Efficient Pedestrian Detection in Urban Traffic Scenes

Dissertation

zur

Erlangung des Doktorgrades (Dr. rer. nat.)

der

Mathematisch-Naturwissenschaftlichen Fakultät

der

Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von

Shanshan Zhang

aus

Jiangxi, V.R. China

Bonn, 2014

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der
Rheinischen Friedrich-Wilhelms-Universität Bonn.

Erstgutachter: Prof. Dr. Armin B. Cremers, Bonn
Zweitgutachter: Prof. Dr. Christian Bauckhage, Bonn
Tag der Promotion: *20.02.2015*
Erscheinungsjahr: *2015*

Abstract

Pedestrians are important participants in urban traffic environments, and thus act as an interesting category of objects for autonomous cars. Automatic pedestrian detection is an essential task for protecting pedestrians from collision.

In this thesis, we investigate and develop novel approaches by interpreting spatial and temporal characteristics of pedestrians, in three different aspects: shape, cognition and motion.

The special up-right human body shape, especially the geometry of the head and shoulder area, is the most discriminative characteristic for pedestrians from other object categories. Inspired by the success of Haar-like features for detecting human faces, which also exhibit a uniform shape structure, we propose to design particular Haar-like features for pedestrians. Tailored to a pre-defined statistical pedestrian shape model, Haar-like templates with multiple modalities are designed to describe local difference of the shape structure.

Cognition theories aim to explain how human visual systems process input visual signals in an accurate and fast way. By emulating the center-surround mechanism in human visual systems, we design multi-channel, multi-direction and multi-scale contrast features, and boost them to respond to the appearance of pedestrians. In this way, our detector is considered as a top-down saliency system.

In the last part of this thesis, we exploit the temporal characteristics for moving pedestrians and then employ motion information for feature design, as well as for regions of interest (ROIs) selection. Motion segmentation on optical flow fields enables us to select those blobs most probably containing moving pedestrians; a combination of Histogram of Oriented Gradients (HOG) and motion self difference features further enables robust detection.

We test our three approaches on image and video data captured in urban traffic scenes, which are rather challenging due to dynamic and complex backgrounds. The achieved results demonstrate that our approaches reach and surpass state-of-the-art performance, and can also be employed for other applications, such as indoor robotics or public surveillance.

Überblick

Fußgänger sind wichtige Teilnehmer im Stadtverkehr-Umgebungen, und damit eine interessante Kategorie von Objekten für autonome Fahrzeuge zu handeln. Automatische Personenerkennung ist allgemein eine wesentliche Aufgabe zum Schutz von Fußgängern aus Kollision.

In dieser Arbeit werden wir neue Ansätze untersuchen und entwickeln, für die Interpretation räumlicher und zeitlicher Eigenschaften von Fußgängern, unter drei verschiedenen Aspekten: Form, Wahrnehmung und Bewegung.

Die besondere menschlicher Körperform, vor allem die Geometrie des Kopf- und Schulterbereichs, ist das diskriminierende Merkmal für Fußgänger im Vergleich zu anderen Objektkategorien. Inspiriert durch den Erfolg der Haar-ähnlichen Funktionen zur Gesichtserkennung mit einer ebenfalls einheitlichen Formstruktur, schlagen wir vor, insbesondere Haar-ähnliche Funktionen für Fußgänger zu entwerfen. Zu einem vorher festgelegten statistischen Fußgängerformmodell werden Haar-Vorlagen mit mehreren Modalitäten entwickelt, um lokale Unterschiede der Formstruktur zu beschreiben.

Kognitionstheorien liefern Erklärungen wie menschliche Seh-Systeme visuelle Signale auf eine genaue und schnelle Weise verarbeiten. Durch die Emulation des Center-Surround-Mechanismus von Seh-Systemen entwickeln wir Multi-Channel, Multi-Richtung und Multi-Skalen-Kontrast-Funktionen, und verbessern sie, um das Aussehen von Fußgängern zu erfassen. Auf diese Weise wird unser Detektor zu einem top-down Salienz-System.

Im letzten Teil dieser Arbeit nutzen wir die zeitlichen Eigenschaften der Bewegung von Fußgängern und verwenden Bewegungsinformationen für den Merkmalsentwurf sowie für die ROIs Auswahl. Bewegungs-Segmentierung der optischen Flussfelder ermöglicht es uns, diejenigen Blobs auszuwählen, die wahrscheinlich Fußgänger enthalten; eine Kombination von HOG Merkmalen und Bewegungsdifferenz-Funktionen ermöglicht weiterhin eine robuste Detektion.

Wir testen unsere drei Ansätze auf Bild- und Videodaten von Verkehrsszenen im Freien, die wegen der dynamischen und komplexen Hintergründe herausfordernd sind. Die erzielten Ergebnisse zeigen, dass unsere Ansätze state-of-the-Art-Leistung erreichen und übertreffen sowie auch für andere Anwendungen, wie z.B. Indoor-Robotik oder öffentlicher Überwachung eingesetzt werden können.

Acknowledgments

Before I started, many friends warned me of great difficulty of studying towards a Ph.D. degree in computer science especially in Germany; but I encouraged myself to take this valuable opportunity and pushed myself to limits. Fortunately, I have come here and I would like to wholeheartedly thank those people, without whose support and cooperation I cannot finish my thesis in time.

First of all, I would like to thank Prof. Dr. Armin B. Cremers for his support during the past years of my study and research and valuable advice during this time. In particular, I want to thank him for giving me the opportunity to meet people and present my work on several international conferences. I furthermore want to express my gratitude to Prof. Dr. Christian Bauckhage, who regularly arranged individual discussions with me, and also helped me improve my scientific English writing. He not only inspired me to new ideas, but also encouraged me to publish high quality papers.

My study towards a Ph.D. degree in University of Bonn was mainly funded by the China Scholarship Council. I am sincerely grateful to the Chinese government for providing me with such a good opportunity to study in a famous university. During my stay in Germany, I improved my research skills, and meanwhile broadened my horizons.

During my Ph.D., I was affiliated with the Intelligent Vision Systems group, headed by Prof. Dr. Armin B. Cremers. I would like to thank my colleagues in the group: PD. Dr. Steinhage, Dr. Simone Frintrop, Dr. Jens Behley, Dominik A. Klein, Dr. Florian Schöler, and Germán Martín García. Moreover, I had a great time working with guest researchers from Huazhong University of Science and Technology: Dr. Ruicheng Yan, Lei Zhu, and Dr. Yin Chen, all supervised by Prof. Dr. Zhiguo Cao, who is the Dean of Department of Automation. I sincerely appreciate their great help in different ways since I came to Germany.

In particular, I want to thank my colleague Dominik A. Klein, who spent a lot of time on discussions with me, and helped me revise several papers, even during his parental leave.

Last but not least, a heartful thank-you goes to my family. My parents were always there when I was in need but I was not able to accompany with them for several years. In order to let me focus on my study, my mother did not even inform me of a very risky surgery in advance. Besides my parents, my husband Guangyu Li also gave me full support. He started his Ph.D. study at the same time as me, so we understood each other very well. We encouraged each other during the hardest days.

Contents

Chapter 1

Introduction

Thanks to the development of chip techniques, computers have become popular and played a more and more important role in our daily lives. Nowadays computers have been successfully and widely used for data processing and communications, which reduce manual labor and provide convenience to a large extent. In addition, people always expect more intelligent computers, which are capable of assisting our lives in an active manner. One may easily think of many applications of using computers as assistants. They watch children at home; they examine abnormal events in public places, such as airports; they warn the drivers of collision, etc. To realize such applications, we have to enable computers to ‘‘see” and to interpret our real world. To see, means to obtain the visual data from our real world. For this purpose, we just need to transfer the captured image data to the CPUs, which can be easily done through current hardware. To interpret, however, is much more complicated. One may argue that even a young child can easily recognize the surrounding environment, so it should be even easier for a modern computer which owns a rather high computational speed. Unfortunately, it is not the case. First, our human visual systems receive analogy signals, while computers receive digital signals in the form of binary bits. Second, each person is exposed to a large amount of visual data since he or she was born, but such an extensive training is hardly for a compute to reach. More important, how human visual systems efficiently learn to recognize various objects from such a large amount of data is still a mystery to computer scientists.

What we are trying to do is to teach computers to recognize different objects in a human-like way. In the last decades, researchers in the computer vision and pattern recognition communities have made great efforts to enable computers to understand the surrounding environment by analyzing and interpreting image or video data. Unfortunately, computers are still far behind human brains in performing such analysis and inference, in terms of

both accuracy and efficiency. There is still a long journey for compute vision scientists to go. However, this research is worthwhile and exciting because it would largely extend the capabilities of computers so as to serve people in a better way.

To achieve the final goal of understanding the real world, one of the most primary and important tasks is to detect objects, which are essential elements for each scenario. The capability of detecting objects would serve as pre-processing in an intelligent vision system, which is expected to identify some specific person, to recognize human behaviors or to understand even more complex social events.

One may ask, how many object categories do we have in our real world? In a popular public image database, namely ImageNet[1], more than $80,000$ object categories are defined according to the WordNet[2] hierarchy. But for many applications, the most interesting category is persons, because human beings actively participate in collected visual data, including private albums, and surveillance video. In particular, the scenario of public places is considered to be more important because it is usually concerned with security and safety. Therefore, we focus our study on people detection in public places.

In public places, people may have different postures, such as sitting on the grass, lying on a chair, or walking along or across the street, among which walking is in majority. Those people travelling on foot belong to the object category of pedestrians, which are important participants in public places, ranging from indoor to outdoor scenarios (see Section 1.1, ‘‘Applications’’ for details).

One may argue that other sensors can accomplish the task of people detection as well. For example, [Behley, 2013] achieved good performance for object classification in outdoor environments by analyzing 3D point cloud data scanned by laser. However, an obvious disadvantage of laser sensors is the high price, especially when precise and dense point cloud is required for real-world applications. Another option may be ultra sonic sensors, which are cheap and convenient to use. Unfortunately, they are too sensitive to noise and can hardly be used in complex outdoor traffic environments. Therefore, we consider cameras as a better choice, because cameras are of low-cost, and are able to acquire rich information of the surrounding environment.

In this thesis we address the problem of localizing pedestrians from image or video data, especially those taken from a camera mounted on a vehicle driving through regular traffic in an urban environment. This task is particularly challenging due to a number of reasons, which will be discussed in Section 1.2, ‘‘Challenges’’.

[1] http://www.image-net.org/
[2] http://wordnet.princeton.edu/

1.1 Applications

Pedestrian detection has attracted wide attention in academic communities over the last decades. This is mainly due to the various possible applications it has in different fields. Here, we list three popular applications:

- Advanced Driver Assistance Systems (ADASs): This is probably one of the most important and difficult applications. Pedestrians are very vulnerable participants in urban traffic. A World Health Organization report [Peden et al., 2004] describes traffic accidents as one of the major causes of death and injuries around the world, accounting for an estimated 1.2 million fatalities and 50 million injuries per year. According to the World Bank website[3], pedestrians account for 65% of the fatalities out of the 1.17 million traffic- related deaths around the world, with 35% of these being children. In China, pedestrians and bicyclists accounted for 27% and 23% of the fatalities, respectively, in 1994, compared to 13% and 2% in the United States [Mock et al., 2004]. Therefore, it is of major importance to protect pedestrians. The first and foremost goal is to detect pedestrians when they are still in a safe distance so as to avoid collision. The purpose of ADASs is to warn drivers of possible dangerous situations as early as possible, so that they have enough time for braking.
- Visual surveillance: Closed-circuit television (CCTV) cameras are commonly installed at important public places, such as airports, shopping malls, and traffic intersections, for the purpose of surveillance. Nowadays those cameras are used to record video data, which are conserved in hard disks for future use, or monitored by security staff. In fact, it is unlikely to monitor all the cameras in a non-automatic fashion. In order to reduce human labor, people are aiming to design intelligent surveillance systems, which are able to accomplish various detection and recognition tasks in an automatic way. Pedestrian detection plays an important part for such systems and, the position information of persons can be used as basic cues for further analysis of abnormal events, such as robbery, stealing or running red lights.
- Human-robot interaction: In the early days of artificial intelligence, robots are designed to execute heavy, tedious or dangerous tasks, in some specific environments where people seldom appear. But nowadays, people are expecting more intelligent robots, which may serve people in a more direct way. For this purpose, human-robot interaction is an important module to be developed. The basic premise of human-robot interaction is to enable robots to accurately localize people walking around them. An essential application may be robot navigation, where the location information of people can be used to avoid collision. Moreover, people expect robots to carry out some more advanced tasks, for instance, assisting people to move or deliver some objects, or even serving people for dinner like waiters in restaurants. To accomplish these complex

[3] `http://www.worldbank.org/html/fpd/transport/roads/safety.htm`

tasks, robots should be able to acquire detailed information about surrounding people, such as locations, sizes, postures and so on.

Among the above three applications, the one of Advanced Driver Assistance Systems (ADASs) is considered to be most challenging. This is because the outdoor scenario usually consists of more complex background, and the moving camera causes significant changes between two successive image frames, even in a short time slot. Both reasons result in notable interference for people detection. By contrast, in the scenario of visual surveillance, the camera is static and the background changes slightly over time; for the application of human-robot interaction, the indoor scenario is usually less dynamic because robots generally move slower than vehicles.

Therefore, we concentrate on the application of Advanced Driver Assistance Systems (ADASs). Nevertheless, the technologies developed in this thesis are not restricted to traffic scenes, but can be employed in other applications directly or after minor modification.

1.2 Challenges

Pedestrian detection in traffic scenes is a challenging task, not only because human physical appearances, clothing, and postures can vary significantly; but also due to the variations from the real-world environments. We briefly introduce the major variations involved in both aspects.

Intra-class variations Pedestrians are of high intra-class variation, considerably more significant than some other object categories, for example, cars or buildings. These intra-class variations make it impossible to distinguish pedestrians from other objects by one single cue, such as color or shape. In the following, we describe intra-class variations in terms of appearances, clothing choices and postures respectively, and we also show some examples in Figure 1.1.

- Appearances: People may look quite different from each other due to genetic properties, which cause various skin colors, hair colors, eye colors, and figures. One can also change his or her original appearance using makeup skills, including changing hair styles. Moreover, one individual usually show distinct appearances at different ages. In Figure 1.1(a), people of different ages, genders and skin colors appear in one single image.
- Clothing choices: People can choose clothes with a wide variety of colors, textures and styles. Sometimes, people also wear different accessories, such as jewelries or handbags. All the above factors lead to few uniform colors or textures in the internal regions of human bodies. Accordingly, [Dalal and Triggs, 2005] pointed out that ‘‘internal regions are unreliable cues”.

(a) (b) (c)

Figure 1.1: *Examples of pedestrian intra-class variations. (a) People with different appearances. Significant differences between female and male; between young and old. (b) The color of people's clothes can be bright or dark; the style can be long or short. (c) People leaning against a wall show a rather different posture from people walking or standing as shown in (a) and (b).*

- Postures: Another source of variations originates from postures the human body exhibits. Pedestrians can be walking across the street, standing at the traffic lights, or leaning against a wall. Since human bodies are non-rigid, a huge number of postures are likely to appear. Each posture shows an identical shape, so people of dissimilar postures look quite different from each other. This means we have to be careful while using shape descriptors for people detection.

Environment variations Urban traffic environment is very complex and less constrained, usually consisting of various object categories, for example, vehicles, buildings, plants, pets, and people. Moreover, the locations of pedestrians relative to the camera result in different viewpoints, occlusion conditions or scales, and thus is also an important factor which determines how pedestrians look in the image or video data.

- Dynamic background: When the camera moves with the vehicle driving, the background becomes dynamic. Compared to a static scenario, where the background can be modeled and moving objects can be easily found by background subtraction, dynamic background is more challenging, because each object may change its location and appearance between two successive frames even in a rather tiny time slot. One may suggest removing the effect of camera motion by employing affine theories, but unfortunately it is a very difficult task especially when the vehicle is driving with a high speed. To this end, it is rather challenging to make use of motion information for pedestrian detection.
- Viewpoints: Basically, viewpoints are determined by the relative location between camera and observed people. In most cases, objects exhibit different appearances when seen from different viewpoints. For example, cars look much wider when seen from the left or right side than seen from the front or back. Fortunately, the aspect ratio of pedestrians is less viewpoint relevant than cars. However, different viewpoints

(a) (b)

Figure 1.2: *Examples of pedestrians from different viewpoints. Walking pedestrians show large angles between two legs when seen from the side view.*

still cause different appearances. For example, we observe larger angles between two legs of walking pedestrians from the side view than from the front or back view. We show two examples in Figure 1.2. Moreover, the shoulder looks wider from the front or back view than from the side view.

