Die approbierte Originalversion dieser Diplom-/Masterarbeit ist in der Hauptbibliothek der Technischen Universität Wien aufgestellt und zugänglich.
http://www.ub.tuwien.ac.at

TU UB WIEN Universitätsbibliothek

The approved original version of this diploma or master thesis is available at the main library of the Vienna University of Technology.
http://www.ub.tuwien.ac.at/eng

FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Random Forest Classification for Fast Multi-class Object Detection using Intensity and Depth Information

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieurin

im Rahmen des Studiums

Visual Computing

eingereicht von

Shu Zhu
Matrikelnummer 0348503

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung
Betreuer: a.o.Univ.-Prof. Dipl.-Ing. Dr.techn. Robert Sablatnig
Mitwirkung: Dipl.-Ing. Dr. Csaba Beleznai

Wien, 05.01.2015

(Unterschrift Verfasser/in) (Unterschrift Betreuer/in)

Erklärung zur Verfassung der Arbeit

Shu Zhu
Grundsteingasse 22, 1160 Wien

„Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.“

Wien, am 05.Januar 2015

Acknowledgements

I owe my deepest gratitude to my supervisor, Csaba Beleznai, for his constant support during the entire course of the development of this thesis. Without his help this thesis would hardly have been completed. I would like to thank Robert Sablatnig for his great feedbacks. I want to thank my colleagues Tim Lammarsch, Ruslan Kamolov, Alistair Jones, Maximilian Lang for sharing their valuable knowledge.

This work is dedicated to my parents and dear friend Markus Laumann who has always believed I could do it.

Work towards this diploma thesis has been partially supported by the FastPass project. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 312583.

Abstract

Object detection is an important problem in the field of computer vision. The main task of object detection is to locate certain objects in image sequences or videos in terms of image coordinator and an estimated scale. This work describes a discriminatively formulated object detection scheme which extracts features from a training set and learns a discriminative classifier to recognize instances of an object category. The feature types, which are also known as object representations, are a key factor to affect the performance and accuracy of an object detection system, thus the choice of features or feature design remain an important research field. Most state-of-the-art object detection systems use features from intensity images. However, developments in sensing technology in the last five years enable inexpensive ways of acquiring depth data, which provide additional cues for object detection, such as scale, depth ordering and foreground-background segmentation. In this work we propose the detection of humans from the combined modalities of intensity and depth, the latter obtained from a passive stereo setup. In order to derive informative features from depth, we use a filtering and interpolation mechanism which substantially addresses the missing data and noise problems which are commonly present in stereo depth data. Furthermore, we propose a scale estimation and validation scheme which successfully suppresses inconsistent detection responses not matching the local depth data. We adopt the ACF strategy of combining multiple features in a single detector. We use a set of features (gradient magnitude and gradient histogram) extracted from the depth data in addition to features from the intensity images to train our RGBD detectors. The system developed is capable to accommodate multiple object classes. The system is tested on two scenarios containing crowded and cluttered situations. We show that using additional features from the depth data improves the detection accuracy, especially in presence of clutter and occlusions.

Kurzfassung

Objekterkennung ist ein wichtiges Problem auf dem Gebiet der Computer Vision. Die Hauptaufgabe der Objekterkennung ist es, bestimmte Objekte in Bildsequenzen oder Videos in Bezug auf Bildkoordinator und geschätzten Umfang (Scale) zu lokalisieren. Diese Arbeit beschreibt ein diskriminativ formuliertes Objekterkennungssystem, das Merkmale aus einem Trainingsset extrahiert und lernt, einen diskriminativen Klassifikator zu Instanzen einer Objektkategorie zu erkennen. Merkmale, die auch als Objektdarstellungen bekannt sind, sind ein Schlüsselfaktor, um die Performance und Genauigkeit eines Objekterkennungssystems zu beeinflussen. Die Auswahl der Merkmale und deren Design sind ein wichtiges Forschungsfeld. Die meisten State-of-the-Art- Objekterkennungssysteme verwenden Merkmale von Intensitätsbildern. Neue Entwicklungen der Sensortechnologie in den letzten fünf Jahren ermöglichen jedoch einen kostengünstigen Erwerb von Tiefendaten, die zusätzliche visuelle Hinweise zur Objekterkennung bieten, wie zum Beispiel Scale, Depth-Ordering und Foreground-Background-Segmentierung. In dieser Arbeit schlagen wir die Erkennung von Personen aus kombinierten Intensitäts- und Tiefendaten vor, wobei letztere aus einem passiven Stereo-Setup erhalten werden. Um aussagekräftige Features aus Tiefendaten abzuleiten, verwenden wir einen Filterungs- und Interpolationsmechanismus, der sich mit fehlenden Daten und mit Noise-Problemen befasst, die gewöhnlich in Stereo-Tiefendaten vorkommen. Außerdem schlagen wir eine Schätzung von Scale-Daten und ein Validierungsschema vor, die erfolgreich inkonsistente Erkennungsreaktionen unterdrücken, die nicht zu den lokalen Tiefendaten passen. Wir übernehmen hier die ACF-Strategie der Kombination mehrerer Merkmale in einem einzigen Detektor. Wir verwenden eine Reihe von Merkmalen (Gradientenmagnitude und Gradientenhistogramm), die aus den Tiefendaten extrahiert wurden, zusätzlich zu den Merkmale aus den Intensitätsbildern, um unsere RGBD Detektoren zu trainieren. Das entwickelte System ist in der Lage, mehreren Objektklassen Rechnung zu tragen. Das System wurde für zwei Szenarien getestet, nämlich in Situationen mit Okklusion und Unordnung. Wir zeigen, dass die Verwendung von zusätzlichen Merkmalen aus den Tiefendaten die Erkennungsgenauigkeit insbesonders in unübersichtlichen Situationen verbessert.

Contents

1 Introduction.. 1

1.1 Problem Definition 1
1.2 Related Literature 3
1.3 Aim and Contributions 5
1.4 Structure 5
1.5 Summary 6

2 Theoretical Background... 7

2.1 Representations and Feature Types 7
  2.2.1 The Pixel and Color Representation 8
  2.2.2 Image Gradients 8
  2.2.3 HOG and the Integral Histogram Histogram 10
  2.2.4 Overcomplete Object Representations and Feature Combination 11
2.2 Fast Feature Pyramid Computation 11
2.3 AdaBoost and Random Forest 13
2.4 Summary 16

3 Datasets..17

3.1 Dataset Description 18
3.2 The Ground Truth Annotation 20
3.3 Summary 23

4 The RGBD Detector: Incorporation of the Depth Cues..24

4.1 The Depth Map Enhancement 25
  4.1.1 The PDE Based Methods 25
  4.1.2 The Joint Bilateral Filter 26
  4.1.3 Comparison of Different Filling Methods 27
4.2 Depth Channel Computation 29
  4.2.1 Depth Gradient Magnitute 29
  4.2.2 Gradient Histogram 30
  4.2.3 The Integral Image 31
4.3 Reduction of the Feature Dimensionality via Gray Level Quantization 32
4.4 Fusion Schemes for the Depth Features and the RGB Features 34
  4.4.1 The RGB Detector 34
  4.4.2 The Aggregated Channels Approach 35
  4.4.3 The Multiple Classifiers Approach 36

4.5 Scale Estimation Using the Depth Information 37
    4.5.1 The Scene Geometry for the Scale Prior 38
    4.5.2 Object Scale from the Reciprocal of the Depth Value 39
4.6 Summary 41

5 Experiments and Results **43**
5.1 Evaluation Protocol 43
5.2 Test Case 1: The RGBD Detector withthe Low-level Feature Fusion Scheme 45
    5.2.1 Ground Truth Filtering 45
    5.2.2 Setting the Treshold for Comparing Two BBs 49
    5.2.3 Comparison of the Detectors 52
    5.2.4 Discussion of the Comparison and Figures 54
5.3 Test Case 2: Multi-class Detection with the High-level Feature Fusion Scheme 57
5.4 Summary 60

6 Conclusion **61**

Bibliography 62

Chapter 1
Introduction

Object detection in computer vision is the task of finding instances of objects in image sequences or videos [66]. To differentiate the two very similar and close related terms *object detection* and *object recognition*, object detection means to find the objects of a certain class, while object recognition means to identify unknown objects and to label their classes [66], [55]. For example one could imagine the picture of a kitchen scene. The object recognition problem can be formulated as, which kind of furniture is in this scene and where is it? The answer can be, for instance, a table in the middle and four chairs around the table. The object detection problem can be formulated as, where are all the chairs? Object detection is related with the task of object tracking by trying to recover the movement of individual objects across consecutive frames [54], [34]. With the information of the object locations, the system may pursue them over time as they move [54], [34].

Object detection has various application fields, varying from image retrieval, video surveillance, robotics to automotive safety, and industrial machine vision [55].

There are many challenges for object detection due to the following ambiguities: illumination effects (shadow, reflection), data noise, low resolution, and partial or full occlusions [36]. Another challenging problem is that each object in the 3D world space can present countless forms in 2D images as its position, pose, lighting conditions change [66], [55]. Figure 1.1 depicts some of the difficulties in object detection. Irrelevant image parts may have similar appearance to the target objects. Occlusion, reflection and shadow are contained in the scene.

1.1 Problem Definition

Learning based object detection considers the object detection problem in a statistical point of view, and transforms it to the classification problem [86], [66].

The principle idea of the learning based approach is to form a classifier using a set of training data, which is then used to classify new input data [66], [18]. For object detection problems, objects are represented by their local features from small patches of the images describing small patches of the images describing local pixel neighborhood [66]. Image patches are those small windows of subimages taken from an image. The patches may overlap, which means that some parts of the images are captured more than once. Each patch is marked as either belonging to the foreground or to the background. Local patches are robust to appearance/shape variations and occlusion because the object is represented by a set of subparts, therefore an occluded object can still be recognized as long as some of its parts are recognized. The limited information of small patches requires that the patches must be sufficiently variable and the amount of patches must be

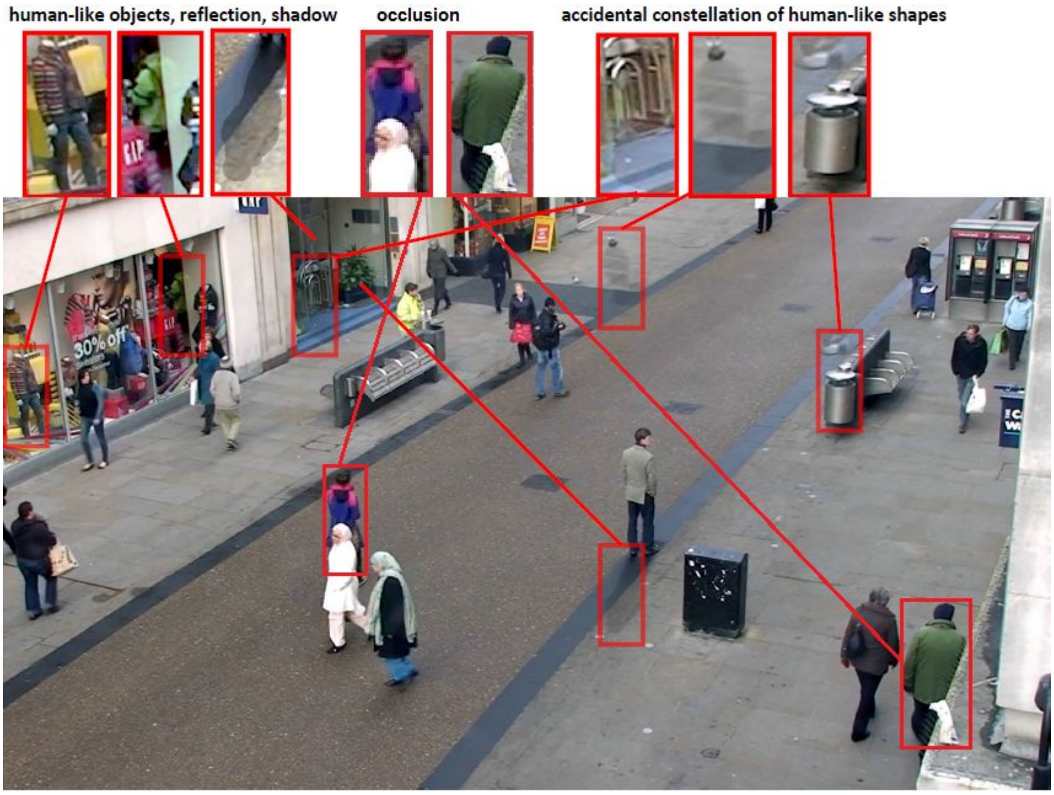

human-like objects, reflection, shadow
occlusion
accidental constellation of human-like shapes

Figure 1.1: A street scene illustrating some of the many difficulties encountered in visual processing. Figure is from [6].

sufficiently large so that the ensemble of features preserves the specificity of the object class. Probability theory provides a powerful framework for combining the partially ambiguous information from individual patches together in a principled manner [85]. From these small patches features are extracted and fed to the learning process, and finally a classifier is trained.

Depending on how the distribution of the image features are modelled, the learning based approach can further be separated into two categories, generative or discriminative approaches [85]. A generative model learns the joint probability distribution $p(x,y)$, where x denotes the input, i.e. the observed visual features, and y denotes the label, i.e. the presence of an object. A discriminative model learns the conditional probability distribution $p(y|x)$ - "the probability of y (presence of an object) given x (observed visual features)". For the discriminative models, when the conditional probability distribution $p(y|x)$, the dependence of the unobserved variable y from the observed variable x, is modelled, then y can be predicted based on the knowledge of x. For generative models, the joint probability can be transformed to the conditional probability by Bayes rule, modeling the probability of objects in presence of certain observed image features.

Generative and discriminative methods have complementary strengths and weaknesses. In [85] it is concluded that although the discriminative learning has lower asymptotic error, a generative classifier approaches its higher asymptotic error much faster. Recent research makes effort to combine the two in the best way [47], [50], [61]. Nevertheless, in classification tasks discriminative models can yield superior performance, and generally outperform the generative models [50].

Most discriminative models require supervised learning [66], [18]. Ground truth is needed to tell the system which image part is the foreground and which is the background, this is done by hand-labelling of annotations on images. The basic workflow of a discriminative object detection system is typically as follows: features are extracted from local patches from both the positive and negative training sets, methods from machine learning are applied for forming an implicitly learned decision function, i.e. a classifier, for example Support Vector Machine (SVM) [66], Random Forest [83], Neural Networks [89]. The trained classifier can thereafter be used to discriminate the target object class from the background in a 2D image [66], [18], [41].

Features, also known as object representations, are a key factor to the accuracy and performance of a discriminative system [15]. Considerable progress has been made focusing on features, schemes are developed for feature design [12], statistical feature selection [86], and sophisticated feature calculation [15].

Multiple types of features can be fused together because observation is made that even poorly performing classifiers may be improved by complementing feature characteristics [21], [66]. Overcomplete object representations and enriched features (e.g. multiple types of features, fine sampling scales, selection from feature sets many times larger than the image space) have proven to improve the detection accuracy dramatically in comparison with features with complete basis and of single types [66], [86], [20].

Not only different types but also different modalities of features, such as intensity, motion, stereo, can be combined together to achieve an improved accuracy [70]. Due to the increased popularity of inexpensive RGBD sensors, there is increased interest among researchers to extract meaningful features from the depth data, to exploit complementary information to the RGB data and provide additional cues for object detection [70], [41], [96], [79], [46], [11], [36]. In this work an object detection scheme is proposed which uses the combination of intensity and depth information for object detection.

1.2 Related Literature

In the last two decades discriminative object detection has been an active research field and has undergone a breathtaking progress [66], [86], [20], [12]. In the year 2000 Papageorgiou et al. presented a trainable system for object detection, using an overcomplete dictionary of Haar-wavelets as object features, which effectively defines a descriptive model for an object class and gains a high detection rate [66]. Partly motivated by Papageorgiou et al. Paul Viola and Michael Jones [86] proposed the first real time object detection framework. Since then, false positive rates have decreased by two orders of magnitude for common object recognition benchmarks such as the PASCAL Visual Object Recognition Challenge [20].

Gradient-based features are widely studied and used in modern detectors for intensity images due to their invariance to illumination. In addition, gradients are descriptive for many object classes, for example humans have a distinctive silhouette shape which majorly comprises vertical gradients. Papageorgiou et al. use the Haar-wavelet transform to compute local, oriented, multiscale differences between adjacent regions. The wavelet features not only preserve all the information from the original image, but also encode the intensity differences at different scales. The uniform areas are encoded as zero, and the strong differences of intensities, or boundaries, are encoded as non-zero. A denser (redundant) transform, the quadruple density wavelet transform, is used to produce an overcomplete dictionary of features [66]. SIFT follows a similar line of thought by being a sparse representation of local gradient orientation histograms, a highly distinctive local feature which is invariant to image scale and rotation and is an effective descriptor for objects under partial occlusion [56]. SIFT has proved to be successful in object recognition and object detection [56], [63], [28]. Variations and extensions of SIFT are developed, such as PCA-SIFT [40], Gradient Location and Orientation Histogram (GLOH) [62], and Histogram of Oriented Gradients (HOG) [12]. In contrast to SIFT, HOG is a dense representation of local gradient orientation histograms on uniformly spaced cells with local contrast normalization [12]. Variations of HOG are for example HOG-LBP [88], GF-HOG [38], gradient histogram [15]. Zhu et al. propose a fast way of computing HOG features using integral images [101]. HOG is extensively applied in object detection, especially in human detection [12], [15], [20], [87], [101].

Regarding the learning algorithms for discriminative models, SVM is a popular classifier widely used in various object detection systems [66], [41], [70]. AdaBoost is able to select a small set of critical features from a large feature pool [100], [81], which is essential for rapid computation. Viola and Jones design a finer feature selection scheme based on AdaBoost, i.e. a cascade structure in the boosting process to reject the negative instances at an early stage [86]. Wu et al. propose a forward feature selection scheme which further improves the cascade learning algorithm of Viola and Jones [94]. Other widely used learning algorithms for object detection systems are discriminant analysis [45], [23], [99], Random Forest [30], and Neural Networks [71], [8].

Different sets of features can be used to train different classifiers separately, or be aggregated to train a single classifier. Researchers have proposed various ways of combining features and classifiers, from low-level fusion of joint feature space to high-level fusion at the classifier-level [21], [70], [41]:

- Parallel combination of classifiers: Classifiers (often of the same type) trained with different feature sets are combined together. The features sets can be subsets of an original feature set, with reduced dimensionality.
- Stacked combination of classifiers: Classifiers of different types (e.g. Neural Networks, nearest neighbour, or parametric decision rule) trained with a same feature set are combined together.
- Combination of weak classifiers: Simple classifiers are trained and combined together. Approaches for this purpose are for example bootstrapping (bagging), boosting and random subspaces. The ACF framework uses bootstapping and boosting, as introduced in Chapter 2.
- Joint feature space: Different kinds of features are combined together to form a descriptive object representation, for example the human detector proposed by Dalal et al. combines HOG based static appearance descriptor with motion based descriptor [13].

Features from other modalities, e.g. depth, motion, can be incorporated into detectors at different levels [18], [70]. On *module-level*, the additional cues can help to reconstruct the scene geometry or generate regions of interest for the subsequent classification step [41]. On *feature-level* the image features from depth or motion are fused with features from intensity, the classifier is learned within this joint feature space [20], [16]. On *classifier-level* features from each modality train a corresponding classifier respectively, in the detection step the outputs of the classifiers are combined together to make a final decision [70], [41]. In terms of detectors with the depth cue, HOG and its variations (in combination with other types of features) are the most popular features for such systems [41], [70], [96], [46], [11].

1.3 Aim and Contributions

The objective of this work is the reliable detection of humans from RGB and depth images in near real-time. Our work includes:

- We design a discriminatively formulated object detection scheme which enables large sets of image n-tuples (RGB+disparity, but even for the more general case, such as RGBD-thermal-motion modalities) to be imported into the system and represented in variables compatible with the existing data representations.
- The processing of the depth data includes data enhancement, annotation, feature extraction, and scale prior estimation. The RGB and depth data are used to support each other, for example the RGB data is used to guide the inpainting of the depth data to preserve the edges of objects, while the depth features are fed to the system in addition to the RGB features in order to enrich the feature pool and provide complementary information.
- The scene information contained in the depth maps can be further exploited in means of estimating a scale prior for the objects. Detected objects whose sizes do not fall into a certain range according to the scale prior will be rejected. Thus we use the depth data to improve the detection result in two ways, the complementary information from the depth maps shall increase true positives, and the scale prior shall reduce false negatives.
- We design two types of detectors which perform the feature fusion at different levels, one applies a low-level fusion of the RGB features and the depth features to form a joint feature space, the other trains a RGB classifier and a depth classifier respectively and combine the detection results together. We evaluate the two RGBD detectors on two scenarios with different variable settings, both detectors achieve improved accuracy in presence of clutter, occlusions and across a wide range of appearance/pose variations in comparison with the RGB detector.
- Common public datasets such as the Daimler Dataset [33] was of very poor quality (in comparison with the INRIA Dataset [12]) focusing on remote, blurred pedestrians as well as in automotive conditions, therefore, an own training and testing dataset had to be created.
- Evaluation was a substantial work, given the large space of possible explorations (representation, fusion type, etc.).

We consider our contribution to be a successful experiment of incorporating the depth cues into conventional detectors which are based on intensity images, and show the potential of new detectors using multiple sensory inputs.

In order to demonstrate the enhanced representational power of combined RGB-D features, we use the fast ACF-features and classification methods from P.Dollár [20], [17] as a basis. This framework does not provide means to add additional modalities, therefore we extended the image I/O and feature computation functions to make them capable of fast RGB-D based classification.

1.4 Structure

The chapters of this diploma thesis are arranged as follows. Chapter 2 introduces the theoretical background and important concepts for a discriminative object detection system. Chapter 3 describes the data sets used in our work. Chapter 4 explains the implementation of the RGBD detector. Chapter 5 describes the test cases and the experimental results, including a detailed description of the evaluation methodology. Chapter 6 gives a conclusion of the main results of this thesis and discusses ideas for future work.

1.5 Summary

In this chapter we give an introduction to the challenges of object detection. We propose a novel multi-modality detector to address some of these difficulties, by integrating additional visual cues into the traditional intensity based detectors. We conduct a literature review of popular features and learning algorithms for state-of-the-art discriminative object detection systems, and methods for combining features from different modalities. We further explain our aim and contributions, followed by a description of the structure of this work.

Chapter 2
Theoretical Background

This chapter gives an introduction of the theoretical background and important concepts for designing a discriminative object detection system. As stated by Piotr Dollár, the learning algorithm and the feature representation are the two key factors to influence the performance of object detection systems [15]. In the following sections more details on these two aspects will be introduced.

This chapter is arranged as follows: Section 2.1 describes the feature types used in our system. Section 2.2 introduces the fast feature pyramid computation method. Section 2.3 gives a description of the learning algorithms AdaBoost and Random Forest. Section 2.4 gives a summary of this chapter.

2.1 Representations and Feature Types

This section introduces the features used in our system. Features, or object representations, are the fundamental concept in designing a discriminative object detection system [66], [15]. Descriptors from different representational concepts have varying degrees of specificity and invariance [10]. As shown in Figure 2.1, specificity and invariance are related in an inverse manner. On the right end of the graph are features based on appearance models, such as image bitmaps. These features are highly specific but sensitive to noise, view, and shape variations. The left end of the graph exhibits features based on aggregated statistics, such as histograms. These features are highly invariant but typically lack the desired extent of specificity. Simultaneous increase of specificity and invariance, as well as the computational efficiency have been a focus of substantial research [22], [2].

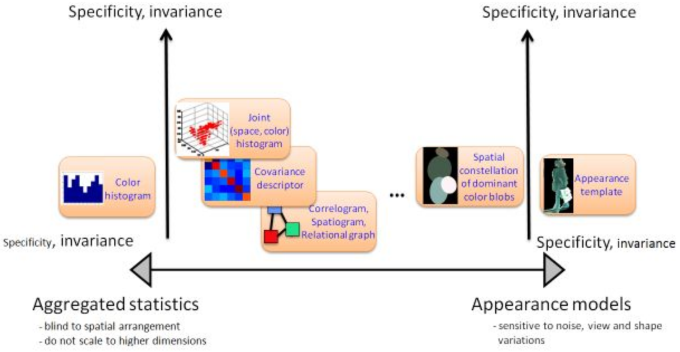

Specificity, invariance
Specificity, invariance
Joint (space, color) histogram
Color histogram
Covariance descriptor
Correlogram, Spatiogram, Relational graph
...
Spatial constellation of dominant color blobs
Appearance template
Specificity, invariance
Specificity, invariance
Aggregated statistics
- blind to spatial arrangement
- do not scale to higher dimensions
Appearance models
- sensitive to noise, view and shape variations

Figure 2.1: Descriptors varying in the degrees of specificity and invariance. Figure is from [5].

2.1.1 The Pixel and Color Representation

A pixel representation in terms of spatial scale is the most local feature which does not capture the mid- and high-level features of the object. For example, image structures like boundaries and edges can not be observed by single pixels. However, the pixel representation is nevertheless used in various object detection systems where local features are needed, for example the face detection [66].

