panoramic image taken from the laser scanner. Although such a separately acquired image could be theoretically utilized to enhance the resolution and thus the level of detail of the 2D information, the main intention in [33] was to improve the lightning conditions of the result images, therefore, they resampled the HDR panoramic image in advance to match the scanner image size. However, for real application scenarios using the DURAARK framework we can not assume the presence of a secondary image source. By convention, the texture images are mapped to a spatial resolution of $1\,mm^2$ per pixel, thus a common light switch of $90\,mm \times 90\,mm$ covers $90\,px \times 90\,px$ on the resulting texture image, which form the final input data of the detection algorithm that is addressed by this thesis. Figure 1.7 shows the generated wall textures of the same sample room of Figure 1.3, which are based on an additionally acquired HDR panoramic image that was aligned with the original scanner panorama.

As discussed in previous sections, it can be observed that also for captured 2D images unfavorable scan positions have a significant impact on the expected data quality. However, in contrast to 3D points where with increasing sparsity sometimes indentations on surfaces are not captured at all, in the case of low resolution 2D images the visual edges are smoothed out without loosing the information entirely. Nevertheless, for many physical locations the wall textures show a very high level of image blur. This is mainly caused by disadvantageous acquisition locations but also by two subsequent processing steps of the original image data, as the panoramic image is in advance already generated by warping and stitching individual images before the panorama is warped and resampled again when calculating the wall projections.

Consequently, well designed 2D descriptors are required to reliably model the visual appearance of wall mounted objects on the resulting texture images. Basically, on the one hand a feature descriptor is constrained to yield similar vectors for the same type of target objects, regardless of varying preconditions during the data acquisition procedure, while on the other hand good features are supposed to group instances in feature space according to their object class as clearly as possible. Matching both demands is especially challenging if already the physical appearance of the classes is characterized by a high inter- and a low intra-class variability.

For practical computer vision applications in general, probably the most successful object detection approaches are based on local patch descriptors calculated on around saliency points [8, 36]. However, in the case of untextured electrical devices exclusively consisting of undistinctive edges and corners, these methods are prone to generate only an insufficient amount of interest points which also have a high chance to form false correspondences in heavily cluttered scenes. Furthermore, when considering our approach of using rectified texture images combined with the physical shape and mounting practices of the target objects, the advantageous properties of algorithms based on local patch descriptors on handling occlusions and most likely a certain degree of invariance to scale, affine and perspective distortions become minor demands.

In contrast, a well suited feature descriptor for our addressed application requires

to yield similar values for images of highly varying quality. Related to this property, another important demand is a high invariance to illumination changes since different wall segments of a room show significantly diverging lightning conditions. Therefore, we favor image descriptors that are designed to model the overall visual appearance of a larger image patch showing an electrical device in its entirety, and focusing on specific arrangements of image gradients while at the same time also tolerating an adequate level of diversity In order to exploit various visual cues that are emphasized differently by certain descriptors, we consider combining multiple algorithms forming a high-dimensional vector.

1.3.4 Classifier Requirements

Whereas choosing proper image descriptors is basically the most crucial step in designing a classical computer vision pipeline, the addressed problem also requires a well considered learning algorithm. As an initial step on defining the requirements on the classifier, a first choice on the actual learning objective has to be made. In general, an automatic data classifier can either directly learn a posterior probability model via parameter fitting according to given data for each class, or, learning a conditional probability model from discrete observations how to best separate the classes. Classifiers where the first case applies are commonly known as generative models, whereas the latter type of classifiers form the group of discriminative models. The first decision that has to be made for selecting an appropriate algorithm is whether a generative or a discriminative approach is better suited for the corresponding task. As shown for example by [41], the common belief that discriminative approaches necessarily perform better than generative models is not always correct. Furthermore, Bishop and Lasserre [3] even proposed a method to combine both, generative and discriminative models.

However, the environment in which our rather undistinctive target objects need to be detected introduce a lot of clutter which can mislead the detection algorithm due to similar looking structures. The high amount of common visual cues between different object categories of electrical components may also result in confusing target object classes. Therefore, we prefer a discriminative approach to learn different object classes and background against each other, because those algorithms typically focus on finding distinctive features that best discriminate the classes, instead of generally modeling the target object appearance whereat the probability of producing a high feature overlap between similar looking object categories is potentially much more likely. Additionally, discriminating classifier models typically require much less training data compared to generative approaches, which facilitates the process of extending and adding object classes.

On the downside, this decision requires establishing a ‘background’ or ‘negative’ class that is theoretically supposed to hold all possible non-target image patches of indoor environments, i.e. basically everything imaginable inside a room that is not a component of electrical installation. Needless to mention, that this class has to contain of a very large amount of samples.

In contrast, for usability purposes introducing new object classes should be feasible even for non-experts while requiring as less effort as possible. Beside other consequences, this aspect also yields a low expectable amount of actual target samples. Thus, whereas the background class needs to contain a huge amount of training samples, only a small amount of training instances of target objects should be sufficient. This results in a highly imbalanced data set. In order to handle imbalanced training data, basically for every classifier approach over- or under-sampling techniques can be applied. However, we assume that re-sampling in an amount-compensating scale of such extreme imbalanced data like ours will exceed its reasonable legitimation relatively fast, as the representativeness of real-world scenarios by the training data would drift away too far from reality. Hence, one key requirement on the classifier is that it can deal intrinsically with extremely imbalanced training data associated with a relatively small absolute amount of target samples.

This context in conjunction with a high dimensional feature vector, another important characteristic of a suitable classifier algorithm can be formulated. As someone would naively expect using more image descriptor approaches yields more valuable data for the classifier and thus necessarily increases classification performance, on the downside also many redundant or generally weakly informative measures are produced. This may compromise the final performance of a classification hypothesis, especially when the classifier is trained from very few data samples and easily struggles with under-determined learning problems in general. Thus, the capability of identifying and focusing on the best features out of a large feature pool while paying less attention to those of less discriminative power is also a desirable property.

Reconsidering non-expert users providing new training data on purpose, other demands on the classifier algorithm are the ability to deal with a certain level of noise in the data, providing a fast re-training procedure and an extensible multi-class capability, all without the need of extensive testing and special parameter tuning.

1.4 Summary and Outlook

Concluding, we are addressing a multi-class object detection problem targeting wall-mounted electrical devices in full-furnished rooms. To ensure an adequate level of assimilation to interior design and additionally serving various customer needs, their 3D structure and their visual appearance is very unobtrusive, but simultaneously showing a high degree of diversity. More generic, the targets can be characterized as untextured, plain, small but individual objects, whose most distinctive parts are a small amount of gradients caused by functional indentations and the outer object boundaries. The data is given by a real-world application and consists of a 3D point cloud and a 2D panoramic image which are captured by a LiDAR scanner utilizing its proprietary software. Since the LiDAR scanner operates on a fixed radial scan grid, more distant and skewed surfaces can show a very low level of detail in the acquired data. These conditions combined with

a desired economical usage of scans limit the expectable acquired details of the target objects in the investigated scenes significantly. In fact, the most discriminative parts of power sockets and light switches are not captured reliably by the available 3D scans, which consequently disqualifies existing object detection methods on 3D data to sufficiently solve our problem. However, the characteristic visual structures have a much higher chance to be captured by the panoramic images although they may show a very high level of blur. Therefore, we propose an object detection approach based on 2D wall texture images that was especially designed for the addressed application and exploits the available information and the properties of our targets the most beneficial way. First, the extracted 3D information about the room geometry is used to map the panoramic image to a rectified, fixed scale representation. Afterwards, a standard computer vision pipeline consisting of well designed feature descriptors and a classifier algorithm is applied under a sliding-window approach to detect the target objects. While the extraction of the texture images is necessarily conducted by software modules assigned to Fraunhofer Austria, this thesis addresses the visual object detection based on those wall texture images. Here, the main challenges lie in reliably modeling object-sized image patches under varying quality and illumination conditions and dealing with highly imbalanced datasets. In order to achieve satisfying detection performance we claim that both, the feature descriptors and the classifier, need to be carefully designed and adapted to each other.

In Chapter 2 we cover state-of-the-art methods on feature descriptors and classifier algorithms which are most suitable for the given problem. The best suited concepts form the basis of our developed approach, which is described in more detail in Chapter 3. Experiments and evaluations of the vision pipeline on real test data are presented in Chapter 4. Finally, thesis is concluded in Chapter 5, followed by a discussion of possible further improvements that may enhance the achieved results.

a) Wall textures (top, bottom) extracted from panoramic image (center)

b) Rectified target objects

Figure 1.7: Figure a) shows all relevant wall textures in their original scales ratio that were generated from a panoramic image and a known wall configuration by projecting the image onto a canonical plane of a fixed spatial resolution. Hence, the geometric scale of all wall-mounted objects is constant over all resulting texture images. Figure b) shows equally sized snippets taken from different texture images that contain power sockets. Due to the spatial rectification the cylindrical notches of all sockets are shown as equally sized circles, whereas the image quality varies significantly.

CHAPTER 2

RELATED WORK

Contents

In this chapter we outline state-of-the-art methods addressing to our object detection problem and discuss their applicability in respect to the derived requirements. We begin with an insight into recent achievements on detecting power sockets in the field of robotics, although they basically have to fulfill different requirements. The main part is divided into three sections. The first part gives the reader a general overview of recent approaches tackling visual detection of untextured objects. The second part covers potential candidates of feature extractor algorithms that fulfill our demands, where each of them represents a diverse, powerful concept of describing the overall appearance of an image patch. Due to the special requirements on the classifier, we complete the main part with an outline of three machine learning techniques that have successfully proven their applicability in many practical applications and discuss their strengths and weaknesses in respect to our problem. Finally, this chapter is concluded by a summary of the acquired knowledge.

2.1 Detecting Power Outlets in Robotics

Considering the task of detecting power sockets, recently this problem gained much attention in the field of robotics in order to build self-charging robots [7, 20, 39]. In [7] and [39] the recognition task is basically done by template matching after image rectification according to the normal vector of the wall. In [20], they segment different hole candidates

of a power outlet by applying a simple version of the *Maximally Stable Extremal Regions* ('MSER') approach [16, 38]. Then they classify them as power holes, ground holes or background by applying the one way descriptor method [27] and finding the nearest neighbor in PCA space, followed by geometric filtering the results.

Although these approaches have successfully enabled autonomous robots to detect power sockets in indoor environments which basically matches also our problem description, however, further investigations reveal significantly different requirements of these methods which disqualify them for our application. Generally, in robotic applications, the remote camera is actively movable and thus able to investigate a wall segment multiple times from different perspectives and positions. This opportunity is deliberately exploited by robotic scientists. Whereas in [20, 39] the robot actively navigates to a known position of a power outlet before visual detection is performed, in [7] only a standardized mounting height is investigated while moving the robot along the wall.

Considered from a computer vision perspective, these methods are rather error-prone to bad image quality, clutter, and a high intra-class variability, e.g. different designs and models of sockets. Furthermore, typically only one specific socket type has to be recognized and false negative detections (missed sockets) on a single image are not critical in the application context. When object detection on fewer or even just one single scan should be performed, much more reliable approaches which are capable of handling a wider range of view angles and appearances become necessary.

2.2 Detection of texture-less Objects

2.2.1 Recent approaches

Since untextured objects mostly require specialized approaches to describe their appearance the most expressive way, texture-less object detection in general became a rather independent research field in computer vision. When objects cannot provide specific textures, their contour typically becomes their most discriminative feature, on which untextured object detection algorithms usually focus on. Thus, a reliable extraction of the object boundaries is a mandatory preprocessing step for many methods. Holzer et al. [28] utilized distance transform templates to detect and estimate the relative pose of untextured objects based on their contour for augmented reality purposes. Their approach requires extraction of almost closed contours, which they compute by applying the Canny edge detector [9]. Likewise, the point pair features of Drost et al. [18] which describe constellations of geometric edges for texture-less object detection, also uses the Canny edge detector in combination with a depth image to extract the edges.

A different approach to acquire and describe object contours, more precisely regions enclosed by contours, is the MSER detector and descriptor [38]. This approach evaluates the evolution of the resulting binary regions when thresholding the input image over the full range of intensity values. Regions are considered as maximally stable when

their appearance does not change significantly over a pre-specified amount of consecutive thresholds. For efficient calculation, a connected graph structure can be utilized to model the evolution of regions over different thresholds and allows using the MSER detector and descriptor even for real-time applications [16].

In an approach less dependent on the clarity of gradients, Hinterstoisser et al. [26] match templates consisting of gradient responses to implement fast and reliable detection of untextured objects in natural scenes. Their concept is based on describing a target object by a set of gradient response maps incorporating the location and the dominant orientation of a gradients for different poses. In order to extract the gradient information, they simply use the maximum derivative of the intensity values over each color channel.

Discussion Considering the challenging input data of our task, in which the visible object boundaries are usually blurred and tend to vanish in the surrounding, we do not consider approaches relying on reproducible results of the Canny edge detector as sufficiently robust. Similarly, although the MSER descriptor and detector is very robust to illumination changes, this detector only retrieves regions that show a sufficiently stable and closed contour. On the other hand, template based approaches are much less vulnerable to missing fragments of the object contour. However, they require a relatively high number of templates for representing one object and are rather restricted to recognize only the exact instances represented in the training data, which is a significant drawback considering the usability of our application.

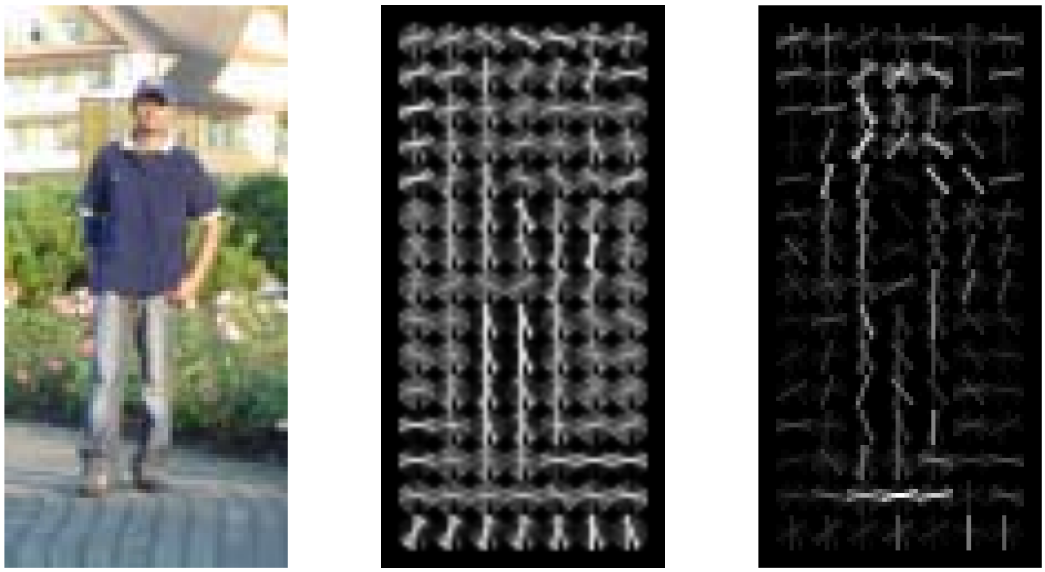
Concluding, in literature we could not find an approach that promises good results when applied directly to our task, mostly due to the special properties of the input data. Therefore, as already mentioned, we aim at developing a standard computer vision pipeline consisting of carefully selected feature descriptors and classifier, of which the most promising concepts are discussed in the next sections.

2.2.2 Feature Descriptors

2.2.2.1 Histogram of Oriented Gradients

The first image descriptor that matches the majority our requirements, is the 'Histogram of Oriented Gradients' (HoG) descriptor of Dalal and Triggs [13]. This very discriminative and at the same time generalizing representation of object silhouettes compiles gradient histograms binned to their direction over certain sub-image-patches (referred to as 'cells') via a weighted voting scheme incorporating the gradient magnitude. These cell histograms are then normalized over (usually overlapping) cell-blocks forming the descriptor values. While this approach also effectively codes the spatial configuration of the gradients strength and orientation, it simultaneously tolerates a certain level of variation given by the adjustable size of the cells and blocks. Undeniably, the discriminative power and thus the success of this descriptor also relies on a well considered preprocessing, including gamma and color normalization, and an aliasing

avoiding interpolation of the histogram votes. Beside the basic algorithm, Dalal and Triggs also described and investigated different variants of their descriptor. In order to exploit color information, they proposed calculating separate gradients for each color channel and considering only the one with the highest magnitude for each pixel. In order to equally model silhouettes that can be formed either by light-dark or dark-light transitions, they defined an undirected version of the HoG descriptor that bins gradients in a range of $[0°, 180°)$ instead of the full range of $[0°, 360°)$. Furthermore, they also investigated rectangular and circular arrangements of the cells to form different block geometries. In the original paper, Dalal and Triggs applied a Support Vector Machine ('SVM') classifier to their descriptor in order to perform detection of humans. Since then, many object detection methods successfully reused the concept of the HoG descriptor combined with a SVM classifier as a basis for more sophisticated algorithms. However, considering our problem of detecting rigid, planar object on a fixed scale, the original HoG implementation fulfills all necessary constraints. A common visualization of the HoG descriptor on the original example of [13] is shown in Figure 2.1.



a) Input image **b)** Hog features **c)** Weighted HoG features

Figure 2.1: Visualization of the undirected, rectangular version of the Histogram of Gradients (HoG) descriptor applied on an image of a human. Figure a) shows the input image. Figure b) illustrates the calculated Histogram of oriented Gradients descriptor, where the length of the white lines in each cell represents the strength of the binned gradients in their corresponding direction. Additionally to b), in Figure c) these lines are weighted by their importance values for the task of a human detector, which were learned by a Support Vector Machine. All three figures are taken from [13].

2.2.2.2 Statistics over local subregions

An other, more generic technique to describe the content of image patches by a few values is to evaluate first- and higher-order statistics over spatially distributed subregions. Based on this concept, a lot of different image, but also interest point descriptors emerged to fulfill various requirements, spanning from very simple but highly efficient binary tests of single pixel values in the BRIEF descriptor [8], over rectangular Haar-like wavelet features [42, 53] that estimate local derivatives, up to more complex weighted combinations of multiple rectangular subregions [15]. The usage of rectangular subregions in this context allows very efficient computation of sums over these regions via integral images [53].

In order to cover as much diverse characteristics of target objects as possible, the majority of the most successful approaches select their feature regions in a highly randomized fashion. Whereas this random generation strategy often brings up very powerful and sometimes unexpected features, also many rather weakly informative values are produced. Therefore, usually a classifier with good feature selection capabilities is required for those image description strategies. Whereas the first upcoming approaches evaluated subregional statistics only on the original intensity images, this technique also enables expressing multiple filter responses or image sources by a common feature vectors in an effective way.

This methodology was intensively used by Dollar et al. in [14] to generate features from a diverse collection of image filters, which they refer to as 'feature channels'. These feature channels were then described by a large amount of randomly generated first order sums of rectangular subregions and more complex weighted combinations, even spanning over different filter responses. In their work, as feature channels they used gray- and color-information, Gabor-filter responses, gradient magnitudes, Canny edges, thresholded images, product-, mean- and maximum-values and gradient histograms, forming a competitive pedestrian detector. A more general object detector [31] combines scene information from different sensors (i.e. a depth image aligned with various filtered responses of color image) and aggregates the feature values by applying Haar-like wavelets over the different scene representations. With the concept of statistically interpreting image information of certain subregions, also histograms can be calculated via first order sums of spatially distributed rectangles applied on quantized versions of the original image. For example, in [56] Zhu et al. used integral images to approximate HoG features.

A graphical illustration of the calculation of first order sums over rectangular subregions via integral images that combine different filer responses to a single feature vector is shown in Figure 2.2.

2.2.2.3 Principal Component Analysis

Instead of extracting visual features as partial quantities directly from the image data, another approach to describe image content is based on interpreting whole images as single data points in a high-dimensional data-space whereat each dimension represents a pixel.

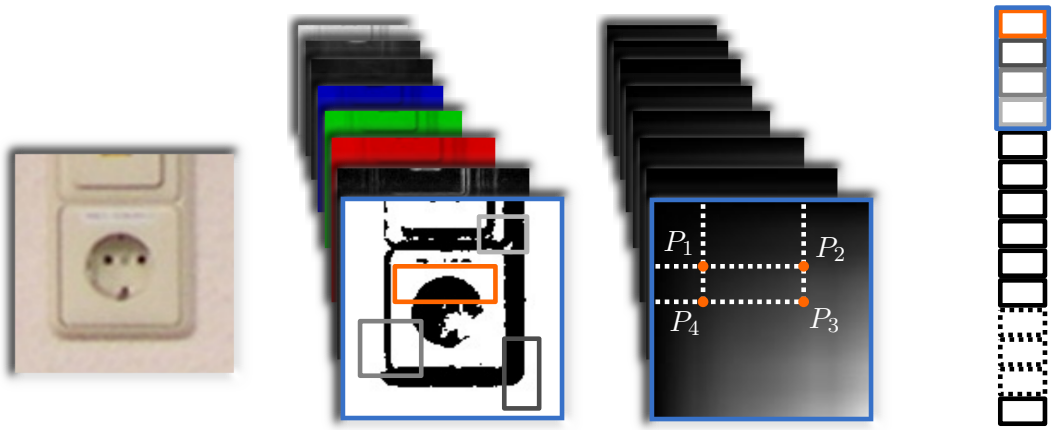


Figure 2.2: Calculation of first order sum features over rectangular subregions via integral images from different channels. Multiple image filters that are applied to the query image to form different feature channels. These feature channels can be effectively described by spatial first order statistics of randomly generated rectangular subregions. For efficient computation, for each filter response image $R(x, y)$, an integral image representation is calculated as $I_R(x, y) = \sum_{i=1}^{x} \sum_{j=1}^{y} R(i, j)$. Using this representation, the sum of a rectangular subregion S_k in image $R(x, y)$ can be determined just by evaluating the values of its 4 corner points in the corresponding integral image as $S_k = I_R(P_3) - I_R(P_2) - I_R(P_4) + I_R(P_1)$. In order to describe the origin image, the calculated sums of all feature channels are concatenated in one feature vector. Also other measurements can form the image description vector, like differences of sums of rectangle pairs or more complex, weighted sums distributed over multiple feature channels.