- Occlusion: In urban traffic scenes, occlusion is very common. Pedestrians can be occluded by other objects or even neighboring pedestrians in a crowded scene. In the first situation, the missing body parts usually change the overall appearance. For example, in Figure 1.3(a), the whole right body part of one pedestrian is occluded by a tree, which makes this pedestrian looks different from other pedestrians in the same image. For those classifiers trained with unoccluded pedestrians, it may be recognized as a non-pedestrian object, thus resulting in high miss rates. In the second case, when several pedestrians are highly overlapped, as shown in Figure 1.3(b), it is difficult to infer each body part belongs to which person. Sometimes, a group of pedestrians are recognized as a single one, while sometimes more hypotheses are made.
- Scales: The object scale mainly depends on the distance between the observed object and the camera, assuming that we ignore the individual difference in height, which is not significant for pedestrians. In this way, those pedestrians which are far away from the camera show fewer pixels than the closer ones in image data. The challenges are twofold. First, some important cues such as connections between body parts may become vague when the number of pixels goes under a certain threshold. Sometimes, we can only see a silhouette, which is even difficult for humans to recognize and is easily to be recognized as a tree at some cases. Second, for sliding window approaches, we have to exhaustively search over scales in order to find pedestrians of different scales. As shown in Figure 1.4, those pedestrians close to the camera are almost ten times bigger than the distant ones.

(a) (b)

Figure 1.3: *Examples of pedestrians under occlusion. In (a), a pedestrian is occluded by a tree; in (b), multiple pedestrians are walking close to each other, resulting in the distant one being severely occluded by the closer ones.*

(a) (b)

Figure 1.4: *Examples of pedestrians of different scales. Generally, those pedestrians which are far away from the camera, are of smaller scales, and vice versa.*

1.3 Contributions of the Thesis

This thesis investigates the challenging problem of pedestrian detection in urban traffic scenes, and proposes novel methods of interpreting the characteristics of pedestrians in three different aspects -shape, cognition and motion, respectively- to enable efficient pedestrian detection.

Shape plays an important role for object detection. On one hand, objects from different categories show different shapes on images; on the other hand, different object instances from the same category generally share some common shape. Through observing a large number of pedestrian images, we find that for pedestrians, the common shape of up-right human body is a distinct characteristic, which not only makes the problem of pedestrian detection

much easier than general person detection, but also distinguishes pedestrians from other objects in urban traffic scenes. Therefore, we look into how to employ this characteristic for more effective pedestrian detection.

To reach or even go beyond human-like recognition is the final goal for intelligent vision systems. Aiming at designing a successful pedestrian detector, which is capable of finding pedestrians as accurate and fast as humans, it is worthwhile to study how human visual systems process visual data and localize objects they are interested in. In this way, we try to design a cognition system for pedestrian detection. By studying the mechanisms of cognition systems, we find the center-surround mechanism for salient object detection very attractive. More deep discussions can be found in [Klein and Frintrop, 2011]. The salient objects are accordingly defined as pedestrians for a pedestrian detection system, where employing the center-surround mechanism for feature design is a direct way to emulate a top-down saliency system.

Motion is also an important cue but has not yet been widely used for pedestrian detection. Unlike the shape cue, which represents spatial information; motion describes the temporal information across successive image frames. We raise three questions: (1) For which purpose can we use motion information? (2) How to interpret temporal information in a reasonable way? (3) Does motion information produce improvements while integrated with spatial information? To answer the above questions, we found out distinct inter-class and intra-class characteristics for pedestrians by observing a large number of motion maps, represented by optical flow vectors. These findings inspire us to solve a more specific problem of moving pedestrian detection.

In the following, we briefly outline the main contributions of this thesis in a more technical way.

- In Chapter 3, ‘‘Informed Multi-channel Haar-like Features’’, we propose a statistical shape model for the up-right human body. Then we design informed Haar-like features tailored to this shape model, so as to represent the special shape structure in terms of local difference. In order to generate more robust descriptors for local difference, we consider multiple image channels, including colors, gradient magnitude and histograms of oriented gradients. We also introduce a ternary modality as supplementary to the traditional binary modality, so as to represent more complicated geometric configurations. These informed features avoid exhaustive searches over all possible configurations and neither rely on a random sampling of a rectangular feature space, thus marking a middle ground. The presented experimental results show that our features reach and surpass state-of-the-art performance and are robust to occlusions. Moreover, our features require less memory and computational time for training than recently proposed competitive detectors, and are expected to reach real-time performance with GPU computation.

- In Chapter 4, ''Center-surround Contrast Features'', we propose local contrast features motivated by the center-surround mechanism in human visual systems, and tune them to respond to the appearance of pedestrians. Unlike previous methods, where contrasts are computed pixel by pixel, we consider contrast values for each region, which is represented by a statistical descriptor instead of by image channel values directly. In order to integrate richer information regarding local difference, we introduce multi-directional contrast vectors, which treat surrounding cells individually rather than as a whole. Another important task is to choose a reasonable distance measurement for those region descriptors. We make a comprehensive comparison of descriptor-distance combinations and find out the optimal one from experiments. Similar to most saliency detection systems, we build a contrast pyramid, representing coarse-to-fine local difference, by varying the cell size.
- In Chapter 5, ''Fast Moving Pedestrian Detection Based on Motion Analysis'', we focus on a more specific task of detecting moving pedestrians, which is an interesting subset for the application of ADASs. Motion information is used in two different ways. On one hand, we implement graph based segmentation on two-dimensional optical flow maps to select regions of interest (ROIs) and select moving objects by blob analysis. On the other hand, since we observe distinct magnitude maps of moving pedestrians from other moving objects, we design motion self difference features accordingly. Finally, to integrate different categories of features into one learning framework, we introduce a two-layer scheme for more reliable classification.

Our approaches achieve and surpass state-of-the-art results from experiments on different data sets. In the following, we summarize some lessons we have learned from our investigation. These notes may be instructive for future research.

- Prior knowledge is more powerful than we thought. In Chapter 3, ''Informed Multi-channel Haar-like Features'', we obtain surprisingly better results than the baseline detector by exploiting prior knowledge for feature design. In fact, prior knowledge enables us to better understand the data we need to handle and thus enhance the capability of recognition. Therefore, it is worthwhile to study how to employ prior knowledge more extensively in the future.
- Mechanisms of human visual systems are helpful for designing more effective intelligent vision systems. This has been recognized by researchers many years ago, but it is still an open question regarding how to integrate these mechanisms with existing computer vision and pattern recognition techniques in an appropriate way.
- Features versus learning methods. There is a debate on which is more important for a successful pedestrian detector: features, or learning methods. In this thesis, we mainly work on feature design, and we find that our methods employing carefully designed features outperform those using rather complex learning techniques. This success indicates that designing features adhere to the statistics of given data is a promising

direction and is expected to achieve even better performance.

1.4 Structure of the Thesis

In the next part, Chapter 2, ‘‘State of the Art’’, we first review the literature, especially focusing on the features that have been used for pedestrian detection in the last decade, followed by a comparison of several state-of-the-art detectors. Then, we introduce several public pedestrian data sets along with experiment settings used in this thesis.

In the subsequent chapters, we cover our contributions in more detail and present experimental results, which exemplarily show the claimed improvements with respect to the state-of-the-art performance on standard public data sets.

In Chapter 3, ‘‘Informed Multi-channel Haar-like Features’’, we investigate compact feature representations based on a statistical shape model.

In Chapter 4, ‘‘Center-surround Contrast Features’’, we propose cognitive vision driven center-surround contrast features. Seeking the strongest feature scheme, extensive experiments on various region descriptors and contrast measurements are implemented.

The following Chapter 5, ‘‘Fast Moving Pedestrian Detection Based on Motion Analysis’’ is then concerned with how to use motion cues for moving pedestrian detection.

At the end of each chapter, we give a summary and point to future directions of research on top of the presented approaches.

Chapter 6, ‘‘Conclusions’’ wraps up the thesis by emphasizing the main insights and by giving prospects of future work and open research questions.

Chapter 2

State of the Art

Vision-based pedestrian detection attracts increasing attention in academic communities in recent years. This chapter covers recent improvements in this field in terms of not only novel approaches being proposed, but also standard public benchmarks being established for experiments.

In the first part of this chapter, Section 2.1, ‘‘Literature Review’’, we review recent literature, focusing on features proposed for pedestrian detection, and also make a comparison of several state-of-the-art detectors. The second part of this chapter, Section 2.2, ‘‘Pedestrian Data Sets’’ introduces three public pedestrian data sets, which are widely used by recent detectors and also used in this thesis. Afterwards, in Section 2.3, ‘‘Experiment Settings’’, we explain the evaluation criteria used in our experiments, which are widely accepted in this area.

2.1 Literature Review

Since the 1990s, an enormous amount of literature has been published on the topic of pedestrian detection in the computer vision and pattern recognition communities. This is not only due to its considerable practical interest, but also because there are still many challenging problems to solve. An early prototype system was proposed as the PROTECTOR system in [Gavrila et al., 2004]. After that, more and more improvements to different system components were introduced. However, the state-of-the-art performance is still quite far from comprehensive real world applications.

In recent years, several survey papers provided reviews of the state of the art with different focuses. [Dollár et al., 2011] compared several pedestrian data sets in terms of pedestrian

scale, occlusion condition and some other important properties. In addition, they evaluated eight most popular detectors by testing them on different data sets. In contrast, [Geronimo et al., 2010] decomposed an entire pedestrian detection system into several components and evaluated various algorithms for each component. A similar attempt is made in [Enzweiler and Gavrila, 2009], which also contained a corresponding experimental study. [Munder and Gavrila, 2006] examined multiple feature-classifier combinations with respect to their performance and efficiency. [Gandhi and Trivedi, 2007] described several pedestrian detection systems which employ different kinds of sensors, not restricted to cameras.

In this section, we review the literature with a focus on feature extraction, in accordance with the main concern of this thesis. Moreover, we compare several popular state-of-the-art detectors in terms of features, learning methods and context information. Some of these detectors are considered as strong baselines for comparisons in the following chapters.

2.1.1 Features for Pedestrian Detection

Inspired by scale-invariant feature transform (SIFT) descriptors [Lowe, 2004], *Histograms of Oriented Gradients* (HOGs) were introduced in [Dalal and Triggs, 2005], considered as arguably the most popular features for visual pedestrian detection. By successfully integrating rich gradients of the special head-shoulder shape in human body, HOG features brought about significant improvements and therefore establish an important baseline. Afterwards, many variants of HOGs were proposed. [Zhu et al., 2006] designed a much larger set of blocks that vary in size, location and aspect ratio, and then used AdaBoost [Freund and Schapire, 1997] to select the most discriminative blocks for detection. A similar approach can be found in [Zini and Odone, 2011]. [Ye et al., 2010] proposed a set of multi-scale orientation features, consisting of coarse and fine HOG-like features. Dominant Orientation Templates(DOT) [Hinterstoisser et al., 2010] were suggested to be explicitly invariant to small translations, because they relied on locally dominant orientations, instead of local histograms. [Tang et al., 2012] presented a new pedestrian detection method applying random forests [Breiman, 2001] on DOT [Hinterstoisser et al., 2010] to achieve state-of-the-art accuracy, and more important, to accelerate runtime speed. [Lin and Davis, 2008] proposed a shape invariant global descriptor, computed from low level features of HOGs, for classifying human/non-human image patterns. [Li et al., 2008] introduced spatial histograms of oriented gradients features consisting of marginal distributions of an image over local and global patches, which preserved shape and contour of pedestrians simultaneously.

In order to improve performance, other researchers tried to combine HOGs with other features. HOG features were combined with *Local Binary Pattern* (LBP) features [Ojala et al., 1996] in order to cope with partial occlusions [Wang and Han, 2009][HogLbp]; spatial and temporal ‘‘granularity-tunable” features were proposed in [Liu et al., 2009], which combined a family of descriptors ranging from edgelets to HOGs; self-similarity features

[Walk et al., 2010] related to color channels [MultiFtr+CSS] as well as motion features [MultiFtr+Motion] were combined with HOG features in order to better interpret the global difference in terms of spatial and temporal information all over the human body.

Deviating from the popular framework of ''HOG+SVM'' computations, another strong baseline [ChnFtrs] was proposed in [Dollár et al., 2009a], which applied integral channel features with respect to colors, gradient magnitudes and histograms of oriented gradients. At that time, [ChnFtrs] outperformed previous detectors significantly in terms of both detection accuracy and efficiency. An immediate extension to this approach has been called the ''Fastest Pedestrian Detection in the West'' [FPDW] [Dollár and Perona, 2010], which enabled real-time multi-scale detection, through approximating channel values of neighboring scales inside each octave. Later, many new variants [Benenson et al., 2012, Dollár et al., 2012] emerged and several authors obtained even better performance by extending the feature pool in various ways. [Benenson et al., 2013] [Roerei] used irregular rectangles resulting in a $748,080$ dimensional feature pool; [Lim et al., 2013] [SketchTokens] added self-similarity features, yielding a $3,202,500$ dimensional feature vector. Due to the extreme sizes of these feature pools, both corresponding detectors require powerful computing hardware and large amounts of memory at training time.

Haar-like features became well-known after a Haar wavelets based system for object detection was proposed in [Papageorgiou and Poggio, 2000]. The epitome of such approaches is found in the work [Viola and Jones, 2004] who used Haar-like features in combination with boosting algorithms to build a successful face detector. In fact, an early attempt of Haar wavelets for pedestrian detection can be found in [Oren et al., 1997] where it was demonstrated that wavelet templates can be used to define the shape of an object. [Alonso et al., 2007] evaluated Haar wavelets and other features, for example, gradients, co-occurrence matrix, to find the most appropriate features for each body part. Unfortunately, Haar-like features, considered as second-order channel features [Dollár et al., 2009a], are not as successful as HOGs and are often discarded in pedestrian detection as they seem not to improve performance when combined with first-order channel features. In a closer analysis as to possible reasons for this behavior, we found that Haar-like templates that perform well for face detection are not necessarily suited for pedestrian detection but may fail to capture visual characteristics of human body.

Shape is another important cue for pedestrian detection. Different shape representations have been proposed for the specific human body shape. [Broggi et al., 2000] represented pedestrians mainly using vertical edges with a strong symmetry with respect to the vertical axis. [Wu and Nevatia, 2007] applied a large pool of short lines and curve segments, namely *edgelet* features, to represent characteristic shapes locally; more globally, *shapelets* [Sabzmeydani and Mori, 2007] were introduced as shape descriptors learned from gradients on local patches. [Bourdev and Malik, 2009] introduced a new notion of parts and poselets, constructed to be tightly clustered both in the configuration space of key points, as well as

in the appearance space of image patches. Then [Bourdev et al., 2010] employed only 2D annotations of key points and used the pattern of poselet activations for people detection. [Pishchulin et al., 2011] explored the possibility to use a 3D human shape and pose model from computer graphics to add relevant shape information for learning more powerful people detection models. Besides using shape features for classification, some researchers directly applied shape matching methods for detection. [Gavrila and Munder, 2007] applied the Hausdorff distance transform and a template hierarchy to rapidly match image edges to a set of shape templates. The proposed method in [Jiang, 2012] directly matched body parts to image regions which were obtained from object independent proposals and successively merged superpixels. Another way is to use shape prior to select the candidate regions, which may contain pedestrians. [Gavrila, 2000] proposed contour features, which were used in a hierarchical template matching approach to efficiently "lock" onto candidate solutions. [Bertozzi et al., 2003] selected interesting regions likely to contain pedestrians by searching for specific characteristics of pedestrians such as vertical symmetry and strong presence of edges.

Texture features interpret local difference patterns and have also been used for pedestrian detection. [Mu et al., 2008] investigated Local Binary Pattern (LBP) features [Ojala et al., 1996] for person detection and proposed two variants of the original LBP: Semantic-LBP and Fourier-LBP. [Zheng et al., 2011] presented a novel feature, termed pyramid center-symmetric local binary/ternary patterns (pyramid CS-LBP/LTP) to capture richer gradient information.

Some researchers explored other kinds of features. [Hong et al., 2010] proposed sigma set implicitly encoding second order statistics of an image region in the form of a point set. [Ren and Ramanan, 2013] computed sparse codes with dictionaries learned from data using K-SVD [Aharon et al., 2006], and aggregated per-pixel sparse codes to form local Histograms of Sparse Codes (HSC).