A pixel can either be represented as a grayscale intensity value, or as color channels. The color channels contain much richer information than the grayscale values. RGB, LUV, LAB, XYZ, YUV are the most common color spaces for color coding, and there are also many other color spaces for specific purposes, for example YIQ, HSV and HIS, GLHS (generalized LHS), and so on [98]. The RGB color space is the default color space for most image formats, and is widely used in color image processing, because other color spaces can be obtained from transformation from RGB. Using the RGB color space has the advantage of requiring no color space conversion. However, different color models are carefully chosen by researchers for different applications [98].

In our work the CIELUV color channels are part of our feature set, because the CIELUV color model decouples luminosity from color, and thus achieves improved invariance with regard to brightness variations [32], [82]. The CIELUV color space was specifically designed for the purpose of perceptual uniformity, which means the perceptual difference between any two colors, in other words the appearance of two colors observed by humans, is represented by the Euclidean distance [32]. The luminance (L) and the chroma (uv) are obtained through a non-linear mapping of the XYZ coordinates [82].

The CIELUV color space is widely used for applications in computer graphics [48]. In object detection, it is often used for skin detection and face detection [9], [51], [66]. It is found that the RGB color space is not suitable for constructing accurate skin color models due to the high correlation between the three components, instead the CIELUV model is often used for the skin color modelling [95]. For example, Lestideau et al. [51] design a face detection algorithm with CIELUV as their predetermined colour space. Cai et al. [9] point out that despite the difference in appearance of people's skin colors, the major difference lies in the intensity rather than in color itself. They use UV values of the CIELUV color space to model the skin colors from more than five hundred images of human skins of different races, while the luminance value is abandoned. They find that the distribution of skin color can be modeled by a Gaussian distribution.

2.1.2 Image Gradients

Dollár et al. [18] state in their survey, that substantial progress of pedestrian detection has been made by the adoption of gradient-based features. Therefore we use two gradient-based features, normalized gradient magnitude and histogram of oriented gradients in our system.

The image gradient is a fundamental concept in computer vision, it is an important image feature for object detection algorithms [56], [12]. The image gradient is computed as a 2D vector of image derivatives in the x and the y directions:

$$\nabla I = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right) \tag{2.1}$$

The image gradients measure how the pixel values of the image are changing. The direction of a gradient indicates the direction in which the image is changing, and the magnitude of a gradient indicates how rapidly the image is changing in this direction.

One of the most common uses of image gradients is edge detection [57]. The gradient magnitude is computed at each pixel of an image, after setting threshold the pixels which have the largest gradient magnitudes are detected as possible edge points. The gradient direction and gradient magnitude can further help to do thinning of the edges or trace the edges in the direction perpendicular to the gradient directions. The image gradient is also used in various algorithms for computer vision, for example feature matching [56], corner detection methods [43], [26] background subtraction [39], the watershed method for image segmentation [78], etc.

Image gradients are used in our system because, firstly, they help to extract the high frequency information from an image, i.e., the contours, boundaries and shapes, as explained in the above paragraph. Secondly, image gradients are more illumination invariant than color values, because the pixels may have completely different values due to different lighting conditions, while the image gradients, encoded by the pixel differences rather than the pixel values themselves, are more robust to lighting and camera changes. Figure 2.2 shows that the gradient-based operators have the property of illumination invariance and the ability to help capture the high frequency information.

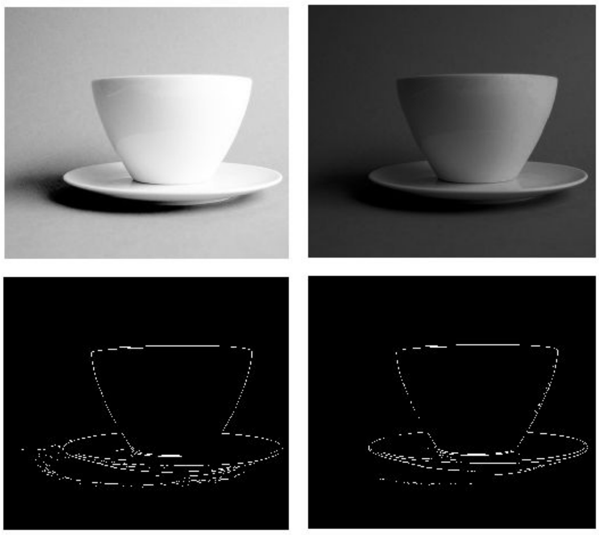


Figure 2.2: The same cup under different illumination has different local pixel intensities, however, the edge detector yields similar contour shapes. The edge images are computed by Matlab Sobel edge detector.

The image gradient is also a basic element for designing other robust feature descriptors. For example, the Scale Invariant Feature Transform (SIFT) proposed by Lowe [56] is designed to detect and describe local features in images, and achieves scale invariance, rotation invariance and affine invariance. The Histograms of Oriented Gradient (HOG) is a basic underlying concept of SIFT, which has proven to be one of the most powerful features in object detection [12], [15], [20], [87], [101]. We introduce more details on HOG in the next section.

2.1.3 HOG and the Integral Histogram

Histogram of Oriented Gradients (HOG) is proposed by Navneet Dalal and Bill Triggs in the year 2005. HOG is a feature descriptor similar to SIFT, with the basic idea that the distribution of local gradients can well describe the object appearance. To obtain the HOG features of an image, the image is tiled into small grids of the same size, called cells. The gradient of each pixel within a cell is computed, and a histogram of the gradient orientations of the pixels is formed. The gradient orientation of each pixel makes a vote to the corresponding bin of the histogram, and each vote is weighted by the gradient magnitude of that pixel. The bins of the histograms are spaced over either 0-180 or 0-360 degrees. Dalal and Triggs find that it works best for the human detection task when the gradient orientations are quantized into N bins evenly spaced over 0-180 degrees. The feature vector is obtained by concatenating all these histograms. To achieve a better accuracy against the local illumination variations or image contrasts, cells are grouped into larger regions, called blocks, and local contrast normalization is performed over all the cells belonging to the same block, by a measure of intensity values of that block. The blocks are overlapping, so that a cell is normalized differently when it belongs to different blocks, and this improves the performance [12]. Figure 2.3 illustrates the principle of the HOG feature extraction.

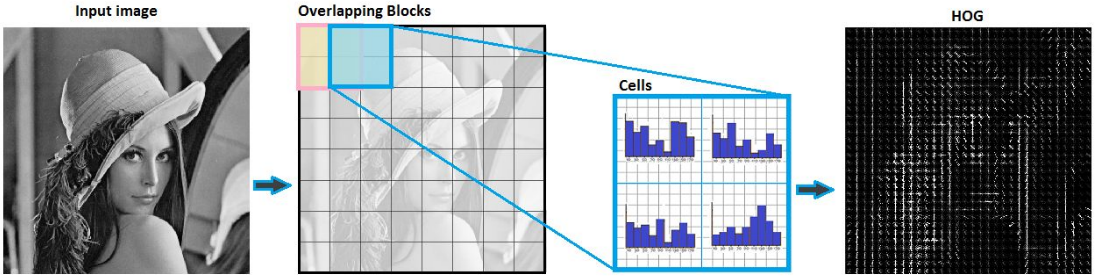


Figure 2.3: HOG feature extraction. The image is divided into overlapping blocks, while each block consists of a set of cells of the same size. The image gradients within each cell form a histogram of gradient orientations. The bins of each histogram are concatenated into a 1 D feature vector. Contrast normalization is performed over the blocks. The feature image is from [77], the illustration of the feature extraction is adapted from [29].

HOG outperforms the wavelet feature, and other existing edge and gradient based descriptors [12]. It differs from the sparse features like SIFT in a way that it is computed over dense grids and uses contrast normalization to achieve the local invariance to geometric and photometric transformations. The first advantage of HOG is that it uses fine-scale gradients, which are especially descriptive for the image information. Dalal and Triggs state that smoothing of the image before the gradient calculation will affect the effectiveness of the resulting HOG feature, because the fine-scale edges represent the most of the image information. The second advantage of HOG is the local invariance to geometric and photometric transformations, which makes it a robust descriptor for shape-based objects, and particularly suited for human detection, because it can ignore the body movements of the pedestrians as long as they maintain roughly upright poses. In the past decade, HOG has been a popular descriptor for the researchers to train the human models in their human detection algorithms [68], [101], [65].

2.1.4 Overcomplete Representations and Feature Combination

In the last decade enriched features, for example overcomplete representations and multiple feature types, are favored by researchers in object detection, because an enriched feature set improves the detection accuracy in comparison with representations of complete basis and single feature types [86], [66], [20].

In signal processing, the signal representation often uses overcomplete bases [53]. The complete representation uses a unique combination of complete bases which have no linear dependence between each other, while the overcomplete representation uses a combination of a subset of the available bases with the number of bases exceeding the dimensionality of the input signal [53]. There are several advantages offered by overcomplete bases: Firstly, it has more flexibility in capturing the structure of the data, because each basis function can capture certain structure in the data. Secondly, it has more robustness against noise, because a complete basis can only describe the data when no noise is introduced. With the presence of noise more bases are needed to express the property of the data. Overcomplete bases can be obtained by combining a set of complete bases, or by adding basis functions to an existing complete basis [53].

In image processing, the amount of complete bases equals the image dimension [86], [66]. When the number of extracted features is greater than the image dimension, they are called overcomplete features, or overcomplete dictionaries [86], [66]. Overcomplete features can be obtained by oversampling the scale, orientation and other kernel properties of multi-resolution or multi-orientation decomposition operations like Fourier, wavelet, and Gabor [20]. For example, in the HOG computation, using overlapping blocks is a method to obtain overcomplete features. Overcomplete features have been used in various research areas of visual computing. For example, the object detection system by Viola and Jones [86] computes 180000 thousand features from each 26×26 image. Lienhart et al. [54] use a set of rotated Haar-like features, which produces up to 117941 features from each 24×24 window. Because using the overcomplete bases can provide more stable and more robust signal representations, in image processing such enriched representations using an overcomplete set of image features can achieve robustness in encoding the image information [20]. Moreover, various types of features can be combined to form an enriched object representation [20], for example research has been done to combine HOG with two to five other features which contain complementary information (e.g. Haar-like features, motion features) [18]. Studies have shown that in a combination of features or classifiers, even poorly performing features or classifiers may contain valuable information to improve the system performance [21], [66].

An overcomplete feature set contains redundancy. How to select the best set of features from the great amount of feature candidates is a big challenge [20]. AdaBoost has proven to be an efficient solution for feature selection [86], [75]. We will study the AdaBoost algorithm in Section 2.3.

2.2 Fast Feature Pyramid Computation

An image pyramid is a collection of images of gradually reduced resolutions [20]. An image pyramid is necessary because the same objects contained in a set of images may appear in different sizes, and objects of different sizes contain different amount of features. For example, if a pedestrian is captured at a far distance, he/she appears as a small object in the image, and the most significant feature is his/her contour. On the contrary, if he/she is captured at a near distance, the object in the image reveals more details like neck, waist, clothes textures, etc.

The image pyramid is constructed by successively downsampling the images at different scales from a single original image. There are two types of image pyramids commonly used in image processing, the Gaussian pyramid, which applies the Gaussian blur to smooth and downsample the images, and the Laplacian pyramid, which applies the Laplacian transform to smooth the images [1]. A feature pyramid is composed of a set of multi-scale image features extracted from the images of an image pyramid. Multi-scale image features are used for detecting objects of various sizes in the images. During the detection process, a sliding window at each scale slides over the images as a template of features to detect objects of various sizes.

Conventionally, a feature pyramid is obtained by first constructing an image pyramid, and then computing the feature images at the corresponding scales. Dollár et al. propose a method for a fast generation of feature pyramids by approximating feature images via extrapolation from existing feature images at nearby scales [20]. The method is based on the phenomenon that natural images (images captured in natural environment, containing trees, rocks, etc.) have fractal statistical characteristics and are scale invariant, meaning that the prominent image structures and the underlying patterns of the natural images are preserved during resampling process [20]. According to the research on the statistical characterization of natural images [72], [73], Dollár et al. discover that the relationship between feature images of different scales follows a power law, and they propose the following formula for the approximation of the feature images across different scales:

$$C_s \approx R(C, s) \cdot s^{-\lambda\Omega} \tag{2.2}$$

Where C denotes a feature image at the original scale, C_s denotes a feature image at scale s, $R(C, s)$ denotes C resampled at scale s, Ω denotes a feature type, and each type has its own λ.

The above-mentioned formula is experimented and verified with different feature types (The computation of λ and other details see [20]). However, there are some exceptions where the above formula is not suited for the approximation of the feature images, for example upsampled images, image regions or images of nearly the same intensity, or sparse feature images where a large number of features have the value of zero [20].

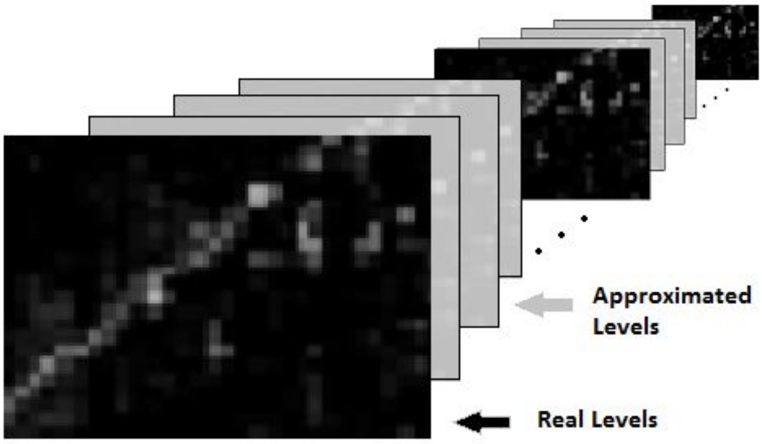


Figure 2.4: Structure of the feature pyramid.

A fast feature pyramid can be constructed using the approximation method above. For example, one can first compute several real feature images at one scale per octave, and approximate the intermediate scales between the octaves by the nearest real feature image. An octave is a term to denote the interval between the scales, increasing the scale by one octave means the image resolution is reduced to 1/2. Hence the feature images at scale 1, 1/2, 1/4, 1/8... are computed from the images downsampled from the original image, and the rest of the feature images in the pyramid are computed by the fast approximation. Figure 2.4 shows the basic structure of an approximated feature pyramid. Using the fast feature pyramid for detection systems makes only sacrifice on the detection accuracy (see Figure 2.1 in Section 2.1), but saves the computational load by the construction of the feature pyramid, and hence increases the speed of the whole detection system.

2.3 AdaBoost and Random Forest

Random Forest is developed by Leo Breiman and Adele Cutler. The definition given by Breiman et al. [30] is as follows:

> " Definition: A random forest is a classifier consisting of a collection of tree-structured classifiers $\{h(\mathbf{x}, \Theta_k), k = 1, \ldots\}$ where the $\{\Theta_k\}$ are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input **x**."

A forest comprises of a set of decision trees. A decision tree is a non-parametric classification model, which uses a directed acyclic graph, i.e. tree, for making decisions [74]. The classification of an example starts from the root node and goes down to the subsequent nodes. There is a test on a particular attribute at each node to determine to which child the decision goes. The children of each node are mutually distinct and exhaustive, and there is exactly one path from the root to each node. Each leaf of the tree bears a class label, when the decision path reaches a leaf, the class label of that leaf is assigned to the example.

If each node of a tree has exactly two children, such a tree is called a binary tree. In object detection, the input for a binary decision tree is a feature vector $v \in R^N$ of an example, each node represents a split function $f_n(v): R^N \to R$ with a threshold $t_n \in R$ to make a binary split. $P_n(c)$, the decision at node *n*, is a histogram of example labels, which is used to calculate the purity of the node and to evaluate the split [42]. Algorithm 2.1 shows the algorithm for training binary trees.

A forest is an ensemble of a large number (a few hundred to several thousand) of decision trees. Such complex models have large variance. A random forest improves generalization through bagging (random selection of samples) and randomised tree learning (random selection of features) [42], [30], [35].

Bootstrap aggregating, also called bagging, randomly selects subsets of examples with replacement from the training set. Trees are trained using these subsets of examples. To further reduce the correlation of the trees, random selection of subsets of features is performed at each node to build the randomised trees [42], [30], [35]. Algorithm 2.2 shows the algorithm for training a Random Forest [83].

In the classification stage, an input feature vector v is running down all T trees in the forest. Each individual tree t makes a decision $P_t(c \mid v)$. The final decision $P(c \mid v)$ is made by averaging the decisions of all the trees

$$P(c \mid \mathrm{v}) = \frac{1}{T}\sum_{t=1}^{T} P_t(c \mid \mathrm{v}) \tag{2.3}$$

or by majority voting.

Boosting is a machine learning method of combining many rough rules of thumb (weak classifiers) to create a single highly accurate prediction rule (the strong classifier), based on the observation that weak classifiers that perform just slightly better than random guessing are much easier to find than a single strong classifier. Multiple rounds of learning are conducted to learn the weak classifiers iteratively, a final strong learner is the combination of all the weak classifiers from each round [75].

AdaBoost is a popular boosting algorithm [83]. The fundamental idea of AdaBoost is to assign weights to the training examples, the weights are initially equal, but after each round of learning, the misclassified examples are assigned larger weights than those examples which are correctly classified, so that in later rounds the classifiers can focus on the hard examples [75]. Algorithm 2.3 shows the algorithm of AdaBoost. AdaBoost is widely used in object detection, for example in the groundbreaking work of Viola and Jones [83], [86], [20]. AdaBoost is able to select critical features from a large feature pool [100], [81], for example in the object detection system by Viola and Jones, each round of boosting selects one feature from the 180000 potential features, the vast majority of features are discarded so that the computation time is decreased [86].

AdaBoost can be used to train the Random Forest with the decision trees as weak learners [83].

Algorithm 2.1: Binary tree training

```
Input:
S: trainig samples, S= $x_i$ (i=1,2,...,n),  $x_i \in R^K$  with labels $y_i \in \{\pm 1\}$
Variables:
k: feature index,    t: threshold value
For each node:
1.  for k = 1:K
        for t = 1:T
            t = a random value in range ( min(S(k)), max(S(k)) )
            Calculate the impurity of the node based on t.
        end for
    end for
2.  Select the best feature with the lowest impurity. Store the corresponding k and t for
    the current node.
4.  Split the samples based on t, let $S_l$ feed the left child, $S_r$ feed the right child.
5.  If the node impurity is small enough or the leaf node is reached
        Store the posterior probability distribution for the current node.
        Return.
    Else
        Continue.
```

Algorithm 2.2: Random Forest algorithm [83]

```
Input: S: trainig samples
        f: number of input instances to be used at each of the tree
        B: number of generated trees in random forest
1. E is empty
2. for b=1 to B
3.         $S_b$ = bootstrapSample(S)
4.         $C_b$ = BuildRandomTreeClassifiers( $S_b$ ,f)
5.         $E = E \cup | C_b |$
6. Next b
7. Return E
```

Algorithm 2.3: AdaBoost

```
Input:
Sample images S: $(x_1, y_1)$ ... $(x_n, y_n)$, where $y_i \in \{\pm 1\}$
Weights of the samples: $w_1$, $w_2$, ..., $w_n$
1. Initialize the weights: $w_i \mathbf{1}_{\{y_i=1\}} = \frac{1}{n_1}$, $w_i \mathbf{1}_{\{y_i=-1\}} = \frac{1}{n_0}$
    where $\mathbf{1}_{\{y_i=-1\}}$ is the indicator function. $n_1$ is the number of positive samples and
    $n_0$ is the number of negative samples.
2. for t = 1 to T
3.         Normalize the weights $w_i^t = \frac{w_i^t}{\sum_{j=1}^{n} w_j^t}$
4.         For each feature k, train a classifier $h_k$, apply the classifier to all training samples
           and calculate the classification error $\varepsilon_k = \sum_i w_i \mathbf{1}\{y_i \neq h_k(x_i)\}$
5.         Store the classifier $h_t$ with the lowest classification error $\varepsilon_t$, set the classifier
           weight $\alpha_t = \frac{1}{2} \log \frac{1-\varepsilon_t}{\varepsilon_t}$
6.         Update the sample weights $w_i^{t+1} = w_i^t \exp\{-\alpha_t y_i h_t(x_i)\} / Z_t$, where $Z_t$ is
           chosen so that the weights sum to one.
7. Output: $H(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$
```

2.4 Summary

In this chapter we focus on two fundamental aspects of discriminative object detection systems, the feature representations and the learning algorithms. We introduce the feature types to be used in our system. We explain the concept of using overcomplete representations and feature combinations to improve detection accuracy. To compensate the performance loss due to the increased number of features, we use a fast feature pyramid computation method in our system. We also give a description of the learning method in our system, AdaBoost, binary tree and Random Forest.

Chapter 3
Datasets

We use four datasets in our work: AIT-Dataset1, AIT-Dataset2, the NYU-Depth V2 Dataset [80], and the INRIA Dataset [12].

AIT-Dataset1 and AIT-Dataset2 are obtained from an in-house developed sensor from the Austrian Institute of Technology (Figure 3.1). The equipment sets up three monochrome cameras in a rig, with a baseline of 0.4*m* between the two cameras at the two ends. The sensor has a resolution of 1280×1024 pixels, resampled to 1150×920. Calibration of the cameras are carried out offline. The census-based stereo matching algorithm computes the stereo maps which are congruent to the rectified intensity images [4].

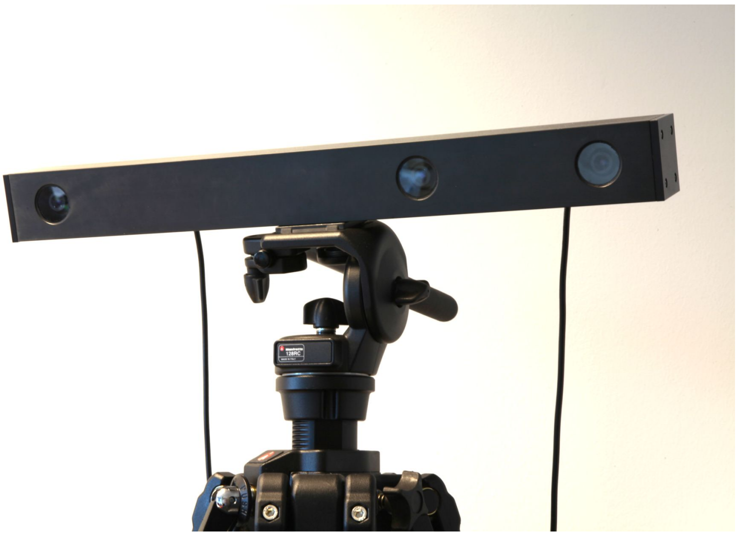


Figure 3.1: The in-house developed sensor from the Austrian Institute of Technology.

The stereo maps from the sensor are converted to depth maps by the following fomula

$$Z = \frac{fB}{D} \tag{3.1}$$

Where f denotes the focal length of the camera, B denotes the baseline, D denotes the disparity value at a pixel, Z denotes the corresponding depth value at the same pixel.

The stereo maps contain unknown values in some areas (shown as holes in the images), because the stereo matching algorithm fails to find the corresponding pixels of a pair of images in these areas while performing the similarity comparison. Such areas which are challenging for the stereo matching algorithm are, for example the textureless regions, or occlusions where pixels can only be seen by one camera [4]. The depth maps converted from the stereo maps retain those holes. In Chapter 4 we introduce two image inpainting methods, the Joint Bilateral Filter and the Partial Differential Equation Systems, for the enhancement of the depth maps.

3.1 Dataset Description

This section gives a description of all the datasets used in our work. Important information of the datasets includes: types of scenes, occlusion of the foreground objects, stationary or moving camera, image amount, image resolution, etc.

(a) AIT-Dataset1

This dataset consists of 237 pairs of grayscale and depth images of an indoor scene. The objects in the scene are ~20 people. The objects show various scales and poses. A majority of the objects are densely cluttered and occluded. This dataset is split into two subsets, 121 pairs of the grayscale and depth images are used as the training set, while the remaining 116 pairs of images are used as the testing set. The images have a resolution of 900 × 720. Figure 3.2 shows a pair of example images from the dataset.

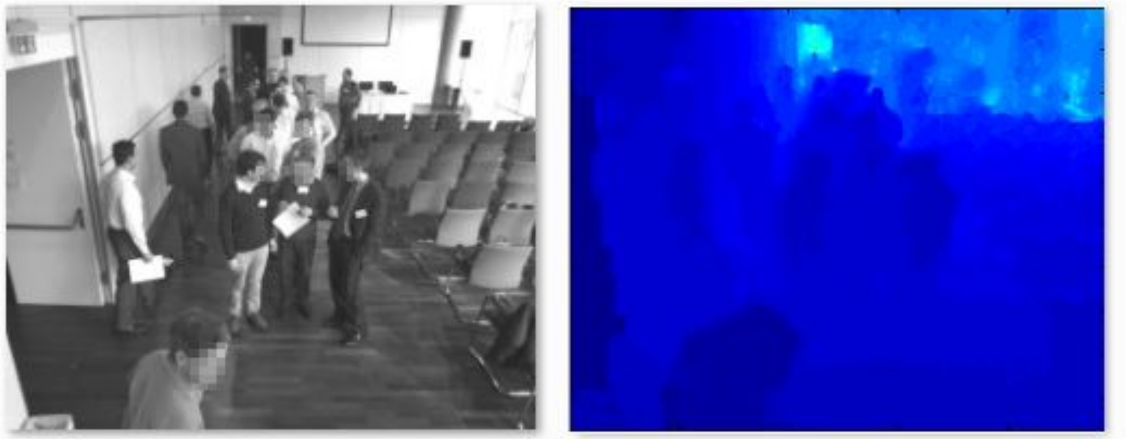


Figure 3.2: Example images of AIT-Dataset1, the depth map in this example is inpainted by the Joint Bilateral Filter (Chapter 4).