While directly classifying these data is usually unfeasible due to the high dimensionality, data compression methods can be applied as a form of a learned feature descriptor that reduces dimensionality by mapping images onto a much lower dimensional feature space. In order to preserve a high level of informational content, this feature space is spanned by precomputed basis vectors representing the most descriptive characteristics of a training set.

One data compression method that was successfully applied to images is the Principal Component Analysis (*PCA*) which finds new orthogonal basis vectors in whose direction the original data points show the highest variance. These directions are calculated by the Eigen-vectors of the covariance matrix containing all input points. After projecting data points from the original data space to their so called Eigen-space, only the dimensions corresponding to the highest Eigen-values are needed to represent the most essential information of the original data. Hence, this method is a lossy data compression method that performs dimensionality reduction in a derived orthogonal subspace, whereat the variance of projection of pre-observed data is maximized and simultaneously their reconstruction error is minimized. Mathematically, for a data matrix

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_0 & \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_{n-1} \end{bmatrix}; \ \mathbf{X} \in \mathbb{R}^{m \times n}$$

holding n m-dimensional data points as column vectors, an Eigen-vector $\mathbf{u}_i$ and its corresponding Eigen-value λ_i must fulfill

$$\mathbf{X}\mathbf{X}^T\mathbf{u}_i = \lambda_i\mathbf{u}_i.$$

Theoretically, there exist m Eigen-vectors where $\lambda_i \neq 0$. In PCA data compression, only a small amount $k \ll m$ of Eigen-vectors $\mathbf{u}_{1..k}$ are considered as new basis, which are ordered by their dominance represented by their corresponding Eigen-values $\lambda_{1..k}$. When concatenating the Eigen-vectors $\mathbf{u}_{1..k}$ to a matrix

$$\mathbf{U} = \begin{bmatrix}\mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 & \dots & \mathbf{u}_k\end{bmatrix};\ \mathbf{U} \in \mathbb{R}^{m\times k}$$

because of their orthogonality (thus, $\mathbf{U}^{-1} = \mathbf{U}^T$), a data point x_i can be

$$\text{projected by: } x_i' = \mathbf{U}^T(x_i - \mu)$$

$$\text{back-projected by: } x_i = \mathbf{U}x_i' + \mu$$

to and from its Eigen-space representation x_i', whereas μ is the mean vector of all data points from which the Eigen-values and Eigen-vectors are calculated. Figure 2.3 graphically illustrates the compression principle for sample two-dimensional data.

In image processing, PCA, or 'Karhunen-Loeve transform', can be utilized to represent whole images as linear combination of a few pre-calculated Eigen-images. When each image is considered as one data point in a pixel data space, the data matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$ then holds all pixels of each training image as column vectors, whereas m is the number of pixels and n is the number of training images. Further, the PCA revealed basis vectors can be considered themselves as images of m pixels, which represent the most significant variations across the training images. Since the projection of a query image onto a new basis is done by calculating the dot-product of the two vectors, the resulting projections can be basically interpreted as cross-correlation scores between the query and the Eigen-images.

2.2.3 Classifier candidates

After collecting good discriminative features, in a classical object detection pipeline a classifier algorithm is utilized to assign a certain class label to the corresponding feature vector generated from an image patch. Although choosing proper feature descriptors is probably the most important step in designing an object detection system, the addressed problem also imposes high requirements on the classifier. Thus, we will outline the Support Vector Machine ('SVM'), AdaBoost and the Random Forest classifier and discuss their applicability for our object detection problem in special respect to imbalanced training data, multi-label classification and feature selection capabilities.

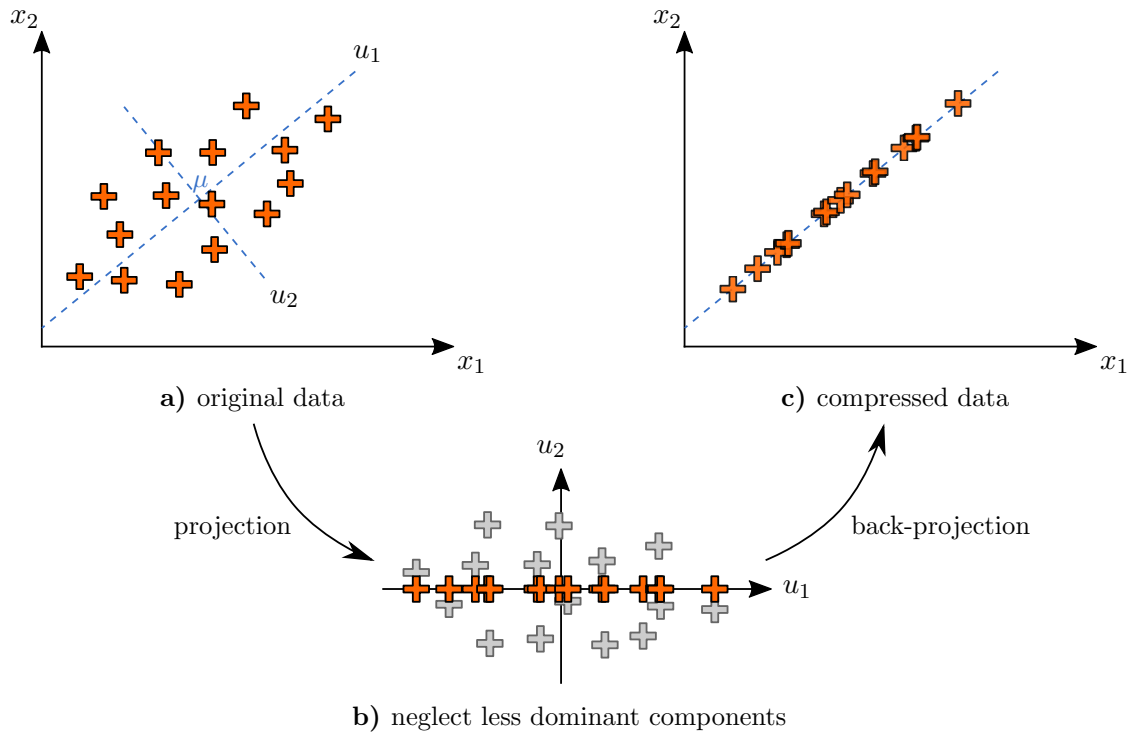


Figure 2.3: Principal components of two-dimensional sample data. Figure a) shows the scatter plot of the original data, whereat their principal component vectors are drawn as blue, dashed lines. Figure b) shows the data points projected to their Eigen-space. Here, the dimension corresponding to the lower Eigen-value is neglected to perform data compression. Figure c) shows the compressed data back-projected to the original data space. The most important information of the data, measured by its value on the axis of the highest variance, was preserved.

2.2.3.1 Support Vector Machine

The first mathematical formulation on Support Vector Machines came up by Boser et al. [4] in the early nineties, as a concept of optimal data separation. Although this approach was already known for a long time, it attracted the attention of Computer Vision about a decade later. Nowadays, in many scientific papers, Support Vector Machines are used as reliable standard classifier algorithm for any type of classification and regression task [13, 22]. Additionally, the clean mathematical formulation of SVMs enables scientists to derive more and more flexible and application-optimized versions.

The basic learning objective of a SVM is to find an optimal decision boundary for training samples of two classes that minimizes the classification error and maximizes the margin between data samples and the learned data separator. Figure 2.4 illustrates this objective of a linear SVM classifier in a two-dimensional feature space.

Considering the training samples as vectors $\mathbf{x}_i \in \mathbb{R}^D$ in the feature space along with their corresponding data label $y_i \in \{-1, +1\}$, then solving a SVM reveals, that the optimal

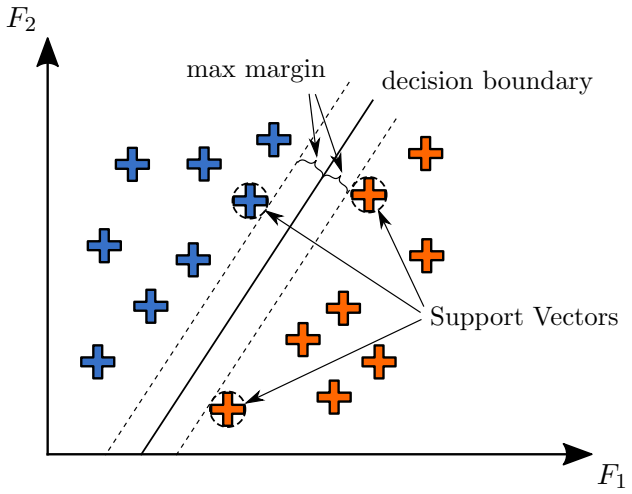


Figure 2.4: Learning objective of a SVM classifier. To discriminate provided training samples of different classes, a (hyper-)plane, respectively a line, is determined that optimally separates the data classes. A solution is considered as optimal when the data samples are separated properly and the margin between the nearest training samples of the confronted classes to the plane/line is maximized. As the found solution only depends on the data points closest to this linear separator, a SVM is completely described by these so called 'Support Vectors'. The classification of a query data point is performed by determining on which side of the separation plane (aka decision boundary) the point is located.

separation plane parameters ω^*, b^* only depend on a small set $\mathcal{S}$ of training samples where $y_s \cdot (\omega^* \cdot \mathbf{x}_s + b^*) = 1,\ s \in \mathcal{S}$. The training samples $\mathbf{x}_s, s \in \mathcal{S}$ are referred to as Support Vectors. This property results in a very efficient classification of a query sample $\mathbf{x}_q$, since only dot-products of and with those Support Vectors $\langle \mathbf{x}_q, \mathbf{x}_i \rangle, i \in \mathcal{S}$ and $\langle \mathbf{x}_i, \mathbf{x}_j \rangle, i, k \in \mathcal{S}$ need to be evaluated. Furthermore, this observation allows mapping the original feature space into another feature space via applying kernel functions with minimal computational costs. As also for the training procedure only dot-products of (projected) samples are required for the calculations, it is unnecessary to ever directly evaluate the mapping function. Thus, usually special mappings are chosen whose dot-products can be calculated very cheaply, therefore, polynomials or radial basis functions became popular kernel functions. These non-linear projections can be particularly useful when no or only an insufficient linearly separating hyperplane can be found in the original feature space.
For practical classification tasks, the so called C-SVM [12] is probably the most common variant of the Support Vector Machine. It extends the original hard-margin classifier by a soft margin classifier that accepts, but penalizes outliers in the calculation of the separation plane. This modification usually leads to better results on the test data by lowering overfitting effects, but often requires extensive searching for the best suited regularization parameter C. In order to rate certain parameters for a specific problem during training, k-fold cross validation is a recommended training procedure.

Imbalanced Training Data Since the soft margin based models minimize a cost function that is basically a trade-off between penalizing single instances and maximizing the margin for all other samples, for extremely imbalanced training data most likely the majority class is undesirably preferred in a reasonable penalty parameter setting. As proposed in [35], a possible solution is to use different penalty parameters for the classes, which significantly increases the complexity of searching their optimal values.

Multi-Class classification Due to the binary nature of Support Vector Machines, in the case of multi-class problems additional strategies must be applied. [29] shows that the one-vs-one approach as described in [32] is one of the best methods to implement multi-label classification for Support Vector Machines. Thus, especially for multi-class problems that have to be split up in multiple binary problems, proper training of a C-SVM classifier can become highly intensive very quickly.

Feature selection A challenging configuration for SVMs is when the number of samples is much lower compared to the number of feature dimensions, especially if the data contains a high ratio of weakly informative features. Instead of cherry-picking the most useful features, a SVM always tries to optimize its decision boundary over all feature dimensions simultaneously which may fail in achieving a sufficient level of generalization due to the lack of training samples. One way to provide a quality measure as a basis to choose a certain feature subset is applying cross validation also in the feature space, however, this would additionally increase the computational costs and amount of required samples for training the classifier. Therefore, we do not consider this approach as an practicable option for our task. In such a case that no additional feature-selection algorithm can be applied, for extremely imbalanced classification problems formulated in a high dimensional and relatively noisy feature space [43] suggests to use one-class SVMs [45] separately trained on the object classes. However, since we suppose that individual one-class SVMs trained on very similar looking objects will necessarily share many joint features and thus imply a higher risk to confuse object classes than discriminative trained models, we probably do not overcome this potential problem of SVMs for our application.

2.2.3.2 AdaBoost

Adaptive Boosting (short 'AdaBoost') is a powerful technique to form a strong classifier from several individual weak classifiers in an iterative fashion [23]. As weak classifier, basically any data separation heuristic can be chosen that supports weighting of its training samples. Due to its greedy iterative training procedure the resulting strong classifier converges when each single weak classifier achieves an prediction accuracy of only $> 50\%$. Considering a binary classification task, this means slightly better than random guessing. Furthermore, it can be proven that with each iteration the error on the training data decreases exponentially fast, while the resulting decision boundary is automatically refined

at each iteration step. However, this only holds when it can be ensured that each weak classifier is independent of the others, which is in practice difficult to achieve since usually the training data provided for each iteration rely on the same database. As additionally the sample weights of each iteration depend on previous calculations, an other practically related drawback of Boosting in general is the lack of possibilities to exploit the full speedup potential of modern hardware for the training procedure. Figure 1 outlines the basic AdaBoost algorithm, whereas Figure 2.6 illustrates the resulting strong classifier of a training example in a two-dimensional feature space.

Imbalanced Training Data In scenarios of imbalanced amounts of training instances per class, just like most of the standard classifiers AdaBoost tend to develop weak classification performance on minority classes. Considering the basic algorithm, there are only a few possibilities to tackle this problem. First, someone could adjust the prior probability of the training data by setting higher initial weights on samples of minority classes. Nevertheless, after a few iterations, the sample weights will converge to similar values compared to using uniform starting weights, thus the corrective effect of non-uniform weight initialization fades with every iteration. The second possibility on handling class imbalance is to use cost-sensitive boosting [21, 37, 48]. However, cost parameter selection is very crucial and these approaches show very diverging performance results on varying datasets. Additionally, the absolute performance gain compared to non-cost-sensitive boosting methods applied on the same data is also very dependent on the dataset and sometimes inconsiderably small even when the best cost parameters were found.

Multi-Class classification For multi-label classification problems, the original AdaBoost framework is much less powerful. First, the weak classifiers have to be capable of processing multi-labeled data, and second, their classification performance also has to reach over 50% prediction accuracy regardless of the amount of classes, which is much more challenging to achieve compared to a binary classification scenario. An extension of the basic boosting algorithm named 'AdaBoost.MH' [44] introduced a much better support for multi-class problems by exchanging each weak classifier with an ensemble of binary classifiers trained on one-vs-all representations of the original problem. The predicted class is then jointly determined by the majority vote.

Another approach to improve multi-class support for the basic AdaBoost framework is the 'SAMME' algorithm [55] which is based on a modified weight calculation of the weak hypothesis, such that the algorithm still converges when the weak classifier performs just slightly better than random guessing also in multi-class scenarios.

However, both SAMME and AdaBoost.MH are prone to reject potentially strong features that could separate similar classes jointly from others very effectively, since only the one-vs-all performance of simple classifiers are taken into account. Especially for applica-

Algorithm 1: Basic AdaBoost Algorithm

Data: $(x_1, y_1), \ldots (x_N, y_N) \;:\; x_i \in \mathbb{R}^D, y_i \in \{-1, +1\}$
initialize sample weight distribution $D_1(i) = \frac{1}{N} \quad \forall i = 1..N$
for $t = 1 \ldots T$ **do**
train/choose optimal weak classifier $h_t(x) \mapsto \{-1, +1\}$ using $D_t(i)$
calculate error on the training set

$$\varepsilon_t = \sum_{i=1}^{N} D_t(i) \left[h_t(x_i) \neq y_i \right]$$

assign weak classifier weight

$$\alpha_t = \frac{1}{2} \, ln \left(\frac{1 - \varepsilon_t}{\varepsilon_t} \right)$$

update sample weights

$$\begin{aligned} D_{t+1}(i) &= \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t}, & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t}, & \text{if } h_t(x_i) \neq y_i \end{cases} \\ &= \frac{D_t(i)}{Z_t} \cdot e^{-\alpha_t h_t(x_i) y_i} \end{aligned}$$

where Z_t is a normalization constant s.t. D_{t+1} yield a distribution
end
Result: Final strong classifier

$$H(x) = sign \left(\sum_{t=1}^{T} \alpha_t h_t(x) \right)$$

Figure 2.5: The basic AdaBoost Algorithm for binary classification problems. For initialization, to each training sample (x_i, y_i) an uniform starting weight $D_1(i)$ is assigned and a first weak classifier $h_1(x)$ is trained with respect to perform best on the training data. According to the prediction of this weak classifier, the weight of each training sample is increased if classified incorrectly, or decreased if the prediction is correct. After renormalization of the training sample weights, the next optimal weak classifier is chosen that minimizes the error on the re-weighted data. The final strong classifier is then formed by a weighted combination of the weak classifiers $h_t(x)$, whose weights α_t are determined by their evaluated classification performance.

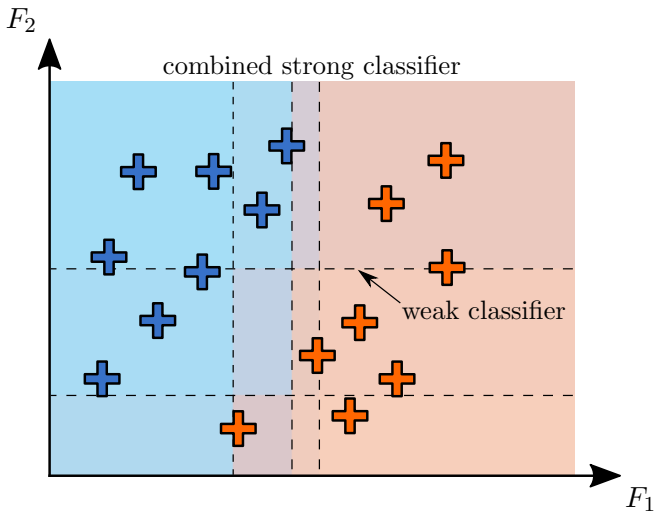


Figure 2.6: The AdaBoost algorithm demonstrated in a two-dimensional feature space. Separate single weak classifiers (here simple decision stumps represented by dashed lines) are combined as weighted linear combination yielding a strong classifier. This simulation shows the result after performing 5 iterations.

tion scenarios where the object classes share most of their visual cues we consider this a suboptimal strategy. Exactly this observation was the initial motivation for another variant of AdaBoost called 'JointBoost' [51, 52]. JointBoost aims at finding the most useful feature that can also be descriptive for more than one class in order to best separate the weighted training samples at each boosting iteration. This is achieved by investigating all possible groupings of the classes that can form a binary problem, not just one-vs-all configurations. On its downside, JointBoost requires a much more extensive training compared to other boosting variants and we have also not found any cost-sensitive modifications in literature to improve handling of class imbalance.

Feature selection Due to its greedy iterative fashion, AdaBoost and all derived algorithms provide effective intrinsic feature selection capabilities. Out of a large pool of weak classifier candidates each representing to a selectable feature, the algorithm will choose the one that best improves the performance on the weighted training samples at each iteration. The sample weights themselves model how well they are already represented by previously selected features, and therefore, redundant features are omitted. These properties were often proven and exploited by many research papers over the last decades which successfully applied Boosting to select the most useful features from a large feature pool [14, 31, 53]. On its downside, the greedy fashion of selecting weak classifiers makes AdaBoost in general error-prone to outliers and overlaps in the training data, a scenario that cannot be excluded for our application considering users adding new samples of similar looking target classes for their purpose and re-training the framework.

2.2.3.3 Random Forests

As final classifier algorithm candidate we discuss random forests [6], which show a remarkable record of successful implementations in practical and even commercial applications [46]. Similar to AdaBoost, its basic principle is to form a strong classifier from an ensemble of weak classifiers, more precisely randomized binary decision trees. However, in contrast to AdaBoost where training follows an iterative, deterministic procedure, random forests gain their strength from compiling randomized, and therefore, non-deterministically created hypothesis. Furthermore, while AdaBoost rates the individual prediction errors of the weak classifiers and thus their influence on the final result, the final decision of a random forest classifier is simply determined by the majority vote of all decision trees, regardless of their single classification performance.

When training a sufficient amount of trees, due to this basic concept of randomized and equally treated predictors the combined ensemble show good generalization performance while the recursive structure of the trees necessarily minimizes the training error. The underlaying strategy of a single binary decision tree that separates given samples by consecutively applying randomly chosen split functions is demonstrated in Figure 2.7.