When it comes to image sequences, motion is another important cue for pedestrian detection. Unlike spatial features, temporal features have not been extensively investigated especially for dynamic scenes, due to significant camera motion, which is difficult to remove. [Viola et al., 2005] employed Haar-like templates on multiple temporal difference images to interpret local difference regarding human body motion, however, this method is restricted to static scenarios. [Jones and Snow, 2008] extended the above work by using more frames as input to the detector thus allowing for a more thorough analysis of motion. [Yamauchi et al., 2008] proposed new spatio-temporal features, representing the state of each pixel as stationary or transient. [Dalal et al., 2006] proposed HOG-like motion features, namely HOF (Histogram Of Flow), based on gradients computed on optical flow field. Afterwards, [Walk et al., 2010] proposed a number of modifications to HOF features, which modestly improved the performance. More recently, [Park et al., 2013] computed temporal differences as features, following weak stabilization based on coarse optical flow estimation over multiple frames.

Table 2.1: *Comprehensive comparison of state-of-the-art pedestrian detectors. Detectors are ordered by the years they were proposed. Each row in this table summarizes information as to features, learning methods, and context information used in a particular approach.*

Detector	Features							Learning		Context
	HOGs	gradients	grayscale	color	texture	self-similarity	motion	classifiers	part based	
VJ [Viola and Jones, 2004]	×	×	√	×	×	×	×	AdaBoost	×	×
HOG [Dalal and Triggs, 2005]	√	×	×	×	×	×	×	linear SVM	×	×
Shapelet [Sabzmeydani and Mori, 2007]	×	√	×	×	×	×	×	AdaBoost	×	×
MultiFtr [Wojek and Schiele, 2008]	√	×	√	×	×	×	×	AdaBoost	×	×
HikSvm [Maji et al., 2008]	√	×	×	×	×	×	×	HIK SVM	×	×
LatSvm-V1 [Felzenszwalb et al., 2008]	√	×	×	×	×	×	×	latent SVM	√	×
PLS [Schwartz et al., 2009]	√	×	×	√	√	×	×	PLS+QDA	×	×
HogLbp [Wang and Han, 2009]	√	×	×	×	√	×	×	linear SVM	×	×
ChnFtrs [Dollár et al., 2009a]	√	√	√	√	×	×	×	AdaBoost	×	×
MultiFtr+CSS [Walk et al., 2010]	√	×	×	×	√	√	×	AdaBoost	×	×
MultiFtr+Motion [Walk et al., 2010]	√	×	×	×	√	√	√	linear SVM	×	×
LatSvm-V2 [Felzenszwalb et al., 2010]	√	×	×	×	×	×	×	latent SVM	√	×
FeatSynth [Bar-Hillel et al., 2010]	√	×	×	×	√	×	×	linear SVM	√	×
MultiResC [Park et al., 2010]	√	×	×	×	×	×	×	latent SVM	×	×
CrossTalk [Dollár et al., 2012]	√	√	√	√	×	×	×	AdaBoost	×	×
VeryFast [Benenson et al., 2012]	√	√	√	√	×	×	×	AdaBoost	×	×
DBN-Isol [Ouyang and Wang, 2012]	√	×	×	×	×	×	×	DeepNet	√	×
AFS+Geo [Levi et al., 2013]	√	×	×	×	√	×	×	linear SVM	√	×
MT-DPM+Context [Yan et al., 2013]	√	×	×	×	×	×	×	latent SVM	√	√
DBN-Mut [Ouyang et al., 2013]	√	×	×	×	×	×	×	DeepNet	√	×
SketchTokens [Lim et al., 2013]	√	√	√	√	×	×	×	AdaBoost	×	×
Roerei [Benenson et al., 2013]	√	√	√	√	×	×	×	AdaBoost	×	×
ACF+SDt [Park et al., 2013]	√	√	√	√	×	×	√	AdaBoost	×	×

Aiming for more efficient training and testing, feature dimensionality reduction is a good solution in many cases. [Felzenszwalb et al., 2010] applied Principal Component Analysis (PCA) [Pearson, 1901] on HOGs and found that the top 11 eigenvectors captured essentially all the information of a 36-dimensional HOG feature. Alternatively, [Schwartz et al., 2009] applied Partial Least Squares (PLS) [Wold, 1985] on HOG features. [Hussain et al., 2010] employed PLS on a large feature pool, consisting of HOGs, LBP and LTP.

2.1.2 Comparison of State-of-the-art Detectors

For the purpose of investigating the research trend of pedestrian detectors, we make a comprehensive comparison of state-of-the-art pedestrian detectors in Table 2.1. We chose 23

state-of-the-art pedestrian detectors proposed between year 2004 and 2013. These detectors are representative for various lines of research, ensuring diversity. In addition, their original performance is publicly available, enabling comparisons for experiments in the following chapters.

In Table 2.1, we compare the detectors in terms of features, classifiers, part-based models, and context information, respectively. In the following, we discuss our insights regarding the above aspects.

- **Features:** Almost all the detectors use HOGs but in different ways. Some detectors use HOGs as their only features, while others combined HOGs with other features. Therefore, HOGs are considered as the most popular features for pedestrian detection in the last decade. In contrast to spatial features, motion information is rarely used. One reason may be that it is computationally expensive to obtain an accurate and dense optical flow map, which directly describes the motion between two successive frames. On the other hand, it is still an open question how to design motion based features, which provide rather different information from colors or gradients. We also find that recent proposed detectors tend to use multiple features in order to integrate richer information.
- **Classifiers:** Consistent with the original [HOG] detector, Support Vector Machines (SVMs) [Cortes and Vapnik, 1995] are used by those detectors which only use HOGs for classification. Linear kernel is used more often due to its efficiency. Histogram intersection kernel SVMs introduced by [HikSvm] [Maji et al., 2008] prove to be more efficient than other non-linear kernels. Latent SVMs were first introduced by [LatSvm-V1] [Felzenszwalb et al., 2008] for partially labeled data training. More recently, deep learning algorithms [Bengio et al., 2013] were employed by [DBN-Isol] [Ouyang and Wang, 2012] and [DBN-Mut] [Ouyang et al., 2013], adaptive to multi-layer part-based models. By contrast, multi-feature based detectors usually use ‘‘Adaptive Boosting” (AdaBoost) algorithm [Freund and Schapire, 1997]. This is because AdaBoost is more efficient to select the most discriminative features from a large feature pool.
- **Part-based models:** The purpose of learning with part-based models is to ensure robust detection of deformable objects, especially under occlusion [Agarwal et al., 2004]. Various strategies have been proposed by different researchers. Unlike [LatSvm-V1][Felzenszwalb et al., 2008] and [LatSvm-V2] [Felzenszwalb et al., 2010], which generated the final detection score from the root filter and several pre-defined part filters, [FeatSynth] [Bar-Hillel et al., 2010] randomly selected image patches as parts, and applied feature selection to select representative features extracted from those parts. Furthermore, [DBN-Isol] [Ouyang and Wang, 2012] and [DBN-Mut] [Ouyang et al., 2013] investigated the challenging problem of modeling the relationship of the visibilities of different parts and combining the responses of part detectors. However, a recent study of finding the weakest link in person detectors [Parikh and Zitnick,

2011] shows that the performance of a detector is more significantly affected by the part detectors than by the use of human or machine spatial models.

- **Context:** It is efficient to use context information as prior knowledge to restrict locations of pedestrians. [Ma et al., 2010] applied notions such as corner, motion, and appearance to localize pedestrians in far-field videos without performing brute-force-search. The corners direct attention to a set of conspicuous locations as the starting points for searching. And motion detection restricts the searching area within the foreground mask. [Wang and Wang, 2011] proposed a new framework of adapting a pre-trained generic pedestrian detector to a specific traffic scene by automatically selecting both confident positive and negative examples from the target scene to re-train the detector iteratively. [Wang et al., 2012] took the context information from motions, scene structures and scene geometry as the confidence scores of samples from the target scene to guide transfer learning. [Yan et al., 2013] built a context model to suppress false positives according to the pedestrian-vehicle relationship in traffic scenes.

In addition, we briefly introduce several other detectors which employ different complex models for detection. [Elzein et al., 2003] applied a wavelet transform computed on the video frames, and multi-stage template matching to determine whether or not a pedestrian is present in the current frame. [Shashua et al., 2004] proposed an approach for single-frame classification based on a novel scheme of breaking down the class variability by repeatedly training a set of relatively simple classifiers on clusters of the training set. [Leibe et al., 2005] combined the local information from sampled appearance features with global cues about silhouette via a probabilistic top-down segmentation. [Andriluka et al., 2008] detected the approximate articulation of each person in every frame based on local features that modeled the appearance of individual body parts. Prior knowledge on possible articulations and temporal coherency within a walking cycle were modeled using a hierarchical Gaussian process latent variable model (hGPLVM) [Lawrence and Moore, 2007].

To briefly summarize, it is promising to integrate multiple features, with respect to colors, gradients and motion, and apply the AdaBoost algorithm for fast training and real-time testing. Moreover, learning with part-based models is an effective way for occlusion handling, given the robust detection for each body part.

2.2 Pedestrian Data Sets

In this section, we introduce several standard pedestrian benchmarks, which are popular in the field of pedestrian detection and are used for experiments in this thesis. The Caltech, KITTI and Daimler Mono data sets comprise mobile videos, while the INRIA data set only consists of still images.

2.2.1 INRIA

The INRIA pedestrian data set[1] is arguably the most popular data set for people detection. This data set was first used in the [HOG] detector [Dalal and Triggs, 2005], [Dalal, 2006] and then widely tested by many later approaches.

The INRIA data set comes along with pre-defined subsets for training and testing. The training subset is provided in two formats:

- Original images with annotations.
- Normalized samples of a uniform size of 96×160.

In the training set, there are $2,416$ positive samples, by mirroring from $1,208$ different pedestrian images, all of which are cropped from 614 natural images; there are also $12,180$ negative samples, randomly cropped from $1,218$ natural images (10 samples per image), where no pedestrian appears. In the test set, there are 288 positive samples, consisting of single or multiple pedestrians; and 453 negative samples with no pedestrians. Consistent with conventions in this area, test is only implemented on the positive samples.

2.2.2 Caltech

The Caltech pedestrian data set [Dollár et al., 2009b][2] is currently the largest and most challenging data set for pedestrian detection. It consists of approximately 10 hours of 640×480 $30Hz$ video taken from a vehicle driving through regular traffic in an urban environment.

About $250,000$ frames with a total of $350,000$ bounding boxes and $2,300$ unique pedestrians were annotated. The annotation includes temporal correspondence between bounding boxes and detailed occlusion labels. Notably, this is currently the only pedestrian data set which provides occlusion labels. These labels enable researchers to analyze their detectors' performance under different occlusion levels.

The first six sets (set00-set05) are defined as training sets, each with 6-13 one-minute long sequence files. The remaining five sets (set06-set10) are defined as testing data. Each set comes along with full annotation information (see [Dollár et al., 2009b] for details).

[1] `http://pascal.inrialpes.fr/data/human/`
[2] `http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/`

2.2.3 KITTI

The KITTI vision benchmark suite [Geiger et al., 2012][3] is a new and large data set, providing stereo image pairs and laser point clouds for a wide range of research interests, including: stereo, optical flow, visual odometry, 3D object detection and 3D tracking.

This data set is captured by driving around the mid-size city of Karlsruhe, in rural areas and on highways. It consists of 7,481 training images and 7,518 testing images, comprising a total of 80,256 labeled objects, including cars, cyclists and pedestrians. In our experiments, we only use the left color images and pedestrian annotations.

2.2.4 Daimler Mono

''Daimler mono pedestrian detection'' benchmark [Enzweiler and Gavrila, 2009] [4] is captured by a monochrome camera installed on a vehicle driving through urban environment. This dataset consists of 21,790 consecutive gray-scale frames (640×480 pixels), along with 56,492 pedestrian annotations.

We use this data set for moving pedestrian detection in Chapter 5, ''Fast Moving Pedestrian Detection Based on Motion Analysis'', because it is captured in an urban traffic environment, and consists of a large number of moving pedestrians, who walk across or along the street. However, annotations for moving pedestrians are not provided by the original data set. Therefore, we manually determine each pedestrian annotation to be moving or static, through observing multiple consecutive video frames before and after the current time point. Then we add an additional label of ''moving'' to the ground truth data, so as to record the movement status for each pedestrian annotation. We publish the re-annotated ground truth data at `http://www.iai.uni-bonn.de/~zhangs/` for public interests.

2.3 Experiment Settings

In this section, we explain evaluation protocols used for experiments in this thesis. These protocols are identical to those explained in [Dollár et al., 2011], and are widely accepted in this field.

Ground truth regulation: The ground truth data is regulated in the following two ways.

[3] `http://www.cvlibs.net/datasets/kitti/eval_object.php`

[4] `http://www.gavrila.net/Research/Pedestrian_Detection/Daimler_Pedestrian_Benchmark_D/Daimler_Mono_Ped__Detection_Be/daimler_mono_ped__detection_be.html/`

		INRIA	Caltech	KITTI	Daimler Mono
Properties	imaging setup	photo	mobile	mobile+stereo	mobile
	color images	√	√	√	×
	video seqs.	×	√	√	√
	occlusion labels	×	√	×	×
Training	# pedestrians	1208	192k	1800	15.6k
	# pos. images	614	67k	3471	-
	# neg. images	1218	61k	-	6.7k
Testing	# pedestrians	566	155k	1962	56.5k
	# pos. images	288	65k	3470	21.8k
	# neg. images	453	56k	-	-

Table 2.2: *Statistics of pedestrian data sets used for experiments [Dollár et al., 2011].*

- Ignored data selection: For each experiment, a subset of all ground truth data is considered according to its specific purpose. Outliers are marked with an *ignore* label, which means they need not be matched, however, matches are not considered as mistakes either. We specify four settings used in this thesis as follows: (1) *Reasonable*: only pedestrians at a resolution of over 50 pixels in height and a visibility of more than 65% are considered. This setting is generally applied without special instructions. (2) *No occlusion*: pedestrians with 100% visibility are considered. (3) *Partial occlusion*: pedestrians with more than 65% visibility are considered. (4) *Heavy occlusion*: pedestrians with 20% - 65% visibility are considered.
- Aspect ratio standardization: Because most of the detectors use windows with a common aspect ratio of 0.5, it is important that the ground truth is annotated in the same way to obtain meaningful performances [Dollár et al., 2011]. We standardize all ground truth bounding boxes by keeping the original height and center while adjusting the width. From our observation, performance of different detectors stays stable for various choices of the aspect ratio.

Detection results filtering: Detection results are filtered out using an expanded filtering method [Dollár et al., 2011], so that detection results far outside the evaluation scale range should not be considered. When evaluating a scale range of $[S_1, S_2]$, only detections in $[S_1/\xi, S_2\xi]$ are considered for evaluation. In our evaluation, we set $\xi = 1.25$.

Matching rules: Filtered ground truth bounding boxes and detection results bounding boxes are annotated by B_{gt}, and B_{dt} respectively. A detected bounding box and a ground truth bounding box match if and only if the ratio of overlap to the union of their areas exceeds a

given threshold of 0.5 [Dollár et al., 2011]:

$$\text{match}(B_{\text{dt}}, B_{\text{gt}}) = \frac{\text{area}(B_{\text{dt}}) \cap \text{area}(B_{\text{gt}})}{\text{area}(B_{\text{dt}}) \cup \text{area}(B_{\text{gt}})} \overset{!}{>} 0.5 \,. \tag{2.1}$$

Performance measurements: We perform full image evaluation instead of per-window evaluation as the former one provides a natural measure of error of an overall detection system. In order to compare different detectors, we plot miss rate against false positives per image (FPPI) curves in logarithmic scales by varying the threshold on the detection confidence of the classifiers. We only plot the curves in FPPI range between $(-\infty, 10^0]$ as more than 10^0 FPPI is unacceptable for intelligent vehicle applications. In addition to these miss rate vs. FPPI curves, we calculate a single, numerical measurement to summarize the overall performance. We use the *log-average miss rate* [Dollár et al., 2011], which is computed by averaging the miss rates at nine FPPI rates evenly sampled in log-space in the range of $[10^{-2}, 10^0]$. This *log-average miss rate* generally gives a more stable and informative assessment of the overall performance for different detectors than the single miss rate at 10^{-1} FPPI [Dollár et al., 2011].

Chapter 3

Informed Multi-channel Haar-like Features

Feature design is of major importance for pedestrian detection. Over the last decade, significant efforts have been made towards the design of new features [Dalal and Triggs, 2005, Dalal et al., 2006, Kao et al., 2012, Walk et al., 2010]. Yet, from looking at the recent literature, it seems that there is a significant general trend: huge feature pools and high dimensional feature vectors are becoming increasingly popular, mainly because they produce reasonable performance by simple integration with classic learning methods, such as boosting, which offers a convenient and efficient way to select from a large number of candidate features. In the future, even better performance is expected by employing complicated models for handling variation from viewpoints, body parts, occlusions, or context.

Unfortunately, ‘‘There’s no such thing as a free lunch’’. Approaches employing a huge feature pool require a large amount of memory and computational time during training; moreover, they rely on the availability of powerful computers and GPU computation to enable real-time applications. Addressing this problem, we aim at more *compact features* which require less memory and computational costs yet guarantee reasonable and robust performance.