(b) AIT-Dataset2

This dataset consists of 177 pairs of grayscale and depth images of an outdoor scene, among which 87 pairs are positive images containing the objects, and 90 pairs are negative images containing only the backgrounds. The scenes are captured by a camera mounted on a vehicle. The scene contains 0-3 people with no occlusion. Due to the few objects contained in this dataset, it is used for testing. The images have a resolution of 1280 × 1024. Figure 3.3 shows a pair of example images from the dataset.

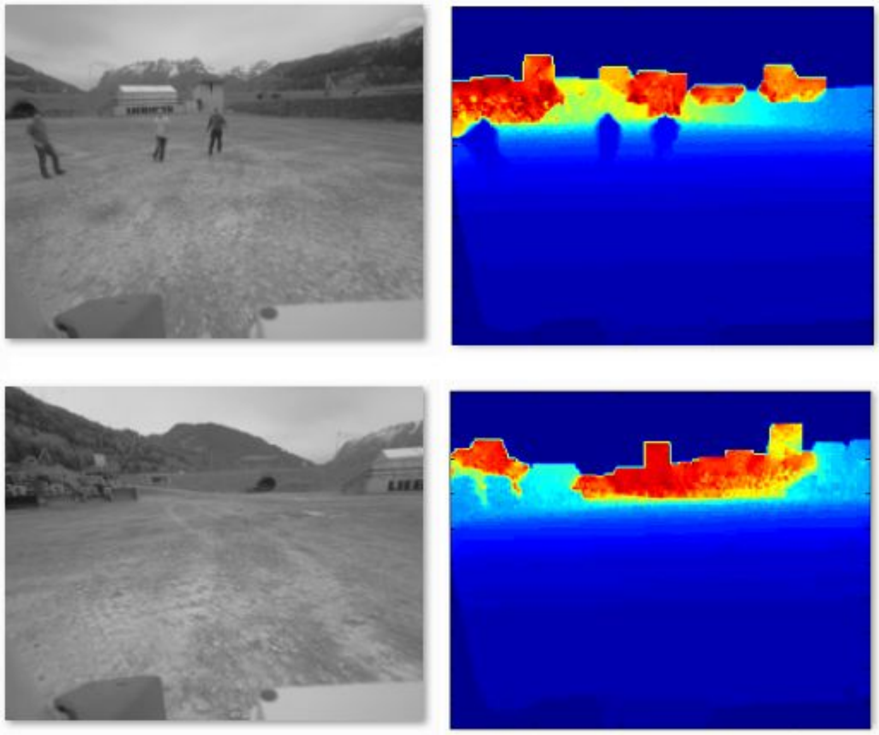


Figure 3.3: Example images of AIT-Dataset2, the depth map in this example is inpainted by the Joint Bilateral Filter (Chapter 4).

(c) NYU-Depth V2 Dataset [80]

The NYU-Depth V2 Dataset contains 1449 pairs of RGB and depth images captured by the Microsoft Kinect camera. The scenes include bathrooms, bedrooms, bookstore, cafe, classrooms, dining rooms, furniture stores, home offices, kitchens, libraries, living rooms, offices, playrooms, reception rooms, and studies. A few of the images contain persons in the scene. We take a subset of 1360 pairs of images from the dataset which contains no persons. The depth maps are filled by the colorization scheme of Levin et al. [52]. We convert the RGB images to grayscale in accordance with the other two depth datasets introduced above. The images have a resolution of 640×480. Figure 3.4 shows a pair of example images from the dataset.

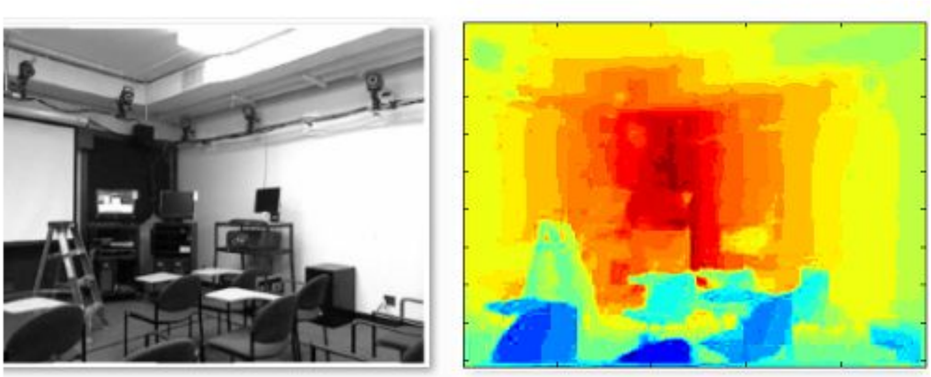


Figure 3.4: Example images of the NYU-Depth V2 Dataset [80]. The depth maps are inpainted by the colorization scheme of Levin et al. [52].

(d) INRIA Person Dataset [12]

The INRIA Person Dataset is collected by the authors of Histogram of Oriented Gradients (HOG) for the purpose of training and testing their human detector using the HOG features. The images in the dataset are taken from different sources, such as GRAZ01 Dataset [64], personal digital image collections, and google images. The positive training set contains 1208 images of persons in more or less upright positions, with a large variation of poses and backgrounds. Persons with height >100 pixels are annotated. We take a subset of 614 images from the positive dataset in our work. The negative training set contains 1218 images from the same data sources as the positive training set, including indoor, outdoor, city, mountain, and beach scenes, with some of the images focused on "hard examples". Figure 3.5 shows examples of the positive and negative images from the dataset.

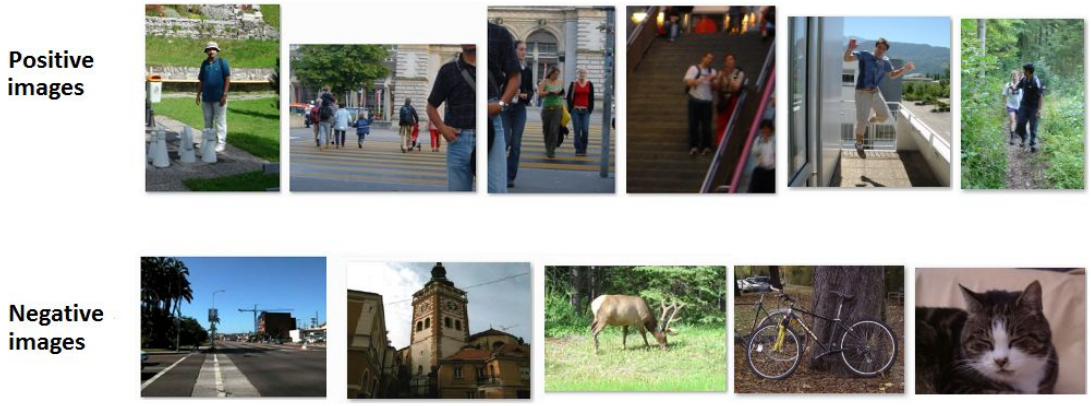


Figure 3.5: Example images of the INRIA Dataset [12].

3.2 The Ground Truth Annotation

The ground truth annotation is done with the help of the AIT annotation tool (Figure 3.6), which enables a video to be loaded and allows any amount of objects in each frame to be annotated, finally it exports all the bounding boxes with the corresponding frame number to a single file. A bounding box is parameterized by its top left coordinates, width and height, and is represented as a tuple (x, y, w, h). A flag can be added to the tuple as a fifth element to indicate whether this bounding box shall be ignored in the evaluation (more detail see Chapter 5).

The AIT-Dataset1 is annotated with head-and-shoulder bounding boxes. In the training set, ~10 persons in the scene are annotated, who are not occluded or partially (less than 50%) occluded. Persons who are heavily (more than 50%) occluded or standing in the back are not annotated. The bounding box is over the head of a person on the top, and the width extends till the shoulder blades are contained if the person is facing the camera, otherwise the width of the bounding box is wider than the body if the person is standing sideways, the height of the bounding box reaches the chest of the person. A total number of 121 images in the training set contains altogether 1183 samples, which are flipped in the training process to enlarge the number of samples.

Figure 3.7 shows an example of one annotated image from the training set, and the positive windows sampled from it. 9 persons are annotated in this image, among which 4 are without occlusion and against the scene background, 2 are without occlusion but with other persons in

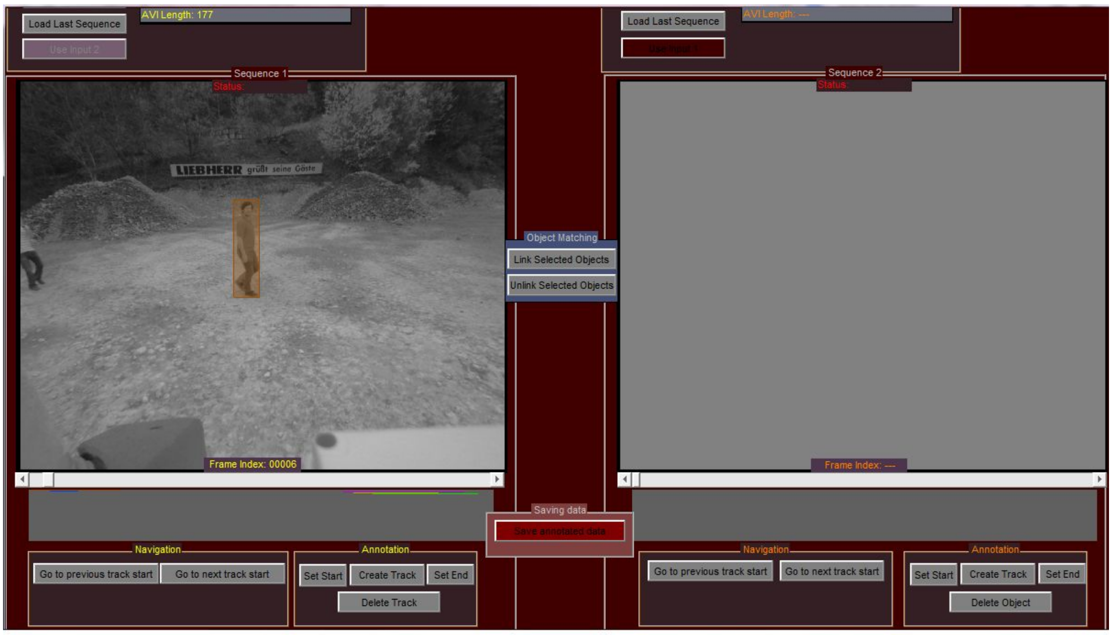


Figure 3.6: The AIT annotation tool.

the background, 3 are partially occluded and with other persons in the background. The sampled windows are padded to a bigger size. The standard model size is defined as 56x41, with 32 pixel margins on each side. The sampling process recalculates the padded size according to the relationship of the proportions as defined above.

In the testing set of AIT-Dataset1, every single person in the scene is annotated because the detector may detect any of them. As Figure 3.8 shows, this produces a clutter of overlapping bounding boxes, which are difficult to evaluate. Because a detected window may match multiple overlapping ground truth bounding boxes, it is hard to distinguish which ground truth is the actual match of the detected object. In Chapter 5 we introduce a filtering method to select a subset of the bounding boxes. Figure 3.8 shows an example of an annotated image from the testing set of AIT-Dataset1.

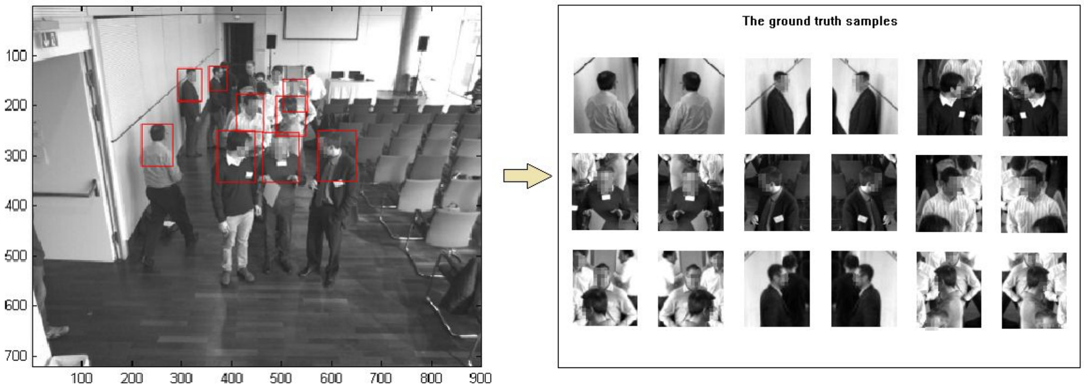


Figure 3.7: An example of an annotated image from the training set of AIT-Dataset1, and the positive windows sampled from it.

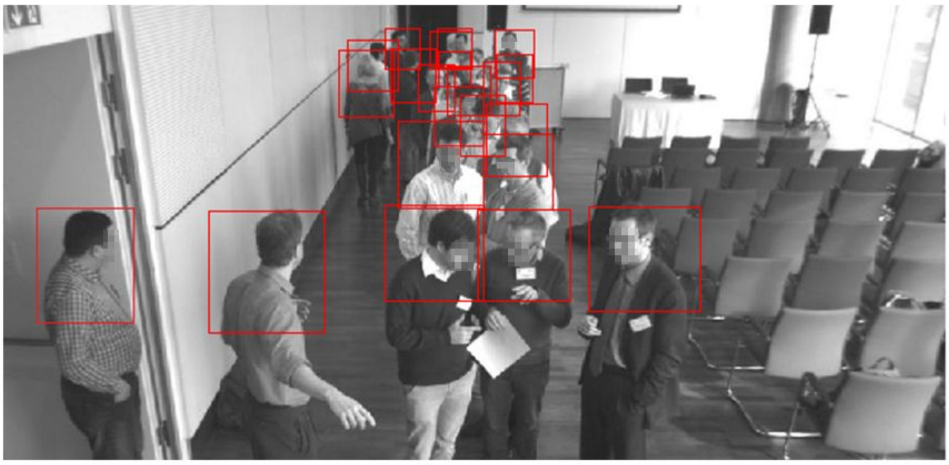


Figure 3.8: An example of the annotation of the testing set of AIT-Dataset1.

The INRIA Dataset and the AIT-Dataset2 are annotated with full-body bounding boxes. The annotation of the INRIA Dataset centers on the torso and body of a person, and leaves arms and legs outside the bounding box if they are extended sideways. The annotation of AIT-Dataset2 is wider than the annotation of the INRIA Dataset, it contains the whole person unless the arms or legs are outstretched. Figure 3.9 shows the examples of annotations for images from the training set of INRIA Dataset and the testing set of AIT-Dataset2.

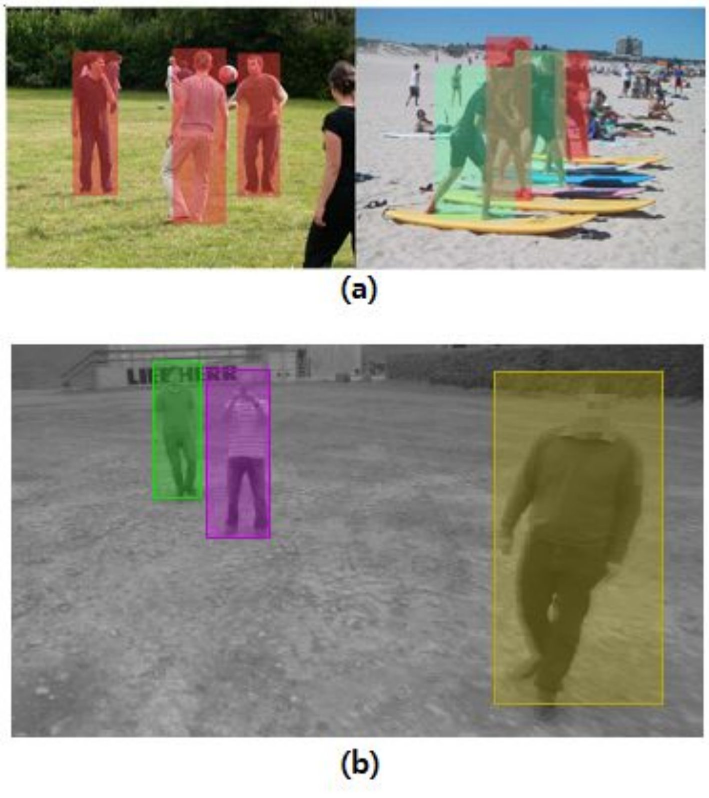


Figure 3.9: An example of the annotation of (a) the training set of INRIA Dataset and (b) the testing set of AIT-Dataset2.

3.3 Summary

In this chapter we provide a description of the preliminary datasets used for our work. The datasets are gathered from different sources. Stereo maps are converted to depth maps. Positive datasets are annotated by the ground truth bounding boxes.

Chapter 4

The RGBD Detector: Incorporation of the Depth Cues

In this chapter we introduce the proposed RGBD Detector. Figure 4.1 shows the workflow of the implementation of the RGBD object detection system. Before we incorporate the depth information into the existing system, the depth maps have to be enhanced in order to fill the missing values and remove the noise. After the refinement of the depth maps, we choose two gradient-based features, gradient magnitude and gradient histogram, for the representation of the depth information. The amount of features can be reduced by a single step of gray level reduction. We incorporate the depth cues into a RGB detector trained with AdaBoost and Random Forest. The depth features and the RGB features can be fused at different levels, in our work we make two experiments, one uses the low-level fusion of features by establishing a joint feature space, the other uses the classifier-level fusion by combining multiple classifiers. We further exploit the scene information embedded in the depth maps and estimate the scales of the objects using the depth values.

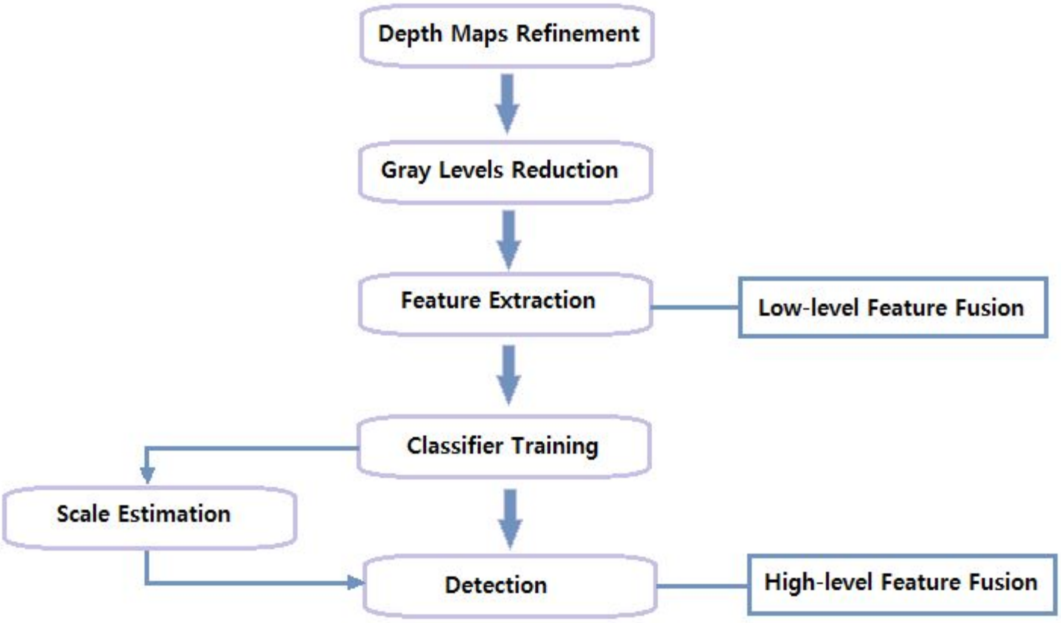


Figure 4.1: The workflow of the incorporation of the depth cues in a RGB detector.

The rest of the chapter is organized as follows: We introduce two image inpainting methods for the refinement of the depth maps in Section 4.1. Section 4.2 describes the two feature types we use for the depth channels. Section 4.3 explains how the amount of features is reduced and the prominent features are enhanced by means of gray level reduction. Section 4.4 describes the feature fusion schemes. Section 4.5 explains the scale estimation method. Section 4.6 gives a summary of this chapter.

4.1 The Depth Map Enhancement

As introduced in Chapter 3, the stereo maps of AIT-Dataset1 and AIT-Dataset2 acquired from the sensor contain invalid data and noise. Therefore the depth maps converted from the stereo maps contain holes which display unwanted patterns and features in the gradient image (see Figure 4.2). Post-processing and noise suppression are compulsory for the depth maps before they can be used for training and detection. The methods for image inpainting can be used for the enhancement of the depth maps. Image inpainting techniques make undetectable modifications to the corrupted parts of images, so that the images are reconstructed seamlessly [7]. In image processing, the inpainting methods use the available data outside and surround the missing parts to fill the values inside the missing parts [7]. In the following part of this chapter we discuss two approaches for inpainting the depth maps using the techniques based on Partial Differential Equations (PDE) and the Joint Bilateral Filter.

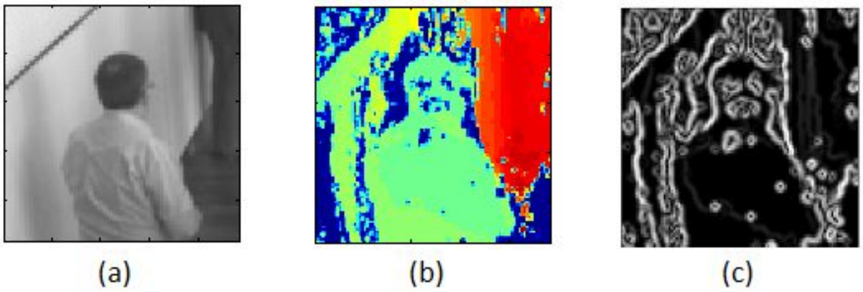


Figure 4.2: (a) Intensity image. (b) The depth image contains holes and noise, which produce unwanted features in the gradient image (c).

4.1.1 The PDE Based Methods

Bertalmio et al. [7] propose a pioneer work of the PDE based solution for image inpainting. Let D denote an image region to be inpainted, their approach propagates the information (the local color smoothness variations obtained by the 2D Laplacian operator) from the outside of D to the inside, through the formulation of the PDE. The propagation is done along the level line (isophote) directions, which are calculated as the perpendicular gradients $(-\partial_y, \partial_x)$ of the pixels at the contour of D [7].

Different kinds of PDEs can be used to formulate the process of the inpainting of the image regions [76]. One of them is the diffusion function:

$$u_t = \Delta u \tag{4.1}$$

Where *u* denotes the temperature value, and Δ is the Laplace operator [10]. This PDE describes how the temperature diffuses across a surface over a time interval. Richard et al. [69] state that because the human visual system can tolerate a certain degree of blurring in non-edge areas, in very small regions of the image let the intensity values represent the temperature values, the information propagation process of the inpainting can be approximated by the diffusion process. [10], [69].

In our work we use the techniques presented by D'Errico [14] to do the PDE based inpainting of the depth maps:

> "... a boundary value solver. The basic idea is to formulate a partial differential equation (PDE) that is assumed to apply in the domain of the artifact to be inpainted. The perimeter of the hole supplies boundary values for the PDE. Then the PDE is approximated using finite difference methods (the array elements are assumed to be equally spaced in each dimension) and then a large (and very sparse) linear system of equations is solved for the NaN elements in the array."

D'Errico has presented several different inpainting methods. One method discretizes the Laplacian equation $u_{xx} + u_{yy} = 0$ of the domain to be inpainted into a linear system of equations. A variation of this method is to use a different PDE model as $u_{xxxx} + 2u_{xxyy} + u_{yyyy} = 0$. Another method uses the spring model. Each pixel is supposed to be connected to its 8 neighbours with springs. The energy of each spring is proportional to its extension. This model can be formulated by linear systems of equations (details see [14]). Figure 4.4 shows the results of the above stated methods.

The main feature of D'Errico's methods is that they are linear solutions, which means that the filling of the missing values is done smoothly. The disadvantage is that it is unable to preserve the edges [76].

4.1.2 The Joint Bilateral Filter

Another technique we employ for the depth map inpainting is the Joint Bilateral Filter proposed by Kopf et. al. [44] for the depth map refinement.

The Bilateral Filter [84], [49] is a non-linear edge preserving smoothing filter. A Bilateral Filter is a combination of a spatial kernel (such as a Gaussian filter) and a similar range kernel. For an image position *p*, the filtering result is defined as:

$$I_p^{filtered} = \frac{1}{W_p} \sum_{q \in \Omega} I_q f(\| p - q \|) g(\| I_p - I_q \|) \tag{4.2}$$

$$W_p = \sum_{q \in \Omega} f(\| p - q \|) g(\| I_p - I_q \|)$$

Where $I_p^{filtered}$ is the resulting filtered image, *I* is the original image, *f* is the spatial kernel, *g* is the range kernel, Ω is the kernel window centered at *p*, and *Wp* is the normalizing factor to preserve the image energy.

The filtering result of the pixel value at p is a weighted average of the values of the nearby pixels. The spatial kernel defines the priority or the weight of a pixel q according to the image plane distance of p and q, then it performs a basic filtering operation. The range kernel determines the weight according to the values of p and q (e.g. intensity, color or depth). Thus pixels which look similar to p are given high weights and pixels which look different are given low weights, it ensures that the sharp edges are preserved in the images.