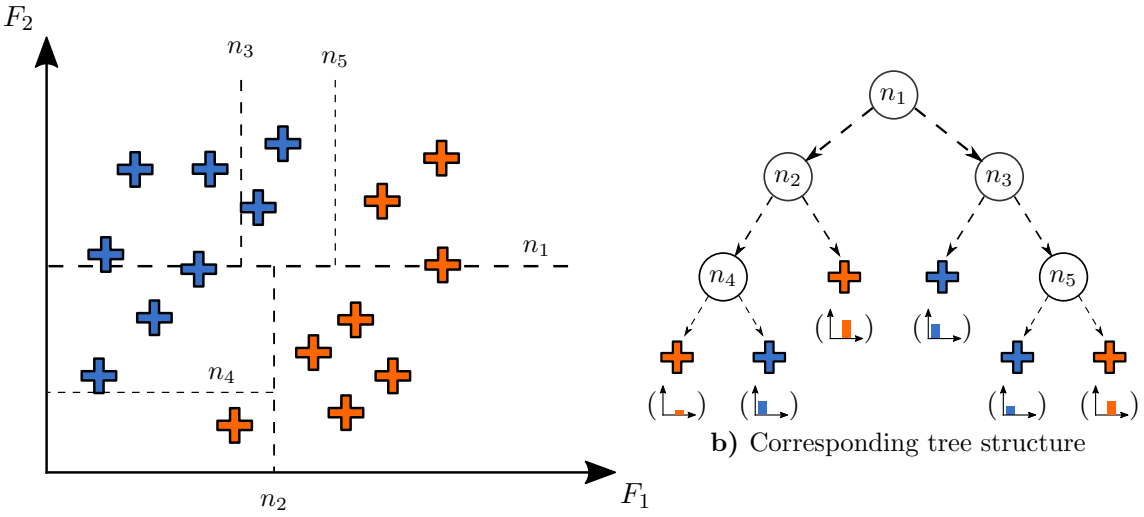
In this illustration, for a training set $\mathcal{S} = \{X_1, X_2, .. X_N\}$ whose samples $X_i = \{\mathbf{x}_i, y_i\}$ consist of a feature vector $\mathbf{x}_i \in \mathbb{R}^D$ and a class label $y_i \in \{1, 2, .. C\}$, the binary split function $f_n(X_i) \mapsto \{L, R\}$ that divides $\mathcal{S}$ into two subsets $\mathcal{S}_L$ and $\mathcal{S}_R$ is defined as

$$f_n(X_i) = \begin{cases} L, & \text{if } \mathbf{x}_i(d) < x^{th} \\ R, & \text{otherwise} \end{cases}$$

where x^{th} is a fixed threshold applied on the d^{th} dimension of $\mathbf{x}_i$. Therefore, $f_n(X_i)$ is fully defined by the tuple $[x^{th}, d]$, $d \in \{1 .. D\}$. As this is probably the most common definition of a split function for binary decision trees, also other heuristics of $f_n(X_i) \mapsto \{L, R\}$ are conceivable.

During the training procedure of a single decision tree for a random forest classifier, the randomness is induced mainly by the choice of the split function. Typically for each newly created node the best split function is selected from a moderately large pool of randomly generated candidates according to a certain scoring measure. In [6], Breiman suggests to select candidates for a split node only from a small subset of the feature space to achieve good generalization, hence d should be chosen from a subset $\mathcal{D}' \in \{1 .. D\}$ where $|\mathcal{D}'| \ll D$. However, also more extreme randomization approaches have been studied [24].

One common quality measure of candidate split functions is the information gain, which describes the relative change in entropy of a data set when it is divided into subsets. The information gain for a binary split function $f_s(X_i) \mapsto \{L, R\}$ that divides $\mathcal{S}$ into two



a) Binary, consecutive decisions in feature space

b) Corresponding tree structure

Figure 2.7: Illustration of the basic principle of a binary decision tree in a two-dimensional feature space. At each 'split node' of the tree, a selected threshold and feature dimension separates the feature space into two parts. The tree structure is build up by recursively performing such binary test functions continuously dividing the feature space into smaller sections and passing the corresponding training samples to the child nodes. In this visualization, the left paths follow the condition of the value of the chosen feature dimension to be lower then the corresponding threshold, otherwise, the right paths are selected. The creation of new child nodes during training is typically aborted at a maximum depth level, when a minimum number of training samples within a child node is reached, or when the training samples show a unary label distribution. Under this conditions a 'leaf node' is formed, that can store a full label histogram of the remaining training samples or simply the label of their majority, finally yielding a fully partitioned feature space. A query sample can then be classified according to the leaf node that is reached after carrying out the subsequent binary tests down the tree.

subsets $\mathcal{S}_L$ and $\mathcal{S}_R$ is defined as

$$IG(f_s(\mathcal{S})) = H(\mathcal{S}) - \sum_{k \in \{L,R\}} \frac{|\mathcal{S}_k|}{|\mathcal{S}|} \cdot H(\mathcal{S}_k) \tag{2.1}$$

where $H(\mathcal{S})$ denotes the entropy of a sample set $\mathcal{S}$ in respect to their labels y_i, that in turn is calculated as

$$H(\mathcal{S}) = -\sum_{c=1}^{C} \frac{|\mathcal{S}_c|}{|\mathcal{S}|} \cdot log \frac{|\mathcal{S}_c|}{|\mathcal{S}|} \tag{2.2}$$

whereat $|\mathcal{S}_c|$ names the cardinality of the subset $\mathcal{S}_c \subseteq \mathcal{S}$ containing all samples in $\mathcal{S}$ with label c.

Focusing on the ensemble of trees, additional randomization techniques can be applied on the selection of samples from which a whole tree is trained. A well-established approach is to generate different training sets for each tree learner by *random selection*

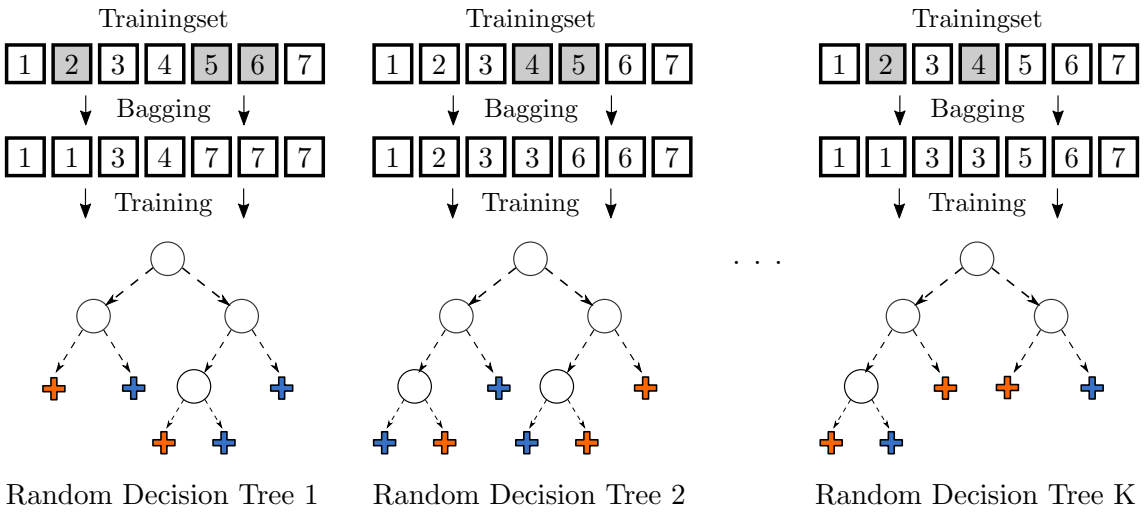


Figure 2.8: Bagging applied on the training procedure of a random forest classifier. For each individual tree the training set is re-sampled via random selection with replacement. Those samples that are left out for a particular decision tree are marked as 'out-of-bag' (gray background color). After training, the generalization error of the final random forest can be estimated by classifying each out-of-bag sample via an ensemble of trees that share this particular sample as out-of-bag data.

with replacement. This method is commonly known as bootstrap aggregating, or short 'bagging' [5]. The hereby generated varying training sets used for each tree produce more decoupled weak hypothesis, and thus, further improve classification performance of the whole ensemble while simultaneously reducing the risk of over-fitting on the training data. Additionally, bagging enables an elegant method for estimating the generalization error of the resulting random forest without the need for an extra validation set. When considering N training samples, the probability of selecting a certain sample when randomly choosing exactly N samples with replacement is $1 - \frac{1}{e} \approxeq 63.21\%$. Thus, about 36.79% of the samples remain unseen for a particular tree, which form the so called 'out-of-bag' data. Each out-of-bag sample can then be evaluated by an ensemble consisting of those trees which share this particular sample as out-of-bag data. Testing all out-of-bag samples by this method gives a good estimation of the expected classification performance of the resulting random forest classifier on unseen samples, without the need for an extra validation dataset. Figure 2.8 graphically illustrates the training procedure of a random forest using bagging.

Additionally, random forests also have a huge advantage in respect to computation time at the training stage. Compared to other classifiers, single decision trees can be trained completely in parallel without the need of exchanging information with other trees.

Imbalanced Training Data The standard implementation of random forests is no exception in terms of favoring the majority classes in the presence of imbalanced data.

However, the algorithm intrinsically provides many possibilities for modifications in order to effectively handle imbalanced data. One approach is to assign class weights according to their occurrence rate in the training data, such that minority samples have a larger impact on the score of a split function and on the vote of a leaf node. This correctively interferes the voting result of single trees, but also influences the creation of the tree structure in favor of imbalanced data. In [10] this approach is referred to as 'Weighted Random Forest'. A different method also described in [10] is to generate balanced data for each single tree by stratified down-sampling and evaluating only a few randomly chosen variables for a split without pruning. As this approach seems to be less problematic than down-sampling the training set for the whole forest in advance, we still consider this method prone to reject too many samples for reaching a reasonable tree depth that can model the complexity of our classification task. Anyway, this approach shows that more randomness in split node selection can improve the stability of the final classifier in the presence of challenging training data.

Multi-Class classification Since the information gain is well-defined for multi class data and the leaf node votes are not restricted to binary values only, the random forest classifier does not require any structural modifications in order to learn and classify multi-labeled data. Furthermore, as the trees are basically unrestricted in their depth, the random forests can process data of an arbitrarily high amount of class labels, without the need of applying data simplification or grouping heuristics in advance.

Feature Selection The hierarchical structure of the single trees in combination with selecting the split function according to their information gain score ensures good feature selection properties of the overall classifier. Whereas for the first nodes the most discriminative features are chosen for the splitting the data, consecutive nodes in deeper tree levels will then automatically neglect redundant split functions. Furthermore, a trained random forest can be directly evaluated to estimate the importance of feature dimensions by accumulating the information gain score over all nodes which use a certain feature variable in its split function and weighting the single terms by the relative amount of training samples that were considered in the particular node.

2.3 Summary

In this chapter, we first investigated existing approaches that explicitly aim at detecting electrical installation devices in indoor environments. However, as these approaches are intended for being deployed on actively moving robot systems and thus basically need to fulfill different requirements, they are not applicable to our detection problem.

We then described other, more general computer vision concepts that successfully tackled detection of untextured objects. Basically, these approaches can be divided into two groups. The first group of approaches relies on the detection of a closed contour, whereas

methods of the second group basically perform template matching using different representations of the gradient information. Considering our problem description, we can identify major drawbacks of both groups. Where on the one hand extracting closed contours can not be reliably achieved due to the difficult input data, on the other hand approaches based on template matching mostly support only distinct appearances of objects.

In order to exploit the given data and constraints of our detection task the most beneficial way, we then discussed feature descriptors and classification algorithms separately. With the histogram of orientated gradients (HoG) descriptor, expressing different filter responses by spatially distributed statistical measures and PCA compression applied on images we described various powerful concepts on describing whole image patches that show untextured objects.

Furthermore, we discussed three well-established classifier concepts for practical applications and analyzed their characteristics in the context of our most critical application requirements. This revealed a clear preference of the random forest classifier due to its intrinsic feature selection capabilities, handling of multi-labeled classes and the straightforward modifications to enable support for highly imbalanced training data.

CHAPTER 3

IMPLEMENTATION

Contents

Summarizing, we face a multi-class detection problem of small and untextured objects on rectified images, which are generated from a rather low-quality panoramic image. Approaches we found in literature which address detection of untextured objects either demand a reliable extraction of the object boundaries, or only support distinct instances while requiring a high amount of training data to cover each possible view angle. However, there exist various separate image descriptor and classifier approaches which allow certain modifications to better meet the special requirements of our application.

In this chapter, we describe the implementation details of our final approach based on such a decision pipeline formed by a separate feature extraction and a classifier stage. Utilizing this classification framework, the detection of electrical devices on the rectified input images is performed by applying a sliding window approach. In the first part of this chapter we address the implementation details of the sliding window, since it also defines the input for the decision algorithm. Afterwards, the applied image feature extractors and their parametrization are described in more detail. Beside the HoG feature descriptor, Haar-like wavelets applied on a set of image filter responses and template matching against auto-generated templates via PCA, we will also introduce a novel feature descriptor named 'Gradient Orientation Features', that is based on concepts across several outlined approaches and is especially designed for our problem. All feature values are concatenated to form a single vector that is passed to a random forest classifier, to which the next section of this chapter is dedicated. Beside discussing the basic algorithm, we will also introduce

some additional modifications to better fit the given application constraints. Finally, this chapter is concluded by a summary of the implemented approach.

3.1 Sliding Window

Given a wall texture image of arbitrary size, the detection of multiple instances of electrical devices is done by a basic sliding window approach. Therefore, around each pixel of the input image an image patch of a fixed size is extracted and passed to the classification pipeline. By collecting the multi-labeled classification result over each center pixel, a response map is created for each object class. In order to gather the final detection results, the response maps are post-processed by a non-maxima suppression stage. Note that in this thesis, for evaluation purposes we treat each object class separately, however, for the finally deployed algorithm only the result assigned to the highest response is reported if at some locations detection results of multiple classes are overlapping.

Due to the fixed spatial resolution of $1mm^2$ per pixel and the fact that instances of the target object classes do not differ much in size, we can omit to search in different scale levels. As a common housing frame for electrical devices is typically $90mm \times 90mm$ and we consider the immediate surrounding as additional valuable information, we decided to use a fixed window size of $128px \times 128px$. This certain size ensures a prominent representation of a single target object in the image patch while capturing a sufficient amount of the surrounding and, furthermore, allows directly applying a wide range of HoG descriptor configurations without the need to rescale the sliding window content.

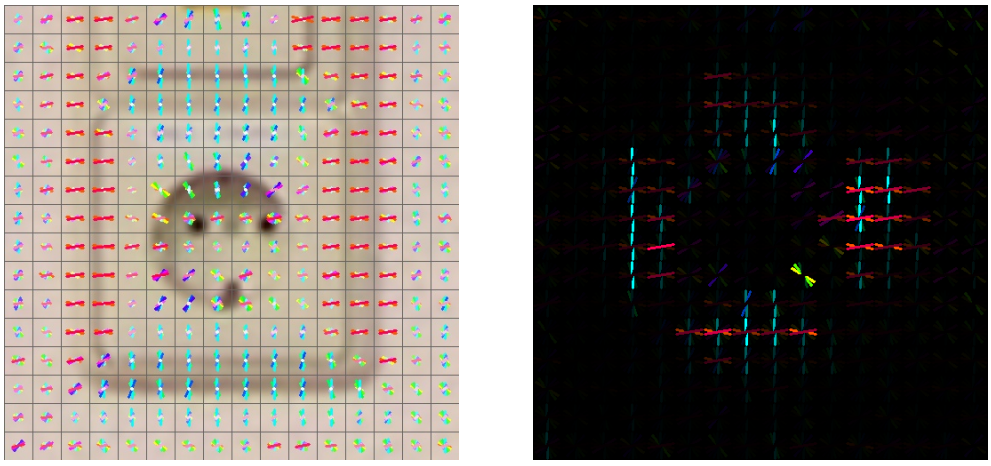
3.2 Feature Extractors

The basis of our derived detection algorithm is a pool of well-selected descriptors, each focusing on an important visual property of our target objects. The formation of a feature pool aims at combining the strengths and compensating the weaknesses of the individual descriptors. Since the silhouette and functional indentations of our target objects are the most valuable visual information, the major part of the added descriptors focuses on modeling image gradients and their constellation. However, we also add features coding other object properties like the color distribution and the strictly horizontal/vertical mounting practice.

3.2.1 Histograms of oriented gradients

Because of its strong properties in modeling the gradients of an image patch, the HoG descriptor is our first choice for the feature pool. However, applying this descriptor is not straight forward, as it provides a relative high amount of parameters that need to be set

carefully. Since objects and background can have different illumination transitions (dark-light, light-dark), the first decision is to use the *undirected* version of the HoG descriptor. Furthermore, we use a *rectangular* shape of the histogram cells. In order to determine the remaining parameters for our use case, we test three configuration settings, representing a fine-structured cell grid, a coarse-structured cell grid and a coarse-structured cell grid with a high overlap of the normalization blocks. The results of the three tested configurations are discussed in more detail in Section 4.3.1. A visualization of the HoG descriptor applied on a target image using the first setting is shown in Figure 3.1.



a) **b)**

Figure 3.1: Figure a) shows an overlay of the corresponding HoG descriptor values on a sample object, where the gradient strength determines the length of the direction lines. For better visibility, directions and strengths are color-coded by applying the Hue-Saturation plane mapped on 180° of the HSV color space. Figure b) illustrates the average importance of the HoG gradients, coded by the value component of the HSV space. The feature variable importance values are determined by a Random Forest classifier trained on multi-class data consisting of 3 classes (background, power sockets and light switches). It can be observed, that the vertical and horizontal edges forming the object silhouettes are the most important features. Additionally, in order to distinguish between power outlets and light switches, also inner, rather diagonally oriented gradients are considered as essential features.

3.2.2 Novel Gradient Orientation Features

When investigating successful feature descriptors for untextured objects of the recent past, we observed that modeling constellations of image gradients is predominantly achieved by heavily quantizing in magnitude, orientation and position [13, 25, 26]. In general, the main advantages these approaches see in quantizing the information is to gain more invariance to various circumstances, and reducing the dimensionality of the resulting feature vector. We agree that suppressing variations in magnitudes can introduce a necessary level of illumination invariance, however, we see in quantizing positions and orientations the

potential to lose and distort valuable information. Especially when targeting rigid objects on rectified images which consist only of a few distinctive gradients, spatial and directional quantization can result in significantly diverging feature representations of similar objects when they show their differences near the quantization levels. In order to provide less distorted information about the underlaying image patch such that the classifier can decide more freely on discriminating thresholds, we propose a feature descriptor that models position and orientation of image gradients with respect to introducing as less quantization mechanisms as possible. This is achieved by representing the gradient orientations via cosine similarity measures evaluated over randomly distributed subregions, which are not aligned to any fixed pattern.

Similar to [25], we utilize a gradient response map as basis for our descriptor, where the gradients are determined in a max-pooling fashion over the color channels. Thus, we calculate gradient images $\nabla I_c(i,j)$ separately for each color channel $I_c(i,j), c \in \{R,G,B\}$ of the input image $I(i,j)$ by applying a Sobel operator in both x- and y-direction:

$$\nabla I_c(i,j) = \begin{pmatrix} G_{x,c}(i,j) \\ G_{y,c}(i,j) \end{pmatrix}$$

where

$$G_{x,c} = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} * I_c \quad \text{and} \quad G_{y,c} = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} * I_c$$

From this filter responses we can construct a gradient direction map $\mathcal{M}(i,j)$ consisting of unit vectors pointing at the corresponding direction by selecting for each pixel the color channel showing the highest gradient magnitude, while suppressing gradients whose magnitude is below a certain threshold th_{mag}:

$$\hat{c}(i,j) = \text{argmax}_{c \in \{R,G,B\}} \|\nabla I_c(i,j)\|_2 \tag{3.1}$$

$$\mathcal{M}(i,j) = \begin{cases} 1 \angle \theta(\nabla I_{\hat{c}(i,j)}(i,j)), & \text{if } \|\nabla I_{\hat{c}(i,j)}(i,j)\|_2 \geqslant th_{mag} \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

where $\theta(\nabla I(i,j))$ retrieves the direction of the gradient on position (i,j).

Using $\mathcal{M}(i,j)$ we generate a cosine-similarity representation by projecting the unit vectors onto a set of reference vectors $\{\mathbf{R}_1 \dots \mathbf{R}_K\}$. In order to cover the full range of possible gradient directions, one would usually utilize 2 orthogonal reference vectors. As specifically for our application we prefer a feature representation that is invariant to the direction of the gradient similar to the undirected version of the HoG descriptor, we demand the cosine-similarity values to repeat each 180°. One naive solution to achieve this would be to map directions between $[\pi \dots 2\pi)$ onto the range $[0 \dots \pi)$ and use the same

2 orthogonal reference vectors. However, this would cause an undesired discontinuity of the descriptor values near orientations of 0, respectively π. To avoid this behavior, we propose to take only the absolute values of the projections on the reference vectors into account. Whereas this strategy applied on 2 reference vectors would yield ambiguous representations each 90°, we instead use 4 reference vectors $\mathbf{R}_1 \dots \mathbf{R}_4$ which differ by an angle of $\frac{\pi}{4}$ (45°). The projections of $\mathcal{M}(i,j)$ form a set of 4 scalar valued image channels $S_k(i,j),\ k = 1..4$:

$$S_k(i,j) \quad = \quad |\mathcal{M}(i,j) \cdot \mathbf{R}_k| \,, \qquad k = 1..4 \tag{3.3}$$

where $\mathbf{R}_1 = 1 \angle 0$, $\mathbf{R}_2 = 1 \angle \frac{\pi}{4}$, $\mathbf{R}_3 = 1 \angle \frac{\pi}{2}$ and $\mathbf{R}_4 = 1 \angle \frac{3\pi}{4}$.

In order to express these gradient projections by a real-valued feature vector which can be further processed by a classifier, we utilize randomly sized and distributed Haar-like features H_i to calculate subregional mean values separately over the 4 orientation channels. These 4 values are then normalized s.t. $\sum_k avg(S_k(H_i)) = 1$. The resulting feature vector of an input image is then composed by applying n_{pairs} pairs of Haar-like features and concatenating their channel-wise differences of the 4 normalized mean orientation values. Because we use Haar-like features, our descriptor can be efficiently calculated via integral images. Figure 3.2 depicts all essential steps of our feature descriptor approach.

Since one descriptor dimension codes the visual difference of 2 subregions, we can emphasize the focus on the symmetry of an object by forcing the Haar-like feature pairs to describe spatial counterparts of an image patch. Given one rectangle, we consider three possible strategies to choose the second rectangle yielding a 'symmetric' constellation: vertical, horizontal and centric. Including pairs where both rectangles are completely chosen at random, this results in a total number of 4 pair selection strategies, which are depicted in Figure 3.3.