We observe that pedestrians generally show a common appearance of up-right human body, which is very distinguishable from other objects. Our motivation is to design informed features by exploiting this characteristic. We looked into prior work on detecting objects of relatively low intra-class variability and noticed the significant success of cascaded Haar-like

This work has been partially published in [Zhang et al., 2014a,b].

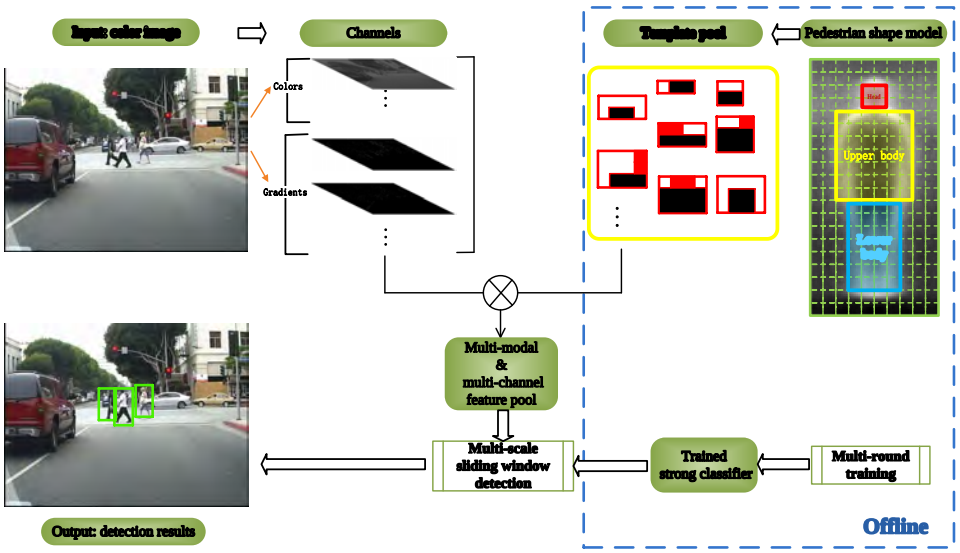


Figure 3.1: *Overview of pedestrian detection based on informed Haar-like features. The dotted blue bounding box indicates the off-line procedure.*

features [Viola and Jones, 2004] on face detection. This indicates that Haar-like features may be an appropriate solution to our problem, because human faces also exhibit a uniform shape structure. However, we note that Haar-like features are not as successful as HOGs and are often discarded in the field of pedestrian detection. Closely analyzing possible reasons for this behavior, we found that Haar-like templates that simply designed for face detection are not necessarily suited for pedestrian detection but may fail to represent shape characteristics of human body, which is obviously more complex than face shape pattern. As a remedy, we propose to design particular Haar-like templates tailored to up-right human body shapes.

The procedure of our pedestrian detector employing informed Haar-like features is shown in Figure 3.1, where we provide three major contributions:

Statistical pedestrian shape model: From an average gradient image computed from statistical data, we find that up-right walking pedestrians share a common visual appearance especially with respect to the geometry of the head and shoulder area of the body. We model pedestrian shape in terms of three rectangles geared towards distinct body parts -- head, upper body and lower body.

Multi-modal Haar-like template pool: Based on the pedestrian shape model, we design a pool of templates that is better tailored to the common pedestrian shape and thus leads to better performance than previous Haar-like templates; on the other hand, these templates only constitute a small subset of all possible rectangular templates so they significantly reduce training time and required memory. Besides the traditional binary modality, we introduce a

ternary modality as a supplement. The ternary modality is specifically proposed to represent corner regions found along the pedestrian shape model so as to enable rectangular features to represent more complex geometric configurations.

Multi-channel Haar-like features: We apply all the templates on multiple image channel maps, to incorporate rich information from given image data. The channels we consider consist of not only colors but also gradient information. Therefore, our multi-channel Haar-like features are more robust to variations of clothes and illuminations.

This chapter contains a description of feature extraction in Section 3.1, ''Feature Extraction'', and our feature selection scheme in Section 3.2, ''Classification''. A thorough set of experiments is presented in Section 3.3, ''Experiments'', where the impact of different parameters is investigated and extensive comparisons to state-of-the-art detectors from the literature are made. Afterwards, we discuss several important issues regarding feature design in Section 3.4, ''Discussions''. Finally, we summarize our contributions and findings and point out several directions for future work in Section 3.5, ''Summary''.

3.1 Feature Extraction

In this section, we describe our feature extraction procedure. First, a statistical pedestrian shape model is defined according to an average edge map computed from statistical image data. Next, a multi-modal Haar-like template pool is generated by sliding rectangles of different sizes all over the shape model. Afterwards, *channel information* in terms of colors and gradients are computed directly or via various transformations from the input color images. Finally, informed multi-channel Haar-like features are extracted by convolution between templates and each channel map.

3.1.1 Statistical Pedestrian Shape Model

Through observing a large number of pedestrian images, we find that pedestrian bodies share a common geometry structure. Then we try to corroborate this assumption based on empirical data. We choose the INRIA pedestrian data set, which consists of cropped image patches showing pedestrians scaled to a height of 96 pixels, and with 12 pixels padded in four directions to include contextual information. Consequently, we perform a statistical analysis on pedestrian images of 60×120 pixels. We compute an average gradient magnitude map based on all sample images, regardless of viewpoints or postures. As shown in Figure 3.2, the resulting average edge map clearly resembles a human body.

Typically, features derived from rectangular image regions can be computed efficiently by employing integral images. Therefore, we decide to base our pedestrian detector on

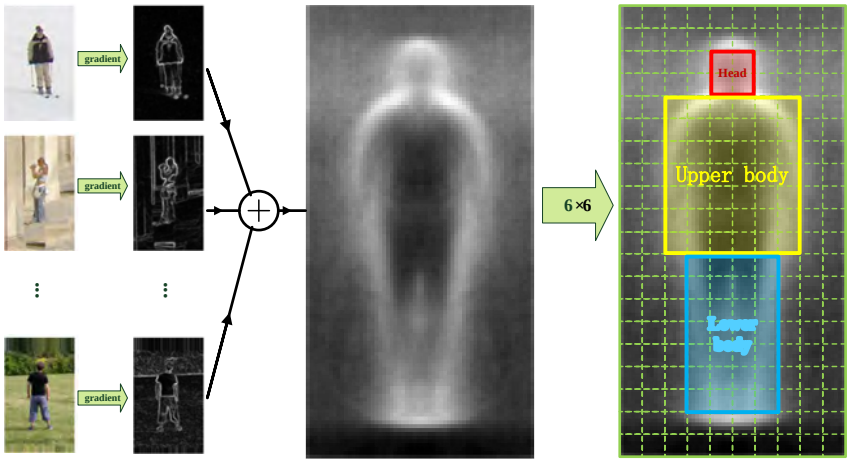


Figure 3.2: *Procedure of our statistical pedestrian shape model (rightmost) generation. We collect all the pedestrian sample images from the INRIA data set and compute an average edge map, as shown in the middle, which is divided by rectangular cells. In this example, cell size is chosen to be* 6×6 *pixels. Three bounding boxes approximately indicate the head, the upper body, and the lower body parts.*

rectangular features. The edge map is then divided into square *cells*, whose sizes may vary. Figure 3.2 shows an example of cells of 6×6 pixels. Given these grids of cells, the whole body can be approximately divided into three distinct parts: the head, the upper body, and the lower body. This is intended to increase robustness as the above three parts generally exhibit different colors and textures in real world images. The boundaries of each part are manually defined according to our prior knowledge on human body parts, as well as the silhouette from the average gradient magnitude map. We vary these boundaries by choosing different cell sizes. In order to obtain the optimal model, we implement experiments with different cell sizes in Section 3.3, ‘‘Experiments’’.

3.1.2 Multi-modal Haar-like Template Pool

In this section, we describe how to generate a multi-modal Haar-like template pool based on the pre-defined statistical pedestrian shape model discussed in Section 3.1.1, ‘‘Statistical Pedestrian Shape Model’’.

We start with explaining the concept of *modality*. The modality of a Haar-like template is determined by how many different weights are involved. For example, traditional Haar-like features are referred to as a binary modality because they only carry two possible weights ($+1$ and -1) for different rectangles. If one template carries three different weights, then it is called a ternary template. We introduce the ternary modality because we find that binary modality is not able to represent cusps or corner-like structures of the human silhouette.

Assume that there are three distinct logical components such as, say, the head, the upper body, and parts of the scene background involved in one rectangle, while we are interested in computing the difference between any two of them at a time, we have to introduce a third weight of 0 to assign to the ignored logical component. This strategy helps to keep all the templates a uniform shape of rectangles, thus enabling subsequent efficient feature computation. An example is given in Figure 3.3, where ternary 2×2 templates capture the local geometry of the image region where head, shoulder, and background meet in joint corners.

Since we employ both binary and ternary modalities, our template pool is multi-modal. An illustration of our template pool generation procedure is shown in Figure 3.3. In the following, we describe the whole procedure step by step in detail.

First, we define a size pool S as follows:

$$S = \{(w, h) \mid w \leq w_m, h \leq h_m, w, h \in \mathbb{N}^+\}, \tag{3.1}$$

where w and h indicate the number of cells along horizontal and vertical directions of a rectangular template; w_m and h_m are used to constrain the overall size of templates since we focus on local image information. Note that our templates are constrained to be of rectangular form, for the purpose of enabling convenient implementation and efficient computation. Statistical variations are handled by using different modalities.

Second, we assign a logical label to each cell based on the pedestrian shape model. As shown in Figure 3.2, images of pedestrians consist of four logical components: background, head, upper body, and lower body. We assign each cell $c(i, j)$ with one label $L(i, j)$ according to which logical component it belongs to.

Next, for each size in the size pool S, we slide a corresponding rectangular window over the whole shape model to generate different templates at different positions. At a certain position (x, y), we first decide the modality by analyzing how many different logical components are involved in the rectangular window. If there are two components involved, the modality is set to be binary and only one ($\binom{2}{1}$) template is generated; if there are three components involved, the modality is set to be ternary and three ($\binom{3}{2}$) templates are generated. Note that it is impossible to have four components in one rectangular window, given the shape model and the constrained template size.

In the following, we denote a template as: $t(x, y, (w, h), W)$ or in a slightly simplified way as: $t(x, y, s, W)$, $s \in S$, where x and y indicate the location of the left top cell of a template with respect to the human shape model, w and h indicate the width and height of a template with respect to cells, and W is a weight matrix with different values assigned to different logical components.

We note that redundancy may appear if we generate templates by employing Algorithm 1. An example is shown in Figure 3.3. At position [4, 4], the 2×1 template is actually identical to the third 2×2 template, although they are of different template sizes. The reason lies in that the lower two cells of the 2×2 template are both assigned the weight of 0, which means that only the upper two cells actually contribute to the feature response during computation, so we can easily simplify it to a 2×1 template. In order to get rid of redundancy, once another identical template is found in the template pool, the current template is discarded. Here comes another problem: how to find out redundancy. We develop a simple method to efficiently check for redundancy for each pair of templates, which locate at the same position. Given two templates: $t_1(x, y, (w_1, h_1), W_1)$ and $t_2(x, y, (w_2, h_2), W_2)$ at the same location (x, y), we define a maximal size $s_{max}(w, h)$:

$$\begin{cases} w = \max(w_1, w_2) \\ h = \max(h_1, h_2). \end{cases} \tag{3.2}$$

Then, we expand two weight matrices to a new size of $s_{max}(w, h)$, by filling blanks with a weight of 0. Next, we simply implement a subtraction between two new weight matrices W_1' and W_2':

$$W_d = W_1' - W_2'. \tag{3.3}$$

Templates t_1 and t_2 are considered to be identical if and only if all the elements of the difference matrix W_d are zero.

To enhance the robustness against individual differences, each template is shifted along four directions with a step of one cell, resulting in a larger template pool. Therefore, for each template $t(x, y, s, W)$, the original template and a group of shifted templates are added to the template pool. We denote this template group as follows:

$$\begin{cases} t(x, y, s, W) \\ t_L(x - 1, y, s, W) \\ t_R(x + 1, y, s, W) \\ t_U(x, y - 1, s, W) \\ t_D(x, y + 1, s, W). \end{cases} \tag{3.4}$$

Notably, some templates at the border of the shape model are infeasible to be shifted along four directions.

Finally, after redundancy removal and shifting, we obtain the full template pool as a set of templates of various sizes, of two modalities and at different positions:

$$T = \{(x, y, s, W) \mid x, y \in \mathbb{N}, s \in S, W \in \mathbb{R}^2\}, \tag{3.5}$$

Algorithm 1 Generating templates for pedestrian shape model through sliding rectangles.

initialize template pool: $T \leftarrow \emptyset$;
for $i = 1$ to $nSize$ **do**
  for $x_1 \in [1, width - w_i]$ **do**
    for $y_1 \in [1, height - h_i]$ **do**
      $label = L(x_1 : x_1 + w_i, y_1 : y_1 + h_i)$;
      if unique(label)==2 **then**
        $W(label == l_1) \leftarrow -1$;
        $W(label == l_2) \leftarrow 1$;
        append $(x_1, y_1, (w_i, h_i), W)$ to T;
      else if unique(label)==3 **then**
        for $iCase \in [1, 3]$ **do**
          $W(label == l_{iCase}) \leftarrow 0$;
          $W(label == l_{(iCase+1)\%3}) \leftarrow -1$;
          $W(label == l_{(iCase+2)\%3}) \leftarrow 1$;
          append $(x_1, y_1, (w_i, h_i), W)$ to T;
        end for
      end if
    end for
  end for
end for
return T

where x and y indicate the location of each template with respect to the human shape model and W is a weight matrix that is determined according to the matrix L of labels for all cells.

3.1.3 Multi-channel Cell Descriptor

Traditional Haar-like features usually compute the feature response only based on the intensity values, yet we consider multiple image channels inspired by the success of integral channel features [Dollár et al., 2009a]. The big advantage of employing multiple image channels is to integrate richer information, for example, colors and gradient magnitude and gradient orientations.

[Dollár et al., 2009a] implemented experiments using various combinations of image channels, and found that the optimal setting is to use a total of 10 different channels: 3 channels for LUV colors, 1 channel for gradient magnitude information, and 6 channels for histograms of oriented gradients. It is also reported in [Dollár et al., 2009a] that pre-smoothing on the input image data with a binomial filter [Haddad, 1971] of radius 1, *i. e.* $\sigma \approx 0.87$ improves the performance, while post-smoothing on channel values has little effect on performance.

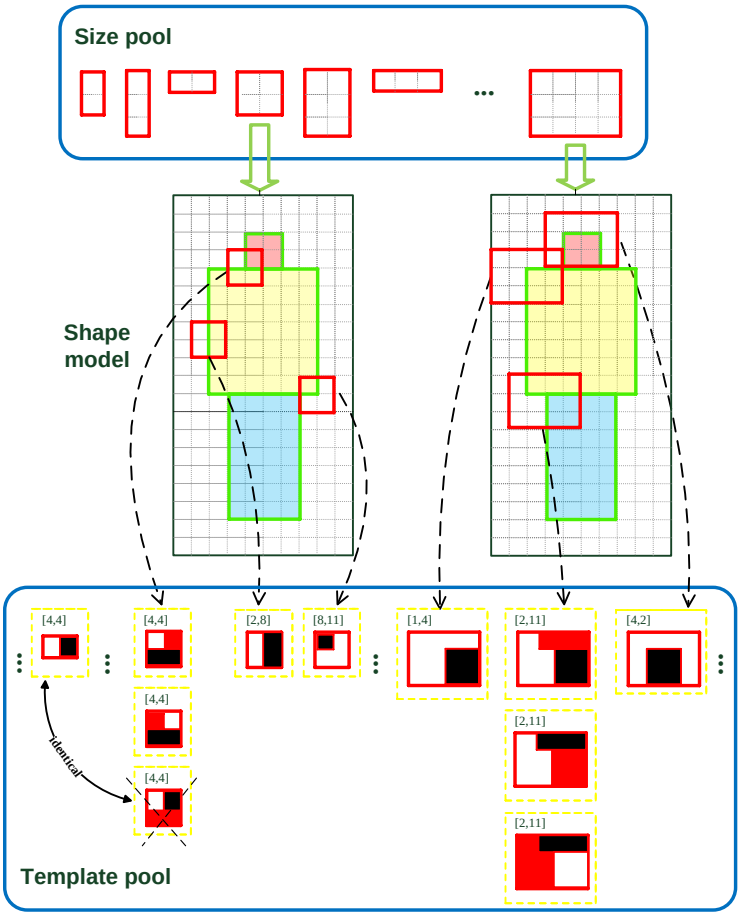


Figure 3.3: *Overview of our Haar-like template pool generation procedure. Note that the number array above each template indicates the x, y coordinates of its left top cell with respect to the shape model; those templates at* $[4, 4]$ *and* $[2, 11]$ *are of ternary modality, which are given the weights of* $+1$*,* -1*, and* 0 *to those white, black, and red areas, respectively. An example of redundancy removal is given: two templates at the same location are considered to be identical during redundancy check procedure, then one of them is discarded.*

We use the above settings recommended by [Dollár et al., 2009a] as our default settings for our primary experiments, but we still discuss in Section 3.3.1, ‘‘Parameter Settings’’ about alternative settings to observe their effects on performance. This is not tedious but oppositely necessary since our features consider local difference rather than absolute channel values. In fact, we come to different conclusions for some parameters.