The Joint or Cross Bilateral Filter[44] is defined as:

$$I_p^{filtered} = \frac{1}{W_p} \sum_{q \in \Omega} I_q f(\| p - q \|) g(\| I_p^{guidance} - I_q^{guidance} \|) \tag{4.3}$$

The only difference to Eq. 4.2 is that the range kernel is applied on a guidance image I' instead of the original image. The filtering process on the original image is assisted with the additional information from the guidance image. For example, when smoothing an image, the depth or normal map can be used as the guidance image to provide the information of the depth boundary or the object geometry. In this way the smoothing is performed with respect to the object geometry. Another example is the joint bilateral upsampling method proposed by Kopf et al. [44]. This method uses the available high resolution image as a prior to upsample the low resolution image. The authors demonstrate the usefulness of this method in applications such as adaptive tone mapping, stereo depth, image colorization, and graph-cut based image composition [44].

For the inpainting of the depth maps we use the intensity image I as the guidance image for the depth map d. The filling of the missing depth values is guided by the similarity of intensity values of the surrounding pixels, under the assumption that pixels which have similar intensity values also have similar depth values.

$$d_p^{filtered} = \frac{1}{W} \sum_{q \in \Omega} d_q f(\| p - q \|) g(\| I_p - I_q \|) \tag{4.4}$$

We use Gaussian filters for both the spatial kernel and range kernel to do the filtering at 3 scales, with the following pairs of standard deviations: (δ_{s1}=12, δ_{r1}=0.2), (δ_{s2}=5, δ_{r1}=0.08), (δ_{s3}=8, δ_{r3}=0.02).

4.1.3 Comparison of Different Filling Methods

Figure 4.3 displays the results from the above stated methods. A subjective evaluation is performed on the result images:

The result image based on the PDE $u_{xxxx} + 2u_{xxyy} + u_{yyyy} = 0$ in panel (e) contains hallucination and artefacts, therefore it is not suitable for our work. The image based on the Joint Bilateral Filter in panel (c) shows a satisfying result since it displays clear shapes of the door at the back and the person on the very right side of the image who is entering the scene. By comparing the result image based on the PDE $u_{xx} + u_{yy} = 0$ in panel (d) and the result image based on the spring model in panel (f), we see that the image in panel (f) is smoother at the wall at the back. We choose the methods from (c) and (f) for our work.

A detailed comparison of the result images from the Joint Bilateral Filter and the spring model PDE is shown in Figure 4.4. As expected, the Joint Bilateral Filter preserves sharp edges while the spring model PDE generates blurred edges.

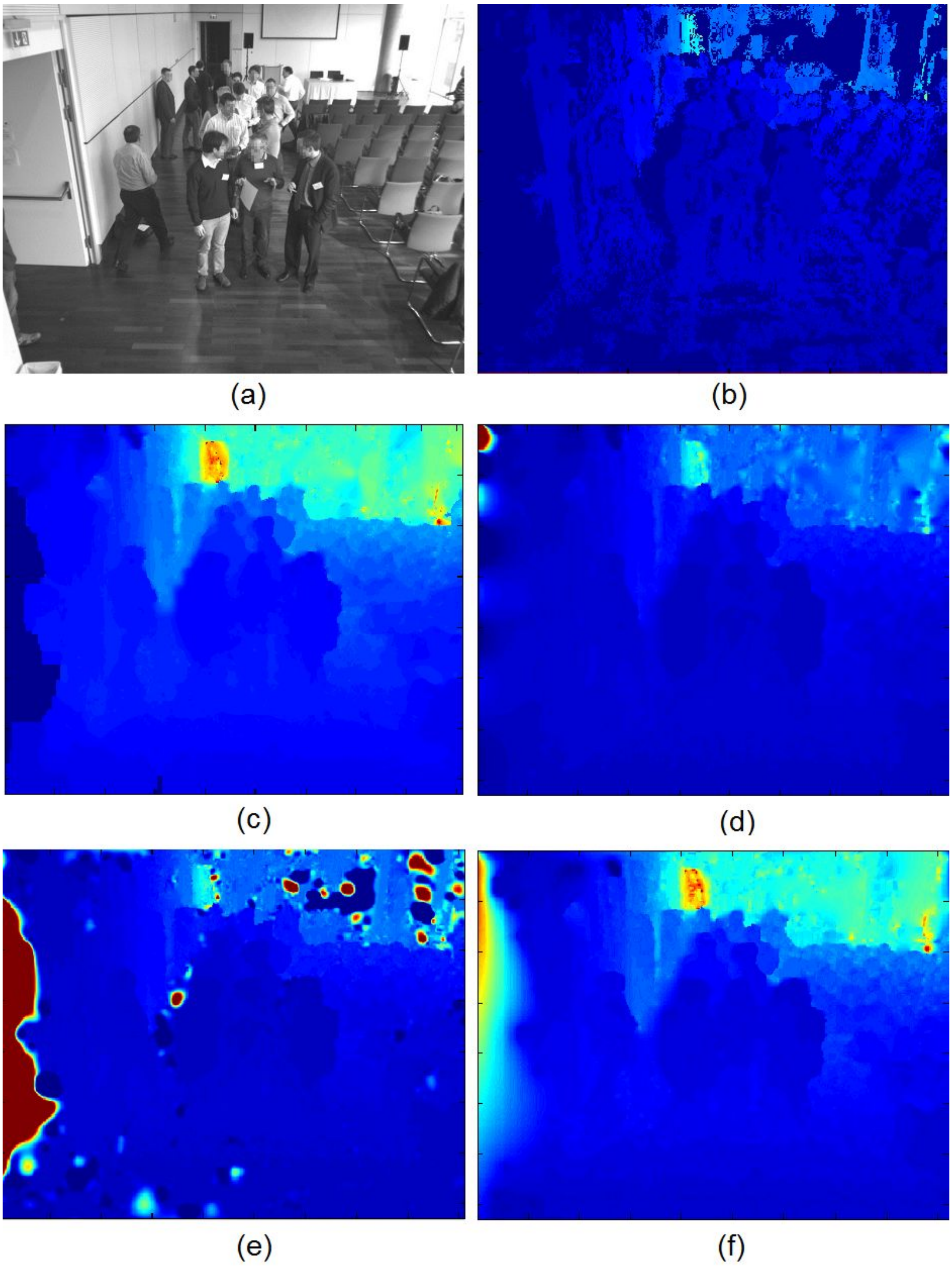


Figure 4.3: (a) The original intensity image. (b) The raw depth map corresponding to (a). The results of different inpainting methods: (c) Joint Bilateral Filter. (d) PDE based method using the equation $u_{xx} + u_{yy} = 0$ (e) PDE based method using the equation $u_{xxxx} + 2u_{xxyy} + u_{yyyy} = 0$ (f) the spring model.

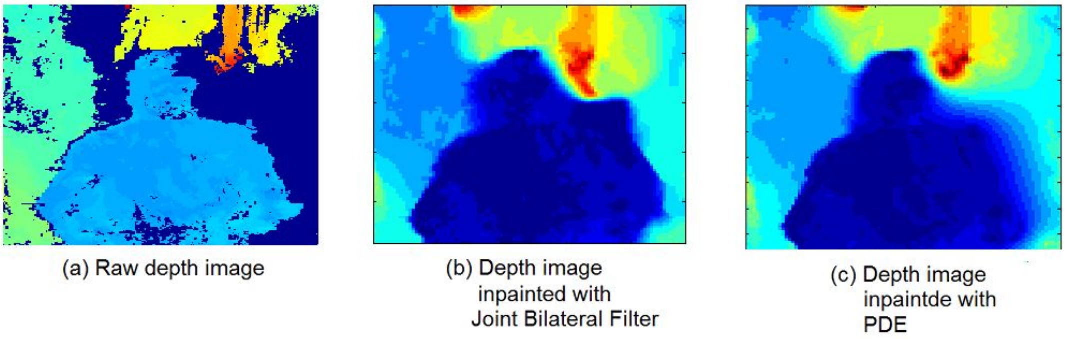


Figure 4.4: Comparison of depth maps inpainted with the Joint Bilateral Filter and the Spring Model PDE.

4.2 Depth Channel Computation

In this section we compute the features from the depth maps. Previous research has exploited the depth information in the following aspects:

- The depth maps contain the additional visual cue, which is useful for reconstructing the 3D scene structure or the 3D object model.
- When the depth map is treated purely as a 2D image, it exhibits the boundary information of the objects. The gradient-based features are well suited for representing the boundary information.

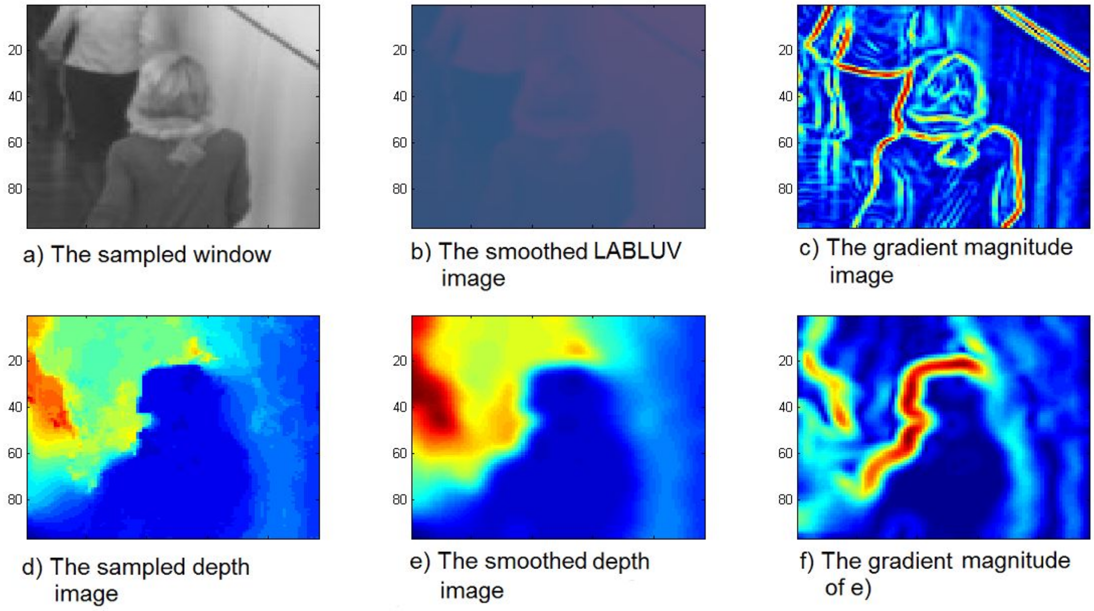
Lai et al. [46] use the HOG features from the RGB and depth images to train a standard sliding window detector. Christoph et al.[41] use the dense stereo in two modules in their pedestrian detection system. Firstly, the stereo information is used to estimate the road profile and the camera parameters for finding the possible pedestrian locations (regions of interest). Secondly, in the training and classification stages, they use the HOG features from the depth maps and the intensity images. They observe that after averaging the SVM weights over the HOG blocks, the intensity images exhibit the strongest features on the head/shoulder and legs of a pedestrian, while in the depth maps the strongest features are the upper body and torso areas. Xia et al. [96] propose a human detection method by locating the head of a human using the depth information. They first find the possible head locations in an image by matching a binary head contour template with the boundary information (edges) from the depth map. Then a 3D head surface model is computed, using the parameters extracted from the depth array, to perform a refined selection in the detected regions. After the head is located, the whole figure is detected by a segmentation method.

In our work we extract the boundary information of the objects by computing the gradient magnitude and HOG features from the depth map. In section 4.5 we make use of the scene information embedded in the depth maps by estimating the scale priors for the detected windows according to the relationship between the size of an object and the depth of its location in the scene.

4.2.1 Depth Gradient Magnitude

The motivation to use the gradient magnitude feature is based on the observation as shown in Figure 4.5. The gradient magnitudes of the intensity image exhibit the details of the scene while the gradient magnitudes of the depth image focus on the contour of the foreground object. The gradient magnitudes of the intensity image show the strongest features around the object contour (the person in the foreground, the person in the background, the line on the wall), together with the other visible features such as the textures of the hair and clothes of the persons, the shadows on the wall, and the shadows and the texture on the floor. In contrast, the gradient magnitudes of the corresponding depth image exhibit the most significant features on the contour of the person in the foreground, with a weaker feature on the contour of the next person in background, while all the other details of the scene are not shown. Note that in the depth image there are areas showing discontinuity of change of the depth values, the shades and flecks, which are caused by the inpainting algorithm performed in the last step and result in weak features like circles and stripes in the gradient magnitude image.

According to the above observation, we use the gradient magnitudes of the depth images to guide the training process to focus on the foreground objects. Each 24 × 24 window yields 576 gradient magnitude features from the depth image.



a) The sampled window

b) The smoothed LABLUV image

c) The gradient magnitude image

d) The sampled depth image

e) The smoothed depth image

f) The gradient magnitude of e)

Figure 4.5: One pair of intensity and depth images sampled from the positive training set, each of which is smoothed before the gradient magnitudes are computed.

4.2.2 Gradient Histogram

Dollár et al. propose a feature type similar to HOG, called the oriented gradient histogram. The computation of the oriented gradient histogram is illustrated in Figure 4.6. Given an input image I of the size $h \times w$, the gradients for the R, G, B channels, R', B', G', are computed respectively. At pixel (x,y), the maximum gradient magnitude of R'(x,y), B'(x,y), G'(x,y) is selected and put into the corresponding bin channel image at the same pixel. Similar to HOG, the bin channel image index of a gradient magnitude is determined by its orientation. The gradient magnitude is either put into the nearest bin, or put into two neighbouring bins by linear interpolation. The bin channel images are tiled into $N \times N$ non-overlapping cells, each cell sum up the gradient magnitudes fallen inside it.

The value of the cells at a same position along the whole range of *N* bins represents a histogram of oriented gradients within that cell. The resulting feature image set is of the size floor*(h*/*N)* × floor*(h*/*N)* × *N*. The oriented gradient histogram is a generalized case of HOG, with normalization by gradient magnitudes it can approximate HOG [15]. Figure 4.7 shows an example of the oriented gradient histogram of a depth map with 6 orientation bins.

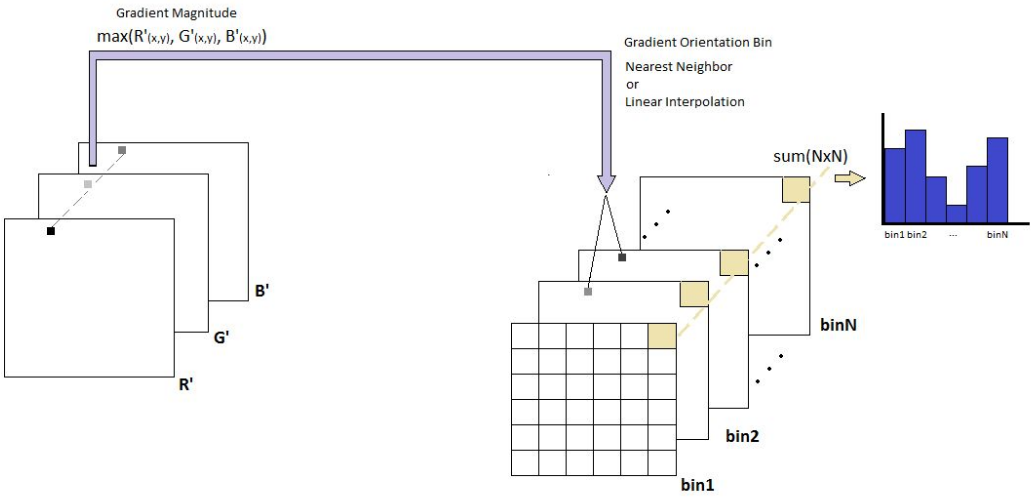


Figure 4.6: Oriented gradient histogram computation based on a gradient magnitude and a gradient orientation image.

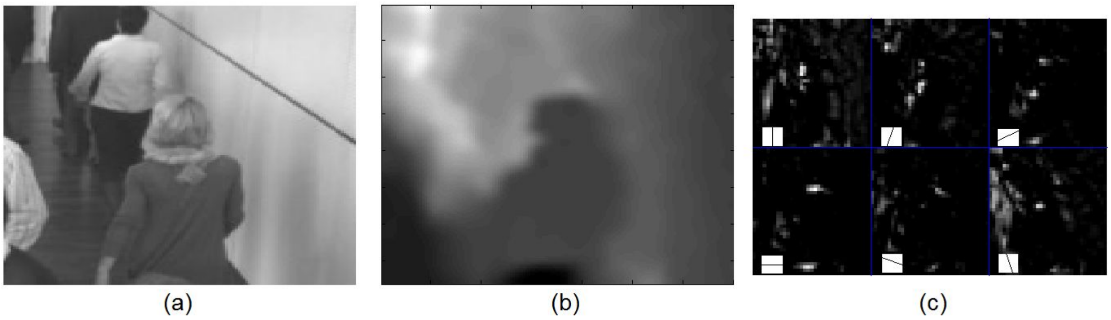


Figure 4.7: Example of the oriented gradient histogram of a depth map. (a) The original intensity image. (b) The corresponding depth map of the intensity image. (c) The oriented gradient histogram with 6 orientations computed from the depth map.

4.2.3 The Integral Image

Because the speed of the feature computation is crucial for a real time detector, we use the integral image to accelerate the computation of features [86], [15], [101]. The gradient histograms and HOG can be computed efficiently by the concept of the integral image for use in a real-time system [101], [15], [67].

The integral image, or summed area table [37], is a representation for storing the sums of pixel values within the rectangular areas of an image. The rectangular areas start from the upper left location and end at any pixel of the original image. With the help of the integral image, rectangular features, like the Haar-wavelet type features, can be calculated very fast, because the sum of any rectangular area in the original image can be calculated by four array lookups in the integral image [86]. The panel (a) of Figure 4.8 illustrates how an integral image is generated.

Let I denote the original image, and S denote the integral image. Any point (x, y) in S can be computed as: $S(x, y) = I(x, y) + S(x-1, y) + S(x, y-1) - S(x-1, y-1)$. The recursive manner allows the integral image to be computed in one pass over the original image. Similarly, panel (b) of Figure 4.8 illustrates that the sum of the yellow area in the middle can be computed as $D\text{-}B\text{-}C\text{+}A$.

In Figure 4.6, the gradient magnitudes are stored in separate bin channel images so that an integral image can be computed for each bin. The sum of $N \times N$ each block can then be obtained by merely 4 array lookups.

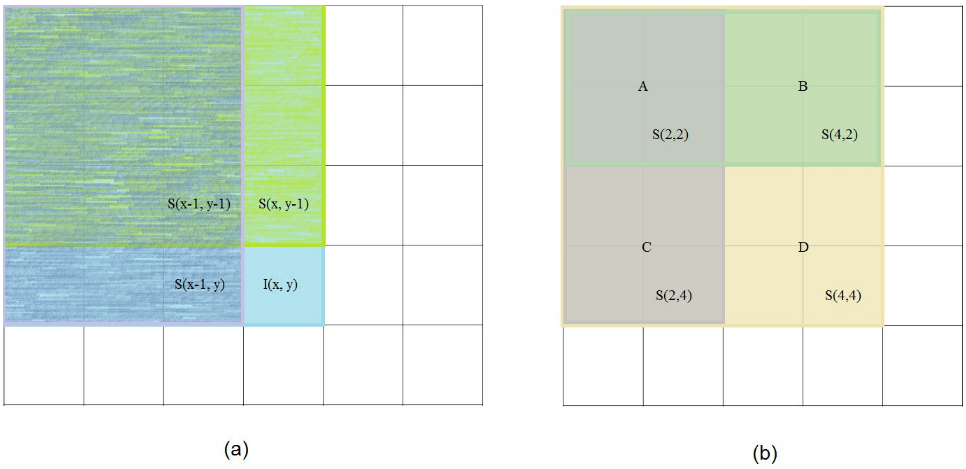


Figure 4.8: (a) The integral image S can be computed from the original image in a recursive way $S(x, y) = I(x, y) + S(x-1, y) + S(x, y-1) - S(x-1, y-1)$. (b) After the integral image is generated, the sum of pixels in any rectangular area can be computed by four array lookups, for example the yellow area in the middle of I can be computed as $D\text{-}B\text{-}C\text{+}A = S(4, 4) - S(2, 4) - S(4, 2) + S(2, 2)$.

4.3 Reduction of the Feature Dimensionality via Gray Level Quantization

The depth images are 16 bits grayscale images. Each pixel has 16 bits storage which encodes 65536 levels of intensities, which means for gradient based feature images that each pixel allows 65535 levels of gradients. As described above, the purpose of using the depth images for training is to extract the boundary information, however Figure 4.9 (a) shows that the gradient magnitude feature image of the 16 bits depth image reveals redundant details for this purpose. In order to eliminate the redundant details and preserve the most prominent edges, we reduce the bit depth of the depth images to 8 bits. Reducing the bit depth decreases the gray levels and groups neighbouring pixels with similar intensity values into homogeneous regions. Figure 4.9 shows a pair of images of a same scene with 16 bits and 8 bits depth respectively.

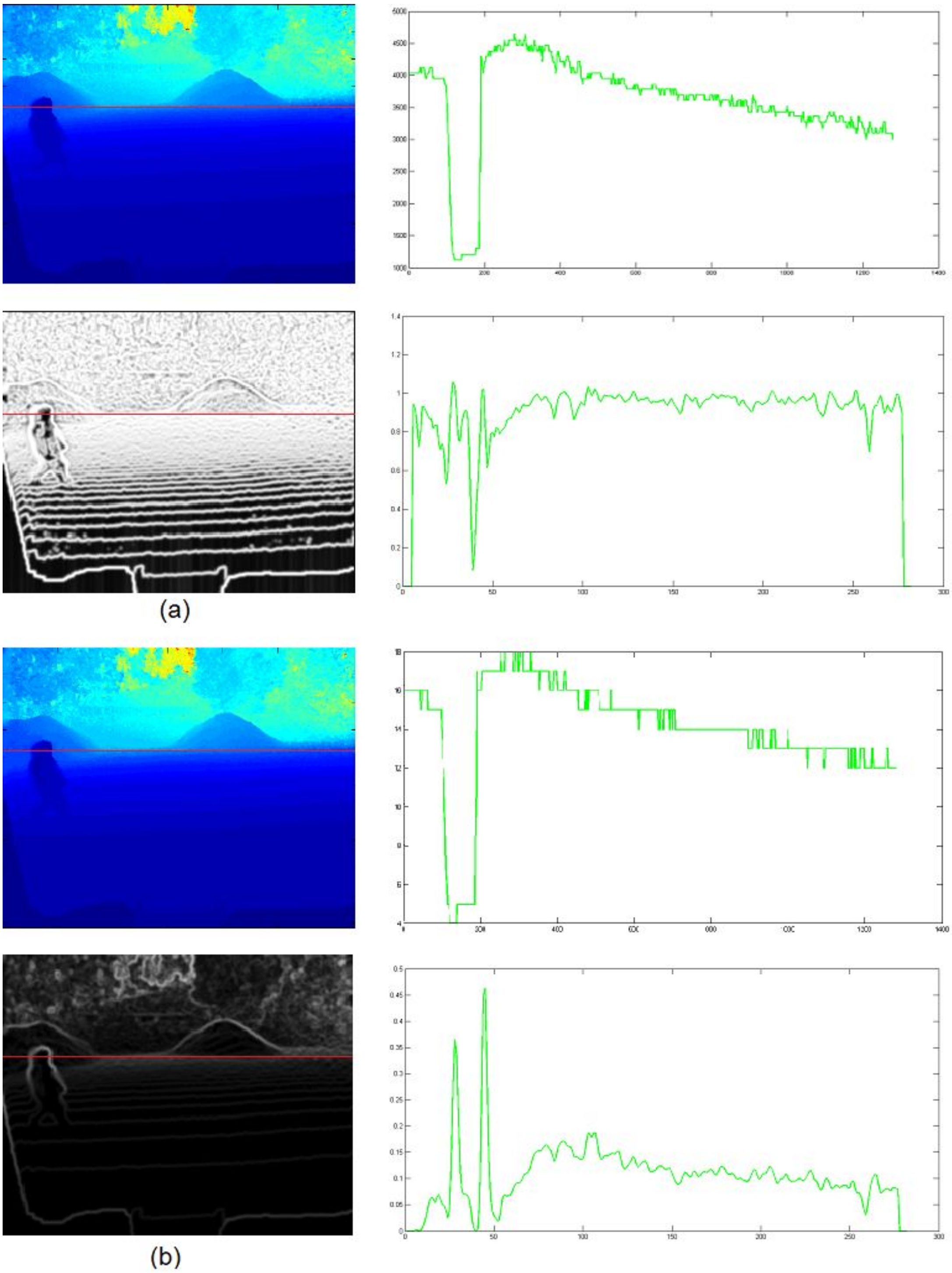


Figure 4.9: Comparison of the gradient magnitude feature images from the original image of the bit depth (a)16 bits (b) 8 bits. Although the original images look similar, the feature image of a higher bit depth reveals more redundant details than the one of a lower bit depth. The the red lines in each image on the left are shown as a function of pixel values against the x coordinates on the right side.