In order to configure our feature descriptor to work best for a certain task, we define a set of parameters. The first parameter defines the number of Haar-like feature pairs n_{pairs}, which directly impacts the resulting length of the feature descriptor as $4 \times n_{pairs}$. Furthermore, we add the threshold th_{mag} to the parameterization, which is applied on the gradient magnitude to suppress image noise. To adjust the granularity of the image description, we introduce sz_{min} and sz_{max} as a lower, respectively, upper limit for the width and the height of the randomly generated Haar-like features, given in percent of the input image size. The parameter set is completed by the symmetry ratio r_{sym} which defines the ratio between symmetric and completely random feature pairs, whereat the probability of choosing a certain type of the three symmetric constellations is always kept equal. All described parameters are summarized in Table 3.1.

3.2.3 Feature channels from image filter responses

With the HoG and the novel gradient orientation features, we already use two very strong approaches for describing the spatial constellation of image gradients.

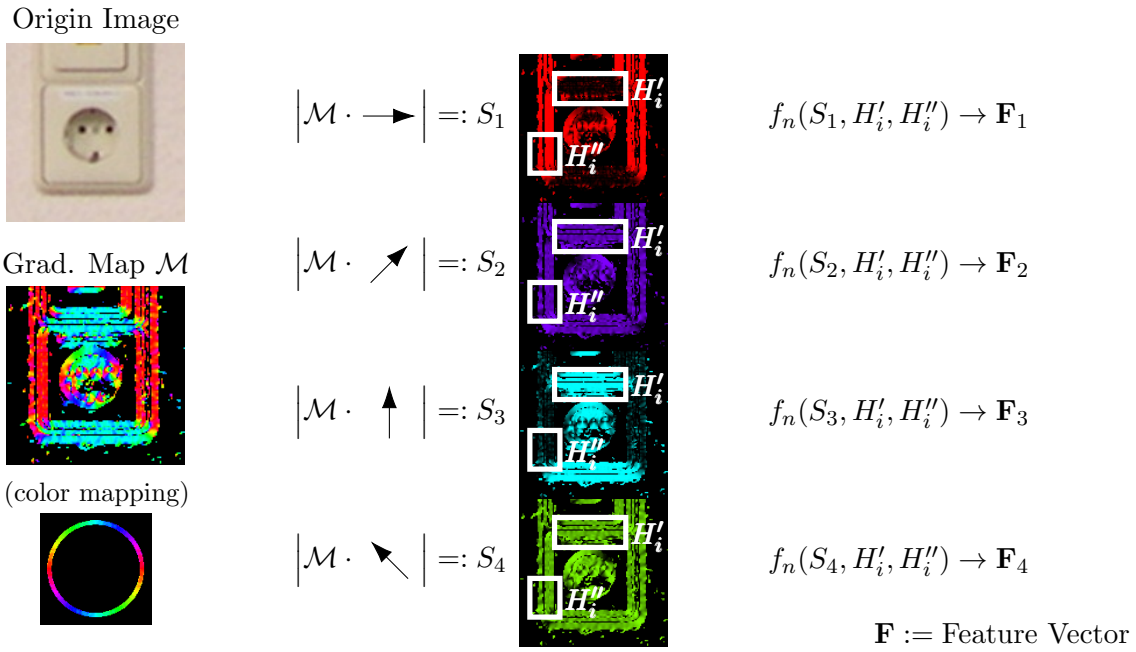
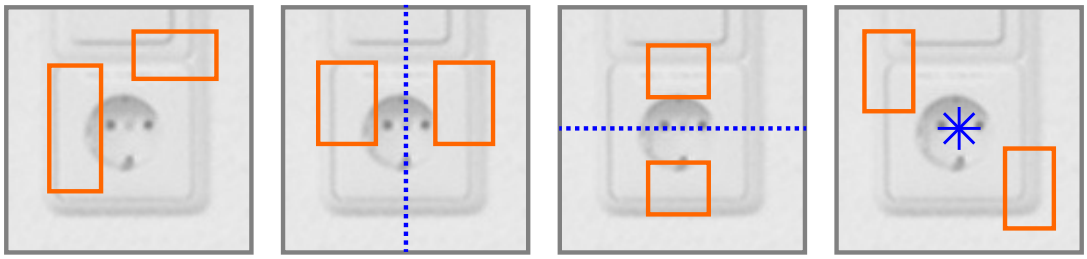


Figure 3.2: Illustration of our gradient orientation features approach. As a first step, a gradient map $\mathcal{M}$ consisting of unit vectors is calculated from the input image. Note that at this stage $\mathcal{M}$ contains directed gradients, but for a better visualization of the undirected gradient modeling property of our descriptor a repetitive color mapping is applied. By pixel-wise projecting the vectors of $\mathcal{M}$ onto a set of 4 reference vectors, we obtain 4 scalar-valued image channels showing the corresponding cosine-similarity values. To achieve a feature representation that is invariant to the direction of the gradient, we only consider the absolute values of the projections. The function $f_n(S_k, H', H'')$ denotes the evaluation of the Haar-like feature pairs H', H'' forming the final descriptor values, which is defined as $f_n(S_k, H', H'') = \frac{avg(S_k(H'))}{\sum_S avg(S(H'))} - \frac{avg(S_k(H''))}{\sum_S avg(S(H''))}$

Parameter	Value Range	Description
n_{pairs}	$\mathbb{N}^+$	Number of Haar feature pairs
th_{mag}	$\mathbb{R}^+$	Magnitude threshold for the gradient calculation
sz_{min}	$(0, 100]\%$	Minimum size of a Haar-feature
sz_{max}	$(0, 100]\%$	Maximum size of a Haar-feature
r_{sym}	$0 - 100\%$	Symmetry ratio

Table 3.1: Gradient orientation features parametrization.

Although this is the most valuable information of our target objects, however, there are also other characteristic and meaningful features that are not yet fully exploited by these descriptors. Therefore, we additionally utilize a set of specifically chosen image filters to generate feature channels and describe their responses by applying Haar-like features, similar to [14]. Considering our application, we aim at modeling the color distribution and the physical alignment of the objects in this descriptor.



a) random **b)** vert. symetric **c)** horiz. symetric **d)** centric symetric

Figure 3.3: Different feature pair selection strategies for our gradient orientation feature descriptor. While a) shows an example of 2 completely random sampled rectangles, c), b) and d) each represent a symmetric constellation. In c) and b) the two selected subregions are each others counterpart when mirrored vertically, respectively, horizontally, where in d) the origin rectangle was flipped in both axis, yielding a symmetry in respect to the center point of the image.

Color One characteristic property of our target objects that is only indirectly expressed by gradient-based descriptors is the color information. Since electrical installation devices are mostly single-colored and sometimes stand out against the color of the background, we consider the color distribution over an investigated sliding window as an additional valuable detection feature. To best differentiate colors by their numerical representation, this feature channel simply consists of the input image converted to the *Luv* color space.

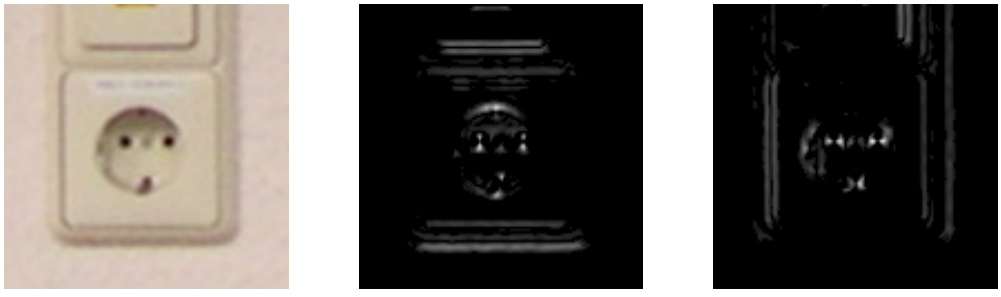
Alignment An other not explicitly handled but descriptive property of the target objects is their strictly horizontal/vertical mounting practice. Since this is a global property of an object and thus could be expressed by a single value describing the mounting angle, one could also incorporate this information when evaluating the posteriori probability of a detection result. However, this angle would require to be measured also somehow visually and would introduce additional uncertainties. Therefore, exploiting that electrical installation devices mainly consist of horizontal and vertical edges too, we propose to feed the information about the presence of strictly horizontal and vertical edges directly to the classifier along with the other visual features. For this feature channel we designed an image filter that solely responds to strictly horizontal and vertical gradients, simply consisting of separate, undirected Sobel filters while suppressing responses below a minimum value.

The result of the designed image filters forming the feature channels for this descriptor are visualized in Figure 3.4 on an example. In order to generate feature values from the image filter responses we basically use the same approach as in our gradient orientation feature descriptor: We apply randomly sized and distributed Haar-like feature pairs and

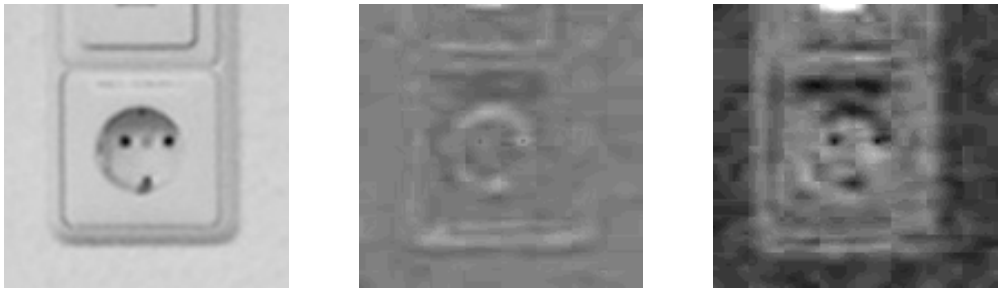
Parameter	Value Range	Description
n_{pairs}	$\mathbb{N}^+$	Number of Haar feature pairs
sz_{min}	$(0, 100]\%$	Minimum size of a Haar-feature
sz_{max}	$(0, 100]\%$	Maximum size of a Haar-feature

Table 3.2: Parametrization used for describing the introduced feature channels. The number of generated Haar-wavelet pairs n_{pairs} directly corresponds to the dimension of the resulting feature vector. Analogue to our gradient orientation features, sz_{min} and sz_{max} can be used to control the granularity of the image description.

obtain the final descriptor values from the pair-wise differences of the mean values of the rectangular subregions. The feature channel on which a certain Haar-wavelet pair is applied, is also chosen randomly from an uniform distribution. This yields a very simple parametrization of the resulting descriptor coding the color distribution and the alignment of the target objects, as listed in Table 3.2.



a) origin image **b)** horizontal gradients **c)** vertical gradients



d) L–channel **e)** u-channel **f)** v-channel

Figure 3.4: The used image filters applied on a sample input image (Figure a)). Figure b) and c) show the response of the horizontal and vertical gradient filter, respectively. The L, u and v channel of the input image is displayed in Figure d), e) and f). For better visibility, for e) and f) we magnified the intensity spectrum by about 6 times. While the u channel seems to share the same value for the power socket and the wall, we can observe a significant difference in the v channel, indicating the slightly diverging color of the wall and the actual object.

3.2.4 PCA for auto-generated templates

In general, the above discussed feature descriptors model the overall appearance of an image patch by concatenating measures of subregions, directly expressing the image content which results in a relatively high dimensional vector. A rather complementary approach that could further enrich this information by a more global representation of the sliding window patch, is matching the whole image patch against templates. Especially when detecting rigid, fixed-sized, untextured objects, where robustness against occlusion plays a minor role, we consider template matching responses as potentially useful features. Since in our application we cannot expect user-provided templates of any form, we propose to apply principal component analysis to obtain auto-generated image templates. When projecting a query image represented as vector to an Eigen-space spanned by pre-calculated Eigen-vectors, each dimension in the Eigen-representation is calculated via the dot-product with the corresponding basis- or Eigen-vector. Since an Eigen-vector can also be interpreted as image, casted in a sliding window approach, the pixel-wise obtained subspace representation yield multi-channel response maps of cross-correlation matching with the Eigen-images acting as templates. In order to generate specialized templates for each class, we calculate the Eigen-spaces of the different object categories separately. The coordinates of a query image transformed to the class-specific Eigen-spaces are then directly added to the feature vector that is passed to the classifier. Figure 3.5 shows the pre-trained Eigen-vectors represented as images for power sockets and light switches.

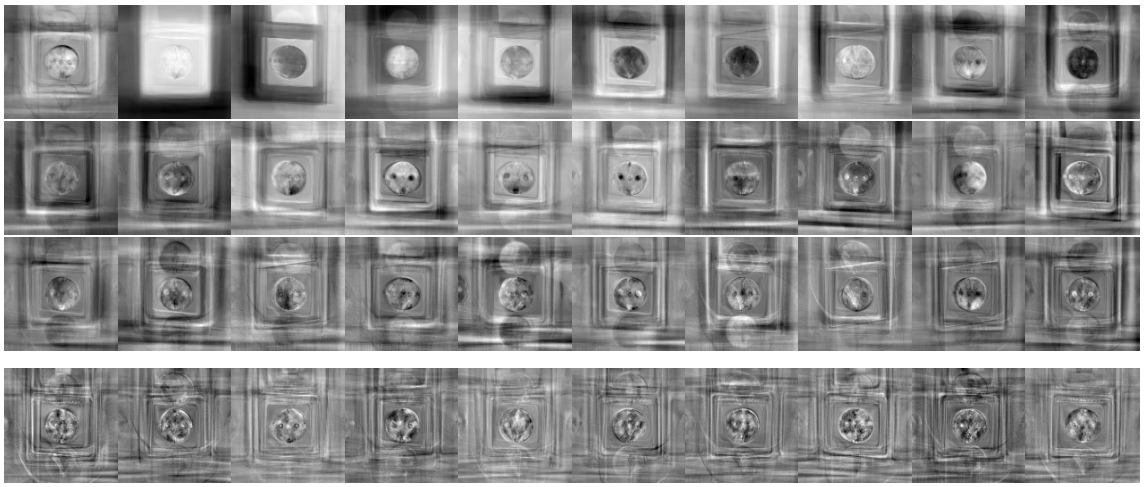
3.2.5 Final selection

Each of the described feature descriptors implements a diverse, powerful concept of representing the untextured target objects on 2D images. However, the final selection and parameterization of the algorithms which works best for our use case needs to be determined by conducted experiments, which are addressed in Chapter 4.

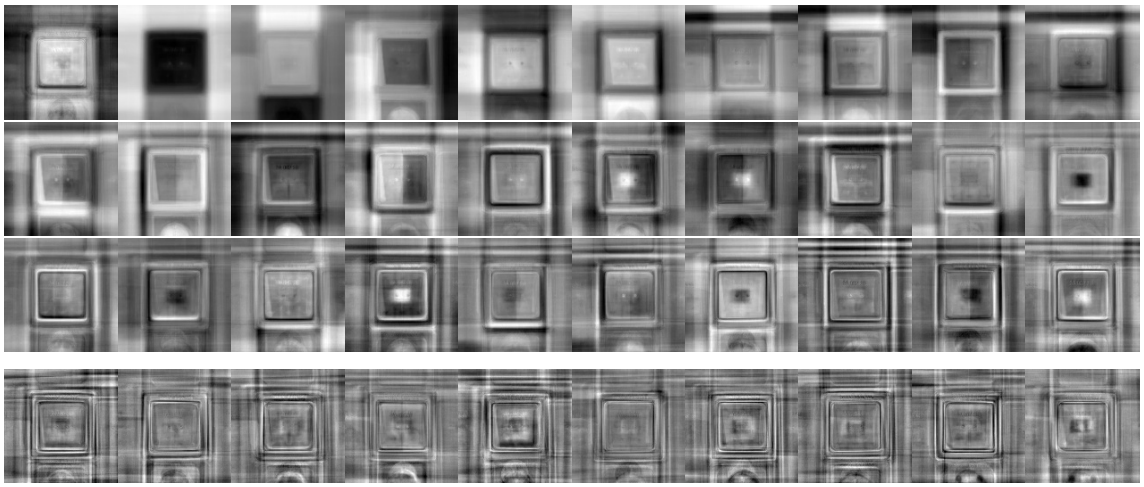
3.3 Classifier

As an undeniable drawback of forming a pool of various, mostly randomized features, the feature-space becomes relatively large while containing many redundant and noisy dimensions. In combination with the very limited and unbalanced amount of training examples, these circumstances state a rather challenging use case for a classifier. To tackle this problem, we utilize a random forest that is especially configured for this use case.

Weighted samples The first modification to handle the high class imbalance in the training data is similar to the idea of a ‘Weighted Random Forest’ in [10], where Chen et



a) Eigen-images of power sockets



b) Eigen-images of light switches

Figure 3.5: These images show the obtained Eigen-images for power sockets a) and light switches b) representing their major visual commonalities, ordered by the corresponding Eigen-value from top-left to bottom-right. The first 3 rows show the first 30 Eigen-templates, whereas each last row holds the 61^{st} to 70^{th} most dominant Eigen-representation of our training data. It can be observed, that the most significant characteristics of the training samples are the varying intensity values between the wall and the overall objects, followed by different intensities of more local parts. The obtained principal components of a higher order contain certain arrangements of the characteristic edges and gradients.

al. introduced class weights that influence the split criterion and the vote of a tree. In their approach, they base their split functions on the Gini impurity measure and propose to determine the weights by the out-of-bag estimate or tune them by hand. In our implementation, we use the information gain criterion based on the Shannon entropy and weight the samples by a class rarity score determined from the training data that is given

to the forest. From a set of samples $\mathcal{S}$ whereat to each instance a class $c \in \mathcal{C}$ is assigned, the weights w_c are calculated as

$$w_c = \frac{1}{Z} \cdot \frac{|\mathcal{S}|}{|\mathcal{S}_c|} \tag{3.4}$$

where $\mathcal{S}_c$ is the subset of $\mathcal{S}$ containing all samples with label c and $Z = \sum_{c \in \mathcal{C}} \frac{|\mathcal{S}|}{|\mathcal{S}_c|}$ is a normalization factor. Using these weights, we can reformulate the entropy $H(\mathcal{S}')$ of a subset $\mathcal{S}' \in \mathcal{S}$ as

$$p(\mathcal{S}', c) = \frac{1}{Z'} \cdot w_c \cdot \frac{|\mathcal{S}'_c|}{|\mathcal{S}'|} \tag{3.5}$$

$$H(\mathcal{S}') = -\sum_{c=1}^{C} p(\mathcal{S}', c) \cdot log\ p(\mathcal{S}', c) \tag{3.6}$$

where $p(\mathcal{S}', c)$ is the weighted occurrence probability of class c in $\mathcal{S}'$ and $Z' = \sum_{c \in \mathcal{C}} w_c \cdot \frac{|\mathcal{S}'_c|}{|\mathcal{S}'|}$ ensures normalization. This weighted formulation of the entropy is applied to determine the information gain when evaluating a split function during training of the forest (Equation 2.1).

At classification stage, the weights w_c also impact the class prediction of the single trees. Whereas usually a tree votes according to the class majority of the training samples $\mathcal{S}'_{\mathcal{N}}$ that reached the corresponding leaf node $\mathcal{N}$, we instead determine the prediction $\mathcal{P}_{\mathcal{N}}$ of a leaf node by evaluating the weighted amount of those samples via

$$\mathcal{P}_{\mathcal{N}} = \text{argmax}_{c \in \mathcal{C}}\ w_c \cdot |\mathcal{S}'_{\mathcal{N},c}|. \tag{3.7}$$

The final prediction outcome of the random forest classifier is then aggregated by collecting the predictions over all trees in a normalized histogram to obtain the probabilities for each class.

In order to additionally enhance generalization, we apply bagging of the initial training data $\mathcal{S}$ to generate diverse training sets $\mathcal{S}'_k$ for each tree k, with $|\mathcal{S}| = |\mathcal{S}'_k|$. Note that for calculating the class weights w_c in Equation 3.4, we use the initial data set, whereas in Equation 3.5, 3.6 and 3.7 the set $\mathcal{S}'$ is based on the bagged data.

Feature noise Weighting the training samples according to their class can tackle the problem of class imbalance, however, it does not solve the challenges of a very high dimensional and rather noisy feature space containing many redundant and non-descriptive dimensions. Especially in combination with an extremely low absolute amount of positive training samples, we consider preventing overfitting the provided data as the biggest challenge. Therefore, in order to enhance generalization, we select the split functions by considering only a rather small subset D' of randomly chosen dimensions of the D-dimensional feature space s.t. $|\mathcal{D}'| = \sqrt{D}$, while at the same time, training a relatively large amount of trees. Furthermore, we let the trees grow rather deep to enhance

the chances of finding more expressive combinations of features in order to compensate the high degree of randomization and the general lack of single well-discriminating features.

Iterative training Beside the described modifications made to the forest itself, we also apply an iterative training procedure inspired by the concept of boosting. This minimizes the error on the training set and encourages the classifier to discriminate even single, visually outstanding instances correctly.

As a first step, we select a subset $\mathcal{S}_1 \subset \mathcal{S}$ of the initial training data randomly without replacement to train the classifier. Then we predict the class labels of all samples. The samples that were predicted wrong, are added to a set of 'hard samples' $\mathcal{S}_H$. In a next iteration, we re-sample the original data and add the hard samples to re-train the classifier. This process is repeated by again predicting the label of all initial samples and adding wrong predictions to the set of hard samples, while this set is never cleared during the whole process. We stop re-training the forest if the error on the training samples drops below a specified threshold, the number of iterations exceeds a certain limit, or if the training error cannot be further improved after 2 retries.

The random sampling of the full training set is done in respect to an inverse class occurrence rate, such that a certain sample belonging to a minority classes is chosen more likely. This approach serves two purposes: The learning problem is much more balanced when training the forest, and redundant instances in majority classes are less dominating over others after fewer iterations. The class-dependent probability of a certain sample s to be chosen is calculated according to

$$r(c) = 1 - \frac{|\mathcal{S}_c|}{|\mathcal{S}|} \tag{3.8}$$

$$p_{sel}(s) = lo + (hi - lo) \cdot e^{-\frac{r(c(s))}{\lambda}} \tag{3.9}$$

where $\mathcal{S}_c$ is the subset of all training samples $\mathcal{S}$ with class label c, $c(s)$ is the class label of a certain sample s and the constants lo, hi and λ are set to 0.2, 0.5 and 0.2, respectively. These settings limit the number of chosen samples from a certain class between 20% and 50% of their respective total amount.