3.1.4 Feature Matrix

Feature values are computed through convolution between each Haar-like template and all channel maps. However, before implementing convolution, we should normalize the weight matrix of each template by size, so as to avoid the effect of template sizes on the final output feature values.

Assume that we are given a template denoted as $t = (x, y, (w, h), W)$. We normalize the weight matrix W for each template by first counting how many cells are assigned with the weights of $+1$ and -1. Those frequencies are denoted as n_{add} and n_{sub}, respectively. That is to say, we have n_{add} additive cells and n_{sub} subtractive cells. Then the normalized average weight matrix can be computed using the following formula:

$$W_{avg} = \frac{sgn(W)}{n_{add}} + \frac{sgn(-W)}{n_{sub}}. \tag{3.6}$$

The final feature pool $\boldsymbol{f}$ consists of those feature values computed from all the templates going through multiple image channels. Assume we have N_t templates in total and consider N_c channels, the size of our feature pool is $N_t \times N_c$. The feature value of any template $t(t < N_t)$ along any channel $k(k < N_c)$ can then be computed as a weighted sum:

$$\boldsymbol{f}(t, k) = \sum_{i=1}^{h} \sum_{j=1}^{w} \sigma(x + i, y + j, k) W_{avg}(i, j), \tag{3.7}$$

where, $\sigma(i, j, k)$ denotes the local sum of $cell(i, j)$ along channel k, which can be computed very efficiently by employing integral images.

3.2 Classification

In order to choose an appropriate classification method, we first look into the characteristics of our features. As proposed in Section 3.1, ‘‘Feature Extraction’’, our features are informed Haar-like features built on multiple channels. The major difference from integral channel features ([ChnFtrs] [Dollár et al., 2009a]) is that, our features interpret local difference between rectangular regions in terms of local sums of channel values, while [ChnFtrs] only considers local sums themselves.

We also analyze our feature size in the following. Given 6×6 cells and templates size ranging from 1×2 to 4×3 cells, we obtain 266 identical templates at different locations after redundancy removal. Shifting each template along 4 directions with a step of one cell yields a template pool of 1276 (some shifts are not possible at image borders); considering

10 channels, the final feature size is 12,760. Due to the large size of our features, we decide to choose a boosting method for classification, since it offers a convenient and fast way to select from a large number of candidate features. Although boosting algorithms are usually efficient enough during testing and can be used for real-time applications, they are very time consuming for training, especially when the feature pool is large. Fortunately, there is a fast version of AdaBoost [Appel et al., 2013], which speeds up the traditional AdaBoost algorithm by an order of magnitude via employing a bound on error to prune unpromising features early in the training process.

An important issue for boosting algorithms is the configuration of weak classifiers. We use decision trees of depth 2 as our weak classifiers and choose the number of weak classifiers to be 2000. Experimental results under different numbers of weak classifiers can be found in Section 3.3, ‘‘Experiments’’. As demonstrated in classic detectors, for example, [HOG] [Dalal and Triggs, 2005] and [ChnFtrs] [Dollár et al., 2009a], a multi-round training strategy produces better performance than applying a simple one-round training procedure with the same number of negative samples. Therefore, we also employ this strategy. Over all training rounds, the positive sample set is consistent; while the negative sample set is expanded by adding those image patches, which do not consist of pedestrians but are mis-classified into pedestrians. We call those samples as hard negative samples. To be specific, for the first round, initial negative training samples are randomly cropped from the negative example images; in the following rounds, hard negative samples are searched using the classifier generated in the previous round, over all negative example images. The above procedure is iterated until no significant performance gains are observed with further retraining. From our experiments, three rounds of retraining were observed to yield optimal performance; additional rounds only show very slight improvements, thus are considered to be unnecessary. We show how performance gains at each training round on the INRIA data set in Figure 3.4. We collect 5,000 negative samples at the first round and select another 5,000 hard negative samples in each following retraining round, resulting in a large negative sample pool of 20,000 in the end.

After boosting, each selected feature is assigned with a single weight. Those features with higher weights are considered to be more discriminative for pedestrians. In order to look into the locations of more informative features, we select the top 100 features with highest weights as a informative subset. Next, we plot an accumulative weight map by adding the weight of each feature in the subset, to the cells it covers. As shown in Figure 3.5, different colors are used to indicate the accumulative weights of cells. We can see that the upper body consists of more high-weight cells than the lower body. Especially, the head-shoulder area of the human body shows to be more discriminative for pedestrian detection than other body parts.

To perform a full image detection, we slide a window with a fixed size of 60×120 pixels, over the whole image and resize the input image to detect pedestrians of different scales.

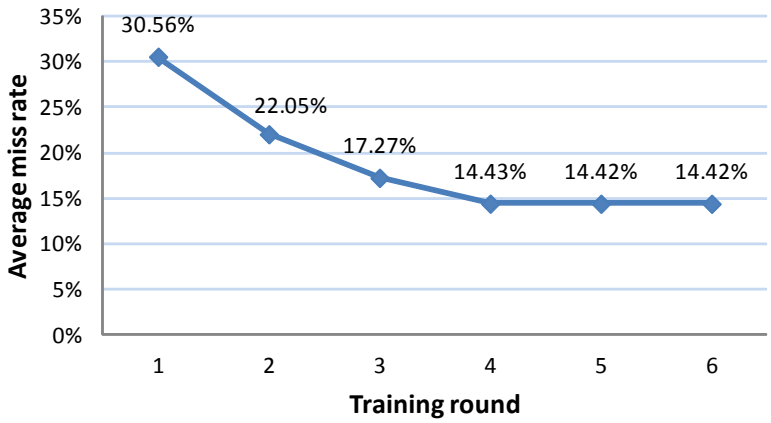


Figure 3.4: *Illustration of how performance gains at each training round on the INRIA data set. After three rounds of retraining, additional rounds do not show significant improvements.*

The spatial step size is set identical to the cell size for speed and the scale step is set to be 1.09 so that there are 8 scales in each octave. Besides, we use a simplified non-maximal suppression (NMS) procedure [Dollár et al., 2009a] to suppress nearby detections with lower confidence scores from the classifier.

3.3 Experiments

In this section, we discuss the impact of parameter settings on performance, compare our optimal detector to other state-of-the-art detectors on different pedestrian data sets, and also provide an analysis on runtimes.

3.3.1 Parameter Settings

In order to analyze the effects of different parameter settings and then find out the optimal one, we implement experiments under various parameter settings on the INRIA pedestrian data set.

Cell size: As shown in Figure 3.2, the boundaries of each body part is determined by the cell size. We present experimental results for cell sizes of 4×4, 6×6 and 8×8 pixels respectively. From Figure 3.6(a), we find that the cell size of 6×6 pixels produces the best results so we choose it to build our optimal shape model.

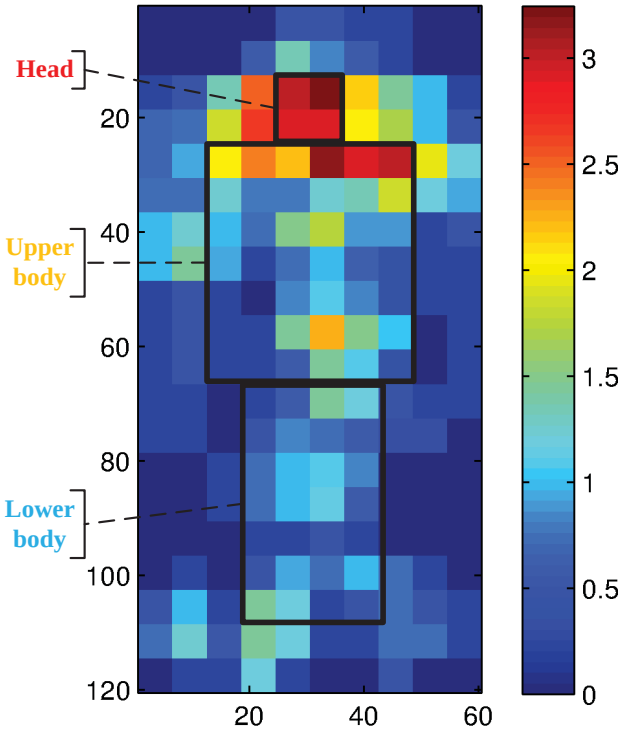


Figure 3.5: *Illustration of locations of representative features. Different colors are used to indicate the accumulative weights of cells after boosting. Three black bounding boxes indicate three body parts of our shape model. The head-shoulder area shows to be more discriminative for pedestrian detection than other body parts.*

Channels: We also use three kinds of channels as in [ChnFtrs]: color channels; gradient magnitude channels; and gradient histogram channels. As gradient histograms have been shown as the most informative channels in [Dollár et al., 2009a], we only try alternatives for color and gradient magnitude channels. We show the performance of various channel combinations in Figure 3.6(b), and we summarize our findings as follows: (1) LUV color channels are more discriminative than HSV channels; (2) using three gradient magnitude channels (one for each color channel) rather than one maximal magnitude channel results in approximately a 4% performance decrease; (3) using two gradient magnitude channels (along the x and y directions respectively) also leads to a slight performance decrease. Therefore, the optimal channel combination is LUV+GM+GH.

Smoothing: From Figure 3.6(c), we can see that pre-smoothing input images with binomial filters of radius 1 improves the performance by more than 3%, similar to [ChnFtrs]; however, from Figure 3.6(d), post-smoothing on channel values does not improve but instead significantly decreases the performance of our features. The reason is that post-smoothing on channel values seems to inhibit characteristic local difference, which is exactly what our

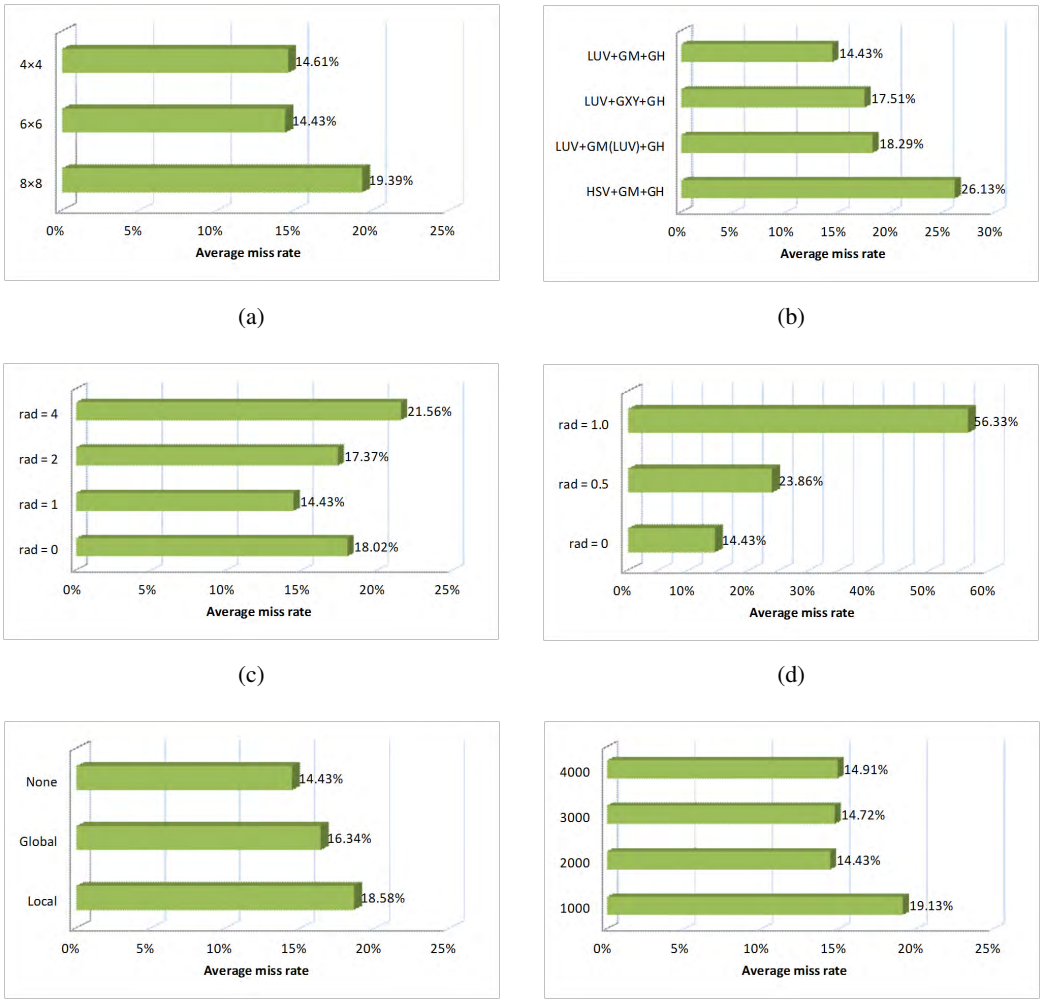


Figure 3.6: *Evaluation of different parameters on the INRIA pedestrian data set. (a) Cell sizes of the pedestrian shape model. (b) Channel combinations with color channels + gradient magnitude channels (GM) + gradient histogram channels (GH). (c) Pre-smoothing of colors with binomial filters of different radii. (d) Post-smoothing of channels with binomial filters of different radii. (e) Image normalization methods. Local intensity normalization is done inside each detection window; global normalization is done for the whole input image. (f) Number of weak classifiers.*

features try to interpret.

Image normalization: We notice that previous work on rectangular features typically employ various ways of normalization: [VJ] [Viola and Jones, 2004] applied local normal-

ization inside each detection window; [Roerei] [Benenson et al., 2013] reported performance improvements by applying global normalization on the input images. Therefore, we also analyze the influence of intensity normalization on our features. However, according to the results in Figure 3.6(e), both forms of normalization decrease the performance of our features and we conclude that our features work best without image normalization.

Number of weak classifiers: Increasing the number of weak classifiers brings more accurate decision boundaries, thus leading to better classification results; on the other hand, given the number of training samples and the dimension of features, a too large number of weak classifiers may lead to overfitting. Hence, we have to make a trade-off. We look for the best choice from experiments. From Figure 3.6(f), we find that detection performance is improved by approximately 5% when using 2000 rather than 1000 weak classifiers but performance starts to decrease slightly when the number of weak classifiers exceeds 2000.

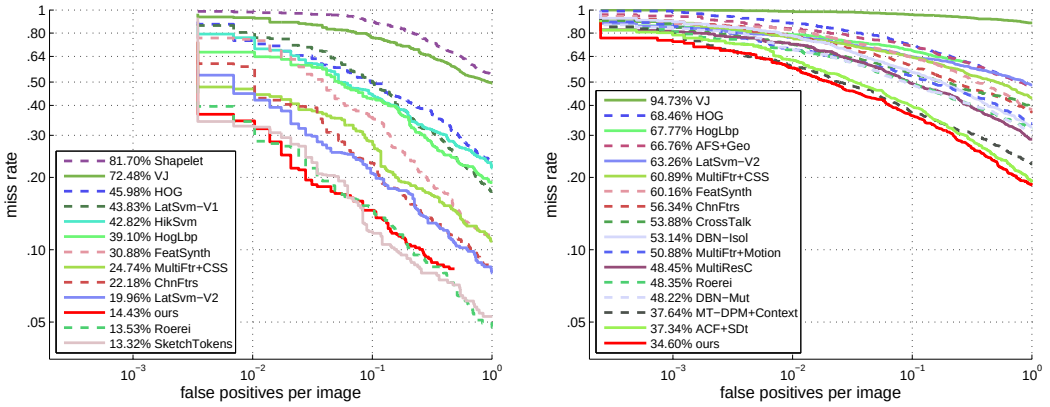
Based on the above analysis, we conclude that the optimal parameter setting of our features are as follows: cell size of 6×6 pixels; channels of LUV+GM+GH; image smoothing with binomial filters of radius 1; no channel smoothing; no image normalization; 2000 weak classifiers. Note that only the above setting is used in the following experiments.