Although the original images do not show apparent differences, the gradient magnitude feature images display significant differences in the amount of details. We take a horizontal line in each image (red) at a same y position and show it as a function of pixel values against the x coordinates on the right side. The upper curve in (a) shows the intensity values of the 16 bits depth image, the valley of the curve corresponds to the person who is nearer to the camera than the background. The noisy fluctuations of the curve indicate the unevenness of the object surfaces, which produces edges in the gradient magnitude feature image, and are shown as troughs and crests in the curve below. The fluctuations in the curve of intensity values in (a) are flattened in (b). In the curve of gradient magnitudes, the edge information at the boundary of the person is much stronger than the rest parts of the image. Although the details in the environment and inside the object are not completely diminished, which is also not our purpose, the boundaries of the objects remain the most prominent features in the scene.

The main benefit of reducing the gray levels is to save the computational cost. Instead of applying a smoothing filter such as the Gaussian filter to the image, only one division is needed for each pixel, i.e., the 16 bits pixel values are divided by 255 and stored in the type uint8. The features are not reduced in dimensionality, but in their density. Hence the discrimination of important and unimportant features is enhanced.

4.4 Fusion Schemes for the Depth Features and the RGB Features

After feature extraction and feature reduction, the feature set derived from the depth maps is to be provided to the classification process. A discriminative object detection system using intensity based features provides a fast and accurate framework, then the depth features are incorporated into the system and fused with the RGB features. We use two approaches of feature fusion in our work, the aggregated channels approach which is based on joint feature space, and the multiple classifier approach which is based on parallel combination of classifiers.

4.4.1 The RGB Detector

We adopt the RGB detector proposed by Dollár et al. [20] as a framework for our work. The RGB detector uses altogether 10 channels of features: normalized gradient magnitude, Histogram of Oriented Gradients (HOG) (6 channels), and LUV color channels. Small windows of a predefined size (default 128×64) are sampled from the training set. The positive windows are sampled from the positive images according to the ground truth annotation, with the annotated objects in the center of the window. The negative windows are sampled from a dense grid of the negative images. An image pyramid of typically 8 scales per octave is created from each sampled window from the negative images. Real feature images are only computed at the power of two scales. The feature images at the 7 intermediate scales within an octave are approximated from a real feature image at the nearest scale.

Feature images are vectorized and fed to the training process. AdaBoost and Random Forest are used for training. N features of each sub-window are trained to form a total number of T depth-two trees for each $m \times n$ window.

Such a framework is capable of achieving real-time speed (> 20 fps) and high accuracy (with an average miss rate $< 20\%$) [20]. Viola and Jones design a cascade structure in their boosting process to reject the negative instances at an early stage to gain performance [86], in contrast, the framework does not apply such a finer feature selection strategy in the learning process, yet it achieves real-time speed. One reason for the superior performance is that the Random Forest as weak classifiers perform simple pixel based comparison. The fast feature pyramid is another key contribution to the system performance. The impact of the approximated feature images on the system accuracy is given in Figure 4.10: The detection accuracy of the framework using exact and fast feature pyramids, with the an average miss rate between 10^{-2} and 10^{0} false positives per image, on four pedestrian datasets, INRIA [12], Caltech [18], TUD-Brussels [91], and ETH [24].

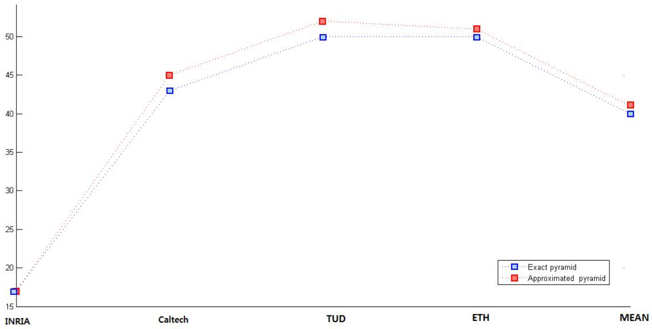


Figure 4.10: The comparison of the accuracy of the framework using exact pyramid (ACF-Exact) and approximated pyramid (ACF) [20]

4.4.2 The Aggregated Channels Approach

In order to construct a joint feature space, we combine different types of features via channels, as illustrated in Figure 4.11. Since the scene images captured by the AIT camera are grayscale images, when they are converted to the LUV space, only the L channel contains the scene information and the UV channels are empty. Feature images of gradient magnitude and gradient histogram are computed from the depth images, as described in Section 2, which add 7 more channels to the existing ones. Each pixel lookup in the aggregated channels forms a 17 dimensional feature vector. The feature vectors are sent to the AdaBoost training process to learn random trees, as explained in Chapter 2.

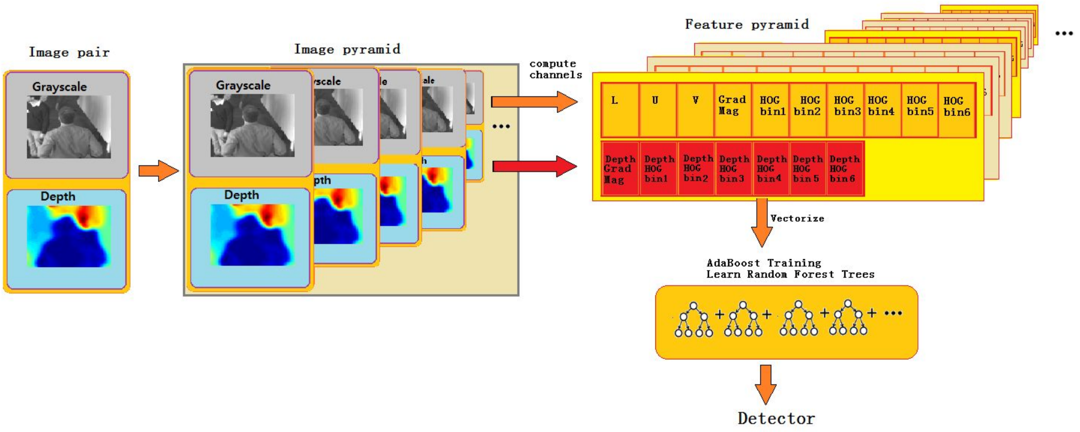


Figure 4.11: The workflow of the aggregated channels approach.

4.4.3 The Multiple Classifiers Approach

Classifier-level combination is a high-level fusion of features, where individual classifiers are trained with different or same feature sets, and the output of each classifier mutually contributes to the decision making process [70], [41]. Duin et al. draw a conclusion from their experiments, that a parallel combination of classifiers is far more effective than a stacked combination, i.e. the combination of information extracted from different feature sets using a same classifier is more effective for classification than the combination of information extracted from a same feature sets using different types of classifiers [21]. Rohrbach et al. and Keller et al. both propose human detectors following a parallel combination of intensity and depth cues [70], [41]. Figure 4.12 illustrates the structure of high-level combination of classifiers.

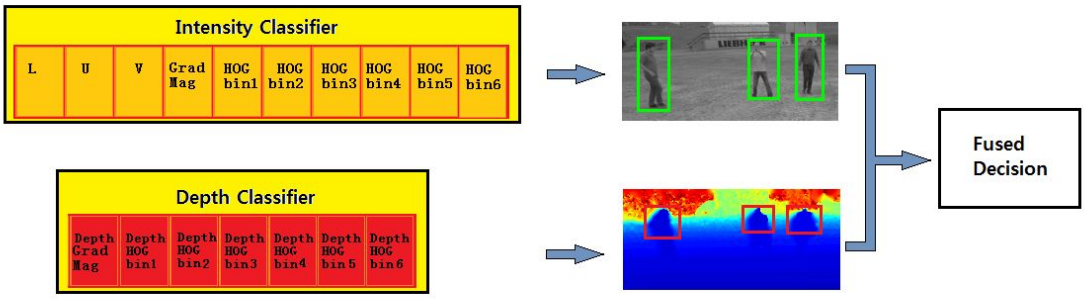


Figure 4.12: The overview of a multiple classifiers people detection system. The intensity images and the depth images are trained separately and fused at the classifier-level. Not only different modalities but also different classes (e.g. full-body, head-and-shoulder) can be combined according to a combination rule.

Following is the decision rule of multiple classifiers presented by Rohrbach [70]: Let a feature vector be denoted as x_k , $k = 1, ..., n,$ a classifier be indexed by i, $i = 1, ..., m$, the object class be denoted by j, $j = 0, 1$ (positive, negative). The posterior probabilities for sample k with respect to object class j and estimated by classifier i is given as $p_{ij}(x_k)$. For each sample, a confidence score with respect to object class j, $q_j(x_k)$, is derived by combining the posterior probabilities $p_{ij}(x_k)$ from all classifiers. The combination rule can be maximum, multiplication, average, or SVM decision value (distance to the hyperplane).

The object class with the highest confidence score is selected as the final decision:

$$\omega(\mathrm{x}_k) = \arg\max_j (q_j(\mathrm{x}_k)) \qquad (4.5)$$

We train an intensity classifier and a depth classifier separately, using the same feature sets as the aggregated channels approach for intensity images and depth images. However, our decision rule differs from that of [70]. Instead of making decision for each data sample, we let the classifiers perform detection independently and make decision for each detected object. The decision rule can be described as:

$$W_{obj} = \mathrm{argmax}_{w \subseteq I}(f_i(W)) \qquad (4.6)$$

Where W denotes the sliding window which slides over the image I, and f_i denotes the evaluation function of classifier *i* which calculates the confidence score for *W*.

We set the intensity classifier as a primary classifier, and make the depth classifier as a complementary secondary classifier. The reason of setting explicit priority to one classifier against the other is because the depth dataset is annotated with head-and-shoulder BBs due to the constraints of the scenes, while the intensity dataset is annotated with full-body BBs, which can better cope with the pose variations of objects and outputs more accurate detection windows. This is done by collecting the detection results from both detectors, and setting the scores of the detected BBs from the RGB detector to a sufficient large number (1000 in our case). Then the non-maximum suppression is performed on the whole set of BBs, so that the BBs from the RGB detector will suppress the BBs from the depth detector when they both detect a same object (see Algorithm 4.1).

Algorithm 4.1: Decision rule for combining the RGB detector and the depth detector

```
    detector = {detectorRGB, detectorDepth}
    for each frame do
        for i = 1 to 2 do
            bbs{i} = conductDetection( detector{i} , current_frame)
            non-maximal suppression on bbs{i}
    end for
    bbs{1}.scores = 1000;
    Enlarge bbs{2}: width *= 1.2; height = width * 3;
    non-maximal suppression on all bbs
end for
```

4.5 Scale Estimation Using the Depth Information

Because the depth values contain the information of the scene structure, in object detection these values can help the detector to conduct detection in consideration of the scene context [36]. Among a variety of scene information which can be reconstructed or obtained by the depth values, the scales of the objects are useful clues for the detector to refine the detection results. Researchers have proposed various methods for obtaining the scales of the objects using the depth values. Scott et al. [36] suggest a method for capturing the scale priors of the scene objects. By rejecting the false positives using the scale priors the average precision of their detector is improved by 0.2 on each dataset they use. Lai et al. [46] construct a scale histogram based on the normalized depth values. Ye et al. [97] update the size of a tracking window in relation to the preceding one by comparing the depth values of these two tracking windows. The essential thought behind these approaches is based on the observation that the scales of the scene objects relate in an inverse manner with their distances to the camera. As we see in Figure 4.13, the persons standing nearer to the camera have larger bounding boxes than the ones standing further away, given a depth value, the sizes of the corresponding bounding boxes fall into a certain range (an estimation with deviation). Motivated by these works, we use scale estimation in our detector to improve the detection accuracy.

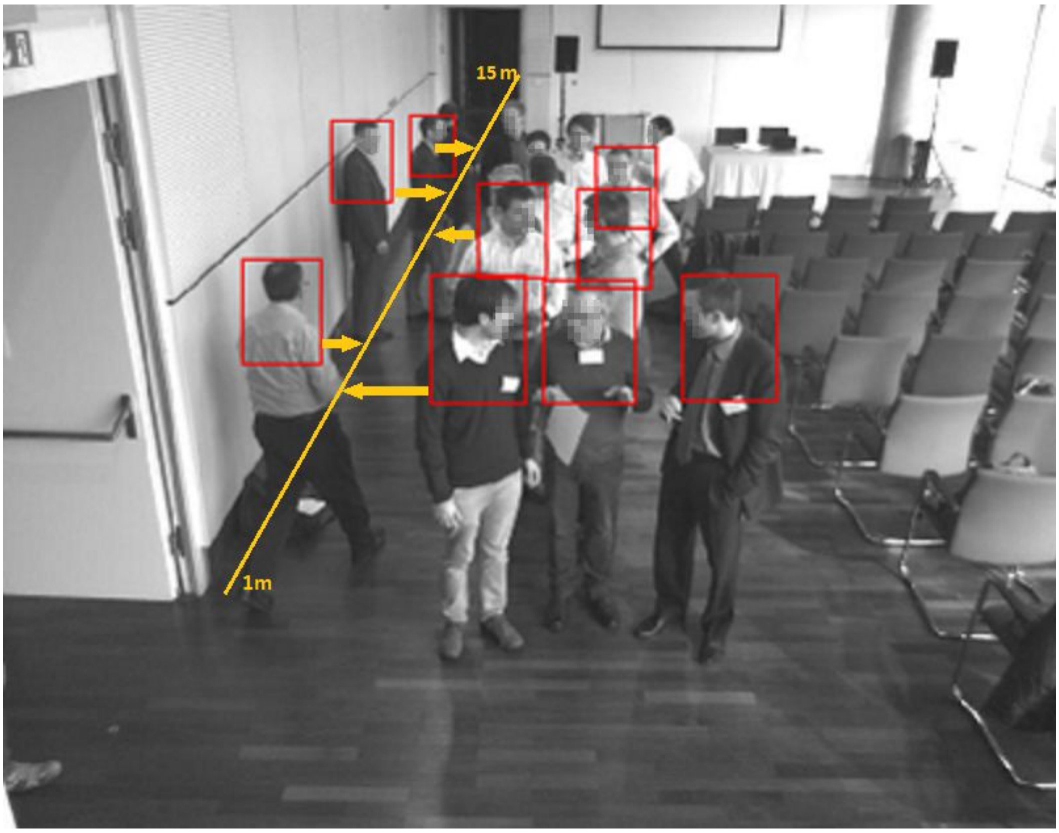


Figure 4.13: An example image from the AIT-Dataset1. The scene in the image exhibits an inversely proportional relationship between the distance of an object to the camera and its size. The bounding boxes have approximately the same sizes at a certain distance.

4.5.1 The Scene Geometry for the Scale Prior

In order to understand the relationship between the scales of objects and their distances to the camera mathematically, we have to study the projective geometry. The projective geometry describes how the objects in the 3D space are projected onto a 2D plane in the pin-hole camera model, where the projection is made by emitting straight lines from the objects and let them pass through a single point [27], [3]. The projective geometry has the following rules [3]:

1. The vanishing points are lying on a straight line called the horizon.
2. Straight lines in the 3D space stay as straight lines in the 2D plane.
3. Any set of parallel lines in the 3D space meet at a vanishing point on the horizon in the 2D plane.
4. A set of parallel lines in the 3D space which are parallel to the 2D plane remain parallel and have no vanishing point in the 2D plane.

Let (x', y') denote the projection of the 3D point (x_0, y_0, z_0) on the 2D image plane, x' can be obtained by x_0:

$$x' = \frac{f}{z_0} x_0 \qquad y' = \frac{f}{z_0} y_0 \tag{4.7}$$

Where f is the focal length of the camera [3].

When we consider an object in the scene, as shown in Figure 4.14, the distance of the object center to the camera origin z_w can be obtained by the following formula [36] :

$$z_w = \frac{f s_w}{s_\theta} \tag{4.8}$$

Where s_w is the scale of the object in the scene, and s_θ is the scale of the bounding box (height or width) in the image plane.

As all points within a bounding box have different depth values, a single depth value has to be chosen to represent the whole bounding box. Scott et al. [36] suggest to use the depth value at the center of the bounding box, or the average of the depth values within a bounding box:

$$\frac{1}{N}_{x \in \theta} \times \mathrm{z}_x \approx \frac{f\,\mathrm{s}}{s_\theta} \tag{4.9}$$

Where N denotes the number of pixels within a bounding box, **z** and **s** are the random variables for the depth values and object scale in the scene. In our work we use the minimal depth value of a bounding box, $\min_{x\in\theta}(\mathrm{z}_x)$, which is the point closest to the camera origin in the scene.

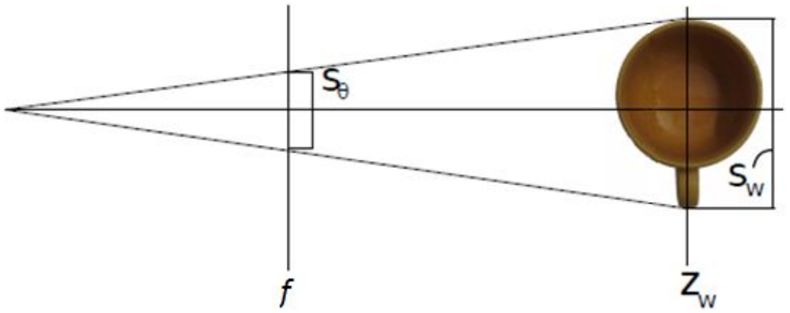


Figure 4.14: Scene geometry of an object. Image adapted from [36].

Noise and uncertainty exist in the above model, for example: the uncertainty in depth measurements, noise due to discretization and the stereo algorithm, the uncertainty in the hand-marked annotations, variations in the scales of the objects (for example heights of persons).

The scale prior of an object class can take any form of statistical distributions [36]. In our work we assume that the scale prior of our object class has a Gaussian distribution.

4.5.2 Object Scale from the Reciprocal of the Depth Value

Now we consider the scene geometry in training data AIT-Dataset1. Figure 4.15 gives a description of the scene geometry in a pin-hole camera model. In our case the aspect ratio of the bounding box is fixed, therefore the scale of the bounding box can be represented by its height. From Eq. 4.10 it can be deduced that the height of a bounding box in the 2D plane can be obtained by:

$$h' = \frac{f}{z_0} h \tag{4.10}$$

Where h is the height of a virtual bounding box of the head-and-shoulder of a person in the 3D space, h' is the height of the bounding box annotated in the 2D image. The height of the head-and-shoulder bounding box is about one third of the height of a person. Suppose all the persons have the same height, i.e. set fh as constant, then the height of a bounding box h' in the 2D image shall follow an inversely proportional relationship with the depth value of its position z_0 . In other words, the height of a bounding box in the 2D image follows an approximately linear relationship with the multiplicative inverse of the depth value of its position $1/z_0$. We take the minimal of the depth values inside a bounding box, which is the nearest point in the head-and-shoulder part of a person facing the camera as the depth value for the bounding box. We model the linear relationship between the height of a bounding box and the minimal depth value within this bounding box by a 1st degree polynomial:

$$H = ax + b \tag{4.11}$$

Where x is the inverse of the minimal depth value within the bounding box, a and b are the two coefficients for the polynomial. The approximately linear relationship between the height of a bounding box and the minimal depth value within this bounding box can been seen in Figure 4.16 when we visualize it from the training data. A polynomial fitting is performed based on these data. As Eq. 4.11 shows, the ideal case shall be a linear polynomial running through the origin of coordinates. However, noise and uncertainty exist in the data, as explained in the above section. The polynomial fitting is calculated in a least squares sense, with a tolerance region of predictions. Given $x = 1/z$, the corresponding value $h = ax + b$ with an error estimate Δ defines a tolerance region [h+Δ   h-Δ] which contains at least 50% of the predictions of future observations at x [59].

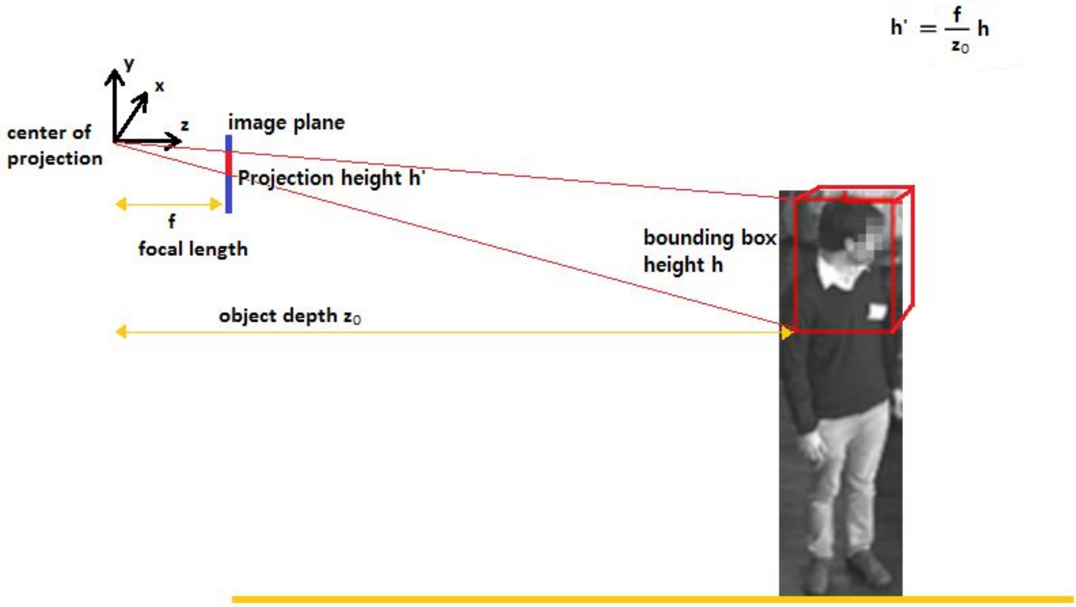


Figure 4.15: Scene geometry in training data AIT-Dataset1.

The polynomial fitting is done in the training stage where the two coefficients are computed. In the detection phase, the bounding box scale estimation is done before performing the non-maximal suppression of detected windows. For each detected window, the minimal depth value within the window is used for evaluating the linear polynomial, and the result is given as *h* and *Δ*. If the actual height of the detected window falls inside the tolerance region, the window is preserved for the next stage of non-maximal suppression, otherwise the window is rejected. In Chapter 5 we will discuss in more details how the scale estimation of the bounding boxes improves the detection accuracy by rejecting the false positives.

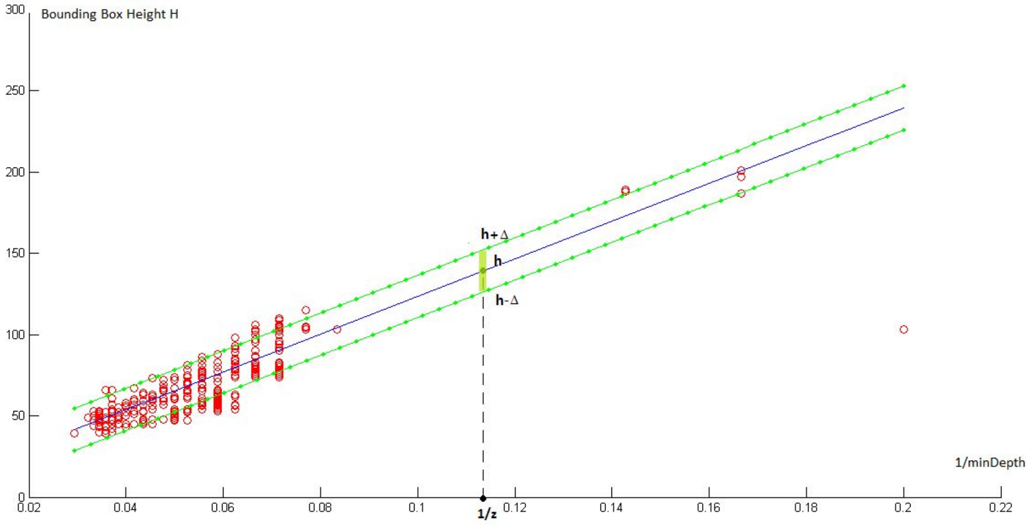


Figure 4.16: Visualization of the relationship between the heights of the bounding boxes and the reciprocal of the minimal depth values within the bounding boxes. The values are computed from the sampled windows of the annotations from the training data. The result exhibits an approximately linear relationship, which can be well fitted by a 1st degree polynomial computed in a least squares sense, with a tolerance region of error estimate. The tolerance region describes the variances in the data, for example the heights of persons, the poses of the persons and the hand annotations, at least 50% of the predictions of future observations are contained in the region [59]. In the detection phase, this polynomial is used for evaluating the predicted scale of each detected window. If the actual size of the detected window falls outside the tolerance region, the window is recognized as a false positive and is rejected.

4.6 Summary

In this chapter we describe how we implement the RGBD detector. A pre-processing step is required for the depth images, in order to enhance the useful information, fill the missing data, and suppress noises. Gradient magnitude and gradient histogram are extracted from the depth data, and the integral image technique is used to accelerate the feature computation. We explain that gray level quantization is a meaningful and cheap way of reducing redundant features. We propose two fusion schemes for the RGB features and the depth features, one scheme fuses the features at a low level, while in the other scheme features are combined at the classifier-level. Finally, we compute the scale prior of objects from the depth data, to help the detector to conduct detection in consideration of the scene context.

Chapter 5
Experiments and Results

In this chapter we evaluate two RGBD detectors developed with different feature fusion schemes. We use the testing sets introduced in Chapter 3 and test the two detectors on two scenarios respectively. We give a description of the evaluation protocol in Section 5.1. In Section 5.2 we evaluate the RGBD Detector using the low-level feature fusion scheme. Emphasis is placed on the comparison of the modalities of features (RGB vs. RGBD), feature types for the depth channels, effectiveness of the scale estimation. The impact of inpainting algorithms to the system accuracy is also discussed. In Section 5.3 we introduce a multiple-class detector which detects different parts of an object and makes decisions at the classifier-level. We show that for both detectors the additional features from the depth data improve the detection accuracy. Section 5.4 gives a summary of this chapter.