The complete algorithm is summarized in Figure 3.6. Drawback of this iterative training procedure, however, is its error-proneness to noise in the training data, e.g. falsely labeled instances.

Assumed prior probability So far, the trained random forest serves as a predictor for the sliding window approach, solely relying on the visual information of the provided image patch. Considering the task of detecting electrical devices based on 3d scans, there are also other, non-visual features that can be exploited in the decision making process. In our approach, we incorporate the knowledge of the physical geometry of the room. Since

Algorithm 2: Iterative training procedure

Data: $\mathcal{S} : \{s_1 \dots s_N\}$
Result: $H(s)$
$\mathcal{S}_H \leftarrow \{\emptyset\}$
while $t = 1 \dots T$ **do**
  sample randomly $\mathcal{S}_t \subset \mathcal{S}$ according to $p_{sel}(s)$
  train classifier $H_t(s)$ using $\mathcal{S}_t \cup \mathcal{S}_H$
  predict label of all training samples; add wrong predicted to $\mathcal{S}_H$:
    $\mathcal{S}_H \leftarrow \mathcal{S}_H \cup s_k$ if $H_t(s_k) \neq c(s_k), \quad \forall k \in \{1 \dots N\}$
end

Figure 3.6: Iterative training procedure algorithm

an electrical installation device is usually mounted in a way that it is easily accessible or is the least distractive when an appliance is permanently plugged-in, we can assume that the mounting height is highly correlated with the actual presence of an object. We can model this property by considering Bayes' theorem:

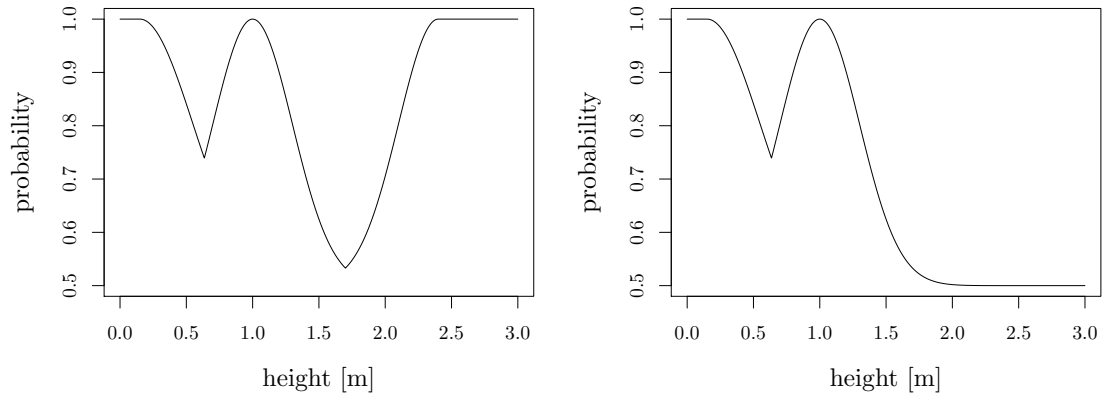
$$P(A|I) = \frac{P(I|A) \cdot P(A)}{P(I)} \tag{3.10}$$

where $P(A|I)$ is the probability of the presence of a device A given an image patch I, $P(I|A)$ the probability of observing a certain image given the information being a device, $P(A)$ is the prior probability of the presence of a target object and $P(I)$ is practically considered as a constant scaling factor. As the random forest is a discriminating classifier trained from labeled examples, it can basically model only the probability $P(I|A)$. This is sufficient in use-cases where the prior probability $P(A)$ is also constant, since then the assumption holds that $P(A|I) \sim P(I|A)$.

However, in this work, we set the prior probability $P(A)$ to model the general relation between the physical height measured from the floor and the presence of an electrical device. Since our training data solely consists of image patches and lacks the information of the mounting height, we approximate the prior by a combination of multiple Gaussian distributions, each representing a characteristic mounting position at floor level, 1 meter above the floor, and, for devices supplying ceiling-mounted appliances, at ceiling level.
In respect to the purpose of an electrical device, we can differentiate two patterns of the expected mounting height, from which we derive two groups of object classes: The first group holds all types of devices which are intended to gain access to the household power grid in a rather permanent fashion, while devices of the second group usually require user-interaction for operation. Typical examples for each group are power outlets, respectively, light switches. Figure 3.7 outlines the two distributions from which we obtain the expected mounting height for a specific object class.

Using these assumed prior probabilities, the final detection result of our algorithm is

determined by the response map of the visual detection from the sliding window approach, multiplied by the value obtained from $P(A)$ for the corresponding class and mounting height.



a) prior for primarily static installations

b) prior for accessible devices

Figure 3.7: Applied prior probabilities modeling the correlation between the mounting height and the presence of electrical devices. Since the expected mounting height varies with the purpose of an electrical device, we group devices that provide access to the electrical wiring as a rather permanent installation (e.g. power outlets, Figure a)), and devices that are supposed to regularly receive user-input (e.g. light switches, Figure b)). While the common mounting height of devices belonging the first group has basically three peaks, we do not assume electrical devices that usually require user-interaction at ceiling level. It may also seem counter-intuitive to expect f.i. switches at floor level, however, form the provided testing data from various countries we could obtain this mounting practice. Note that in order to achieve final prediction probabilities in a proper range of $(0, 1) \in \mathcal{R}$, we scale the prior probability functions s.t. their maximum is equal to 1. Further, to not completely disregard objects that are mounted in less common heights, we ensure that the applied prior probability does not fall below 0.5.

3.4 Summary

Concluding, our algorithm for detecting wall-mounted electrical devices utilizes a sliding window to generate a response map for each trained class. Since the input images are rectified wall textures with a fixed physical spatial resolution and the target objects are also sized consistently, we can omit to search in different scale levels. The values of the response maps for a certain sliding window position are retrieved from a pre-trained random forest classifier. Input for the classifier is a high dimensional feature vector, which is composed by concatenating the results of several feature description approaches in order to combine different visual properties. For the addressed application, the most expressive characteristics are the constellation of visual gradients. Therefore, we apply the 'Histogram of oriented Gradients' and an own, especially designed 'Gradient orientation

Features' descriptor. Other visual features, i.e. the color distribution and the strict horizontal and vertical alignment of the electrical devices, are modeled by image filters forming additional 'feature channels', which are then encoded by Haar-like wavelet pairs. Furthermore, we attempt to add with auto-generated template-matching an additional facet of describing the sliding window content.

Due to the high dimensional and noisy feature space, and the highly unbalanced training data, special attention is required for designing an appropriate classifier. Thus, we let the random forest grow rather deep and use a high degree of randomization in the split selection. Further, we introduce class-dependent sample weights derived from the relative class distribution of the provided training data. An iterative training procedure is applied to additionally enhance the discrimination capabilities between the classes.

In order to incorporate information about the strong correlation between the physical distance to the floor and the presence of an electrical device, we introduce an assumed prior probability with respect to the detected class. The final detection results are determined by applying this prior on the sliding window response maps, followed by a non-max suppression step.

CHAPTER 4

EXPERIMENTS AND EVALUATION

Contents

In this chapter, we evaluate and present the detection performance of our derived algorithm. The first section describes the training and test data we use for the evaluation, followed by an explanation of the applied quality measures and methods to objectively rate the detection performance. Afterwards, we test different parameterizations of the HoG and our novel gradient orientation feature descriptor in order to find their best performing setting for the targeted use case. We proceed with evaluating different combinations of descriptor approaches when concatenating their results to a single feature vector. Using the best performing descriptor setting, we evaluate different values for the maximum depth and number of trees in the random forest, followed by analyzing the impact of the proposed iterative training procedure and the assumed prior probability on the detection result. A subsequent visual examination of the best performing algorithm setup is then conducted to confirm a properly working detection in different scenarios. Finally, this chapter is completed with a summary.

4.1 Dataset

We gathered our testing data from two different sources. On the one hand, we have been provided with 7 relevant wall-texture images derived from real 3D scans of 3 rooms, using the approach of [33]. In order to increase the variety of the test data, we additionally recorded an own set of 24 images showing indoor scenes featuring electrical devices of different design and view angles. To best simulate how these scenes would look in the final DURAARK approach, we took the images with the same resolution as the build-in camera of the scanner and rectified them in respect to the wall geometry using an open-source image processing tool. In total, our test data contains 52 annotated instances of power outlets and 63 light switches, along with their physical mounting height.

For training our algorithm, we collected and annotated 291 images containing 198 power sockets and 401 light switches in a natural surrounding. Each annotation was rectified accordingly to yield squared patches of 128×128 pixel. Additionally, we augmented 4 training samples per annotation by slightly modifying the annotated corners, a random variation of the scale by $\pm 10\%$ and a randomly applied horizontal flip of the resulting image patch. This results in a total amount of 990 and 2.005 training samples for power sockets and light switches, respectively.

In order to form a representative negative set that is supposed to contain everything except electrical devices, we randomly extracted 12 patches from the not selected regions for each annotated image and applied the rectifying homography. Additionally, we extracted random patches all across the SUN2012 dataset of indoor scenes [54]. This yields a total amount of 159.165 negative samples. Figure 4.1 presents some instances of our training data.

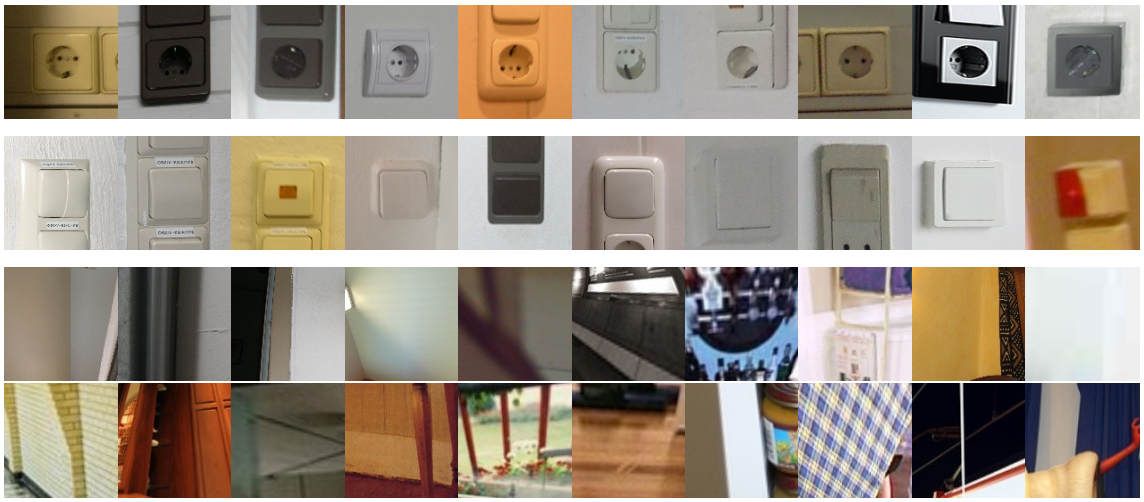


Figure 4.1: Examples of our applied training data. The first row shows some instances forming the class of power outlets, whereas the second row holds examples of light switches. The lower two rows show randomly selected patches of the group of negative samples, which is supposed to contain every observable pattern in indoor scenes except the object classes.

4.2 Quality Measures

In order to objectively measure the quality of certain configurations of our algorithm, we evaluate precision vs. recall curves and compare their F-scores, as described below. Since the training of the classifier and thus the outcoming detection results are relying on a random process, we run each configuration 4 times and accumulate the results to minimize fluctuations caused by the randomization of the algorithm. To further increase the comparability of the different algorithms and their configurations, we set the seed of the random number generator to a distinct value bound to the run count.

Precision vs. Recall A common quality measure for detection algorithms is to plot precision vs. recall in a diagram, while varying the decision threshold. When comparing the detection results against an underlaying ground truth, we can distinguish 4 quantities:

		ground truth	
		positive	*negative*
predicted	*positive*	true positive (TP)	false positive (FP)
	negative	false negative (FN)	true negative (TN)

Table 4.1: The 4 possible entities of a detection result compared to a ground truth.

If an instance is predicted where also a corresponding ground truth entry is present, it is considered to the group of true positives (TP). Predicted objects lacking a ground truth entry are categorized as false positives (FP). Undetected, or missed objects, which are present in the ground truth belong to the group of false negatives (FN). The last group of true negatives (TN) contains all entities with negative detection result, which are also not present in the ground truth. Considering an object detection task whose result space is basically infinite, this last category is rather hypothetical since the quantity of correctly missing items is difficult to determine.

Given the quantities of the described categories of a detection run on the full testset, the *precision* is calculated as the ratio of correctly predicted instances to the total number of predictions

$$precision = \frac{TP}{TP + FP}$$

On the other hand, the *recall* is defined by the ratio of correctly predicted instances to all real existing instances

$$recall = \frac{TP}{TP + FN}$$

Both, *precision* and *recall* are thus bounded in the range of $[0, 1] \in \mathbb{R}$. We plot these two measures in a diagram with $1 - precision$ on the abscissa versus *recall* on the ordinate. Therefore, an optimal detection result would reach the upper left corner of the diagram.

However, under some circumstances (higher detection probability for wrong predictions as for correct ones) the raw plot of precision vs. recall can wiggle in a way that it gets rather difficult to interpret and compare. Thus, we plot the left sided convex hull of the data points. Additionally, we manually add the top right point representing the theoretical decision threshold of 0% and interpolate the missing segments by a spline to get a well comparable and least distorting visualization of the classifier performance.

F-Score The F-score is a real-valued measure of the relation between *precision* and *recall*, and is defined as

$$F = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

Therefore, the F-score is the *harmonic mean* of *precision* and *recall* and is also bound by $[0, 1] \in \mathbb{R}$, where 1.0 is the highest achievable value for an optimal classifier. For evaluation, we compare the maximum F-score observed over varying the decision threshold, which we also highlight on the corresponding precision vs. recall curves as circle.

4.3 Feature Parameters

Utilizing the described evaluation procedure and data, we determine the best suited configuration of the algorithm for our task. Therefore, we systematically investigate different settings of parameters one after another, beginning from the entity we attribute having the biggest impact on the overall detection performance. We begin with investigating different configurations for the HoG descriptor, followed by testing various parameters for our *gradient orientation features*, further abbreviated as 'GoF'. Since the HoG and the GoF descriptor target the main features, i.e. the gradient information with spatial context, we omit individual testing of other outlined descriptors that focus more on supplementary features. Hence, we continue directly with evaluating different combinations of descriptors. In order to evaluate all feature tests already with the best performing classifier setup, we use a random forest consisting of 250 trees and a maximum depth of 22. As discussed in 3.3, the rather depth growing trees should compensate the increased randomization in the split selection, which we proposed to tackle the relatively noisy feature space. Additionally, we expect a higher number of trees preventing a loss of generalization due to the low total amount of positive samples when applying bagging on the dataset given to a single tree. Furthermore, we train the random forest using the introduced iterative training procedure with a maximum of 10 iterations and apply the assumed prior probability on the result. The impact on the detection performance of those actions is investigated separately in Section 4.5.

Parameter	Config 1	Config 2	Config 3
Window size	128 px	128 px	128 px
Cell size	8 px	16 px	16 px
Block size	16 px	32 px	32 px
Block stride	8 px	16 px	8 px
Number of bins	9	9	9
Overlap	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{3}{4}$
Resulting Dimension	8100	1764	6084

Table 4.2: Investigated parameterizations of the HoG descriptor, ordered by the objectives of modeling a fine-structured cell grid, a coarse-structured cell grid and a coarse-structured cell grid with a high overlap of the blocks.

4.3.1 HoG Parameterization

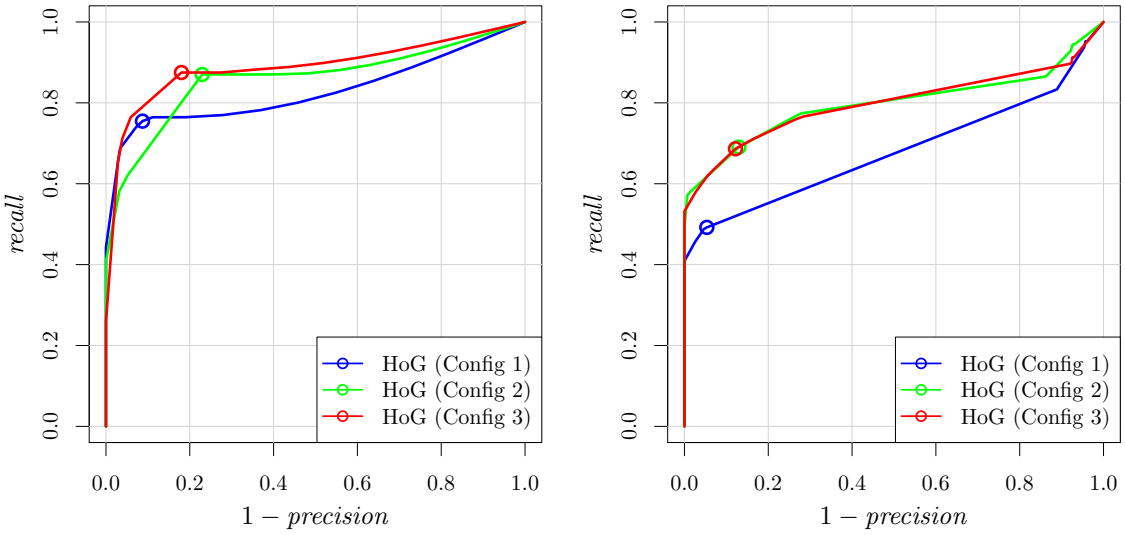
To configure the HoG descriptor properly for our application, we test 3 different parameterizations, each aiming at implementing a certain objective. In fact, we test the performance of the HoG descriptor using a fine-structured cell grid, a coarse-structured cell grid and a coarse-structured cell grid with a high overlap of the normalization blocks. Table 4.2 lists these parameter settings in the described order.

Figure 4.2 shows the precision vs. recall curves observed for the 3 examined configurations along with a comparison of the maximum achieved F-scores. We can observe a significant difference in performance between the configurations, while the third configuration representing a coarse-structured cell grid with a high overlap of the normalization blocks yielded the best results for both object categories.

4.3.2 GoF Parameterization

In this section we determine the best suited parameters also for our *gradient orientation features* descriptor individually. In contrast to the HoG approach where window size, cell size, block size and block stride need to comply with a coherent setup, the parameters for our GoF descriptor can be set more independent. Thus, we determine the parameters sequentially in the order of their considered impact on the detection performance, starting with the size of the randomly selected rectangles sz_{min} and sz_{max}, followed by the symmetry ratio r_{sym} and the number of selected rectangle pairs n_{pairs}.

Rectangle size For determining the best performing setting of of sz_{min} and sz_{max}, we define 5 meaningful configurations as listed in Table 4.3. For the remaining parameters, we fix for all tests the number of randomly chosen rectangle pairs as $n_{pairs} = 1000$ and the symmetry ratio to $r_{sym} = 0\%$. The results are presented in Figure 4.3. In general, it can be observed that settings featuring only larger rectangles perform inferior to settings



a) detection result for power sockets

b) detection result for light switches

F-scores

Category	Config 1	Config 2	Config 3
Power sockets	0.826	0.817	**0.847**
Light switches	0.648	0.770	**0.771**
Average	0.737	0.794	**0.809**

c) Observed maximum F-scores

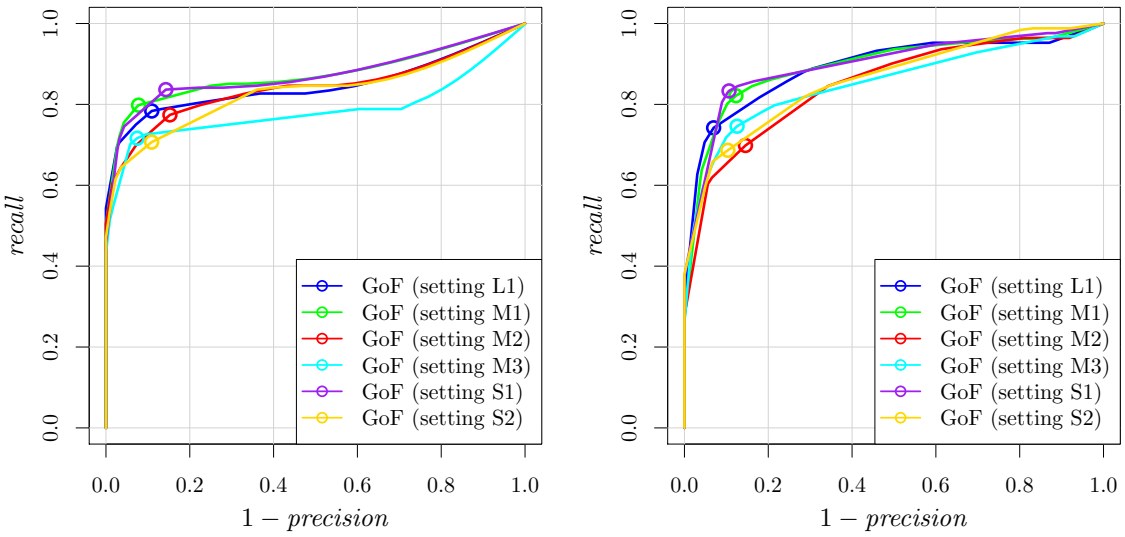
Figure 4.2: Detection performance of using the HoG descriptor only, comparing the parameterizations of Table 4.2 separately for power sockets a) and light switches b). The hereby observed maximum F-scores marked by a circle are tabularly summarized for both object categories along with their average in c).

which allow also smaller regions. Passing features covering the full range of rectangle sizes to the classifier performs significantly better, however, the best detection results are achieved with the two configurations aiming at small regions only, M1 and S1, where the more restricted setting S1 performs slightly better. Hence, we continue testing the other parameters with fixing sz_{min} and sz_{max} to the S1 configuration.