3.3.2 Comparisons with State-of-the-art Detectors

For evaluation, we compare the performance of our detector using the optimal setting to other state-of-the-art detectors whose results are publicly available[1], using the experimental protocol explained in Section 2.3, ''Experiment Settings'', on the INRIA, Caltech and KITTI pedestrian data sets.

INRIA data set. The results in Figure 3.7(a) show that our detector outperforms the baseline detector [ChnFtrs] by more than 8% and reaches the state-of-the-art performance. Although our detector [Informed-Haar] obtains slightly higher average miss rates than two recently proposed detectors [Roerei] and [SketchTokens], it actually shows better performance in the FPPI range of $[10^{-2}, 10^{-1}]$, which is more interesting for some applications, for example, ADASs. Notably, the slightly better overall results of [Roerei] and [SketchTokens] comes at a price of 50 times larger feature pools and about 100 times more training time than ours.

[1] `http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/`



(a) (b)

Figure 3.7: *Experimental results of different detectors on the (a) INRIA and (b) Caltech pedestrian data sets under reasonable evaluation settings.*

Caltech data set. The Caltech data set is more challenging than the INRIA data set, due to the lower resolution and much more complex background in a real urban traffic environment. As shown in Figure 3.7(b), our detector not only outperforms the baseline detector [ChnFtrs] by around 20% but also obtains the overall best performance, and consistently performs better than the second and third best detectors [ACF+SDt] and [MT-DPM+Context] over the whole FPPI range. Particularly, we note that it even outperforms those detectors which integrate additional motion information with spatial information, such as [Walk et al., 2010] and [Park et al., 2013] .

We further compare the performance under different occlusion conditions. The experimental results shown in Figure 3.8 are implemented on the Caltech data set, since it provides occlusion labels and annotations for visible area, which enable us to calculate the percentages of occlusions. Observing the curves in Figure 3.8, we obtain the following conclusions: (1) The performance of all the detectors drops significantly as occlusion increases. This trend indicates that occlusion is an important factor which affects the performance, thus the importance of robustness against occlusions. (2) Our detector seems least affected by occlusion because it shows stably high ranks over all occlusion levels. (3) In fact, our detector achieves the best performance among all tested detectors for the cases of no and heavy occlusion, and we conclude that the informed design of our features yields robustness against occlusions. (4) Notably, our detector even outperforms those detectors that employ explicit occlusion handling strategies, for example, [DBN-Isol] and [DBN-Mut], for all levels of occlusion.

In addition, we show several detection examples of our detector under different scenarios

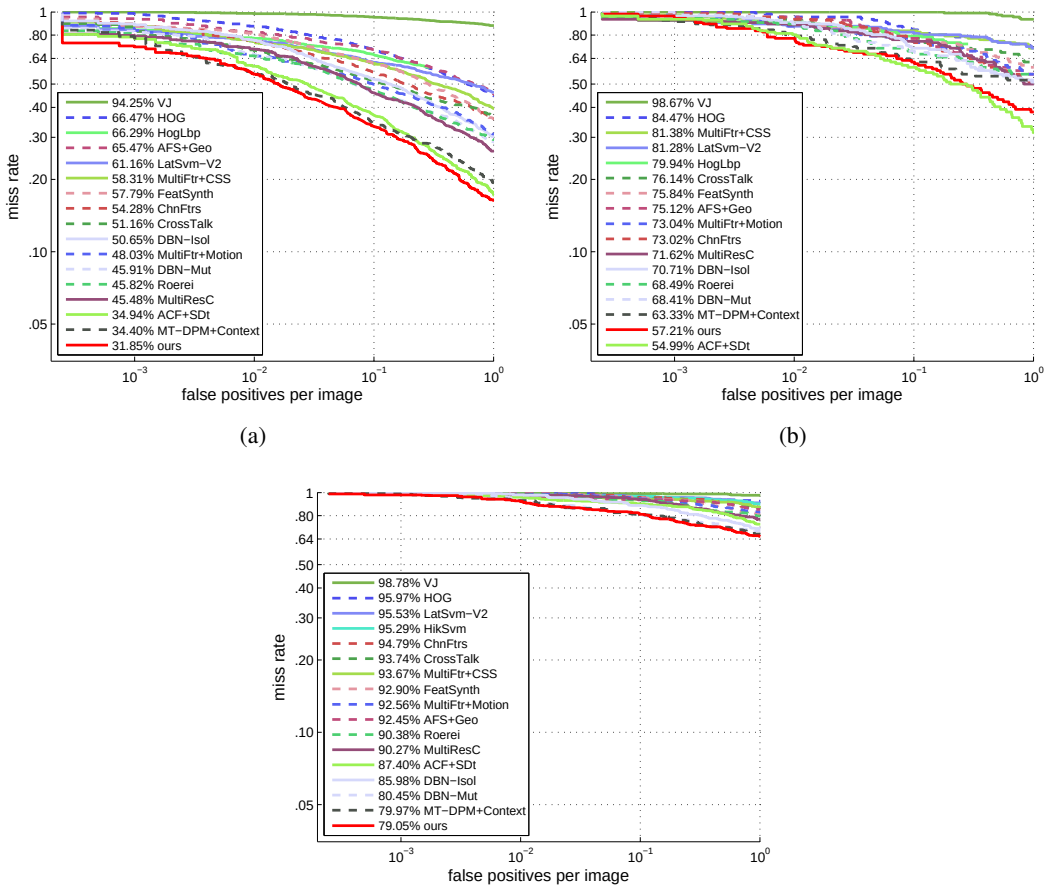


Figure 3.8: *Evaluation results under different occlusion conditions on the Caltech pedestrian data set. (a) No occlusion. (b) Partial occlusion(1-35% occluded). (c) Heavy occlusion(35%-80% occluded).*

from the Caltech pedestrian data set in Figure 3.11.

KITTI-Train data set. The KITTI-Train data set is considered as a more difficult data set, and experimental results are shown in Figure 3.9. Unfortunately, we are not able to make as extensive comparisons as on the INRIA and Caltech data sets, due to the unavailability of results from other state-of-the-art detectors. However, we still notice a significant improvement of our approach, compared to our baseline detector [ChnFtrs].

We provide a more comprehensive comparison for 22 state-of-the-art detectors and ours with respect to detector components as well as performance in Table 4.2. First, as HOG

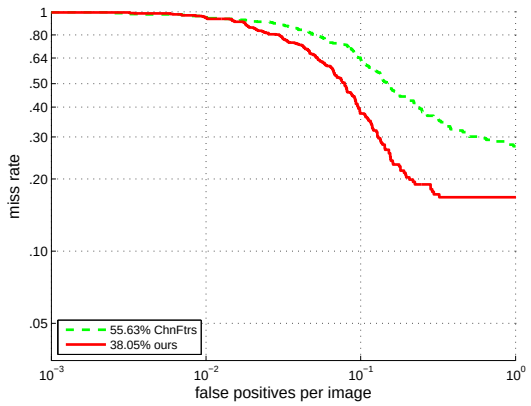


Figure 3.9: *Experimental results on the KITTI-Train data set.*

and channel features are the most popular features for pedestrian detection, we group all the detectors into three categories: HOGs based, channels based and others, according to which kind of features they employ in major. Then we indicate which classifiers they use, and whether they apply motion information in the third and fourth column for all the detectors considered in this chapter. In the last two columns, corresponding performance on the INRIA and Caltech pedestrian data sets are demonstrated in terms of average miss rates. We summarize our insights as follows:

(1) Most detectors use HOG features in various ways, hence, HOGs are still an established strong baseline after being proposed for around 10 years. After integral channel features being proposed in 2009, more recent detectors tend to focus on channel features, which obtain better performance as well as higher speed.

(2) In terms of classifiers, most HOGs based detectors utilize SVM, while channels based ones all use AdaBoost. The reason is that channel features are usually of higher dimensions, and boosting methods are more efficient to select the most discriminative ones from a large number of candidate features.

(3) While considering the problem of pedestrian detection over frame sequences, motion is an important cue, as supplementary to spatial information. However, motion information is rarely used by the state-of-the-art detectors. One reason lies in that, it is computational expensive to obtain accurate and dense optical flow maps, which directly describe the motion information between successive frames. On the other hand, it is still an open problem about in which way motion may help for pedestrian detection. We can easily understand that pedestrians exhibit special appearance than other objects, but is there also some inter-class variability with respect to motion? Still, a recent success of [ACF+SDt] indicates that it is promising to exploit motion information for pedestrian detection.

Category	Detector	Classifier	Motion	Average miss rate	
				INRIA	Caltech
HOGs based	HOG [Dalal and Triggs, 2005]	linear SVM	×	45.98%	68.46%
	MultiFtr [Wojek and Schiele, 2008]	AdaBoost	×	36.50%	68.62%
	MultiFtr+CSS [Walk et al., 2010]	AdaBoost	×	24.74%	60.89%
	MultiFtr+Motion [Walk et al., 2010]	linear SVM	√	/	50.88%
	HikSvm [Maji et al., 2008]	HIK SVM	×	42.82%	73.39%
	HogLbp [Wang and Han, 2009]	linear SVM	×	39.10%	67.77%
	LatSvm-V1 [Felzenszwalb et al., 2008]	latent SVM	×	43.83%	79.78%
	LatSvm-V2 [Felzenszwalb et al., 2010]	latent SVM	×	19.96%	63.26%
	FeatSynth [Bar-Hillel et al., 2010]	linear SVM	×	30.88%	60.16%
	MultiResC [Park et al., 2010]	latent SVM	×	/	48.45%
	AFS+Geo [Levi et al., 2013]	linear SVM	×	/	66.76%
	MT-DPM+Context [Yan et al., 2013]	latent SVM	×	/	37.64%★
	DBN-Isol [Ouyang and Wang, 2012]	DeepNet	×	/	53.14%
	DBN-Mut [Ouyang et al., 2013]	DeepNet	×	/	48.22%
Channels based	ChnFtrs [Dollár et al., 2009a]	AdaBoost	×	22.18%	56.34%
	CrossTalk [Dollár et al., 2012]	AdaBoost	×	18.98%	53.88%
	VeryFast [Benenson et al., 2012]	AdaBoost	×	15.96%	/
	SketchTokens [Lim et al., 2013]	AdaBoost	×	13.32%★	/
	Roerei [Benenson et al., 2013]	AdaBoost	×	13.53%★	48.35%
	ACF+SDt [Park et al., 2013]	AdaBoost	√	/	37.34%★
	Informed-Haar	AdaBoost	×	14.43%★	34.60%★
Others	VJ [Viola and Jones, 2004]	AdaBoost	×	72.48%	94.73%
	Shapelet [Sabzmeydani and Mori, 2007]	AdaBoost	×	81.70%	91.37%

Table 3.1: *Comprehensive comparisons for state-of-the-art pedestrian detectors. Each row in this table summarizes information as to classifiers, whether motion information is used in a particular approach, and displays the corresponding performance in terms of average miss rates on the INRIA and Caltech pedestrian data sets. The approach proposed in this paper [Informed-Haar] yields state-of-the-art performance on the INRIA data set and consistently better results than previously reported on the Caltech data set. We annotate the top three detectors for each data set with a ★ following each average miss rate.*

3.3.3 Runtimes

Speed is another important factor to evaluate detectors, because we are aiming to real-time applications. Unfortunately, it is an extremely difficult task to provide a comprehensive comparison of runtimes among all state-of-the-art detectors considered in this chapter, because different detectors are implemented on different machines, some even heavily rely on GPU computations, for example, [VeryFast] [Benenson et al., 2012] and [Roerei] [Benenson et al., 2013]. By contrast, our detector is implemented in Matlab, on an Intel Core-i7 CPU (3.5GHz). Therefore, it does not make much sense to list runtimes from different computing architectures.

In the following, we explain the runtimes of our detector using the optimal parameters as

illustrated in Section 3.3.1, ‘‘Parameter Settings’’ on the Caltech data set. For training, it takes around one hour for four rounds; for testing, it takes approximately 1.6 seconds for a 640×480 image. While looking into the sources of computational costs, we notice that besides channel computation, the main cost comes from local sums and subtraction, both of which can be parallelized for further speed-up. Therefore, our detector is expected to reach real-time efficiency running on a powerful machine and with GPU computation enabled.

3.4 Discussions

In this section, we discuss several important issues regarding our feature design, so as to argue that our features are reasonably designed and the reported good performance is essentially convincing.

Compactness of features. We call our features as *compact features* at the beginning of this chapter because the proposed rectangular features do not use randomly selected rectangles, but employ a relatively small template pool designed based on a statistical shape model. Compared to those recently proposed detectors, for example, [Roerei][Benenson et al., 2013] and [SketchTokens][Lim et al., 2013], whose performance is close to ours, our feature pool is more than 50 times smaller. The compactness of our features is proved by the competitive results obtained from a smaller feature pool.

Intra-class variations. One may question that how our single shape model can adapt to intra-class variations of pedestrians, mainly coming from different viewpoints and postures. Looking back at the pedestrian body shape shown in Figure 3.2, it looks like being observed from the front or back view. One may doubt that how could we detect those pedestrians seen from a side view, for example, those pedestrians who are crossing the street, which are an important concern for safety and show quite different shapes from the average pedestrian body shape in Figure 3.2. To answer the above questions, we first declare that from our everyday knowledge, the upper body (including the head) of pedestrians shows much fewer variations than the lower body (including the feet). For example, the head is always above the shoulder, but two feet can be crossed while walking. Fortunately, this is also automatically learned by our classifier. From the weight map shown in Figure 3.5, the most informative features selected by our classifier are always found in the head-shoulder area, although our candidate features distribute evenly all over the body parts. Therefore, our features are largely invariant against intra-class variations, because they are able to capture the specific common shape of head-shoulder structure of the human body.

Figure 3.10: *Three examples of occlusions happen at lower body. Example images are from the Caltech pedestrian data set, and red dashed bounding boxes are used to indicate the occluded body parts. In the given examples, occlusions are caused by an dustbin, a moving car and another walking-by pedestrian, respectively.*

Robustness against occlusions. We show our robust results under different occlusion conditions in Figure 3.8, but the reasons are still need to be investigated. From our observation, occlusions happen more at the lower body (see Figure 3.10 for examples); by contrast, our classifier emphasizes more on the upper body and those features from the lower body are automatically ranked as less informative. Therefore, the occluded lower body do not have a significant negative effect on the final confidence score output from our classifier.

Redundant first-order channel features. Considering the success of first-order channel features in [ChnFtrs], one expect better performance by combining the second-order and first-order features, as the latter ones describe the uniform texture inside each body part. However, from our experiments, the combination of first-order features does not bring any improvements but decreases the performance slightly. Thus, we decide to exclude the first-order features and assume them to be redundant. We attribute this to the ability of our templates to represent uniform texture. As an ensemble, our templates cover the whole body after shifting and uniform texture on clothing can be represented as minimal feature values from those templates only cover one body part.

3.5 Summary

In this chapter, we proposed a particular approach, which was in a different line from a current trend of employing feature pools of ever increasing sizes, in the field of pedestrian detection. From the perspective of recognition accuracy, it is not necessarily guaranteed that additional efforts spent on computing high dimensions pay off in terms of accuracy, because redundancy may exist among the feature values. From the perspective of computational cost, those large feature pools necessitate the use of powerful hardware in order to guarantee real time capability. We therefore explored more compact features, which are of lower dimension but still guarantee the state-of-the-art performance. This goal can be achieved if our features

Figure 3.11: *Detection examples of our detector under different scenarios from the Caltech pedestrian data set. Green solid bounding boxes, yellow dotted bounding boxes and red dotted bounding boxes indicate true positive, false positive and false negative (missed) results, respectively. (a) Small scale pedestrians walking along the street. (b) One missed pedestrian due to heavy occlusion (> 70%). (c) Complex scenario at one intersection with one false positive occuring at one tree. (d) Pedestrians with occlusions. (e) Multiple pedestrians walking across the street. (f) One motorcyclist falsely detected as a pedestrian. (g) One pedestrian of low contrast. (h) Pedestrians with pets. (i) One traffic sign falsely detected as a pedestrian.*

are specially designed for the pedestrian category, which shows high inter-class yet low intra-class variations in terms of appearance.

Computed from a large number of pedestrian images, an average gradient magnitude map shows a clear up-right human body shape. This stable geometrical structure enables us to divide the human body into distinct parts, which usually exhibit different colors or textures. Local difference is a suitable measurement to interpret high contrasts at conjunctions of different body parts, and low contrasts inside one single part as well.

We compute the local differences on rectangle level instead of on pixel level, because local sums inside rectangles are more tolerant to noises and rectangular features are efficient to compute by employing integral images. Therefore, the shape model is covered with grids of cells and Haar-like templates are generated by sliding rectangular windows of various sizes all over the model. After this procedure, a set of location specific, binary and ternary Haar-like templates are created. Next, each template is filled with multiple channel values, thus generating a pool of multi-modal & multi-channel Haar-like features.