5.1 Evaluation Protocol

As the "Piotr's Image and Video Matlab Toolbox" (PMT) [19] provides an evaluation method based on the modified pascal criteria [18], we adopt it for the evaluation of our detection results. The pascal criteria [25] is given as:

$$a_o = \frac{area(dtBB \cap gtBB)}{area(dtBB \cup gtBB)} > threshold \tag{5.1}$$

Where *gtBb* denotes the ground truth bounding box, and *dtBb* denotes the detected window. A certain threshold is given for the determination whether a ground truth bounding box and a detected window match. If a_o is above the threshold, *gtBb* and *dtBb* are determined as matched, otherwise they are determined as unmatched. If a detected window matches a ground truth bounding box, both of them are marked as true positives. If a ground truth bounding box is not matched by any of the detected windows, it is a false negative. If a detected window makes no match to any of the ground truth bounding boxes, it is a false positive.

However, the modified pascal criteria [18] allows a ground truth bounding box to be marked by a flag as "ignore". The modified pascal criteria is given as:

$$a_o = \frac{area(dtBB \cap igBB)}{area(dtBB)} > threshold \tag{5.2}$$

Any detected window that makes no match to the standard (non-ignore) ground truth bounding boxes, but has an intersection with a ground truth bounding box marked as "ignore", then this detected window will also be marked as "ignore". The "ignored" ground truth bounding boxes and detected windows will not be taken into consideration in the evaluation of the system accuracy. Figure 5.1 illustrates the modified pascal criteria. The "ignore" flag provides advantage to reduce false positives in special areas of a scene, such as the occluded areas. We will explain the benefit of using "ignore" flags in more detail in Section 2.

For displaying the system performance we use the Receiver Operating Characteristic (ROC) curve. Instead of the two commonly used operating characteristics for ROC curves, the true positive rate and the false positive rate [31], we plot the miss rate against False Positives Per Image (FPPI) at various thresholds, using log-log plots (see Figure 5.7). The Log-Average Miss Rate (LAMR) is computed for an assessment across the range of 10^{-2} to 10^{0} in log-space, by averaging the miss rate of evenly spaced samples in log-space of the FPPI [92].

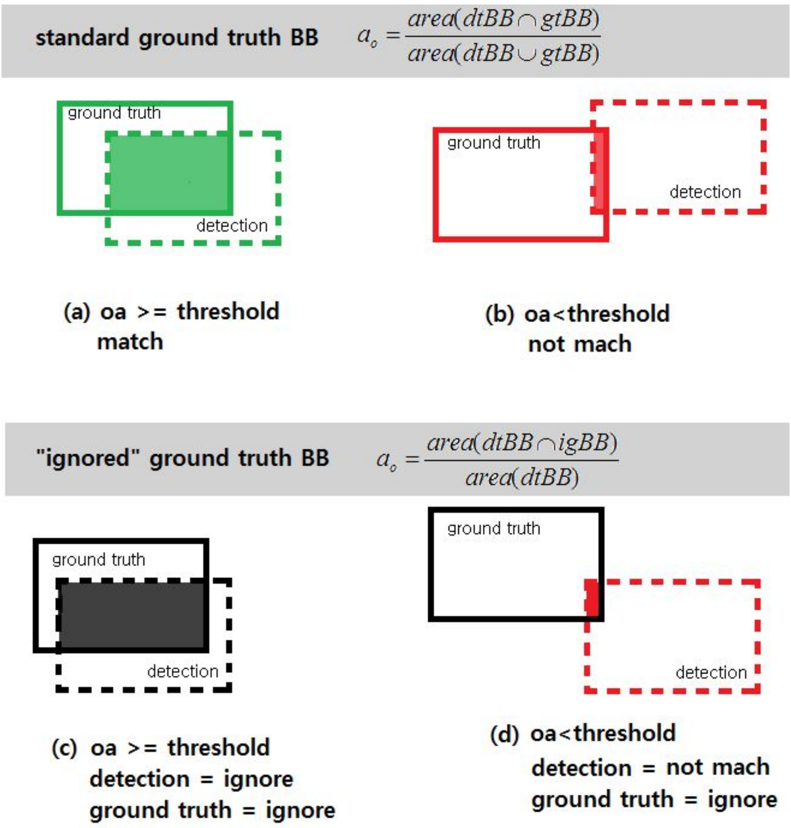


Figure 5.1: The modified pascal criteria for the determination of whether a detected window matches a ground truth bounding box. For a standard (non-ignore) ground truth bounding box, (a) if the overlap area between a ground truth bounding box and a detected window is big enough (oa > threshold), then these two windows are marked as matched, (b) otherwise they are marked as unmatched. (c) For a detected window which have no match to any standardRGBD Detectorh bounding box, but has an intersection with a ground truth bounding box marked as "ignore", then this detected window is also marked as "ignore", and will not affect the evaluation of the system accuracy.

5.2 Test Case 1: The RGBD Detector with The Low-level Feature Fusion Scheme

In this section we compare a set of detectors developed with the low-level feature fusion scheme.

The AIT-Dataset1 is split into two subsets, 121 frames for the positive training set and 116 frames for the testing set. We use the NYU-Depth V2 Dataset [80] as the negative training set. All the detectors in this section are trained on the same training set and tested on the same testing set.

The detectors use 10 channels of 3 feature types (LUV, gradient magnitude, gradient histogram) for the RGB data. The features (gradient magnitude, gradient histogram) extracted from the additional visual cue, i.e. the depth maps, are added to the RGB channels.

We use two types of depth maps which are filled by two different inpainting algorithms, the Joint Bilateral Filter and the Partial Differential Equation systems, in order to investigate the impact of inpainting algorithms to the performance of the detector. We use Bilateral and PDE to denote the two types of depth maps respectively. Because the gradient histogram is a variation of the HOG feature, we refer to it as HOG for the easiness of understanding.

Below is a list of the detectors to be evaluated:

1. RGB
2. RGBD-Bilateral GradMag
3. RGBD-Bilateral HOG
4. RGBD-Bilateral GradMag+HOG
5. RGBD-PDE GradMag
6. RGBD-PDE HOG
7. RGBD-PDE GradMag+HOG

Training Set		**Testing Set**
Positive: AIT-Dataset1 Number: 121	Negative: NYU-Depth V2 [80] Number: 1360	AIT-Dataset1 Number: 116

Table 5.1: The training sets and testing set used for Test Case 1: The RGBD Detector with the low-level feature fusion scheme.

5.2.1 Ground Truth Filtering

Since the evaluation of the detection results depends on the intersection between the detected windows and the ground truth bounding boxes, the ground truth annotation is a significant factor to affect the evaluation result. When we apply the testing set with the full set of hand annotations for the evaluation of the detection results, false positives and false negatives are reported by the computational evaluation process where actual matches take place if judged by human eyes. In order to reduce the "false false positives" and the "false false negatives" resulting from the ground truth annotation and to make the computational evaluation result to be best in accordance with the evaluation by human eyes, we have to address the following problems:

1) The Sizes of the Ground Truth Bounding Boxes

The sizes of the ground truth bounding boxes are arbitrary due to the hand annotation, while the detected windows are set to a fixed aspect ratio. A match between a detected window and an object may be judged as a mismatch when the ground truth bounding box is too big. To this end the ground truth bounding boxes have to be adjusted to the same aspect ratio as the detected windows. The heights of the ground truth bounding boxes remain as the hand annotation and the widths are recalculated according to the aspect ratio of the detected windows.

2) Missing Values in the Depth Maps

The very left regions of the depth maps have undefined depth values. In the inpainted depth maps, the objects standing in this region either have a sharp undefined boundary (depth map inpainted by the Joint Bilateral Filter) or a boundary merging into the background (depth map inpainted by the PDE systems). Detection of objects in this region behave in an unstable manner, because the detection may fail due to the incomplete or distorted information of the depth channel, or the detection may be successful based on the information provided by the RGB channels together with the partial information from the depth channel. Therefore the ground truth annotations from this region (x<100 pixel) shall be marked as "ignore", i.e. the detected windows that have intersection with the ground truth bounding boxes from this region, or the ground truth bounding boxes themselves, whether they have any detected window matched to them or not, will be ignored in the evaluation of the system accuracy.

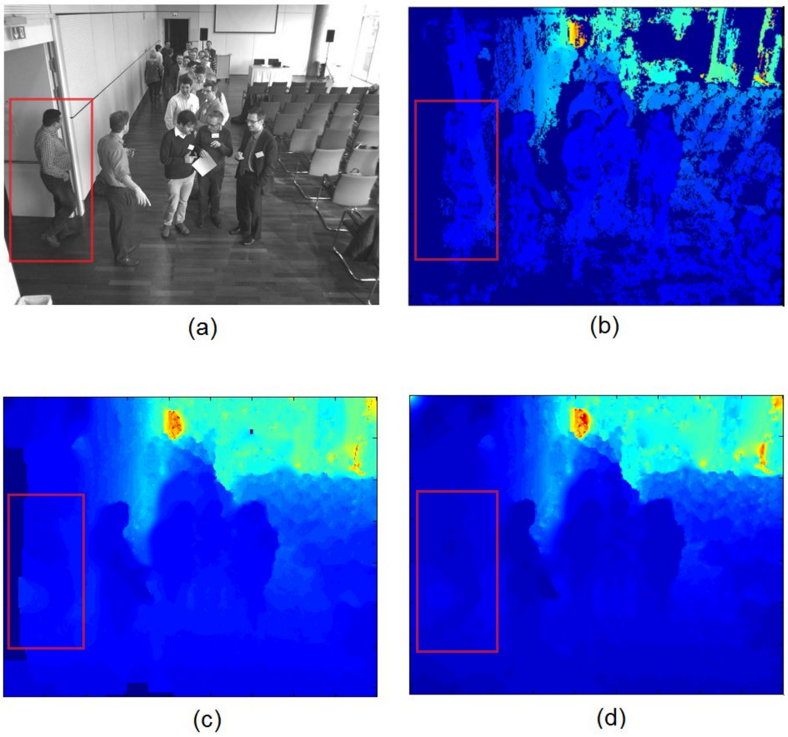


Figure 5.2: (a) The intensity image. (b) The unfilled depth map. (c) The depth map inpainted by the bilateral filter. (d) The depth map inpainted by the linear equation system of the spring model. The very left region of the depth map has undefined values. This region is shown as a sharp undefined boundary in the depth maps inpainted by the Joint Bilateral Filter, or a boundary merging into the background in the depth maps inpainted by the PDE systems.

3) Occlusions and Cluttered Regions

Since the training set provides samples with occlusions (see Chapter 3), the detector is able to detect occluded objects. However, in the image regions where occlusion and clutter take place, the annotation of a ground truth bounding box as well as the evaluation of a detected window become difficult. Small and occluded objects are hard to distinguish even with human eyes. Evaluation using the full set of annotations where each single person is annotated results in a high amount of false positives and false negatives in the cluttered region, as shown in Figure 5.3 panel (a). In such regions the following questions arise: Which objects shall be annotated? Which detected window is a true positive and which is a false positive?

In order to reduce the amount of false negatives in the cluttered region, the non-maximal suppression is applied to all the ground truth bounding boxes. Figure 5.3 panel (c) and (d) show the ground truth bounding boxes before and after the non-maximal suppression. One can observe that the non-maximal suppression eliminates the bounding boxes in the middle of the cluttered region, but preserves those at the border of the region. The objects contained in those unsuppressed bounding boxes exhibit more distinguishable forms than the suppressed ones. We select those bounding boxes as challenging examples for the detector.

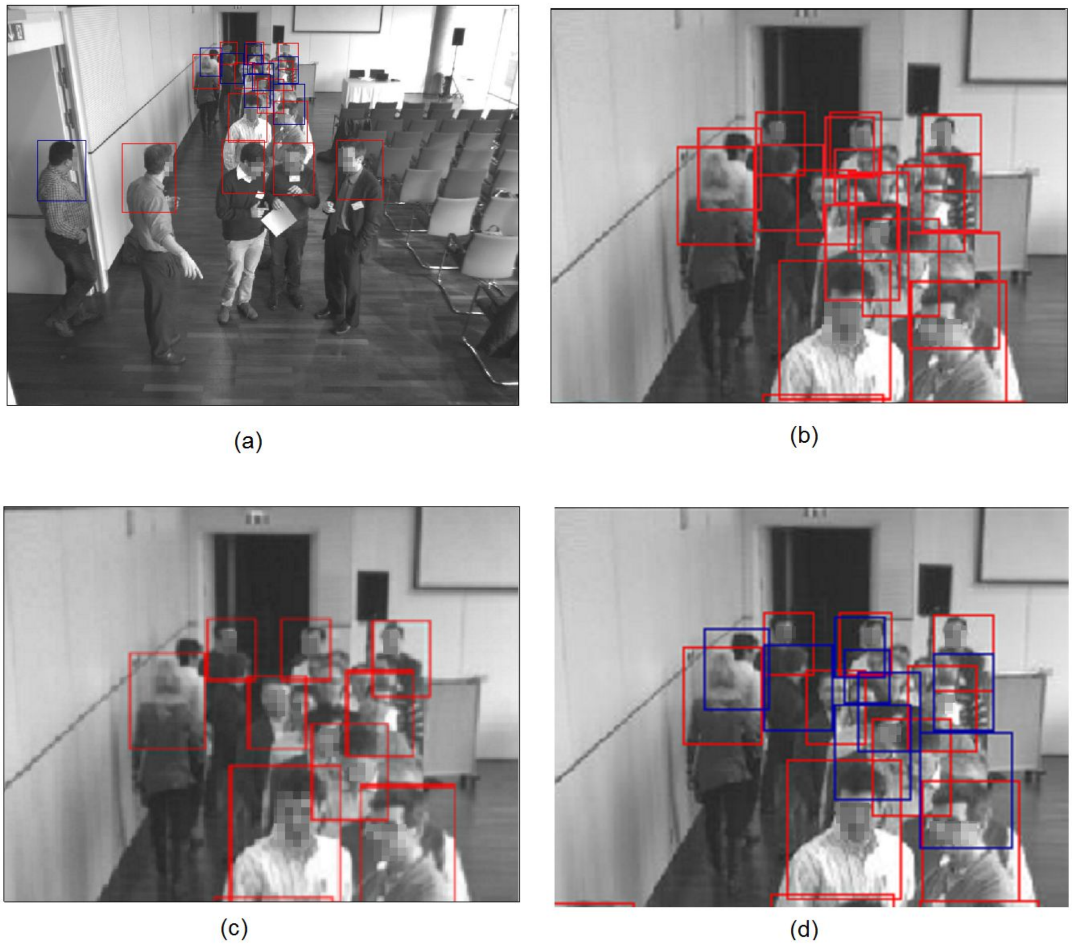
The amount of false negatives from the detected windows in the middle of the cluttered region can further be reduced by setting the "ignore" flag to the suppressed ground truth bounding boxes. Because the suppressed ground truth bounding boxes locate in the cluttered region, detected windows in such regions can be ignored. Figure 5.4 panel (a) shows the evaluation result using the reduced set of annotations, with the suppressed ground truth bounding boxes marked as "ignore", one false negative from the cluttered region turns to be ignored. Panel (b) shows the details of the mismatched cases. The undetected windows containing one person with partial occlusion are the challenging examples, therefore they are marked as false negatives. A detected window containing a group of 9 persons is determined as a false positive. A detected window containing a group of 3.5 persons located in the ignored region and is hence marked as "ignore".

Setting the flags of "ignore" to some of the ground truth annotations at certain regions aims to reduce the "false false positives", i.e. regions where the objects are successfully detected by the detector but reported as false positives by the evaluation. If some detected windows match the hard examples in the occluded and cluttered regions, nevertheless they are not counted for the evaluation. Therefore, the actual system accuracy shall be slightly higher than the computationally evaluated accuracy, taken into consideration that some potentially correct detections may be filtered out.

Algorithm 5.1 shows how the ground truth bounding boxes are filtered.

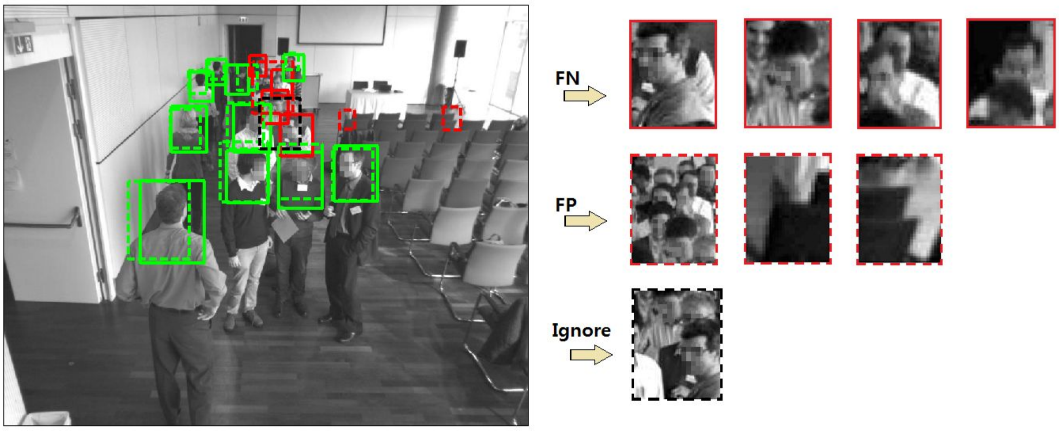
Algorithm 5.1: Ground truth filtering

```
For each frame
    Load all ground truth bounding boxes GT_BBs
    Resize GT_BBs: height = height, width = height/aspect_ratio
    non-maximal Suppression on GT
        Set the suppressed GT_BBs to ignore
        Set the very left GT_BBs(x<100) to ignore
End for
```



(a) (b) (c) (d)

Figure 5.3: Non-maximum suppression of the ground truth annotations and setting "ignore" flags on the cluttered region. (a) The full set of ground truth annotations annotate each single person in the scene. This set is split into two categories: standard (red), and "ignore" (blue). After performing non maximum suppression on the full set of bounding boxes, the unsuppressed bounding boxes are the standard bounding boxes, the suppressed bounding boxes together with the bounding boxes on the very left side of the image (x<100 pixel) are marked as "ignore". (b) Details of the cluttered region with the full set of annotations, these bounding boxes are the challenging examples for the detector. (c) The standard BBs on the cluttered region. The non maximum suppression preserves the bounding boxes at the border of the clutter, and suppresses the bounding boxes inside. (d) Details of the cluttered region with filtered annotations. Only the red BBs in this region have to be matched, any detection that matches to the blue ones will be ignored.



(a) (b)

Figure 5.4: An example of the evaluation result based on a filtered set of ground truth annotations. (a) After performing the non maximum suppression on the full set of ground truth annotations, the unsuppressed bounding boxes are marked as "non-ignore", while the suppressed bounding boxes and the very left bounding box (x<100 pixel) are marked as "ignore". Detected windows which intersect those ground truth bounding boxes are also marked as "ignore" (black dotted), and will not affect the evaluation of the system accuracy. This further reduces the false positives in the occluded or cluttered regions. (b) shows the details of the mismatches in the left frame. The false negatives are those persons standing at the border of the clutter, which the detector fails to detect. These objects are the challenging examples for the detector. The false positives are two windows containing chairs and a window containing 9 persons. The ignored detection is a window containing 3.5 persons.

5.2.2 Setting the Treshold for Comparing Two BBs

By comparing the detectors, the very first question we are interested in is the overview of the performance of the detectors. Questions can be answered, for example does the additional cue of depth maps bring a higher detection rate? However, as Figure 5.5 shows, the detection accuracy represented as LAMR varies strongly as the threshold for comparing two BBs changes.

Dollár et al. [18] perform an extensive evaluation of the state-of-the-art detectors based on RGB channels, and state the following observation regarding the BBs overlap threshold and the LAMR:

> "The evaluation is insensitive to the exact overlap threshold used for matching so long as it is below ~0.6. This implies that nearly all detections that overlap the ground truth overlap it by at least half. However, as the threshold is further increased and higher localization accuracy is required, performance of all detectors degrades rapidly."

In Figure 5.5 the LAMR of the RGB detector displays a curve which stays almost unchanged below the threshold 0.6, as described above. However, all the RGBD detectors show a tendency that the LAMR increases as the threshold gets higher. This indicates that the RGBD detectors are very sensitive to the tolerance of the localization requirement. In order to address the problem of how to "reasonably" evaluate the performance of the detectors [18], a proper BBs overlap threshold shall be chosen to enable the LAMR to best describe the detection results.

In Figure 5.5 a prominent change of the curves of the detectors occurs between the threshold 0.3 and 0.5. When the BBs overlap threshold is below ~0.37, the RGB detector has the highest LAMR in comparison with the other RGBD detectors, however, when BBs overlap threshold is above 0.5, the RGB detector shows the lowest LAMR among all the detectors. We examine the evaluation results from the RGB detector and the RGBD-Bilateral HOG detector at the BBs overlap threshold 0.3 and 0.5 to analyse the reason for the increase of the LAMR of the RGBD detectors.

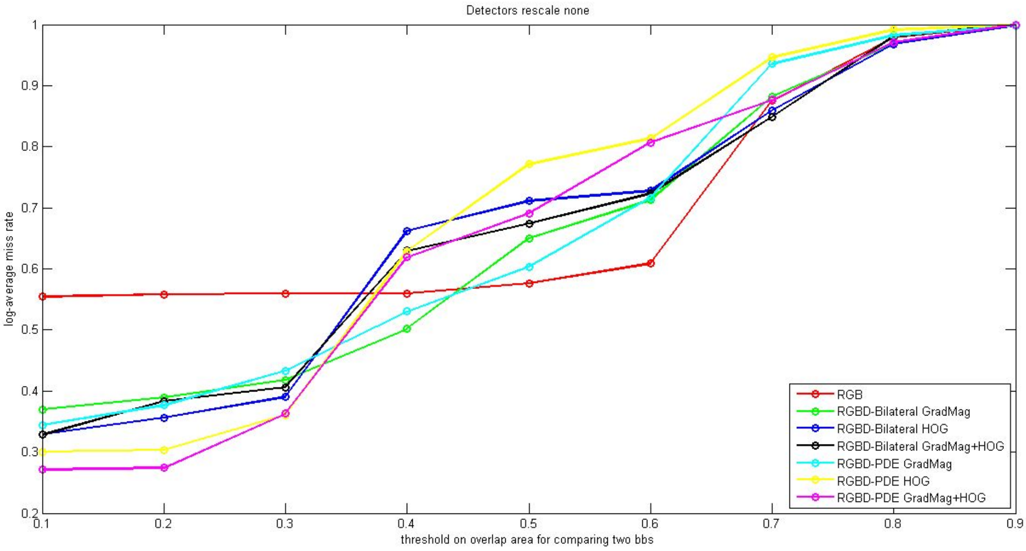
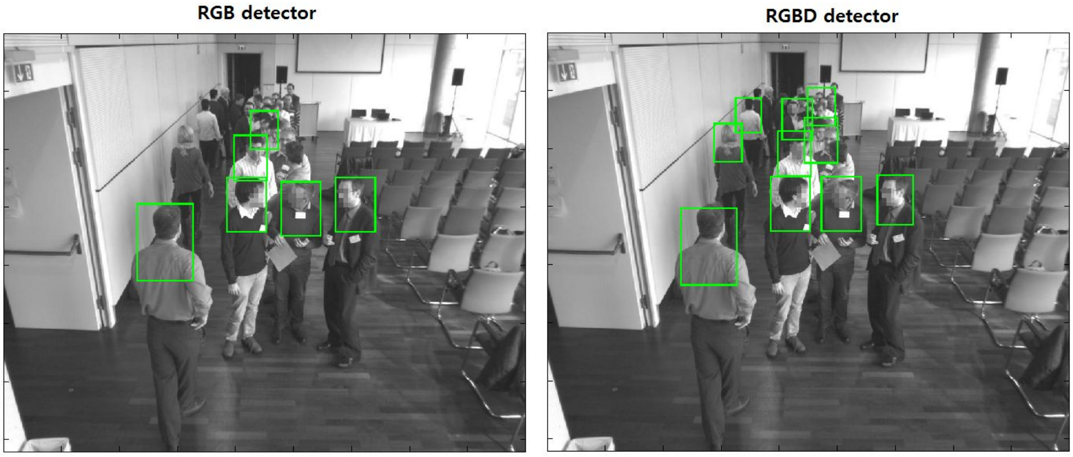


Figure 5.5: The LAMR as a function of threshold for comparing two BBs.