Symmetry Ratio As a next step the impact of symmetry ration r_{sym} is investigated. Therefor, we compare the performance for r_{sym} set to 0%, 10%, 20%, 30%, 40% and 50%. For this tests, we keep the remaining parameter n_{pairs} fixed to 1000. Figure 4.4 shows these experiments. At a first glance, we can confirm that the impact of the symmetry ratio on the detection performance is overall less than for the rectangle size. Analyzed in more detail, higher symmetry ratios tend to perform better for power sockets, whereas

Identifier	sz_{min}	sz_{max}	Description
L1	3%	100%	full size range
M1	3%	60%	medium range of rather small rectangle sizes
M2	20%	80%	medium range of rather medium sized rectangles
M3	40%	100%	medium range of rather large sized rectangles
S1	3%	40%	small range of small sized rectangles
S2	30%	70%	small range of medium sized rectangles

Table 4.3: Tested configurations for sz_{min} and sz_{max} for the random generated rectangles. 'L1' refers to the largest range for selecting the size of the rectangles, most encouraging the classifier to select the best discriminating features itself. The medium range configuration 'M1', 'M2' and 'M3' bound the size of the rectangles in between 60% of the input patch, while allowing rather small, medium and large sized regions, respectively. The small range settings 'S1' and 'S2' limit the size more strictly to an interval of 40%, again subject to rather small and medium sized rectangles.



a) detection result for power sockets

b) detection result for light switches

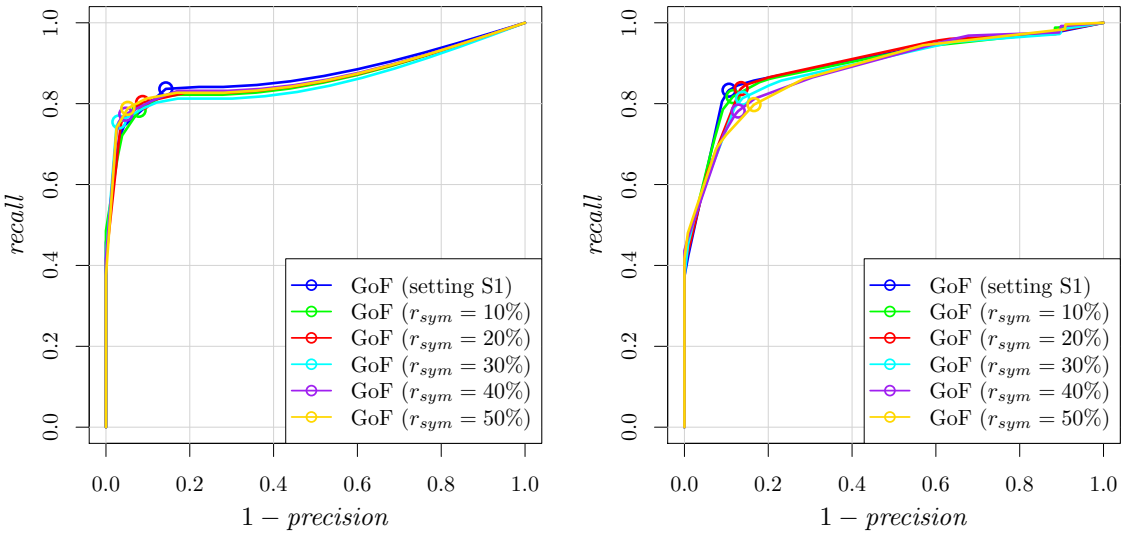
F-scores

Category	setting L1	setting M1	setting M2	setting M3	setting S1	setting S2
Power sockets	0.834	**0.856**	0.809	0.808	0.847	0.788
Light switches	0.826	0.848	0.769	0.805	**0.862**	0.778
Average	0.830	0.852	0.789	0.806	**0.855**	0.783

c) Observed maximum F-scores

Figure 4.3: Detection performance of different sz_{min} and sz_{max} settings as listed in Table 4.2 separately for power sockets a) and light switches b). The observed maximum F-scores are summarized in c).

this trend follows the opposite relation for light switches. Thus we decide purely based on the average F-score, which is highest for a symmetry ratio of 0%.



a) detection result for power sockets

b) detection result for light switches

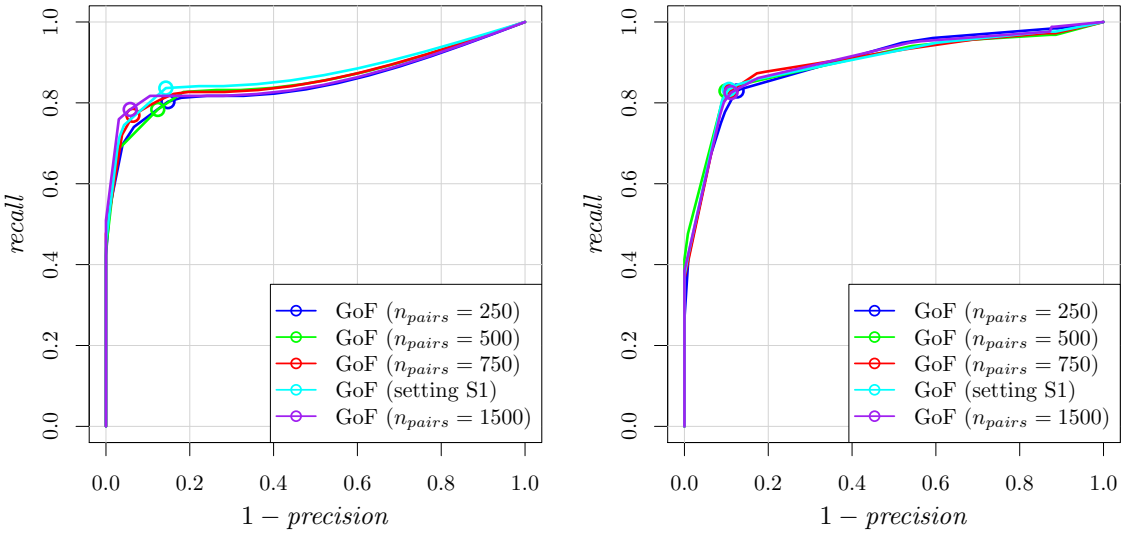
F-scores

Category	setting S1	$r_{sym} = 10\%$	$r_{sym} = 20\%$	$r_{sym} = 30\%$	$r_{sym} = 40\%$	$r_{sym} = 50\%$
Power sockets	0.847	0.847	0.854	0.849	0.854	**0.861**
Light switches	**0.862**	0.849	0.851	0.837	0.824	0.815
Average	**0.855**	0.848	0.853	0.843	0.839	0.838

c) Observed maximum F-scores

Figure 4.4: Detection performance of different values for r_{sym}, separately for power sockets a) and light switches b). Note that the blue line represents $r_{sym} = 0\%$, however, we kept the notation of Figure 4.3 to emphasize that it is the same configuration. The observed maximum F-scores are summarized in c).

Number of Rectangles The last parameter of the GoF descriptor we want to investigate is the number of randomly generated rectangle-pairs n_{pairs}, which is the only parameter that defines the dimension of the resulting feature vector. Again, we compare 5 discrete values for n_{pairs}, namely 250, 500, 750, 1000 and 1500. We can observe that the impact of n_{pairs} on the overall detection performance is rather low for $n_{pairs} \geqslant 750$. However, the best average F-score was achieved by setting $n_{pairs} = 1500$ yielding a 6000-dimensional feature vector, which is also a legitimate decision in order to balance the amount of dimensions when concatenating with the best performing configuration of the HoG descriptor to a single feature vector.



a) detection result for power sockets

b) detection result for light switches

F-scores

Category	$n_{pairs} = 250$	$n_{pairs} = 500$	$n_{pairs} = 750$	setting S1	$n_{pairs} = 1500$
Power sockets	0.827	0.827	0.844	0.847	**0.856**
Light switches	0.851	0.864	0.858	**0.862**	0.856
Average	0.839	0.846	0.851	0.855	**0.856**

c) Observed maximum F-scores

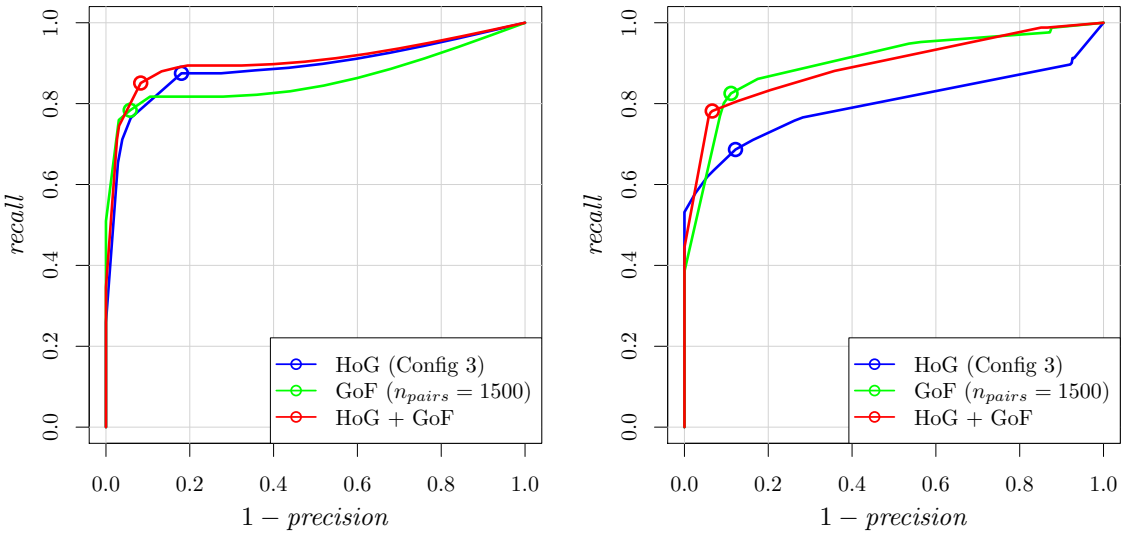
Figure 4.5: Detection performance of ascending values for n_{pairs}, separately for power sockets a) and light switches b). Again, the cyan line represents $n_{pairs} = 1000$, which is the same parameterization S1 of Figure 4.3 and 4.4. The observed maximum F-scores are listed in c).

4.4 Feature Combinations

In this section, we investigate the detection performance of various descriptors combined in one large feature vector that serves as input for the classifier. Considering that the most important visual features, the constellation of gradients, are only well represented by the HoG and the GoF descriptor, we use them as basis set for testing combinations with other descriptors that focus more on supplementary visual information. Additionally to the precision vs. recall curve and the hereby observed maximum F-scores, we analyze the relative variable importance averaged over the 4 executed runs of each descriptor.

HoG + GoF The first feature combination we investigate is the HoG and GoF descriptor, both in their individually best performing configurations. Figure 4.6 shows the results of concatenating the feature vectors of both descriptors to a 12.084 dimensional

vector that is passed to the classifier. It can be observed that the random forest is able to complement the performance of both descriptors. As the combination of HoG and GoF clearly outperforms the individual detection performance for power sockets, however, for light switches, the results are slightly worse compared to the individual GoF descriptor. We mainly contribute this effect to the initial significantly inferior performance for this category of the HoG descriptor. The observation that the GoF descriptor can provide more valuable information to the random forest is also confirmed by the distribution of the variable importance scores on the descriptors, as presented in Figure 4.7. However, the average F-score over both categories is significantly improved by combining both feature descriptors.



a) detection result for power sockets

b) detection result for light switches

F-scores

Category	HoG (Config 3)	GoF ($n_{pairs} = 1500$)	HoG+GoF
Power sockets	0.847	0.856	**0.883**
Light switches	0.771	**0.856**	0.851
Average	0.809	0.856	**0.867**

c) Observed maximum F-scores

Figure 4.6: Evaluation of the combined HoG and GoF descriptor compared to their individual performance. As the combination outperforms both single descriptors for power sockets (a)), the individual GoF descriptor performs best for light switches (b)). The combination of both descriptors however clearly reaches the highest F-score averaged over the categories, as listed in c).

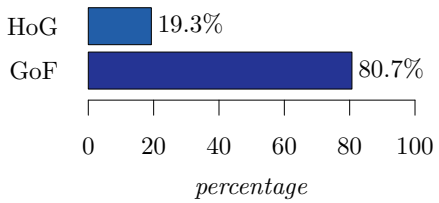
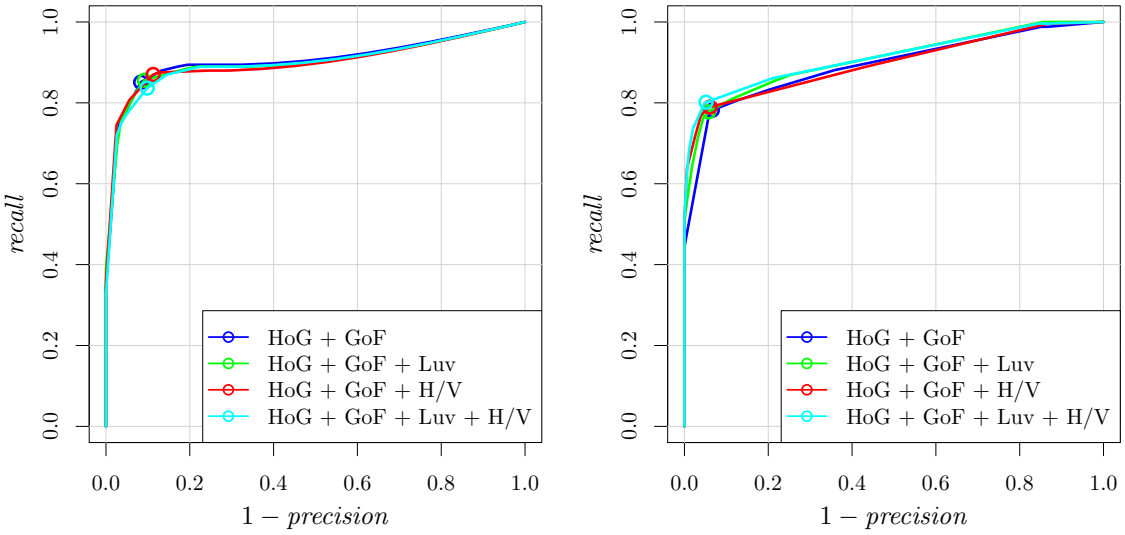


Figure 4.7: The distribution on a percentage basis of the accumulated variable importance per descriptor, averaged over the 4 test runs. The importance of a feature variable is determined by the calculated information gain of the according split nodes, weighted by the relative amount of training samples reaching that node.

HoG + GoF + Image Filters The first supplementary descriptors we evaluate when added to the set of HoG and GoF are the additional image filters as introduced in Section 3.2.3. For generating the rectangular subregions to describe the image filter responses, we tested the equivalent size settings 'L1', 'M1' and 'S1' as listed in Table 4.3. The overall best results were achieved by the setting 'L1', hence we set $sz_{min} = 3\%$, $sz_{max} = 100\%$. To get a better understanding of the impact of the separate image filters we form two groups of feature channels, namely 'Luv' composed by the L, u and v color channel filter, and 'H/V' consisting of the strictly horizontal and vertical gradient filter. To ensure a relevant amount of these feature channels in the accumulated vector, we generate $n_{pairs} = 1500$ rectangle pairs for each group.

Figure 4.8 shows the evaluation results. Whereas the detection performance for power outlets decreases with adding additional feature channels, the ratings increase in the case of light switches. We can explain this effect by the different appearances of the two categories. Power sockets have with the cylindrical notch a very significant visual feature that can be modeled best by gradient based descriptors. In this case, adding other feature channels just introduces additional noise to the vector representation. On the other hand, the detection of light switches, which in general consist of fewer distinctive edges, slightly benefits from the additional information about color and the presence of strictly horizontal or vertical gradients. However, the general low impact of the additional feature channels is also confirmed by the analysis of the variable importance of the HoG+GoF+Luv+H/V configuration, as depicted by Figure 4.9. Comparing the importance of H/V and Luv features, the random forest incorporates the H/V channel significantly more as the Luv group, which also reflects the observed differences of the detection performance results. Although the average F-scores are rather unaffected for the different feature compositions, we prefer the combination of HoG+GoF+Luv+H/V since it most balances the detection performance between the object classes.

HoG + GoF + PCA The last experiment we evaluate on the feature side is adding the cross-correlation scores of matching with the pre-trained Eigen-images obtained by PCA to the feature vector. Since the PCA representation of the search window is rather low-dimensional, we replicate the resulting vector several times when combining with the HoG and GoF descriptor in order to provide fair chances of a PCA-feature to be selected in a split node of the random forest. In our experiments, we have added the PCA based features to the basic HoG+GoF configuration, as well as to the best performing feature



a) detection result for power sockets

b) detection result for light switches

F-scores

Category	HoG+GoF	HoG+GoF+Luv	HoG+GoF+H/V	HoG+GoF+Luv+H/V
Power sockets	**0.883**	0.881	0.879	0.868
Light switches	0.851	0.852	0.858	**0.869**
Average	0.867	0.867	**0.868**	**0.868**

c) Observed maximum F-scores

Figure 4.8: Evaluation of the feature channels 'Luv' and 'H/V'. Figure a) and b) show precision vs. recall for power outlets and light switches, respectively. The maximum F-scores over varying the decision threshold are listed in c).

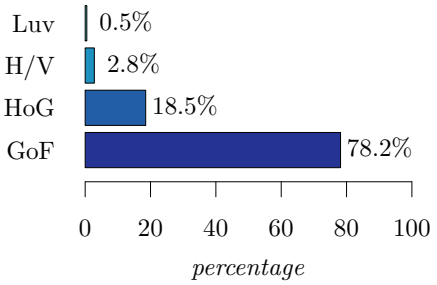
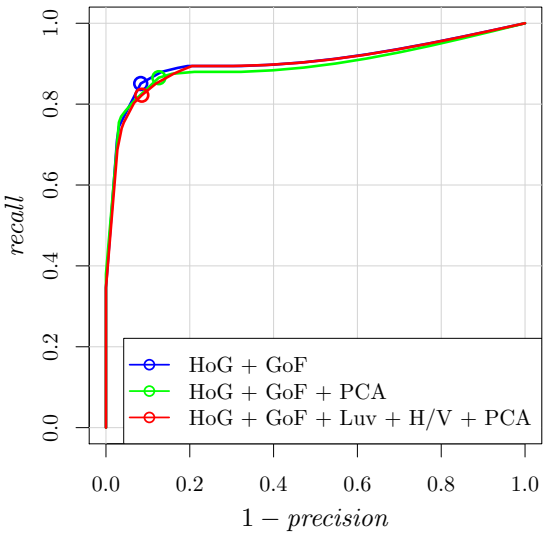


Figure 4.9: Percentagewise distribution of the determined variable importance per descriptor of the HoG+GoF+Luv+H/V feature combination in ascending order.

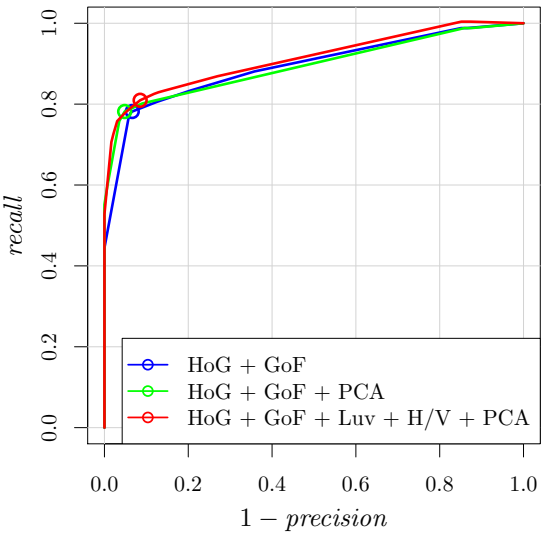
set so far, HoG+GoF+Luv+H/V.

The resulting detection performance is shown in Figure 4.10, Figure 4.11 depicts the relative feature importance of HoG+GoF+PCA determined by the classifier. It can be observed that adding the Eigen-space representation to the basic HoG+GoF features slightly increases the detection rating for light switches, however, it decreases the performance for

power sockets significantly. Similar observations can be made when adding the PCA based feature to the full set of HoG+GoF+Luv+H/V, but lacking the small boost for the class of light switches. Summarized, all combinations incorporating the PCA features perform inferior to the HoG+GoF+Luv+H/V feature configuration. As reasonable conclusion also in accordance with the determined feature importance, we consider the PCA representation not able to provide any additional valuable information to the HoG+GoF+Luv+H/V features since it may be less effective in coding the spatial constellation of gradients than the HoG and GoF descriptors.



a) detection result for power sockets



b) detection result for light switches

F-scores

Category	HoG+GoF	HoG+GoF+PCA	HoG+GoF+Luv+H/V+PCA
Power sockets	**0.883**	0.870	0.866
Light switches	0.851	0.858	**0.859**
Average	**0.867**	0.864	0.862

c) Observed maximum F-scores

Figure 4.10: Evaluation of the PCA space representation added to the feature pool.

4.5 Classifier Evaluation

In this section we evaluate different settings for the maximum depth and number of the trees in the forest, as well as the introduced iterative training procedure and the application of an assumed prior probability, as outlined in Section 3.3. For comparing the performance

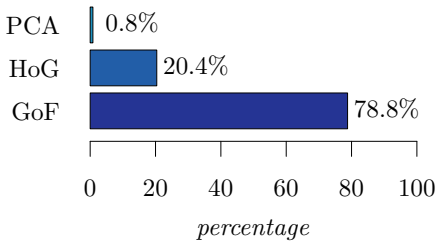


Figure 4.11: The relative variable importance per descriptor for the PCA Eigen-representations added to the HoG+GoF feature set. Although the split selection was artificially encouraged to consider all features, the PCA based descriptor seem not to provide any additional discriminating information.

of training and evaluating the classifier with the different options, we utilize the best performing feature setup, i.e. HoG+GoF+Luv+H/V.