We summarize the advantages of our features as follows.

1. Easy to implement. One just need minor modification to the implementation of integral channel features, which can be downloaded at `http://vision.ucsd.edu/~pdollar/toolbox/doc/`.
2. Easy to train. A simple boosting method can be used for selecting the most informative features.
3. Fast to apply. Besides channel computation, the computational costs include local sums and subtraction, both of which can be parallelized to reach real-time capability.

The weighting scheme provided us with a simple mechanism of generating multi-modal & multi-channel Haar-like features and we applied boosting to determine the most informative ones. As our approach does not require computing all possible configurations of rectangles within a sliding window nor is based on random sampling of rectangle features, it marks a middle ground among recently published similar approaches.

From extensive experiments on standard benchmark data sets, we found our detector to achieve state-of-the-art performance on the INRIA pedestrian data set and, for the Caltech pedestrian data set, we found it to outperform all previously proposed approaches considered in our tests. In addition, our model-based rectangular features proved to be highly robust to occlusions and even outperformed several methods that design explicit mechanisms for occlusion handling.

Future Work. Given the reasonable results obtained by informed Haar-like features, it appears promising to further explore model driven design of efficient rectangular features. Immediate extensions of the approach presented in this chapter could be to incorporate additional channels such as motion information. Also, we see more challenging extensions, *e. g.* to define multiple shape models with respect to parts or viewpoints for one object category, thus enabling more shape-variant object detection.

Chapter 4

Center-surround Contrast Features

Looking at previous features designed for pedestrian detection, prior knowledge or complex image processing technologies have been used in various forms. Yet, one must acknowledge that the state-of-the-art performance still lags behind human vision, in terms of both accuracy and speed. As we all know, while given a task of finding pedestrians, human vision is capable of precisely and rapidly localizing pedestrians of a huge range of scales, postures, occlusion levels and contrasts. Consequently, we are motivated to analyze how the human visual systems deal with input visual information. The mechanisms found from the analysis can be used to instruct the design of novel features for pedestrian detection. In this chapter, experimental results are presented to show that employing biologically inspired mechanisms can indeed aid recognition and improve the performance.

In human visual systems, analog visual signals are first converted into electronic signals at photoreceptive cells. After that, the retinal tissue begins to process the information. In the first layer of bipolar cells, electrical membrane potentials are locally aggregated and grouped bipolar cells report to different types of ganglion cells. In this procedure, a center-surround fashion is found to modulate the electronic signals to enhance contrasts by a lateral wiring of so called horizontal respectively amacrine cells, at the transitional synapses between photoreceptive and bipolar cells, and also from bipolar to ganglion cells. It was found that the output of certain ganglion cells can be simulated by a simple difference of Gaussian (DoG) filter responses [Rodieck, 1965] or more complex oriented Gabor filter [Jones and Palmer, 1987]. For more details about retinal cell types and their wiring, please refer to an in-depth survey, for instance, [Lee et al., 2010].

This work has been partially published in [Zhang et al., 2015, 2014c].

In later processing stages in human brains, the center-surround mechanism is also found to guide human *attention* and thus affects how people recognize objects of interest. To numerically measure human attention, saliency maps are employed in recent computational approaches, where the psychophysical theory of center-surround have been widely used [Frintrop et al., 2010]. However, unlike visual attention, which corresponds to bottom-up, model-free analysis of signals from the environment, visual search for designated class of objects requires top-down saliency, which tunes the scoring of basic features to the expected appearance.

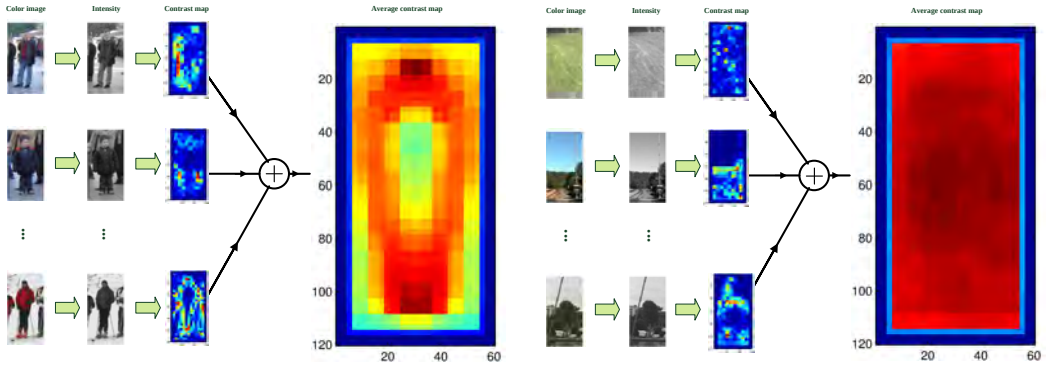
In this chapter, we propose center-surround contrast features motivated by the human visual systems and to tune them towards characterizations of pedestrian appearance. Our contribution can be summarized as follows:

Statistical multi-channel cell descriptors: For each cell region, *i. e.* local image patch, multiple channel information including colors and gradients are considered, in order to cope with challenging variations of clothing or articulations of the human body. Instead of using the channel values directly, we describe each cell with two kinds of statistical descriptors: (1) mean and variance values, which ensure maximum entropy for a continuous Gaussian distribution [Cover and Thomas, 2006]; (2) a series of frequencies, observed over discrete intervals (bins) for a bilinear interpolated histogram.

Multi-direction and -scale contrast vectors: For the purpose of incorporating more specific information between central and surrounding cells, adjacent image regions are treated in different directions individually rather than as a single surrounding region, resulting in multi-direction contrast descriptors; according to the general architecture of most visual saliency systems, a contrast pyramid is built by computing statistical features at different cell sizes.

Extensive evaluations under various configurations: In order to find out the optimal feature scheme for pedestrian detection, we implement various contrast measurements for both descriptors and at different scale structures. From extensive evaluations on the INRIA data set, we find that the optimal scheme is to use a Gaussian-$\mathcal{W}_2$combination and a 4-6-8-10 scale structure.

This chapter is proceeded as follows: Section 4.1, ‘‘Related Work’’ introduces related work on difference based features for pedestrians and contrast measurements used for visual saliency systems; Section 4.2, ‘‘Overview of Feature Extraction’’ presents an overview as to our feature extraction procedure. Two key components of this procedure, namely statistical descriptors and contrast measurements are explained in Section 4.3, ‘‘Statistical Cell Descriptors’’ and Section 4.4, ‘‘Contrast Measurements’’, respectively. Our classification procedure is presented in Section 4.5, ‘‘Classification’’, followed by a discussion of thorough and extensive experiments in Section 4.6, ‘‘Experiments’’, where we evaluate different feature schemes and compare state-of-the-art detectors on standard benchmarks. Finally,



(a) (b)

Figure 4.1: *Illustration of average center-surround contrast maps generated from positive and negative samples from the INRIA pedestrian data set. The warmer colors indicate higher contrast values, and vice versa. (a) Average contrast map for pedestrians; (b) Average contrast map for non-pedestrians.*

we summarize our findings and propose several directions for future work in Section 4.7, ‘‘Summary’’.

4.1 Related Work

Since in this chapter we propose center-surround contrast features, which numerically interpret local difference using appropriate contrast measurements, we focus the following literature review on difference based features for pedestrians, and center-surround contrast measurements used by computational visual attention approaches.

4.1.1 Difference Based Features for Pedestrian Detection

Local and global differences are reasonable representations for texture information, which are often characteristic for different categories of objects. Therefore, in the field of pedestrian detection, aiming for robust classification, many features are designed based on difference, with respect to intensity or colors, in various forms. We group these features into pixel-wise and patch-wise ones and briefly introduce some examples in the following.

Pixel-wise difference based features.

Gradients characterize pixel difference with respect to intensity or colors between neighboring pixels by two numerical elements: magnitude and orientation. Gradients can be used directly as features yet more often act as basic elements for high level features. For instance,

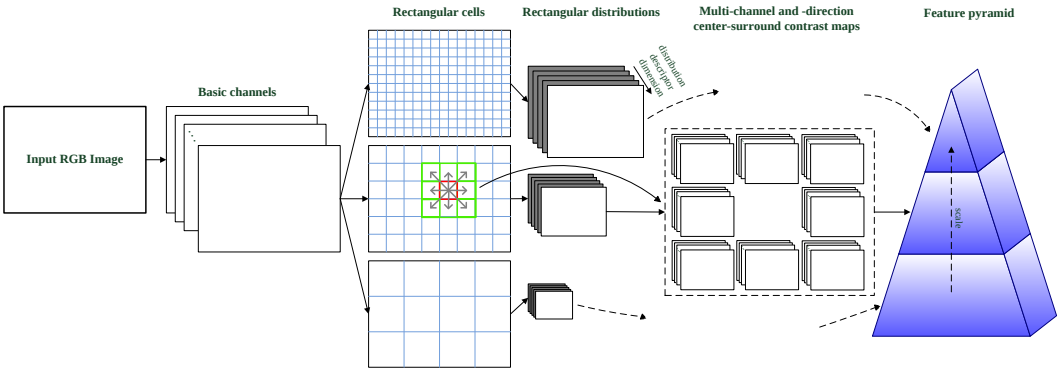


Figure 4.2: *Flow chart of center-surround feature extraction. Here, we consider a three-scale structure as an example but different scale structures can be used as well.*

the arguably most popular features HOGs [Dalal and Triggs, 2005] for pedestrian detection are built on gradients.

LBP features [Ojala et al., 1996] are another kind of pixel-wise difference based features. Unlike gradients, which describe local difference in a precisely numerical way, LBP features just compare the intensity values of neighboring pixels and encode their relationships with binary codes. This rather coarse strategy loses detailed information in a way, but on the other hand, it ensures the robustness to noises, which usually exist in real world data.

Based on the above comparison, it may be a wise choice to combine gradients and LBP features in appropriate ways. In fact, [Wang and Han, 2009] found that such a combination was able to cope with occlusions successfully; [Ma et al., 2013] proposed a set of edge orientation histogram (EOH) and oriented LBP based features to describe cell-level and block-level structure information.

Patch-wise difference based features.

Haar-like features [Viola and Jones, 2001] compute local sums of intensity values over image patches and use the subtraction between two sum values to represent local difference. [Viola et al., 2005] employed Haar-like features on both intensity and motion information for pedestrian detection.

Color Self Similarity (CSS) features proposed by [Walk et al., 2010] describe each image patch by one color histogram and then represent global difference by computing the distances between pairs of color histograms. Compared to sole HOG features, the integration of CSS features brought about significant improvements, since CSS features allow for representing uniform textures found in people's clothing.

From the above discussions on two kinds of difference based features, we acknowledge that extensive efforts have been made to interpret difference in various forms. Unfortunately, the

state-of-the-art performance is still far behind humans' recognition accuracy. In this way, we are motivated to look into how human brains process the input visual data, and then design human vision driven features for more robust pedestrian detection.

To our best knowledge, the first attempt was found in [Montabone and Soto, 2010], which designed human vision inspired features dedicated to pedestrian detection. They compute difference between a central pixel and its surrounding pixels with respect to intensity. Our features proposed in this chapter can be considered as a significant extension to this early method, and we summarize the difference as follows:

- Local difference is considered patch-wise rather than pixel-wise. In our approach, patches are defined as square regions.
- Center-surround contrasts are computed in multiple channels (not only on colors but also on gradients).
- Each square region is represented by a statistical descriptor instead of by channel values directly.
- Surrounding regions are treated individually rather than as a whole. As a result, more detailed information regarding local difference is incorporated.

4.1.2 Center-surround Contrast Measurements

Various contrast measurements have been proposed in computational visual attention approaches. The most popular one is the response of DoG-filters or approximations of these [Itti et al., 1998]. More recently, some researchers tried to capture more information about image patches by representing the central and surrounding regions in terms of feature distributions, rather than intensity values. Color histograms were employed in [Klein and Frintrop, 2011], and normal distributions were chosen in [Klein and Frintrop, 2012]. The above two distributions are discrete and continuous respectively. To evaluate the difference between two distributions, various contrast measurements can be computed according to some properties of given distributions. For instance, Kullback-Leibler divergence [Kullback and Leibler, 1951] was used in [Klein and Frintrop, 2011] to measure the distance between two histograms.

It is an option to choose the best previous measurement for our approach, however, there is no evidence about superiority in the literature. On the other hand, the previous measurements were only used for rather simple scenarios, for example, a big red ball lying on the green grass. In fact, in our case, the background shows to be much more complex and the previously proposed measurements are not guaranteed to perform well. Consequently, it is necessary to evaluate different contrast measurements, and then find out the optimal one for our applications.

4.2 Overview of Feature Extraction

In this section, we explain how we extract center-surround contrast features.

Before feature extraction, we seek the evidence of center-surround contrasts being discriminative for the category of pedestrians, from statistical data. Pedestrian (2,416) and non-pedestrian (5,000) images are collected from the positive and negative samples of the INRIA pedestrian data set respectively. To simplify, all the source images are converted to gray scale images, in other words, we only consider one channel of intensity here for an example. Next, each image is divided into 4×4 and 6×6 pixel square regions, resulting in a two scale structure. Then we compute the difference between each cell and its surrounding cells with respect to mean value of intensity as the contrast value, and add this value to the pixels belonging to the central cell. After iterations on each cell and over two scales, two average contrast maps are generated for the pedestrian class and non-pedestrian class, respectively. In Figure 4.1, we can recognize the shape of human body on the average contrast map for pedestrians, while no clear texture is shown on the average contrast map for non-pedestrians. In this way, we demonstrate that center-surround contrasts are discriminative for pedestrians.

Therefore, we design center-surround contrast features for pedestrian detection. An illustration of our feature extraction procedure is shown in Figure 4.2. The whole procedure can be divided into the following four steps:

- Channel computation. We compute multiple channel values (*e. g.* colors and gradients) pixel by pixel, resulting in multiple channel maps for the input image.
- Statistical description. Each channel map is divided into square cells of a fixed size and each cell is described by some statistical descriptor.
- Contrast computation. We compute the difference using an appropriate contrast measurement, between each cell and its eight nearest surrounding cells individually so as to obtain a multi-direction contrast vector. This computation is repeated over all channel maps.
- Iteration over multiple scales. We repeat the second and third step with different cell sizes and thus obtain a multi-scale contrast pyramid for the whole image.

The final feature vector is generated by concatenating all the contrast values computed from different scales.

4.2.1 Channels

Inspired by the success in Chapter 3, ‘‘Informed Multi-channel Haar-like Features’’, we also consider a total of 10 different channels: 3 channels for LUV colors, 1 channel for gradient magnitude, and 6 channels for histograms of oriented gradients.

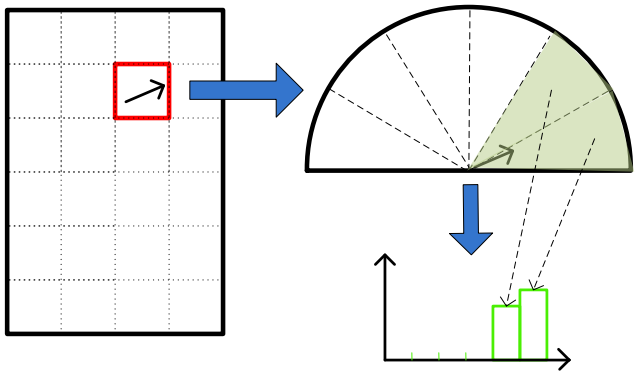


Figure 4.3: *Illustration of a histogram of oriented gradients computed for a single pixel with gradient magnitude and orientation.*

Note that here we compute the histograms of oriented gradients in a different way. In traditional channel computation approaches, only one histogram is computed for a group of pixels inside one image region. But in this chapter, we computer one histogram for each pixel, which is convenient for further statistical description for each cell. In fact, this is easy to implement. We just simply quantize the gradient magnitude of each pixel into two out of six orientation bins, by employing bilinear interpolation, see Figure 4.3 for an illustration.

For the purpose of removing noise, input images are smoothed with a binomial filter [Haddad, 1971] of radius 1, *i. e.* $\sigma \approx 0.87$, before channel computation. In contrast, post-smoothing on channel values is not applied as a decrease on performance was observed.

4.2.2 Center-surround Neighborhood Patterns

In order to design center-surround cell pairs in a more reasonable way, four patterns are proposed and explained in the following.

C_1S_8 *pattern:* Since we divide the channel maps into square regions, each cell is equally surrounded by eight nearest neighboring cells, which are considered as surrounding cells and denoted as $[C_1^s, C_2^s, ..., C_8^s]$. These eight cells can be treated as a whole, denoted as C_1S_1 pattern, similar to traditional computational visual attention approaches. But we propose to treat them separately, denoted as C_1S_8 pattern. This is because we find that C_1S_8 pattern obtains a significantly better performance than C_1S_1 pattern (cf. Figure 4.7) from our experiments. Thus, we use this C_1S_8 pattern in our approach, so as to integrate local difference information in eight directions respectively, resulting in a multi-direction contrast vector for each cell along one channel.