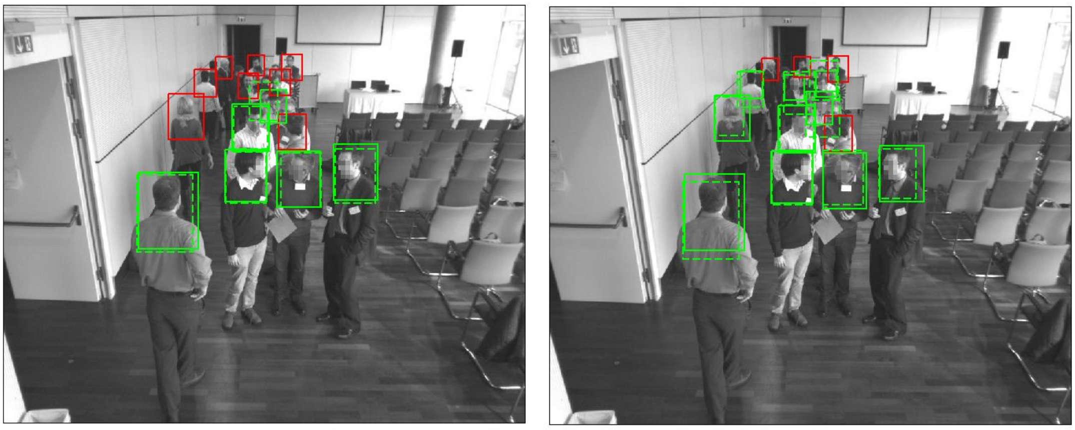
As Figure 5.6 shows, the RGBD-Bilateral HOG detector detects more objects than the RGB detector from the same scene. The objects detected by the RGB detector are almost all detected by the RGBD Detector (except one in the cluttered region of the scene), the RGBD Detector has an extended set of detection on objects in smaller scales and objects in the cluttered region of the scene. When the BBs overlap threshold is raised, detection windows for objects without occlusion are not affected, as observed by Dollár et al., nearly all detections that overlap the ground truth overlap it by at least half. The most affected detection windows are in the cluttered region of the scene. Less tolerated BBs overlap threshold causes the detection in the cluttered region to be evaluated as false positives, and at the same time, the unmatched ground truth BBs are evaluated as false negatives, hence the LAMR is increased in two ways. Since the RGB detector detects maximum one object in the cluttered region, the performance evaluation of the RGB detector under the threshold 0.6 stays almost unchanged, and starts to increase when the threshold is over 0.6 due to the higher requirements of localization accuracy. The RGBD detectors, on the other hand, are sensitive to the tolerance of the match criteria, because they are able to detect more objects in the cluttered region than the RGB detector.

In Figure 5.6 we observe that the evaluation results at the BBs overlap threshold 0.3 most satisfyingly describe the detection result. Taking into consideration that lower threshold may cause more false positives to be judged as true positives, in the following experiments we choose the BBs overlap threshold at 0.3.


RGB detector

RGBD detector

(a) Detection result


(b) Evaluation result with BBs overlap threshold 0.3

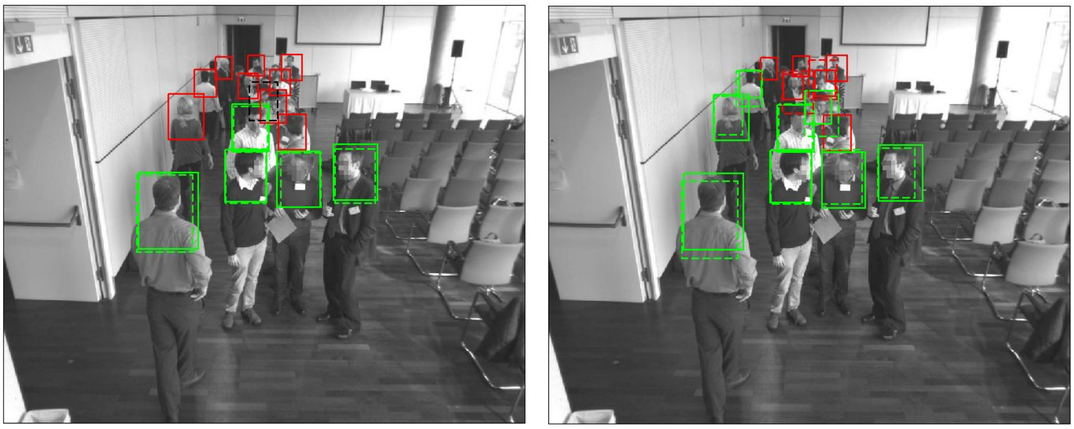

(c) Evaluation result with BBs overlap threshold 0.5

Figure 5.6: Evaluation of the RGB detector and the RGBD-Bilateral HOG detector at different BBs overlap thresholds. (a) The direct visual evaluation shows that the RGBD detector outperforms the RGB detector by being able to detect a higher amount of objects, especially the objects in the cluttered region of the scene. (b) Setting the BBs overlap threshold at 0.3 produces a satisfying computational evaluation result. (c) Setting the BBs overlap threshold at 0.5 causes the detection in the cluttered region to be evaluated as false positives and at the same time the unmatched ground truth BBs are evaluated as false negatives, hence the overall LAMR is raised in two ways.

5.2.3 Comparison of the Detectors

We rescale the entire pre-trained detectors, so that small objects in the scene can be detected. We make experiments with detectors at three rescale ratios: 1.0 (unchanged), 0.75, 0.5. The experiments at a each single scale is done twice, once with the scale estimation for all the detectors, and once without. Figure 5.7 displays the ROC curves of the performance of the detectors. Table 5.2 shows a list of the best performance in LAMR of each detector at the corresponding scales. Figure 5.8 displays the detection results of each detector at their best performance.

Detector	With Scale Estimation		Without Scale Estimation	
	Best Performance in LAMR (fps)	Detector Scale	Best Performance in LAMR (fps)	Detector Scale
RGB	55.97% (1.74fps)	1.0	**52.62%** (2.80fps)	1.0
RGBD-Bilateral GradMag	**41.79%** (0.50fps)	1.0	47.53% (2.11fps)	1.0, 0.75
RGBD-Bilateral HOG	**28.65%** (0.30fps)	0.75	38.14% (1.77fps)	1.0
RGBD-Bilateral GradMag+HOG	**33.79%** (0.28fps)	0.75	35.12% (1.65fps)	1.0
RGBD-PDE GradMag	**43.32%** (0.38fps)	1.0	54.17% (1.85fps)	1.0
RGBD-PDE HOG	**27.49%** (0.26fps)	0.75	47.96% (1.75fps)	1.0
RGBD-PDE GradMag+HOG	**29.34%** (0.26fps)	0.75	39.58% (1.68fps)	1.0

Table 5.2: The best performance of each detector in Log-Average Miss Rate (LAMR), with the corresponding scales and speed (frame per second), with and without the scale estimation. The numbers marked in bold are the lowest LAMR achieved by each detector. The best two detectors with the lowest LAMR among all the detectors are the RGBD-Bilateral HOG detector (28.65%) and the RGBD-PDE HOG detector (27.49%), both of them are RGBD detectors using the HOG feature and the scale estimation, rescaled to 0.75. The table shows that using the scale estimation reduces the detection speed significantly, but on the same time increases the detection accuracy.

We analyze the results in the following aspects:

RGB vs. RGBD:

Figure 5.7 shows that under the same settings (detector scale and scale estimation) with a total number of 6 tests, 75% (27 out of 36) of the RGBD detectors outperform the RGB detector by the LAMR. Table 5.2 shows that the best performance of all the 6 RGBD detectors outperform the best performance of the RGB detector, with a maximum decrease of the LAMR by 28.48%, and an average decrease of the LAMR by 18.55%. However, the false positive rate of the RGB detector is much lower than the RGBD detectors, this can be observed in Figure 5.7, the ROC curves of the RGB detector stop earlier at a low FPPI than the RGBD detectors.

Figure 5.8 provides an example of the detection results in the scene. One observes that the RGB detector shows limited ability to detect objects with occlusions (maximum 1 object with occlusion in each frame). The RGB detector performs best at the scale 1.0 (see Table 5.2), where the biggest objects are detected, and objects in smaller scales are not detected. In our experiments, rescaling the RGB detector to smaller scales increases the detection of small objects, but at the same time loses the detection of big objects, this gives rise to an increased LAMR (see Figure 5.7). The RGBD detectors produce more satisfying detection results by extending the BBs detected by the RGB detector in two ways: (1) With a tuning of the detector scale, the RGBD detectors can detect small objects in the back of the queue. (2) The RGBD detectors can detect the occluded objects in the cluttered region of the scene.

Scale estimation:

The overview of the performance of the detectors (Figure 5.7) indicates that the scale estimation on the detection brings a significant increase on the overall performance of the detectors. At scale 1.0, the RGB detector has a decrease of performance by 3.35%, the RGBD-Bilateral HOG detector has a decrease of performance by 0.88%, and the RGBD-Bilateral GradMag+HOG detector has a decrease of performance by 5.47%, while all the other detectors have an average increase of performance by 7.93%. At scale 0.75, the RGB detector maintains the same performance after performing the scale estimation, while all the other detectors have gained an average increase of performance by 24.61%. At scale 0.5, all the detectors have gained an average increase of performance by 24.36%. The best performance of each detector (Table 5.2, the numbers marked in bold) shows that, except that the RGB detector has a decrease of the best performance by 3.35% after performing the scale estimation, all the other 6 detectors have gained an average increase of their best performance by 9.68% after performing the scale estimation.

Depth maps filling algorithm Joint Bilateral Filter vs. PED:

In Figure 5.7 we observe that at the detector scale 1.0 and 0.75, the ROC curves of a pair of RGBD-Bilateral detector and the RGBD-PDE detector which have the same depth feature are close to each other or exhibit similar behaviors, with or without the scale estimation. Moreover, Table 5.2 shows that the best performance of a pair of RGBD-Bilateral detector and the RGBD-PDE detector corresponding to a same depth feature are similar to each other, with a difference of 1.53% for the feature type GradMag, 1.16% for the feature type HOG, and 4.45% for the combination of GradMag and HOG.

The above reported difference of LAMR can be observed in Figure 5.8. The RGBD-Bilateral detectors detect more amount of BBs than the RGBD-PDE detectors, the number of false positives detected by RGBD-Bilateral detectors are higher than the RGBD-PDE detectors. This difference can be seen also in Figure 5.7 at the detector scale 1.0 and 0.75. Before performing the scale estimation the ROC curves of the RGBD-Bilateral detectors and the RGBD-PED detectors all stop at a FPPI of ~9, however, after performing the scale estimation the ROC curves of the RGBD-Bilateral detectors stop earlier at lower FPPIs than the RGBD-PED detectors, indicating that the scale estimation eliminates more false positives for RGBD-Bilateral detectors than for the RGBD-PED detectors.

Feature types for the depth channels:

We use two types of features for the depth channels, the gradient magnitude (referred to as GradMag), the histogram of gradients (referred to as HOG), and a combination of these two features (referred to as GradMag+HOG). Table 5.2 shows that without the scale estimation, the best performance of all detectors is obtained by the GradMag+HOG feature, followed by the HOG feature, and last by the GradMag feature. With the scale eatimation the best performance of all the detectors is obtained by the HOG feature, followed by the GradMag+HOG feature, and last by the GradMag feature. The rankings stay the same for both the RGBD-Bilateral detectors and the RGBD-PED detectors. Dollár et al. state that no single feature outperforms HOG, and a combination of HOG and other types of features can gain a performance improvement over pure HOG [18]. This statement holds true in our experiments for detectors without the scale estimation. However, HOG performs better than GradMag+HOG by LAMR with the scale estimation.

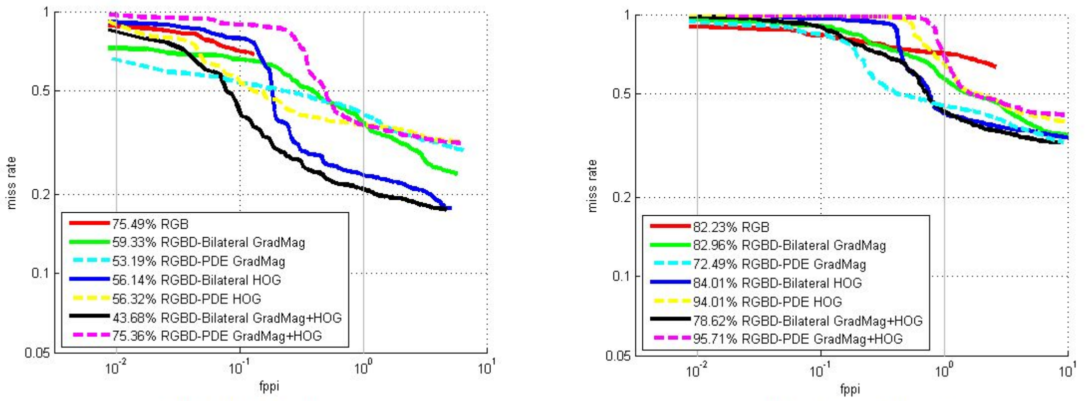
Speed:
Table 5.2 shows the following aspects regarding the detection speed. The RGB detector is faster than the RGBD detectors. Using the scale estimation reduces the detection speed significantly. Using the HOG feature for the depth maps is slightly faster than using the GradMag+HOG feature, but slower than using the GradMag feature. When the GradMag feature is used for the depth maps, the RGBD-Bilateral detectors are faster than the RGBD-PDE detectors, otherwise the RGBD detectors using these two inpainting algorithms have almost the same detection speed.

5.2.4 Discussion of the Comparison and the Figures

Based on the comparison and figures of the previous section, we can draw the following conclusions:

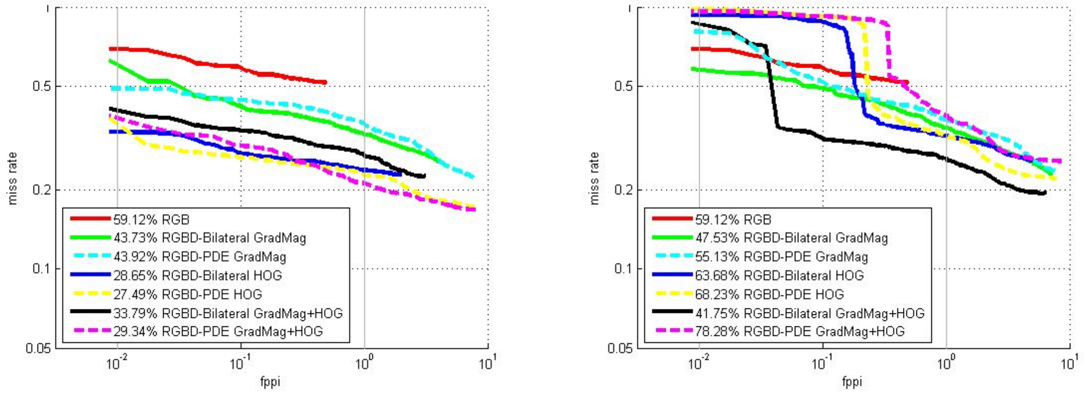
- The additional information from the depth channels improves the detection results from the detectors using purely RGB features. The RGBD detectors have extended ability in detecting objects with occlusions and objects in finer scales in comparison with the RGB detector.
- The scale estimation makes use of the spatial relationship and scene structure from the depth maps. In our experiments the training data and testing data are captured from the same scene, hence the camera setting is unchanged, which enables us to use directly the inversely proportional relationship between the size of the object and its distance to the camera for the scale estimation without modelling the other parameters of the camera and the scene.The amount of false positives is significantly reduced, because the sizes of the detection windows are limited to a range corresponding to the distances of the objects to the camera.
- The inpainting algorithm for the depth maps is also an influential factor for the performance of the detection, because the local features from the inpainted depth maps may show variations in details due to the different inpainting algorithms. A multitude of inpainting algorithms are developed for depth images because of the popularity of the low cost Microsoft Kinect sensor. Those algorithms are especially designed for the depth maps, aiming to refine the depth maps by reducing noise, preserving boundaries and filling holes [49], [60]. A detailed discussion is beyond the scope of our work. However, we believe that improving the inpainting algorithm for the depth maps can bring a better performance for the RGBD detectors in our future work.
- HOG proves to be an effective feature for the depth maps in our experiments. A combination of HOG and other types of features is favored by the researchers [20], [90], [58], [93], [88], [102]. We use gradient magnitude as an additional feature in combination with HOG. Without performing the scale estimation, the combination of GradMag+HOG outperforms HOG. However, HOG performs better than GradMag+HOG after the scale estimation. This motivates us to make experiments with other supporting features in addition to HOG in our future work.

(a) Detector rescaled to 0.5



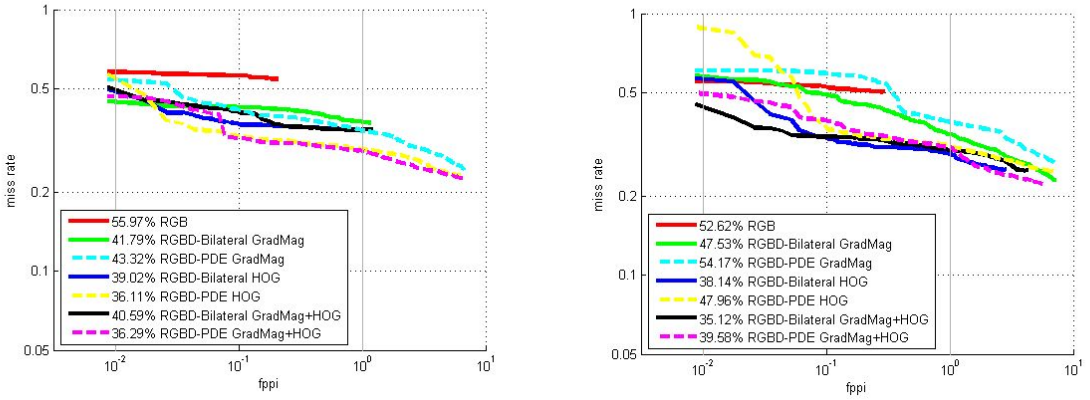
with Scale Estimation **without Scale Estimation**

(b) Detector rescaled to 0.75



with Scale Estimation **without Scale Estimation**

(c) Detector scale unchanged



with Scale Estimation **without Scale Estimation**

Figure 5.7: The ROC curves of the performances of the detectors at three scales: 1.0 (unchanged), 0.75, 0.5.

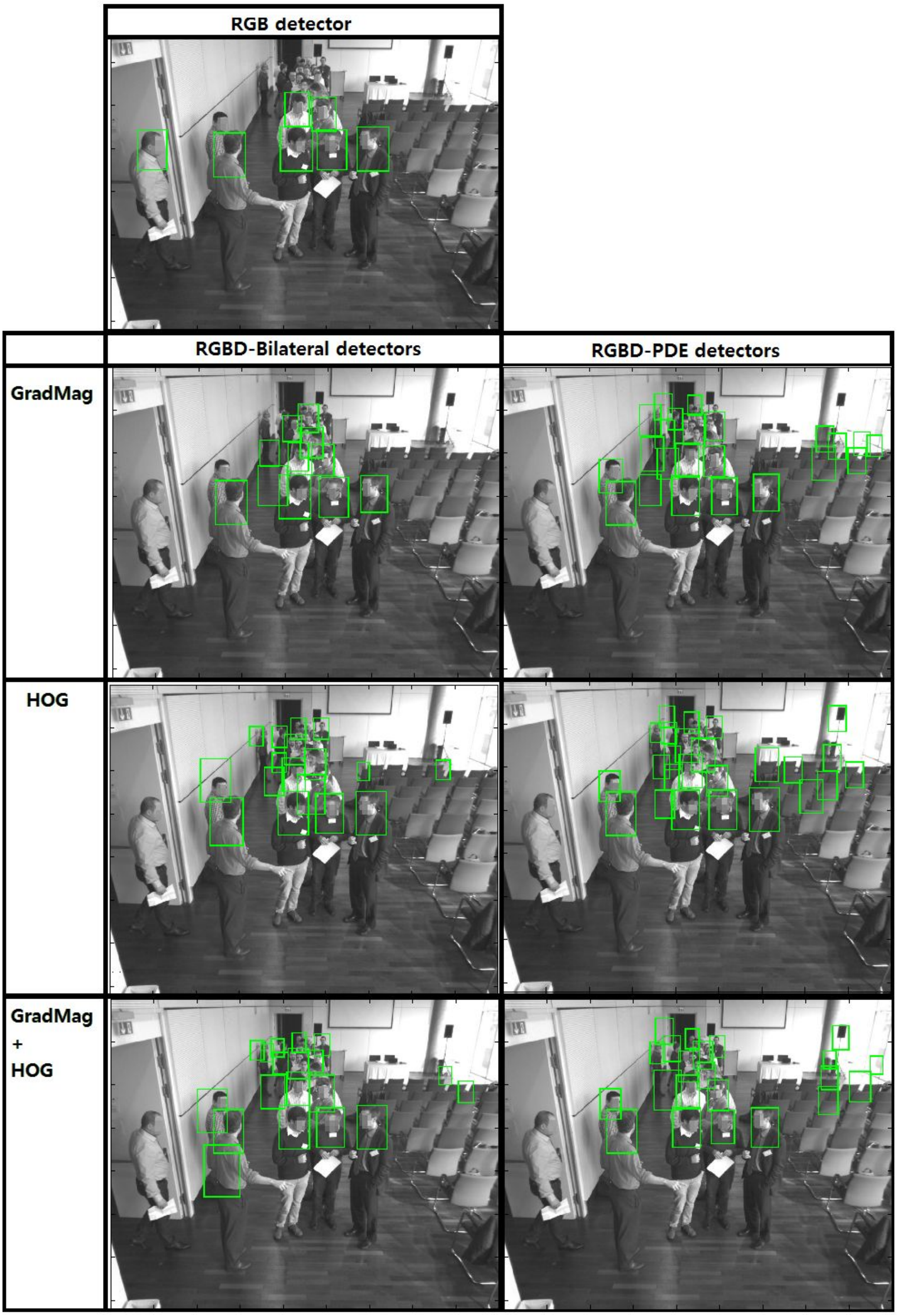


Figure 5.8: Demonstration of the best performance of each detector in the same scene. The RGB detector detects objects in big scales and without occlusion. The RGBD detectors outperform the RGB detector by being able to detect the objects in a wider range of scales and objects with occlusions.

5.3 Test Case 2: Multi-class Detection with the High-level Feature Fusion Scheme

In the last section, the RGBD detectors trained by the AIT-Dataset1 show satisfactory performance when tested on the testing data of the same scene. The RGBD detectors tested in the last section integrate the depth information as additional channels, i.e. the features from the RGB channels and the depth channels are fused together during the training stage.

In this section, we use AIT-Dataset2 as a second testing set. We construct a multi-class detector which combines two detectors trained independently on two different training sets, one is a RGB detector trained on a RGB training set, the other is a depth detector trained purely on the depth maps. The purpose of the multi-class detector is to exploit the limited source of data available for our work. As Table 5.3 shows, the testing set AIT-Dataset2 requires a detector which is able to detect full-body objects in the outdoor scene with depth information. None of the other three data sets can fully fulfill this requirement. Therefore we combine the information from different datasets by means of combining independently trained detectors. Table 5.4 gives an overview of the datasets used for each component detector of the multi-class detector:

Dataset	Image Modality	Number of Frames		Ground Truth Type	Scene Type
		pos	neg		
AIT-Dataset1	Grayscale&Depth	237	-	Head-and-shoulder	Indoor
AIT-Dataset2	Grayscale&Depth	87	90	Full-body	Outdoor
NYU-Depth V2 [80]	Grayscale&Depth	-	1360	-	Indoor
INRIA[12]	RGB	614	1218	Full-body	Indoor&Outdoor

Table 5.3: An overview of the details of the four datasets available for our work.

Multi-class Detector	Training Set		Testing Set
1) Depth Detector	Positive: AIT-Dataset1 Number: 121	Negative: AIT-Dataset2 Number: 90	AIT-Dataset2 Number: 177
2) RGB Detector	Positive: INRIA [12] Number: 614	Negative: INRIA [12] Number: 1218	

Table 5.4: The training sets and testing set used for the multi-class detector.

The RGB detector is trained on the INRIA Dataset, which provides rich samples of persons with different poses in various kinds of scenes. The detected objects are full-body persons. The advantage of this detector is that it is capable of detecting persons in different poses, and has a false positive rate close to 0 (see Figure 5.9). The weak point is that it has a higher LAMR than the depth detector, meaning that it detects only a limited number of the objects.

The training set for the depth detector takes the positive depth images from the AIT-Dataset1 and the negative depth images from the AIT-Dataset2. The negative images provide a rough description of the background of the scene. We use them directly in the training to provide the background information, in order to reduce the false positive in the background region. The detected objects are head-and-shoulders, which fit closely to the upper part of the body and appear narrower than the full-body BBs from the ground truth. In order to approximate the full-body BBs, we enlarge the width of the detected windows by a ratio of 1.2, and set the height of the detected windows as three times the length of the width. This general approximation can only fit part of the objects closely, because the objects have different aspect ratios of their bounding boxes, and they may appear in different poses which changes the aspect ratios of their BBs (see Figure 5.10). The advantage of the depth detector is that it has a higher detection rate than the RGB detector. The weak point is that the approximation of the full-body bounding boxes are unable to fit the objects correctly due to the various poses of the objects.

To combine the advantages of the RGB detector and the depth detector, the multi-class detector is constructed in a way that the detection result from the RGB detector is accepted in first priority. On objects in which the RGB detector fails, the detection result of the depth detector is accepted (see Section 4.4).

The multi-class detector achieves the lowest LAMR in comparison with its component detector. However, it is worth noting that the detection speed is sacrificed (see Table 5.5).

Detector	Performance LAMR (fps)
Depth Detector	11.53% (2.15fps)
RGB Detector	50.87% (1.90fps)
Multi-class Detector	8.70% (1.00fps)

Table 5.5: Speed of the multi-class detector and its component detectors.