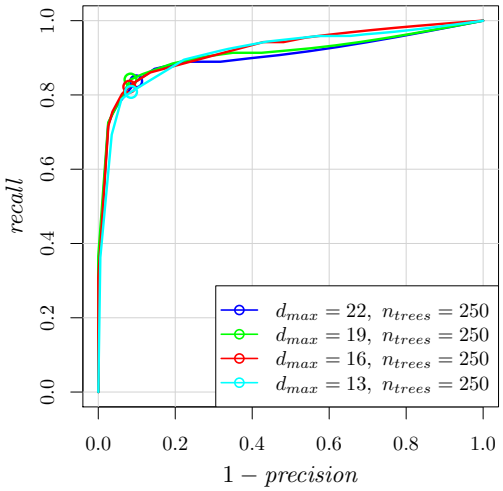
Depth and Number of Trees To demonstrate the impact of the maximum depth d_{max} and total number of trees n_{trees} in the random forest, we evaluate both parameters separately. Figure 4.12 shows the conducted experiments. In general, we can observe that higher values for both, d_{max} and n_{trees} yield better overall performance. However, the impact is less significant than expected. When separately analyzing the object classes, the classifier parameters show a higher influence on the detection performance for light-switches. This could be explained by more extensive combinations of features required to model the appearance of light switches due to their less distinct visual cues.

The overall best results were achieved with $d_{max} = 22$ and $n_{trees} = 250$, for our evaluation data these numbers may seem a bit overstated though. Nevertheless, our experiments show that an overestimated depth and amount of trees does not harm the performance. Considering the expected growth of the DURAARK training database over time, we decide for the setting of $d_{max} = 22$ and $n_{trees} = 250$ in order to preserve future compatibility.

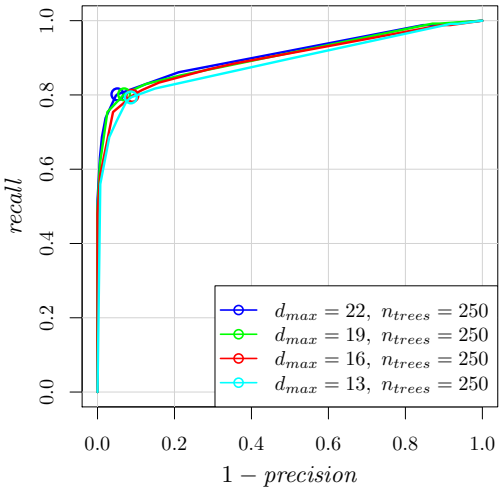
Iterative Training and Prior Probability For investigating the impact of the iterative training procedure, we compare with training the random forest as usual with no iterations and using all available training samples. To show the effect of the assumed prior probability, we also add precision vs. recall when applying the decision threshold directly on the reported certainty of the classifier. Figure 4.13 shows the results for the described test settings. It can be observed that the assumed prior probability significantly increases the overall detection performance, however, the lower recall around 0.5 precision rate for power outlets indicates that some individual instances that may not comply with the estimated common mounting height are harder to recognize.

Comparing the normal training with the iterative procedure, the performance difference is less significant. Investigating the maximum observed F-scores, in fact the two methods achieve equal performance. However, the overall higher recall of the iterative training procedure indicates that the classifier can better recognize more difficult instances.

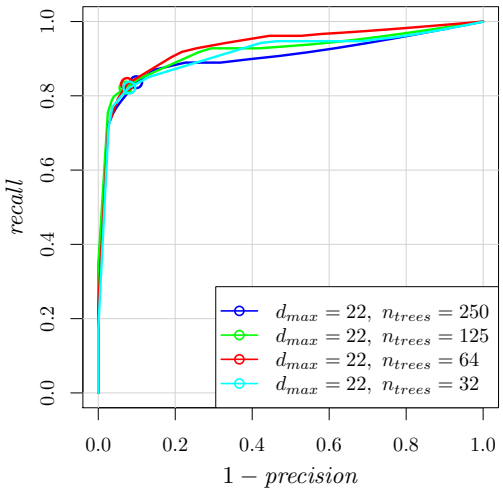
In Figure 4.14 we additionally compare the prediction error on the full training set for the normal and the iterative training procedure. We can see that in the first iteration



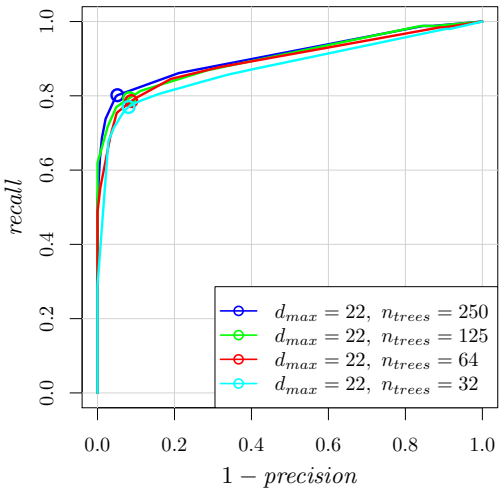
a) detection result for power sockets



b) detection result for light switches



c) detection result for power sockets



d) detection result for light switches

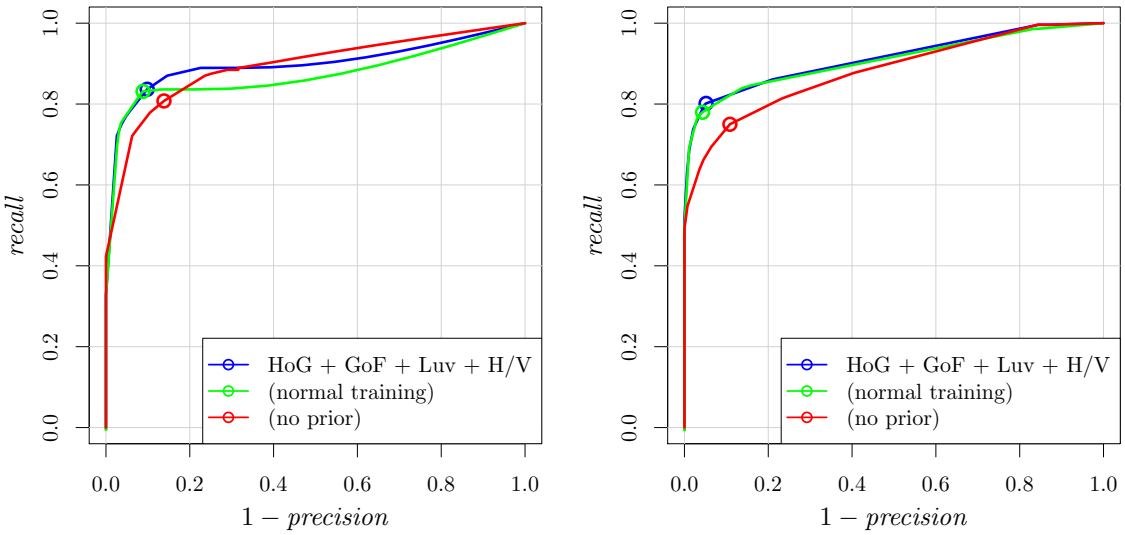
F-scores

Category	n_{tree}	250				125	64	32
	d_{max}	13	16	19	22			
Power sockets		0.858	0.868	0.875	0.868	0.872	**0.876**	0.868
Light switches		0.849	0.850	0.861	**0.869**	0.851	0.844	0.838
Average		0.854	0.859	**0.868**	**0.868**	0.862	0.860	0.853

e) Observed maximum F-scores

Figure 4.12: Evaluation of the maximum depth d_{max} and number of trees n_{trees} of the random forest. Note that the setting $d_{max} = 22$ and $n_{trees} = 250$ corresponds to the same configuration everywhere else denoted as HoG+GoF+Luv+H/V.

the amount of wrong classified samples is about 3 to 4 times larger than using the full training set at once. Anyway, after only a few iterations the iterative approach is able to significantly reduce the classification error on the training set. Although in one run the error could not be reduced below 10 samples in 2 more attempts, in general the iterative training procedure achieves almost a classification rate twice as good as the normal approach on the training set.



a) detection result for power sockets

b) detection result for light switches

F-scores

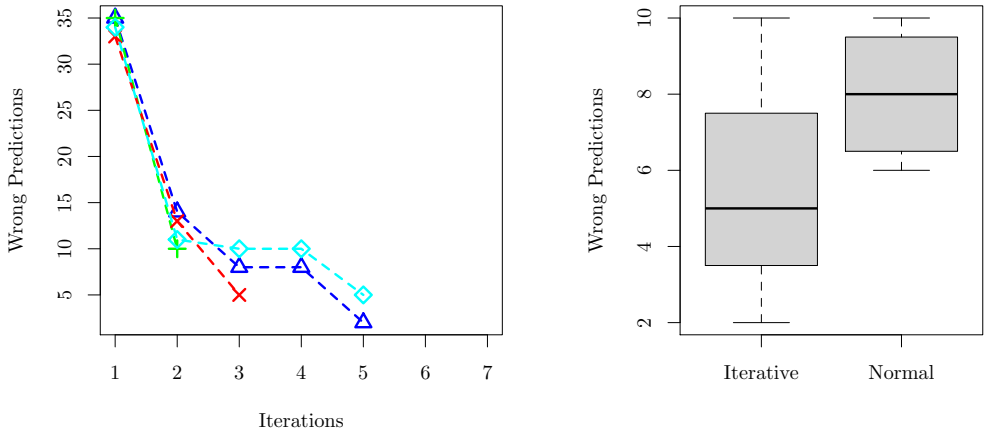
Category	HoG+GoF+Luv+H/V	(normal training)	(no prior)
Power sockets	0.868	**0.870**	0.834
Light switches	**0.869**	0.859	0.815
Average	**0.868**	0.864	0.824

c) Observed maximum F-scores

Figure 4.13: Evaluation of the introduced iterative training procedure and applying a prior probability. The blue line shows precision vs. recall of applying both methods, which were already used for determining the best feature set. The observed detection performance of normal training with applying the prior probability is visualized in green. The red line shows precision vs. recall using iterative training, but without applying the assumed prior probability.

4.6 Qualitative Evaluation

In order to give a better impression of the actual detection capabilities of our derived algorithm on real images, we present some selected results from our test-set, each demon-



a) Evolution of the training error **b)** Comparison of the training error

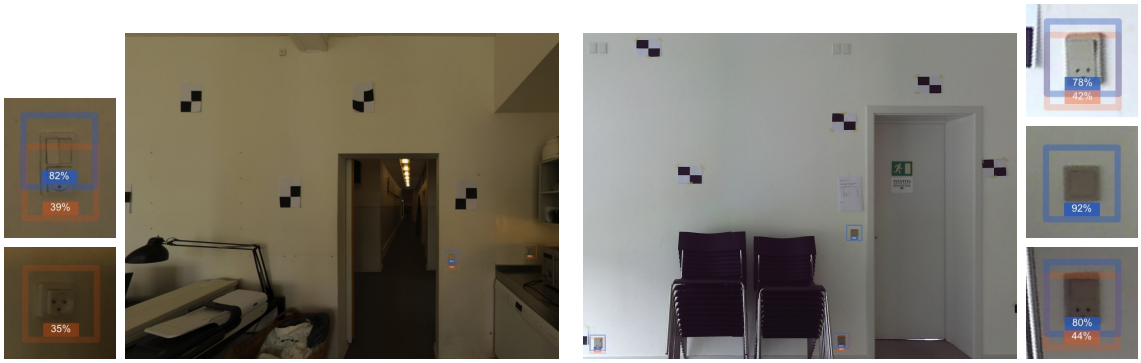
Figure 4.14: Analysis of the training error of the iterative training procedure compared to the normal approach. Figure a) shows the evolution of the training error over the iterations using the HoG + GoF + Luv + H/V feature configuration for the 4 performed runs. The according mean, standard deviation and extrema values are depicted as box plot in Figure b), along with the same statistics observed by the normal training procedure.

strating a special condition. For this evaluation we use the best performing setup and applied the decision thresholds for each category where the highest F-score was obtained. Figure 4.15a) and 4.15b) show correctly detected electrical devices in a very cluttered environments, without hallucinating false positives somewhere else. Also in low light conditions, our algorithm is able to correctly detect the target objects, as demonstrated in Figure 4.15c). Even on very bad image quality and the lack of the characteristic notch for power outlets, we are able to detect and classify all instances correctly in Figure 4.15d). Figure 4.15e) shows a scenario where our detection algorithm missed some instances. Reasons are most likely the highly distorted shape due to the steep view angle and the high image blur at the corresponding location. A wrong detection can be observed on Figure 4.15f), where the round, framed button of the elevator was confused with a power outlet. Figure 4.15g) depicts a scenario where the classifier was even able to detect power outlets where an appliance is plugged in, but also classified an Ethernet connector as light switch. This can be explained by the very similar appearance along with the lack of training data for Ethernet connectors.

4.7 Summary

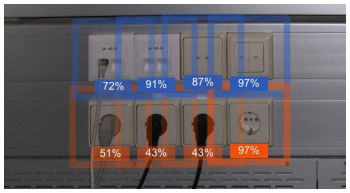
In this chapter, we addressed determining the best-suited configurations and evaluating the performance of our derived detection algorithm. In the first part, we outlined our test and training data, which we gathered from various sources. The final test-set consists of 31 wall texture images, containing 52 annotated instances of power outlets and 63 light

a) Cluttered scene **b)** Noisy texture



c) Low-light condition **d)** Bad quality, no notch

e) Large view angle **f)** Similar appearance



g) Altered targets

Figure 4.15: Qualitative evaluation on various examples. The large images show the full scenes cropped to fit the same aspect ratio with the overlayed detection results. The small images on the sides show magnified snippets of the detection locations to get a better impression of the image quality and the actual appearance of the targets. Blue squares mark detected light switches, whereas power sockets are highlighted orange. The certainty of a detection is printed below each target and also indicated by the opacity of the drawn boxes.

switches along with their physical mounting height. For training our detection pipeline, we collected, annotated, rectified and augmented several images ourselves, yielding a total amount of 990, 2.005 and 159.165 training samples for power sockets, light switches, and the negative class, respectively. For comparing the performance of different setups of the detection algorithm, we utilize precision vs. recall charts and the hereby maximum obtained F-score of these measures. For each tested configuration, we ran the whole training and detection process 4 times and combined the results to minimize fluctuations due to the random processes in the algorithm.

To find the best performing feature setup within the vast amount of parameters and possible configurations, we investigated the HoG and the specially designed GoF descriptor separately and tested their parameters in the order of their impact on the detection performance. Afterwards, we observed the results of different combinations also with the introduced supplementary features. Our experiments show that combining the HoG and GoF descriptor yielded significantly better results compared to their single performance. The impact of adding different supplementary features is less significant but noticeable. Whereas including additional feature channels coding the color distribution and the presence of strictly horizontal and vertical gradients can increase the performance for one category, the opposite is the case for the other object class. However, adding these features results in better leveling the detection rate between the categories. A similar finding was made by incorporating matching against PCA autogenerated templates, whereas we concluded that these features do not create any additional benefit to the detection pipeline overall. Furthermore, we investigated the impact of the maximum depth and number of trees in the random forest, as well as the introduced modifications of the training and evaluation procedure. The conducted experiments on our data attest the depth and size of the forest only a minor influence on the detection performance, whereat the best results were achieved with a rather high number for both values. A much bigger impact was observed by utilizing an assumed prior probability to model the common mounting practices for electrical devices which vastly enhances the results. Moreover, the iterative training procedure of the classifier also improves the overall performance, along with significantly reducing the prediction error on the training set. With the best performing configuration consisting of the HoG+GoF+Luv+H/V features and both modifications on the classifier side, we achieve at the maximum observed F-scores a recall of $83,7\%$ and $90,2\%$ precision for power sockets, and $80,2\%$ recall rate at $94,8\%$ precision for light switches.

A finalizing qualitative evaluation of the results confirmed a good detection performance in various difficult situations. Conditions our algorithm struggles the most are a high blur or bad quality of the query images. Additionally, we could observe some wrong detections triggered by various objects that share many visual features with our targets. However, this could be prevented by explicitly adding such instances to the negative class.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

In this thesis we derived and evaluated a detection algorithm for wall-mounted electrical devices in indoor scenes, which is implemented in a practical application. In fact, the final approach is part of the EU-funded DURAARK framework for analyzing and digitalizing building structures, more precisely to estimate the electrical wiring inside walls. Therefore, our developed algorithm needs to fulfill special demands and requirements concerning input data, usability and extensibility.

The original input data for processing a single room consists of a 3D point cloud and a recorded panoramic image, which serves as good basis for reconstruction of buildings in very high detail. However, our analysis revealed an insufficient spatial resolution of the point cloud in exposed areas for reliable detection of the plain and undistinctive targets. Hence it was decided to introduce a pre-processing step that generates rectified wall-texture images by projecting the panoramic image onto detected 3D planes, which defines the input for our algorithm. This decision also serves a more user-friendly usability and extensibility of the corresponding application component, since the user is required to provide images only in order to extend the training database. Anyway, in general the benefit of having access to a full 3D point cloud for the addressed detection task turned out to be less than expected. Despite the rectified representation and the known real scale information, the partially low quality of the panoramic image, as well as the high variability and lack of distinctive features of the target objects result in a very challenging detection task. Additionally, the amount of available training data is very limited, which excluded the possibility to train an end-to-end detector like the currently hyped convolutional neural networks (CNN).

In order to solve this specific problem, we decided for a classical detection pipeline consisting of handcrafted features evaluated by a classifier. Therefore, we investigated

different descriptor approaches for detecting plain, untextured targets. We fused the most promising concepts in a novel feature descriptor named 'gradient orientation features' (GoF), which is especially designed for our use-case. Additionally, we evaluated concatenating the representations of the outlined descriptor algorithms in various combinations, aiming at combining the strengths of the different approaches. However, joining various feature descriptors can result in a very noisy and redundant vector representation of the underlaying image. This demands a classifier with strong feature selection capabilities, along with handling the multi-class requirement of the application and a highly imbalanced training dataset. Therefore, we chose a random forest classifier and added some modifications to the basic algorithm. In order to handle the imbalanced training data, we introduced sample weights impacting the split node creation and the vote of a single tree. Further, an additional iterative training procedure specifically addresses the low total amount of training samples for the object classes. Last but not least, we modeled common mounting heights of electrical devices by an assumed prior probability applied on the result of the classifier.

Our conducted experiments show that our GoF descriptor outperformed the well-established 'histograms of oriented gradients' (HoG) descriptor in a direct comparison. Furthermore, we were able to show that combinations of feature descriptors can perform significantly better than single descriptor setups, however, the performance decreases when adding features not providing any additional value. The evaluation of the iterative training procedure attests a better detection of more difficult instances while enforcing a lower prediction error on the training set. However, this approach makes the random forest more vulnerable to mislabeled training data. The biggest improvements on the classifier side were achieved by the assumed prior probability, while the maximum depth and number of trees in the forest show only a minor impact on the results. Using the best performing feature and classifier setup, we achieved a recall of $83,7\%$ at $90,2\%$ precision for power sockets, and $80,2\%$ recall at a precision of $94,8\%$ for light switches, both datapoints measured at the maximum observed F-scores per object category.

5.2 Future Work

Currently, in the final implementation detection is performed on every plain wall segment that was extracted by a plane fitting algorithm, also retrieving structures that do not necessarily contain electrical wires like doors, windows, or large furniture like tables or wardrobes. Since the semantic classification of those structures is also part of the DURAARK tools, one could convey this information to exclude certain textures from the visual detection in advance. However, this semantic information is anyway already incorporated by the grammar shape models that estimate the invisible electrical installation.

Another improvement of the detection results could be of course accomplished by directly passing the 3D information of an investigated region to the classifier, either in the form of a descriptor modeling the arrangement of the 3D points, or a derived depth image

described by 2D approaches. However, this would be only possible if also the training data would provide the same information. We propose two possibilities in gathering this training data in the future. First, one could simply manually collect and annotate user data that was uploaded over time to the DURAARK application. The second possibility would require the user to take multiple images from different angles for one training instance, from which a 3D reconstruction could be calculated and sampled to best match the appearance of the point cloud captured by the LiDAR scanner. However, the reconstruction of the mostly texture-less objects is not trivial and most probably would require high effort to make it reliably working with user-provided data. Furthermore, pursuing this approach would at some point require to either generate 3D data also for the negative class, or implementing a classification deciding on a similarity score or matching distance without the need for a negative class.

APPENDIX A

CALCULATION OF EFFECTIVE 3D SCAN RESOLUTION

This chapter investigates the theoretically achievable scan resolution of a 3D LiDAR scanner that is equipped with an angle based deflection unit in a cuboid room. The walls of the room are modeled as separate upright Cartesian planes.

When recording a point cloud, a scanner emits multiple laser beams by altering its azimuth and elevation angle by a constant step value, while measuring the euclidean distance towards a reflecting surface. Hence, the measurement points are organized within a regular pattern of azimuth and elevation angles in a polar coordinate system. The amount of measurement points is fixed, therefore, we are interested in the achievable point density at each position on a Cartesian plane, which is obviously a non-linear function.

To model the relation between the internal 2D Cartesian coordinates of an arbitrary upright plane and the deflection angles of the laser scanner, we consider a plane ε defined by

$$\varepsilon: \; a \cdot x_{3D} + b \cdot y_{3D} + c \cdot z_{3D} + d = 0, \quad \text{with} \quad a = c = 0, \; b = 1, \; d = \tilde{y} \tag{A.1}$$

(i.e. a plane spanned by the x_{3D}- and z_{3D}-axis at a distance $\tilde{y}$ to the origin). Considering the model shown in Figure A.1, the transformation between 3D polar $[r, \varphi, \vartheta]$ and 3D Cartesian coordinates is given by

$$x_{3D} = r \cdot \cos\vartheta \cdot \sin\varphi \tag{A.2}$$

$$y_{3D} = r \cdot \cos\vartheta \cdot \cos\varphi \tag{A.3}$$

$$z_{3D} = r \cdot \sin\vartheta \tag{A.4}$$

whereat r represents the euclidean distance, φ and ϑ denote the azimuth and the elevation

angle of the deflection unit, respectively.

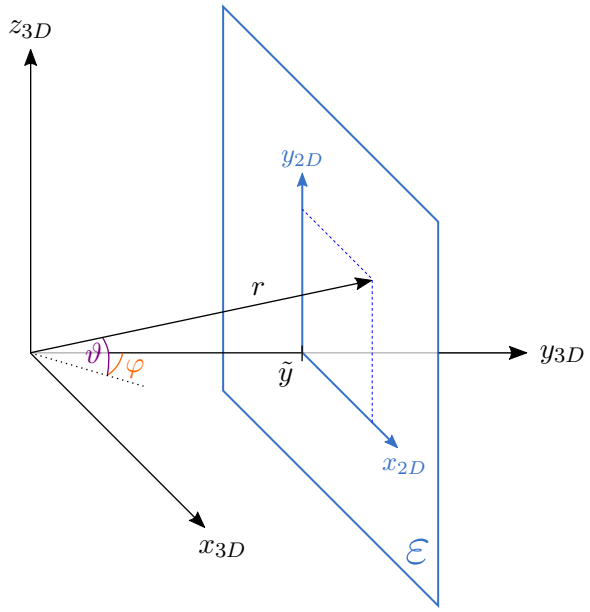


Figure A.1: Geometric model for effective scan resolution calculation

We can calculate the Cartesian 3D points that are part of ε in respect to φ and ϑ by expressing r implicitly as $r = \frac{\tilde{y}}{\cos\vartheta \cdot \cos\varphi}$. Hence,

$$x_{3D} = \tilde{y} \cdot \frac{\sin\varphi}{\cos\varphi} \tag{A.5}$$

$$y_{3D} = \tilde{y} \tag{A.6}$$

$$z_{3D} = \tilde{y} \cdot \frac{\sin\vartheta}{\cos\vartheta \cdot \cos\varphi} \tag{A.7}$$

with the bounds $\varphi \in (-\frac{\pi}{2}, +\frac{\pi}{2})$ and $\vartheta \in (-\frac{\pi}{2}, +\frac{\pi}{2})$. Assuming a mapping of the internal 2D coordinate system of the plane by $x_{2D} = x_{3D}$ and $y_{2D} = z_{3D}$, we can model the transformation between the azimuth and elevation angle of the deflection unit and the internal 2D coordinates of ε by

$$\begin{aligned} x_{2D} &= \tilde{y} \cdot \frac{\sin\varphi}{\cos\varphi} = \tilde{y} \cdot \tan\varphi \\ y_{2D} &= \tilde{y} \cdot \frac{\sin\vartheta}{\cos\varphi \cdot \cos\vartheta} = \tilde{y} \cdot \frac{\tan\vartheta}{\cos\varphi} \end{aligned} \tag{A.8}$$

As the area of an arbitrary segment $\mathcal{S}(x_{2D}, y_{2D})$ on the plane ε is defined by $A = \iint_{\mathcal{S}(x_{2D},y_{2D})} 1 \, dx_{2D} dy_{2D}$, we can apply a coordinate transformation $(x_{2D}, y_{2D}) \rightarrow (\varphi, \vartheta)$ of the area integral via the functional determinate

$$\begin{aligned}
\det\left(\frac{\partial(x,y)}{\partial(\varphi,\vartheta)}\right) &= \det\begin{pmatrix}\frac{\partial x}{\partial\varphi} & \frac{\partial x}{\partial\vartheta}\\ \frac{\partial y}{\partial\varphi} & \frac{\partial y}{\partial\vartheta}\end{pmatrix}\\
&= \det\begin{pmatrix}\tilde{y}\cdot\frac{1}{\cos^2\varphi} & 0\\ \tilde{y}\cdot\frac{\tan\vartheta\cdot\tan\varphi}{\cos\varphi} & \frac{\tilde{y}}{\cos\varphi}\cdot\frac{1}{\cos^2\vartheta}\end{pmatrix}\\
&= \frac{\tilde{y}}{\cos^2\varphi}\cdot\frac{\tilde{y}}{\cos\varphi\cdot\cos^2\vartheta}\\
&= \frac{\tilde{y}^2}{\cos^3\varphi\cdot\cos^2\vartheta}
\end{aligned} \tag{A.9}$$

resulting a formula for the area of a segment $\mathcal{S}'(\vartheta,\varphi)$ on ε, defined by the deflection angles of the laser beam hitting the plane ε:

$$A = \iint_{\mathcal{S}'(\vartheta,\varphi)} \frac{\tilde{y}^2}{\cos^3\varphi\cdot\cos^2\vartheta}\; d\vartheta\; d\varphi \tag{A.10}$$

By setting proper integration limits, we can calculate the area associated with exactly one discrete scan point on ε, in respect to 2 concrete deflection angle values $(\vartheta\prime,\varphi\prime)$. Therefore, we set

$$A(\vartheta\prime,\varphi\prime) = \int_{\varphi\prime-\frac{\Delta\varphi}{2}}^{\varphi\prime+\frac{\Delta\varphi}{2}}\int_{\vartheta\prime-\frac{\Delta\vartheta}{2}}^{\vartheta\prime+\frac{\Delta\vartheta}{2}} \frac{\tilde{y}^2}{\cos^3\varphi\cdot\cos^2\vartheta}\; d\vartheta\; d\varphi \tag{A.11}$$

whereat $\Delta\varphi$ and $\Delta\vartheta$ denote the constant step size values of the horizontal and the vertical component of the deflection unit, and $\varphi\prime$ and $\vartheta\prime$ are the absolute azimuth and elevation angle values of the corresponding discrete measurement point, respectively. For the $A(\vartheta\prime,\varphi\prime)$ we get

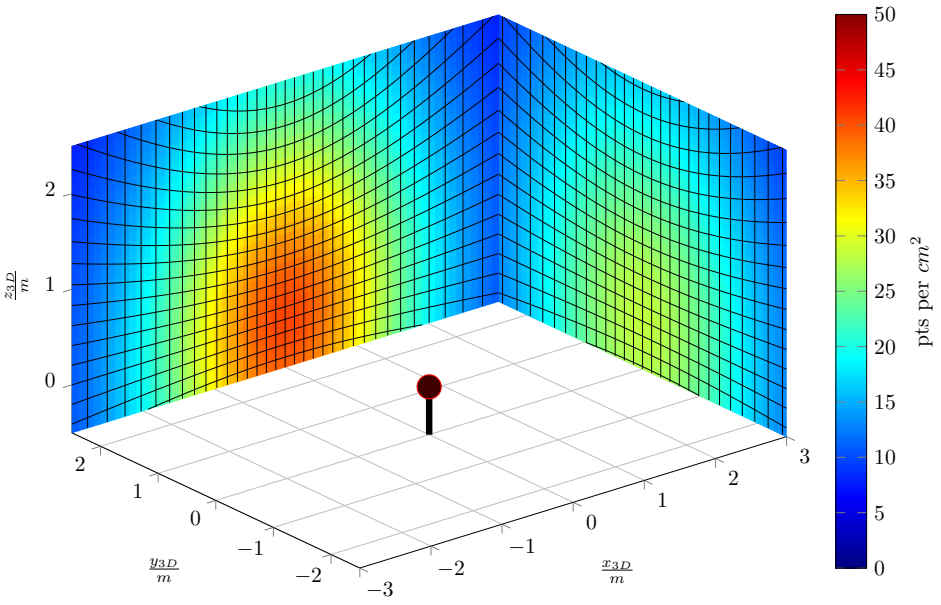
$$\begin{aligned}
A(\vartheta\prime,\varphi\prime) &= \tilde{y}^2\cdot\int_{\varphi\prime-\frac{\Delta\varphi}{2}}^{\varphi\prime+\frac{\Delta\varphi}{2}}\int_{\vartheta\prime-\frac{\Delta\vartheta}{2}}^{\vartheta\prime+\frac{\Delta\vartheta}{2}} \frac{1}{\cos^3\varphi\cdot\cos^2\vartheta}\; d\vartheta\; d\varphi\\
&= \tilde{y}^2\cdot\int_{\varphi\prime-\frac{\Delta\varphi}{2}}^{\varphi\prime+\frac{\Delta\varphi}{2}} \frac{1}{\cos^3\varphi}\cdot\tan(\vartheta)\Big|_{\vartheta\prime-\frac{\Delta\vartheta}{2}}^{\vartheta\prime+\frac{\Delta\vartheta}{2}}\; d\varphi\\
&= \tilde{y}^2\cdot\tan(\vartheta)\Big|_{\vartheta\prime-\frac{\Delta\vartheta}{2}}^{\vartheta\prime+\frac{\Delta\vartheta}{2}}\cdot\int_{\varphi\prime-\frac{\Delta\varphi}{2}}^{\varphi\prime+\frac{\Delta\varphi}{2}} \frac{1}{\cos^3\varphi}\; d\varphi\\
&= \tilde{y}^2\cdot\tan(\vartheta)\Big|_{\vartheta\prime-\frac{\Delta\vartheta}{2}}^{\vartheta\prime+\frac{\Delta\vartheta}{2}}\cdot\left(\frac{\sin(\varphi)}{2\cos(\varphi)^2}+\frac{1}{4}\ln\left|\frac{1+\sin(\varphi)}{1-\sin(\varphi)}\right|\right)\Bigg|_{\varphi\prime-\frac{\Delta\varphi}{2}}^{\varphi\prime+\frac{\Delta\varphi}{2}}
\end{aligned} \tag{A.12}$$

Due to the constant step size of the polar azimuth angle φ, the result holds not only for our particular plane ε, but for any upright plane with an arbitrary rotation around

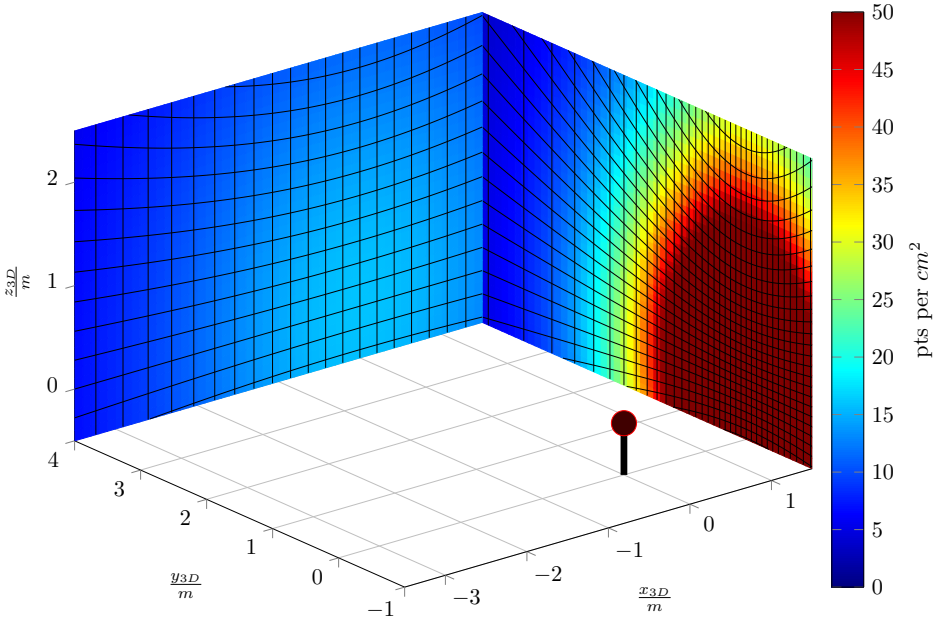
the z-axis. The inverse value of this function gives the expected scan resolution on a 2 dimensional plane that is scanned by a radial deflected laser beam in pts/m^2.

Via the derived formula we can calculate different acquisition setups for rooms consisting of upright, plane walls. Figure A.2 shows simulations of the achievable effective point cloud resolution on the wall surfaces for two different scenarios. The parameters of this simulations were matched to the reference scanner *Faro Focus 3D*, as described in Section 1.3. We assumed the scanners default noise reduction enabled, which reduces the number of measurements by a factor of 16. Obviously, the effective scan resolution of the projected equidistant points of a 3D polar coordinate system to a 2D plane varies significantly with the position on the wall. Furthermore, we can observe strong distortions of the spatial order of scan points in regions, where the scanner show a skewed viewpoint relative to the walls surface. Hence, although the scanner acquires a distance measure each $\Delta\varphi = \Delta\vartheta = 0.009°$ per deflection axis resulting in a total number of about 1.6 billion points, less dense sampled areas of simulation A.2a) feature only a resolution of about 6 to 7 pts/cm^2, or one point each 3 to 4 mm. A standard European electrical appliance providing a square area of about 25 cm^2 that is mounted in a less dense sampled corner, would got sampled by only about 13×13 points. In this simulation we assume the scanner to be placed in the exact center of the room.

The simulation shown in A.2b) is based on the real-world example of Figure 1.3. Due to the off-centered placement of the scanner, the effect of the scan grid distortion is much more distinctive, as illustrated by the equiangular lines. This distortion contributes to the minimum achievable resolution dropping down to 3.63 pts/cm^2, which yield a standard socket to be digitalized by only 90 points on average. Beside the loss of data density, the distorted scan raster leads to additional difficulties when interpreting the measurement data. In our example, a square area of 25 cm^2 in the least densest region would be sampled by about 6×15 points, instead of 9×9. This encourages the chance of missing important structure information.



a) Centered placed scanner



b) Off-centered placed scanner

Figure A.2: Simulation of the achievable scan resolution in 2 different scenarios with a LiDAR scanner. The round marker represents the center of the scanners measurement unit. In order to visualize the distortion of the scan point grid, the black lines indicate scan positions every 3° of equal azimuth and elevation angle values. The simulation of a) shows a room of size $6 \times 5\ m$, where the scanner is placed at the center of the floor. Towards the corners, the lowest achievable resolution on the walls decreases significantly. Figure b) simulates the scenario of a real example discussed in Section 1.3. The off-centered scanner position reinforces the loss of resolution at certain regions and additionally introduces strong distortions of the scanning grid.

Bibliography

[1] Bariya, P. and Nishino, K. (2010). Scale-hierarchical 3d object recognition in cluttered scenes. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1657–1664. IEEE. (page 9)

[2] Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359. (page 9)

[3] Bishop, C. and Lasserre, J. (2007). Generative or discriminative? getting the best of both worlds. 8:3–23. (page 13)

[4] Boser, B., Guyon, I., and Vapnik, V. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM. (page 24)

[5] Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140. (page 32)

[6] Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32. (page 30)

[7] Bustamante, L. and Gu, J. (2007). Localization of electrical outlet for a mobile robot using visual servoing. In *Electrical and Computer Engineering, 2007. CCECE 2007. Canadian Conference on*, pages 1211–1214. IEEE. (page 17, 18)

[8] Calonder, M., Lepetit, V., Strecha, C., and Fua, P. (2010). Brief: Binary robust independent elementary features. In *Computer Vision–ECCV 2010*, pages 778–792. Springer. (page 12, 21)

[9] Canny, J. (1986). A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8(6):679–698. (page 18)

[10] Chen, C., Liaw, A., and Breiman, L. (2004). Using random forest to learn imbalanced data. *University of California, Berkeley*, pages 1–12. (page 33, 43)

[11] Chua, C. S. and Jarvis, R. (1997). Point signatures: A new representation for 3d object recognition. *International Journal of Computer Vision*, 25(1):63–85. (page 9)

[12] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297. (page 25)

[13] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE. (page 19, 20, 24, 37)

[14] Dollár, P., Tu, Z., Perona, P., and Belongie, S. (2009). Integral channel features. (page 21, 29, 40)

[15] Dollár, P., Tu, Z., Tao, H., and Belongie, S. (2007). Feature mining for image classification. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE. (page 21)

[16] Donoser, M. and Bischof, H. (2006). Efficient maximally stable extremal region (mser) tracking. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 553–560. (page 18, 19)

[17] Dorai, C. and Jain, A. K. (1997). Cosmos-a representation scheme for 3d free-form objects. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(10):1115–1130. (page 10)

[18] Drost, B. and Ilic, S. (2012). 3d object detection and localization using multimodal point pair features. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*, pages 9–16. IEEE. (page 10, 18)

[19] Drost, B., Ulrich, M., Navab, N., and Ilic, S. (2010). Model globally, match locally: Efficient and robust 3d object recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 998–1005. IEEE. (page 10)

[20] Eruhimov, V. and Meeussen, W. (2011). Outlet detection and pose estimation for robot continuous operation. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 2941–2946. (page 17, 18)

[21] Fan, W., Stolfo, S. J., Zhang, J., and Chan, P. K. (1999). Adacost: misclassification cost-sensitive boosting. In *Icml*, pages 97–105. (page 27)

[22] Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645. (page 24)

[23] Freund, Y. and Schapire, R. E. (1995). A desicion-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, pages 23–37. Springer. (page 26)

[24] Geurts, P., Ernst, D., and Wehenkel, L. (2006). Extremely randomized trees. *Machine learning*, 63(1):3–42. (page 30)

[25] Hinterstoisser, S., Cagniart, C., Ilic, S., Sturm, P., Navab, N., Fua, P., and Lepetit, V. (2012). Gradient response maps for real-time detection of textureless objects. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(5):876–888. (page 10, 37, 38)

[26] Hinterstoisser, S., Holzer, S., Cagniart, C., Ilic, S., Konolige, K., Navab, N., and Lepetit, V. (2011). Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 858–865. IEEE. (page 10, 19, 37)

[27] Hinterstoisser, S., Kutter, O., Navab, N., Fua, P., and Lepetit, V. (2009). Real-time learning of accurate patch rectification. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2945–2952. (page 18)

[28] Holzer, S., Hinterstoisser, S., Ilic, S., and Navab, N. (2009). Distance transform templates for object detection and pose estimation. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1177–1184. (page 18)

[29] Hsu, C.-W. and Lin, C.-J. (2002). A comparison of methods for multiclass support vector machines. *IEEE transactions on Neural Networks*, 13(2):415–425. (page 26)

[30] Johnson, A. E. and Hebert, M. (1999). Using spin images for efficient object recognition in cluttered 3d scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(5):433–449. (page 9)

[31] Klein, D. A., Schulz, D., and Frintrop, S. (2009). Boosting with a joint feature pool from different sensors. In *Computer Vision Systems*, pages 63–72. Springer. (page 21, 29)

[32] Kreßel, U. H.-G. (1999). Pairwise classification and support vector machines. In *Advances in kernel methods*, pages 255–268. MIT Press. (page 26)

[33] Krispel, U., Evers, H. L., Tamke, M., Viehauser, R., and Fellner, D. W. (2015). Automatic texture and orthophoto generation from registered panoramic views. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-5/W4:131–137. (page 6, 11, 12, 52)

[34] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105. (page 8)

[35] Lin, Y., Lee, Y., and Wahba, G. (2002). Support vector machines for classification in nonstandard situations. *Machine learning*, 46(1-3):191–202. (page 26)

[36] Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee. (page 9, 12)

[37] Masnadi-Shirazi, H. and Vasconcelos, N. (2011). Cost-sensitive boosting. *IEEE Transactions on pattern analysis and machine intelligence*, 33(2):294–309. (page 27)

[38] Matas, J., Chum, O., Urban, M., and Pajdla, T. (2004). Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 22(10):761–767. (page 18)

[39] Meeussen, W., Wise, M., Glaser, S., Chitta, S., McGann, C., Mihelich, P., Marder-Eppstein, E., Muja, M., Eruhimov, V., Foote, T., et al. (2010). Autonomous door opening and plugging in with a personal robot. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 729–736. IEEE. (page 17, 18)

[40] Mian, A. S., Bennamoun, M., and Owens, R. (2006). Three-dimensional model-based object recognition and segmentation in cluttered scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(10):1584–1601. (page 10)

[41] Ng, A. Y. and Jordan, M. I. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In Dietterich, T. G., Becker, S., and Ghahramani, Z., editors, *Advances in Neural Information Processing Systems 14*, pages 841–848. MIT Press. (page 13)

[42] Papageorgiou, C. P., Oren, M., and Poggio, T. (1998). A general framework for object detection. In *Computer vision, 1998. sixth international conference on*, pages 555–562. IEEE. (page 21)

[43] Raskutti, B. and Kowalczyk, A. (2004). Extreme re-balancing for svms: a case study. *ACM Sigkdd Explorations Newsletter*, 6(1):60–69. (page 26)

[44] Schapire, R. E. and Singer, Y. (1999). Improved boosting algorithms using confidence-rated predictions. *Machine learning*, 37(3):297–336. (page 27)

[45] Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., and Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471. (page 26)

[46] Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., and Blake, A. (2011). Real-time human pose recognition in parts from single depth images. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1297–1304. (page 30)

[47] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*. (page 8)

[48] Sun, Y., Kamel, M. S., Wong, A. K., and Wang, Y. (2007). Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognition*, 40(12):3358–3378. (page 27)

[49] Sun, Y., Paik, J., Koschan, A., Page, D. L., Abidi, M., et al. (2003). Point fingerprint: a new 3-d object representation scheme. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 33(4):712–717. (page 9)

[50] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9. (page 8)

[51] Torralba, A., Murphy, K. P., and Freeman, W. T. (2004). Sharing features: efficient boosting procedures for multiclass object detection. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–II. IEEE. (page 29)

[52] Torralba, A., Murphy, K. P., and Freeman, W. T. (2007). Sharing visual features for multiclass and multiview object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(5):854–869. (page 29)

[53] Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511. IEEE. (page 21, 29)

[54] Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., and Torralba, A. (2010). Sun database: Large-scale scene recognition from abbey to zoo. In *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*, pages 3485–3492. IEEE. (page 52)

[55] Zhu, J., Zou, H., Rosset, S., and Hastie, T. (2009). Multi-class adaboost. *Statistics and its Interface*, 2(3):349–360. (page 27)

[56] Zhu, Q., Yeh, M.-C., Cheng, K.-T., and Avidan, S. (2006). Fast human detection using a cascade of histograms of oriented gradients. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1491–1498. IEEE. (page 21)