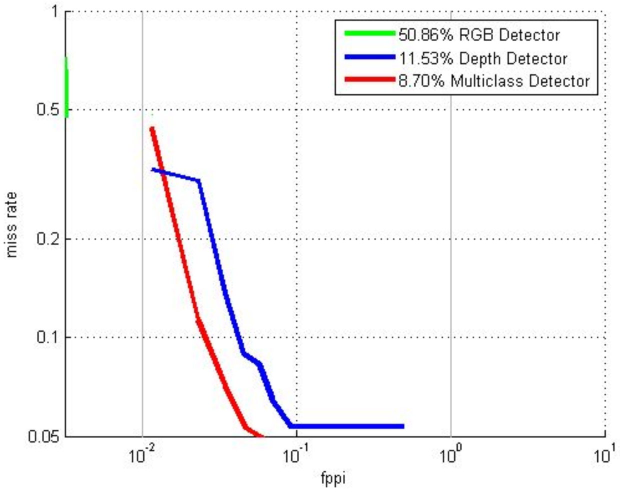


Figure 5.9: The ROC curves of the RGB detector, the depth detector, and the multi-class detector.

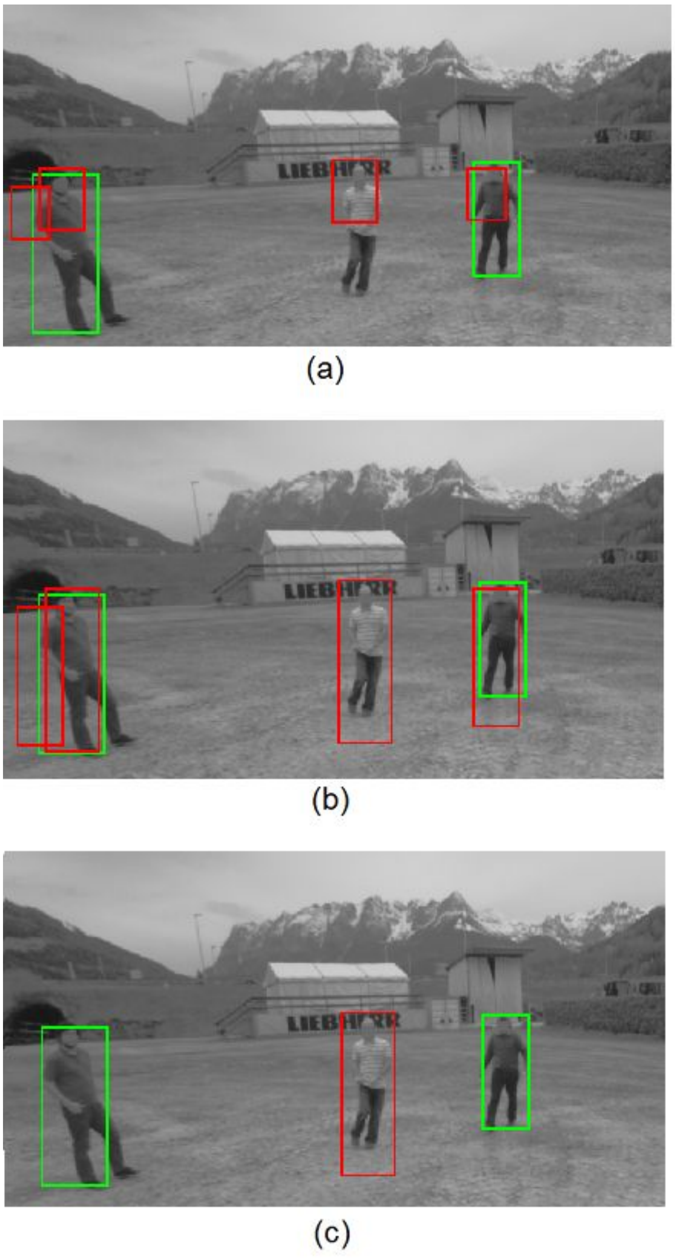


Figure 5.10: The combination of the RGB detector and the depth detector as a multi-class detector. (a) The multi-class detector detects head-and-shoulders (red) by the depth detector, and the full-body persons (green) by the RGB detector. (b) The BBs of the head-and-shoulders are enlarged to approximate the full-body BBs. However, due to the variety of poses and different aspect ratios of the objects, the approximation is not able to fit each individual object perfectly. The BBs from the RGB detector show a better result in fitting the objects. In simple terms, the RGB detector is more accurat than the depth detector, and the depth detector detects a larger amount of objects than the RGB detector. (c) Non-maximum suppression is performed on the detection set of (b). The BBs detected by the RGB detector are assigned a large score (1000) to suppress the BBs detected by the depth detector, so that the detection results of the RGB detector are accepted in first priority, while the depth detector supplements the detection results with further objects.

5.3 Summary

In this chapter we evaluate the RGBD detectors proposed in the previous chapters. We test the RGBD detectors on two scenarios. The RGBD detector with a low-level feature fusion scheme is tested on a scenario containing a crowded and cluttered situation. We show that using additional features from the depth data improves the detection accuracy in presence of occlusion. In the second scenario we construct a multi-class detector which combines a RGB detector and a depth detector in the classifier-level. The final detector outperforms each of its component detectors in detection accuracy. Both test cases show that the additional cues from the depth data improve the detection accuracy of the traditional intensity-based detectors.

Chapter 6
Conclusion

In this work we present a set of RGBD detectors extended from the Aggregated Channel Features (ACF) framework by Dollár et al. [20]. We exploit the information contained in the depth maps in the following two ways: We provide the training process with additional features from the depth maps and use the depth values to estimate the scales of detected objects. The depth features are incorporated into the intensity based object detection system at two levels, in the first test case the detectors combine all the features at a low level by constructing a joint feature space, in the second test case the detector trains two classifiers of different classes using the intensity features and the depth features separately and combines the decision at the classifier-level. We validate the assumption that the depth cue improves the detection accuracy, especially in presence of clutter and occlusions.

Scale estimation has proven to be effective for reducing the false positives by rejecting detected windows whose sizes are not within the estimated range. However, in our work we use the simplified model of an inversely proportional relationship between the distance of an object to the camera and its size, and we use the minimal depth value within a bounding box to represent the depth of that bounding box. The simplified model does not take into consideration the distribution of the scales of the object class and the uncertainty in depth measurements. Therefore, as represented in our first test case, the estimation function is best applicable to a testing set which has the same scene as the training set, with the same group of objects, and a camera fixed to a static position. To develop a more generalized scale estimation function, we will establish a more sophisticated statistical model proposed by [36] in our future work.

The inpainting algorithms of the depth maps have slight influence on the detection accuracy as discussed in Chapter 6. We have shown that detectors with the same settings but using depth maps inpainted with different algorithms make a difference in the detection accuracy. We believe that improved inpainting algorithms will improve the performance of the detection system.

Using HOG as the main feature for the depth maps has achieved satisfying performance in our experiments. We also have tried to combine HOG with gradient magnitudes. In our future work, we will make experiments with other supporting feature types in addition to HOG. Moreover, we will integrate additional cues to the detection system, for example cues from the thermal camera.

To further improve the performance of the system, a cascade-of-rejectors design can be added to the learning algorithm, as proposed by [86], [94] and [101].

Bibliography

[1] Adelson, E. H.; Anderson; C. H.; Bergen, J. R.; Burt, P. J.; Ogden, J. M., "Pyramid methods in image processing." In *RCA engineer*, 29.6 (1984): 33-41.

[2] Agarwal A.; Triggs B., "Hyperfeatures – multilevel local coding for visual recognition." In *ECCV 2006*, pp. 30–43.

[3] Banerjee, S. (2008). "Projective geometry, camera models and calibration". In *Department of Computer Science and Engineering Indian Institute of Technology Delhi* [Online]. Available: http://www.cse.iitd.ernet.in/~suban/vision/geometry/geometry.html (last accessed on 2014, August 18).

[4] Beleznai, C.; Gemeiner, P.; Zinner, C., "Reliable Left Luggage Detection Using Stereo Depth and Intensity Cues." In *ICCVW*, pp. 59-66, 2013.

[5] Beleznai, C. (2014), "Visual Surveillance of Humans." Unpublished manuscript.

[6] Beleznai, C. (2015), "Challenges in object detection." Unpublished figure.

[7] Bertalmio, M.; Sapiro, G.; Caselles, V.; Ballester, C., "Image inpainting." In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 2000.

[8] Broussard, R. P.; Rogers; S. K.; Oxley, M. E.; Tarr, G. L., "Physiologically motivated image fusion for object detection using a pulse coupled neural network." In *IEEE Transactions on Neural Networks,* 10.3 (1999): 554-563.

[9] Cai, J.; Goshtasby A., "Detecting human faces in color images." In *Image and Vision Computing*, 18.1 (1999): 63-75.

[10] Cannon, J. R. (1984). The one-dimensional heat equation (Vol. 23). Cambridge University Press.

[11] Choi, W.; Pantofaru, C.; Savarese, S., "Detecting and tracking people using an rgb-d camera via multiple detector fusion." In *ICCV Workshops*, pp. 1076-1083, 2011.

[12] Dalal, N.; Triggs, B., "Histograms of oriented gradients for human detection." In *CVPR*, Vol. 1. pp. 886-893, 2005.

[13] Dalal, N.; Triggs B.; Schmid, C., "Human detection using oriented histograms of flow and appearance." In *Computer Vision–ECCV*, Vol. 1, pp. 886-893, 2006.

[14] D'Errico, J. "inpaint_nans" *Matlab Central* [online]. 13 Aug. 2012. Available: http://www.mathworks.com/matlabcentral/fileexchange/4551-inpaint-nans (last accessed on 10 Jul. 2014).

[15] Dollár P.; Tu Z.; Perona P.; Belongie S., "Integral Channel Features." In *BMVC*. Vol.2, No.3, pp.5, 2009.

[16] Dollár P.; Belongie S.; Perona P., "The Fastest Pedestrian Detector in the West." In *BMVC*. Vol.2, No.3, pp.7, 2010.

[17] Dollár P.; Appel R.; Kienzle W., "Crosstalk Cascades for Frame-Rate Pedestrian Detection." In *Computer Visio-ECCV 2012*, Springer Berlin Heidelberg. pp.645-659, 2012.

[18] Dollár P.; Wojek C.; Schiele B.; Perona P., "Pedestrian detection: An evaluation of the state of the art." In *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 34.4 (2012): 743-761.

[19] Dollár, P., "Piotr's image and video Matlab Toolbox (PMT)." *Software available at: http://vision. ucsd. edu/~ pDollár/toolbox/doc/index. html* (2013).

[20] Dollár P.; Appel R.; Belongie S.; Perona P., "Fast Feature Pyramids for Object Detection," In *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 36.8 (2014): 1532-1545.

[21] Duin R. P. W.; Tax, D. M. J., "Experiments with classifier combining rules." In *Multiple Classifier Systems*, pp. 16–29, 2000. Springer Berlin Heidelberg.

[22] Elgammal A., "Probabilistic tracking in joint feature-spatial spaces." In *IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 1, pp. I-781, 2003.

[23] Etemad, K.; Chellappa, R., "Discriminant analysis for recognition of human face images." In *JOSA A* 14.8 (1997): 1724-1733.

[24] Ess, A.; Leibe, B.; Gool, L. V., "Depth and appearance for mobile scene analysis." In *ICCV 2007*, pp. 1-8.

[25] Everingham, M.; Van Gool, L.; Williams, C. K.; Winn, J.; Zisserman, A., "The pascal visual object classes (voc) challenge." In *International journal of computer vision*, 88.2 (2010): 303-338.

[26] Fang, J. Q.; Huang, T. S., "A Corner Finding Algorithm for Image Analysis and Registration." In *AAAI*, pp. 46-49, 1982.

[27] Faugeras, O., "Three-dimensional computer vision: a geometric viewpoint." In *MIT press*, 1993.

[28] Ferrari, V.; Tuytelaars, T.; Van Gool, L., "Simultaneous object recognition and segmentation by image exploration." In *Computer Vision-ECCV 2004*, pp. 40-54. Springer Berlin Heidelberg.

[29] Farshbaf, B. (2011) "HOG (Histogram of Oriented Gradients) with Matlab Implementation." In *BFD* [Online]. Available: http://farshbafdoustar.blogspot.co.at/2011/09/hog-with-matlab-implementation.html (last accessed on 2014, August 01).

[30] Breiman, L., "Random forests." In *Machine learning* 45.1 (2001): 5-32.

[31] Fawcett, T., "An introduction to ROC analysis." In *Pattern recognition letters*, 27.8(2006): 861-874.

[32] Fedorovskaya, E. A.; Blommaert, F. J.; Ridder, H. D., "Perceptual quality of color images of natural scenes transformed in CIELUV color space." In *Color Imaging Conf., IS&T/SID*, pp. 37–40, 1993.

[33] Flohr F.; Gavrila D. M., "PedCut: an iterative framework for pedestrian segmentation combining shape models and multiple data cues." In *Proc. BMVC,* pp. 66-1, 2013.

[34] Gall J.; Yao A.; Razavi N.; Van Gool L.; & Lempitsky V., "Hough forests for object detection, tracking, and action recognition." In *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 33.11 (2011): 2188-2202.

[35] Geurts, P.; Ernst D.; Wehenkel L., "Extremely randomized trees." In *Machine learning* 63.1 (2006): 3-42.

[36] Helmer, S.; Lowe, D., "Using stereo for object recognition." In *ICRA*, pp. 3121-3127, 2010.

[37] Hensley, J.; Scheuermann, T.; Coombe, G.; Singh, M.; Lastra, A., Fast Summed ‐ Area Table Generation and its Applications. In *Computer Graphics Forum*, Vol. 24, No. 3, pp. 547-555, 2005.

[38] Hu, R.; Barnard, M.; Collomosse, J., “Gradient field descriptor for sketch based retrieval and localization.” In *ICIP*, pp. 1025-1028, 2010.

[39] Javed, O.; Khurram S.; Mubarak S., "A hierarchical approach to robust background subtraction using color and gradient information." In *IEEE Workshop on Motion and Video Computing*, pp. 22-27, 2002.

[40] Ke, Y.; Sukthankar, R., “PCA-SIFT: A more distinctive representation for local image descriptors.” In *CVPR 2004.* Vol. 2, pp. II-506.

[41] Keller, C.G.; Enzweiler, M.; Rohrbach, M.; Fernandez Llorca, D.; Schnorr, C.; Gavrila, D.M., "The Benefits of Dense Stereo for Pedestrian Detection," In *IEEE Transactions on Intelligent Transportation Systems,* 12.4 (2011): 1096-1106.

[42] Kim, T.; Shotton, J.; Stenger B., “Boosting and random forest for visual recognition.” In *ICCV 2009.*

[43] Kitchen, L.; Azriel R., "Gray-level corner detection." In *Pattern recognition letters* 1.2 (1982): 95-102.

[44] Kopf J.; Cohen M. F.; Lischinski D.; Uyttendaele M., “Joint Bilateral Upsampling.” In *Acm Transactions On Graphics*, Vol. 26. No. 3. Pp.96, 2007.

[45] Lachenbruch, Peter A. In *Discriminant analysis*. John Wiley & Sons, Inc., 1975.

[46] Lai, K.; Bo, L.; Ren, X.; Fox, D., “A large-scale hierarchical multi-view rgb-d object dataset.” In *ICRA,* pp. 1817-1824, 2011.

[47] Lasserre, J. A.; Bishop, C. M.; Minka, T. P., “Principled hybrids of generative and discriminative models.” In *CVPR*, Vol. 1, pp. 87-94, 2006.

[48] Levkowitz, H., “Color theory and modeling for computer graphics, visualization, and multimedia applications.” Vol. 402. Springer Science & Business Media, 1997.

[49] Le, A. V.; Jung, S. W.; Won, C. S., "Directional Joint Bilateral Filter for Depth Images." In *Sensors*, 14.7 (2014): 11362-11378.

[50] Lester, J.; Choudhury, T.; Kern, N.; Borriello, G.; Hannaford, B., “A Hybrid Discriminative/Generative Approach for Modeling Human Activities.” In *IJCAI*, Vol. 5, pp. 766-772, 2005.

[51] Lestideau, F., "Face detection in color images with complex background." U.S. Patent No. 7,035,456. 25 Apr. 2006.

[52] Levin, A.; Lischinski, D.; Weiss, Y., “Colorization using optimization.” In *ACM Transactions on Graphics*, Vol. 23, No. 3, pp. 689-694, 2004.

[53] Lewicki, M.; Sejnowski T., "Learning overcomplete representations." In *Neural computation* 12.2 (2000): 337-365.

[54] Lienhart, R.; Liang, L.; Kuranov, A., “A detector tree of boosted classifiers for real-time object detection and tracking”. In *ICME 2003.* Vol. 2, pp. II-277, *2003.*

[55] Liter, J. C.; Bülthoff, H. H., “An introduction to object recognition”. In *Zeitschrift fur Naturforschung C-Journal of Biosciences*, 53.7 (1998): 610-621.

[56] Lowe, David G. "Distinctive image features from scale-invariant keypoints." In *International journal of computer vision* 60.2 (2004): 91-110.

[57] Marr, D.; Ellen H., "Theory of edge detection." In *Proceedings of the Royal Society of London. Series B. Biological Sciences* 207.1167 (1980): 187-217.

[58] Mori, G.; Belongie, S.; Malik, J., "Efficient shape matching using shape contexts." In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27.11(2005): 1832-1837.

[59] The MathWorks, Inc. (1994-2015). "91". In *MathWorks* [Online]. Available: http://www.mathworks.de/de/help/matlab/ref/91.html (last accessed on 2014, August 18).

[60] Matsuo, T.; Fukushima, N.; Ishibashi, Y., "Weighted Joint Bilateral Filter with Slope Depth Compensation Filter for Depth Map Refinement." In *VISAPP (2)*, pp. 300-309, 2013.

[61] McCallum, A.; Pal, C.; Druck, G.; Wang, X., "Multi-conditional learning: Generative/discriminative training for clustering and classification." In *AAAI*, Vol. 21, No. 1, pp.433, 2006.

[62] Mikolajczyk, K.; Schmid C., "A performance evaluation of local descriptors." In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27.10 (2005): 1615-1630.

[63] Opelt, A.; Fussenegger, M.; Pinz, A.; Auer, P., "Weak hypotheses and boosting for generic object detection and recognition." In *Computer Vision-ECCV 2004*, pp.71-84. Springer Berlin Heidelberg.

[64] Opelt, A.; Pinz, A., Graz 01 data set. On Web, 2004. URL http://www.emt.tugraz.at/~pinz/data/GRAZ 01/.

[65] Pang, Y.; Yuan, Y.; Li, X.; Pan, J., "Efficient HOG human detection." In *Signal Processing* 91.4 (2011): 773-781.

[66] Papageorgiou, C.; Poggio, T., "A trainable system for object detection." In *IJCV*, 38.1 (2000): 15-33.

[67] Porikli, F., "Integral histogram: A fast way to extract histograms in cartesian spaces." In *CVPR 2005.* Vol. 1, pp. 829-836.

[68] Rao, M. A.; Vázquez, D.; López, A. M., "Color contribution to part-based person detection in different types of scenarios." In *Computer Analysis of Images and Patterns*, pp. 463-470, 2011. Springer Berlin Heidelberg.

[69] Richard, M. M. O. B. B.; Chang, M. Y. S., "Fast digital image inpainting." In *VIIP 2001*, pp.106-107.

[70] Rohrbach M.; Enzweiler M.; Gavrila D. M., "High-level fusion of depth and intensity for pedestrian classification." In *Pattern Recognition*, pp.101-110, 2009. Springer Berlin Heidelberg.

[71] Rowley, H. A.; Baluja, S.; Kanade, T., "Neural network-based face detection." In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20.1 (1998): 23-38.

[72] Ruderman, D. L. "The statistics of natural images." In *Network: computation in neural systems* 5.4 (1994): 517-548.

[73] Ruderman, D. L., "Origins of scaling in natural images." In *Vision research* 37.23 (1997): 3385-3398.

[74] Safavian, S. R.; Landgrebe D., "A survey of decision tree classifier methodology." In *IEEE transactions on systems, man, and cybernetics* 21.3 (1991): 660-674.

[75] Schapire, R. E., "The boosting approach to machine learning: An overview." In *Nonlinear estimation and classification*, pp. 149-171, 2003. Springer New York.

[76] Schönlie, C. B. (2012). "Applying Modern PDE Techniques to Digital Image Restoration." In *MathWorks* [Online]. Available: http://www.mathworks.de/company/newsletters/articles/applying-modern-pde-techniques-to-digital-image-restoration.html (last accessed on 2014, July 23).

[77] The scikit-image development team. "Histogram of Oriented Gradients." In *Scikit-image* [Online]. Available: http://scikit-image.org/docs/0.10.x/auto_examples/plot_hog.html (last accessed on 2015, Feb 13).

[78] Shafarenko, L.; Petrou, M.; Kittler, J., "Automatic watershed segmentation of randomly textured color images." In *IEEE Transactions on Image Processing,* 6.11 (1997): 1530-1544.

[79] Shotton, J.; Sharp, T.; Kipman, A.; Fitzgibbon, A.; Finocchio, M.; Blake, A.; Moore, R., "Real-time human pose recognition in parts from single depth images." In *Communications of the ACM*, 56.1(2013): 116-124.

[80] Silberman N.; Kohli P.; Hoiem, D; Fergus R., NYU-Depth dataset V2. On Web, 07.04.2013. URL: http://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html. (last accessed on 2014, June 10).

[81] Sun Y.;, Wang Y.; Wong, A. K. C., "Boosting an associative classifier." In *IEEE Trans. Knowledge and Data Engineering*, vol. 18, pp. 988-992, 2006.

[82] Tkalcic, M.; Tasic, J. F., "Colour spaces: perceptual, historical and applicational background." In *Eurocon*, pp. 304-308. 2003.

[83] Thongkam, J.; Xu, G.; Zhang, Y., "AdaBoost algorithm with random forests for predicting breast cancer survivability." In *IJCNN 2008*, pp. 3062-3069.

[84] Tomasi, C.; Manduchi, R., "Bilateral filtering for gray and color images." In *Computer Vision*, pp. 839-846, 1998.

[85] Ulusoy, I.; Bishop, C.M., "Generative versus discriminative methods for object recognition." In *CVPR,* Vol. 2, pp. 258-265, 2005

[86] Viola, P.; Jones, M., "Rapid object detection using a boosted cascade of simple features." In *CVPR.* Vol.1, pp.I-511, 2001.

[87] Walk, S.; Majer, N.; Schindler, K.; Schiele, B., "New features and insights for pedestrian detection." In *CVPR*, pp. 1030-1037, 2010.

[88] Wang, X.; Han, T. X.; Yan, S., "An HOG-LBP human detector with partial occlusion handling." In *IEEE 12th International Conference on Computer Vision*, pp. 32-39, 2009.

[89] Wöhler C.; Anlauf J. K., "A time delay neural network algorithm for estimating image-pattern shape and motion." In *Image and Vision Computing*, 17.3 (1999): 281-294.

[90] Wojek C.; Schiele, B., "A Performance Evaluation of Single and Multi-Feature People Detection." In *Pattern Recognition*, pp. 82-91. Springer Berlin Heidelberg.

[91] Wojek, C.; Walk, S.; Schiele, B., "Multi-cue onboard pedestrian detection." In *CVPR 2009*, pp. 794-801.

[92] Wojek, C.; Walk, S.; Roth, S.; Schiele, B., "Monocular 3D scene understanding with explicit occlusion reasoning." In *CVPR 2011*, pp. 1993-200.

[93] B. Wu and R. Nevatia, "Optimizing Discrimination-Efficiency Tradeoff in Integrating Heterogeneous Local Features for Object Detection," In *CVPR*, 2008.

[94] Wu, J.; James M. R.; Mullin, M. D., "Learning a rare event detection cascade by direct feature selection." In *Advances in Neural Information Processing Systems*, pp. None. 2003.

[95] Wu, Y. W.; Ai, X. Y., “Face detection in color images using AdaBoost algorithm based on skin color information.” In *WKDD 2008*, pp. 339-342, 2008.

[96] Xia, L.; Chen, C. C.; Aggarwal, J. K., “Human detection using depth information by kinect.” In *CVPR Workshops*, pp. 15-22, 2011.

[97] Ye, Y.; Ci, S.; Liu, Y.; Wang, H.; Katsaggelos, A. K., "Binocular video object tracking with fast disparity estimation." In *AVSS*, pp. 183-188, 2013.

[98] Zhang, J.; Liu, Y.; Ha, S. W., “A novel approach of face detection based on skin color segmentation and PCA.” In *ICYCS 2008*, pp. 1006-1011.

[99] Zhao, W.; Krishnaswamy, A.; Chellappa, R.; Swets, D. L.; Weng, J., "Discriminant analysis of principal components for face recognition." In *Face Recognition*. Springer Berlin Heidelberg, pp.73-85,1998 .

[100] Zhou, M.; Wei, H., “Face Verification Using GaborWavelets and AdaBoost.” In *the Eighteenth international Conference on Pattern Recognition*, pp. 404-407, 2006.

[101] Zhu, Q.; Yeh, M. C.; Cheng, K. T.; Avidan, S., “Fast human detection using a cascade of histograms of oriented gradients.” In *CVPR*, Vol. 2, pp. 1491-1498, 2006.

[102] Ojala T.; Pietikainen M.; Maenpaa T., “Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns.” In *Pattern Analysis and Machine Intelligence*, 24.7 (2002): 971-987.