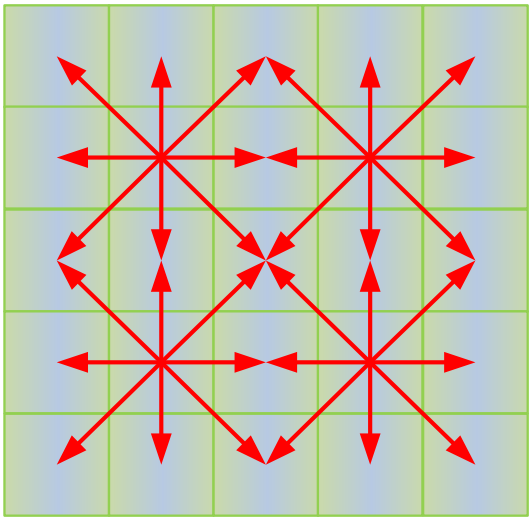


Figure 4.4: *Sparse neighborhood map. Each red arrow points from the central cell to one of its neighboring cells.*

Sparse pattern: If we densely iterate the above C_1S_8 pattern on each cell, significant redundancy will emerge, because each adjacent pair of cells is counted twice. To cope with this problem, we use a cell step of two cells along both image width and height directions, which means we have one cell treated as the central cell for every two cells along each direction. This sparse pattern results in a sparse neighborhood map as shown in Figure 4.4. One may question that this map is too sparse, where many neighboring relationships are discarded. The percentage of information loss would be extremely high when the number of cells is small. Therefore, we propose the following shift pattern to eliminate this negative effect.

Shift pattern: We propose a shift mechanism where we define two cell layers for each cell size and iterate the above C_1S_8 and sparse patterns on each respectively. The first layer is defined starting from the left top pixel; while the second layer is defined starting from an alternative point, which is shifted with a bias number times the cell size along both horizontal and vertical directions. According to the Nyquist-Shannon sampling theorem [Shannon, 1949], this bias number is chosen to be 0.5. Both layers are divided into square cells with the same cell size. To ensure the number of cells to be integers along both directions, some pixels at the right or bottom borders may be discarded. An illustration of the shift pattern is shown in Figure 4.5, where the second layer does not cover the whole image.

Multi-scale pattern: Finally, in order to describe local difference in both coarse and fine manners, we divide the channel maps with different cell sizes to build a contrast pyramid which is in accordance with the general architecture of most computational visual attention

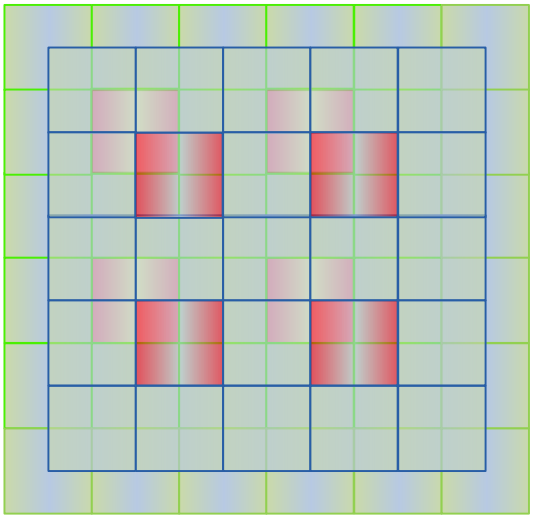


Figure 4.5: *Illustration of the shift pattern. Two layers of cells are denoted with green and blue grid lines. The red cells denote central cells with eight nearest neighboring cells.*

systems.

4.2.3 Center-surround Contrasts

The most important part of our feature extraction is to represent the difference between two square regions. This problem involves two key elements: one is the representation of channel values; the other is the contrast measurements. In Section 4.3, ‘‘Statistical Cell Descriptors’’, we introduce two kinds of distributions for channel values to build a statistical descriptor for each cell; and in Section 4.4, ‘‘Contrast Measurements’’, we consider corresponding contrast measurements to numerically describe the difference between two given descriptors. In order to find the strongest center-surround contrast features for pedestrian detection, we conduct extensive experiments and comprehensive comparisons on various combinations of distributions and contrast measurements. Experimental results under different schemes are presented in Section 4.6, ‘‘Experiments’’, and the optimal descriptor-measurement combination is chosen accordingly.

4.3 Statistical Cell Descriptors

The distribution of channel values inside each cell is unknown, like a ‘‘black box’’. To estimate it, we consider both continuous and discrete statistical descriptions:

- A Gaussian distribution.
- A bilinear interpolated histogram.

In our following discussion, we denote the channel map for the whole image along channel i as P^i and the channel values for a specific cell c as a vector $P_c^i = [v_1^i, v_2^i, ..., v_p^i]$, where p indicates the number of pixels inside one cell.

4.3.1 Gaussian Distributions

Gaussian distribution is a reasonable choice to estimate image data. We notice that it is applied in many classic low-level vision models, for instance, in [Horn and Schunck, 1980]. A big advantage of choosing Gaussian distribution is that normality makes many mathematic formulas convenient to solve, which can be found not only in this section, but also in Section 4.4.1, ''Measurements for Gaussian Distributions''.

To obtain a numerical descriptor for each Gaussian distribution, we apply maximum likelihood (ML) estimation of the parameters and obtain mean and variance values as

$$\hat{\mu}_c^i = \frac{1}{p}\sum_{k=1}^{p} v_k^i = \overline{P_c^i}, \tag{4.1}$$

and

$$\hat{\Sigma}_c^i = \frac{1}{p}\sum_{k=1}^{p}(v_k^i - \overline{P_c^i})^2 = \overline{(P_c^i)^2} - \overline{P_c^i}^2. \tag{4.2}$$

From Equation 4.1 and Equation 4.2, we can see that the estimation has been simplified to computation of two local averages: $\overline{P_c^i}$ and $\overline{(P_c^i)^2}$. Given the region size, to acquire the average values for all cells, we just need to obtain sum values, which can be computed efficiently by using integral images. Therefore, we employ two integral images along each channel: one for the original channel image P^i and the other for the squared channel image $(P^i)^2$. This strategy avoids extensive summations per individual cell, and thus significantly reduces computational complexity.

After the estimation, we represent the channel values inside cell c along channel i by using the mean and variance values, resulting in a two dimensional descriptor, denoted as:

$$G_c^i = [\mu_c^i, \Sigma_c^i]. \tag{4.3}$$

4.3.2 Histograms

Histogram is a kind of discrete representation of distributions, and has been widely used for image data. One big advantage of using histograms is that no prior assumption of the underlying statistics has to be made, which ensures its applicability to represent various kinds of data. Histograms are represented by a series of frequencies, in other words, they count the amount of observed data appear in discrete intervals. Another advantage of using histograms is its tolerance to noise and minor intra-class difference. Moreover, the degree of tolerance can be adjusted by choosing different numbers of bins. Generally, a finer description of the original data can be obtained by using a larger number of bins, and vice versa.

It is rather time consuming to individually compute one histogram for each cell from different scales. Thus, we consider employing integral histograms proposed in [Porikli, 2005]. An integral histogram can be decomposed into several integral images, each corresponding to one bin value. Each integral image counts the number of pixels that fall into the current bin, from each pixel to the top left one.

We implement bilinear interpolation instead of simple voting for histograms to eliminate bias, which happens when many values fall close to the interval borders. In this way, each value contributes into two nearest bins rather than one bin which it exactly falls into. The contributions are weighted using the distance between the given value and the bin center.

The data scales of bin values from different histograms may vary significantly due to various data magnitudes of different channel values, and various cell sizes. To eliminate these effects, we normalize each local histogram for each cell along one channel so that it sums up to 1. In the end, given b bins, we obtain a histogram H_c^i as a descriptor vector for channel vector P_c^i:

$$H_c^i = [h_c^i(1), h_c^i(2), ..., h_c^i(b)], \sum_{k=1}^{b} h_c^i(k) = 1. \tag{4.4}$$

Note that b is an important parameter and is further discussed in Section 4.6, ‘‘Experiments’’.

4.4 Contrast Measurements

In this section, we introduce multiple contrast measurements for each statistical descriptor. The combination of a descriptor and a corresponding measurement forms one specific scheme for feature extraction. To summarize, we explore six different combinations:

- Gaussian-$\mathcal{W}_2$
- Gaussian-L^2
- Gaussian-SGrd

- Histogram-KLD
- Histogram-Hellinger
- Histogram-HI

In the following, the channel values for a central and a surrounding cell are denoted as P_c^i and P_s^i, respectively. The contrast vector $\overrightarrow{cst}(P_c^i, P_s^i)$ is computed using different measurements.

4.4.1 Measurements for Gaussian Distributions

Three different contrast measurements are introduced to compute the difference between P_c^i and P_s^i, each represented by the descriptor in Equation 4.3. We compare the results of those three measurements in Section 4.6, ''Experiments''.

$\mathcal{W}_2$ distance

In mathematics, the Wasserstein distance is a function defined between two probability distributions on a given metric space M. Intuitively, if each distribution is considered as some amount of ''mud" piled on M, the metric is the minimum cost of turning one pile into the other along M. This cost can be computed by multiplication of the amount of mud that needs to be moved and the distance it has to be moved. In this way, the L^1 norm Wasserstein distance is also known as ''earth mover's distance''.

The $\mathcal{W}_2$ distance (2nd Wasserstein distance) was first introduced as a measurement for center-surround contrast by [Klein and Frintrop, 2012] and achieved reasonable results for saliency detection.

In our case, the space of M is generalized to be $\mathbb{R}$ for P_c^i and P_s^i, and the definition of $\mathcal{W}_2$ distance can be written as:

$$\mathcal{W}_2(P_c^i, P_s^i) = \left[\inf_{\gamma \in \Gamma(P_c^i, P_s^i)} \int_{\mathbb{R}\times\mathbb{R}} |x-y|^2 \, \mathrm{d}\, \gamma(x,y) \right]^{\frac{1}{2}}, \tag{4.5}$$

where $\Gamma(P_c^i, P_s^i)$ indicates the set of all couplings of P_c^i and P_s^i.

It would be intractable to compute the integral in Equation 4.5 in case of arbitrary distributions. However, it can be solved analytically for the Gaussian distribution [Givens and Shortt, 1984]. The $\mathcal{W}_2$ distance between one central cell distribution $P_c^i \sim N(\mu_c^i, \Sigma_c^i)$ and its neighboring cell distribution $P_s^i \sim N(\mu_s^i, \Sigma_s^i)$ along channel i indeed amounts to:

$$\mathcal{W}_2(P_c^i, P_s^i) = \left[\|\mu_c^i - \mu_s^i\|_2^2 + \Sigma_c^i + \Sigma_s^i - 2\sqrt{\Sigma_c^i \Sigma_s^i} \right]^{\frac{1}{2}}. \tag{4.6}$$

L^2 distance

The descriptors of the central and surrounding cells, denoted as (μ_c^i, Σ_c^i) and (μ_s^i, Σ_s^i) respectively, can be treated as two points in a 2D space. Then the distance between two distributions can be simplified to the L^2 distance between the two points:

$$D_{L^2}(P_c^i, P_s^i) = \sqrt{(\mu_c^i - \mu_s^i)^2 + (\Sigma_c^i - \Sigma_s^i)^2}. \tag{4.7}$$

Signed Gradient matrix (SGrd)

Here we do not treat each descriptor as a whole, but consider the mean and variance values separately. We propose a new measurement namely Signed Gradient matrix (SGrd), which computes the signed gradient for the mean and variance values individually, and forms a contrast vector by simply concatenating these two gradient values.

The contrast vector between one central cell distribution $P_c^i \sim N(\mu_c^i, \Sigma_c^i)$ and its neighboring cell distribution $P_s^i \sim N(\mu_s^i, \Sigma_s^i)$ along channel i can then be expressed as follows:

$$\overrightarrow{SGrd}(P_c^i, P_s^i) = \left[\mu_c^i - \mu_s^i, \Sigma_c^i - \Sigma_s^i\right]. \tag{4.8}$$

In the feature space, the contrast vector in Equation 4.8 is treated in terms of two separate values, enabling a more convenient training procedure.

4.4.2 Measurements for Histograms

We consider three different distance measurements which have been commonly used for histograms. In the following, the histograms for a central and a surrounding cell along channel i are denoted as H_c^i and H_s^i. We compare the results of the three measurements in Section 4.6, ''Experiments''.

Kullback-Leibler divergence

Kullback-Leibler Divergence (KLD) [Kullback and Leibler, 1951] is a similarity measurement between two probability distributions P and Q that indicates the information loss when Q is used to approximate P.

Using arguments from information theory, one can represent the Kullback-Leibler divergence of Q from P as follows:

$$D_{KL}(P||Q) = \int_{-\infty}^{\infty} p(x) \ln \frac{p(x)}{q(x)} dx, \tag{4.9}$$

where $p(x)$ and $q(x)$ denote the probability densities of P and Q. The more P differs from Q, the higher the KL divergence.

Here we can also see KL divergence is a non-symmetric measurement, which means the KL divergence from P to Q is generally not the same as that from Q to P.

To specify, the KL divergence from H_c^i to H_s^i can be calculated using the following formula:

$$D_{KL}(H_c^i||H_s^i) = \sum_{k=1}^{b} \ln\left(\frac{h_c^i(k)}{h_s^i(k)}\right) h_c^i(k). \tag{4.10}$$

Hellinger distance

Let P and Q be two continuous probability distributions with respect to a parameter λ; the Hellinger distance is a measure that represents the difference between them. The square of the Hellinger distance has a particularly simple form and is defined as [Hellinger, 1909]:

$$H^2(P, Q) = \frac{1}{2} \int \left(\sqrt{\frac{dP}{d\lambda}} - \sqrt{\frac{dQ}{d\lambda}} \right)^2 d\lambda. \tag{4.11}$$

Note that this definition does not depend on parameter λ, in other words, the Hellinger distance between P and Q does not change even if λ is replaced with a different probability measure with respect to which both P and Q are absolutely continuous.

For two discrete probability distributions H_c^i and H_s^i that represent P_c^i and P_s^i, the Hellinger distance is then computed as the contrast between P_c^i and P_s^i:

$$H^2(H_c^i, H_s^i) = \frac{1}{\sqrt{2}} \sqrt{\sum_{k=1}^{b} \left(\sqrt{h_c^i(k)} - \sqrt{h_s^i(k)} \right)^2}. \tag{4.12}$$

Scales	4-6	4-6-8	4-6-8-10
Feature size	$20,320D(\overrightarrow{cst})$	$23,440D(\overrightarrow{cst})$	$25,040D(\overrightarrow{cst})$

Table 4.1: *Illustration of feature size under different configurations. All the contrast measurements considered in this chapter are one dimensional, except SGrd, which is two dimensional.*

Histogram intersection

Histogram intersection (HI) is another popular similarity measure for histograms. Given two histograms H_p and H_q with b bins, it is defined as:

$$\mathrm{HI}(H_p, H_q) = \frac{\sum_{k=1}^{b} \min(H_p(k), H_q(k))}{\sum_{k=1}^{b} H_p(k)}. \tag{4.13}$$

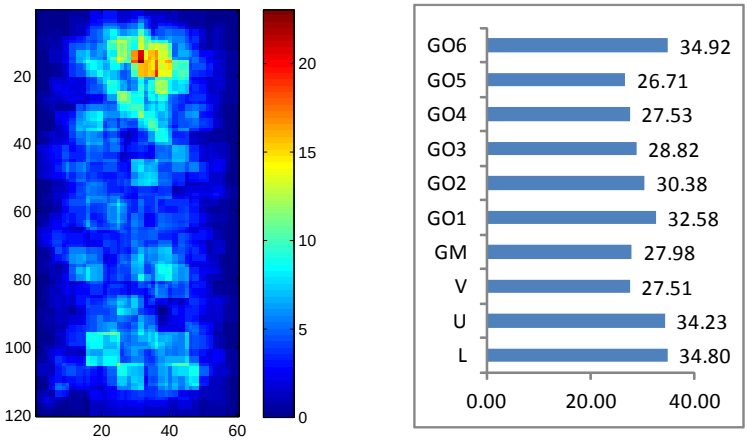
Since the histograms considered in this chapter are all normalized so that they sum up to 1, the histogram intersection between H_c^i and H_s^i can be further simplified to:

$$\mathrm{HI}(H_c^i, H_s^i) = \sum_{k=1}^{b} \min(h_c^i(k), h_s^i(k)). \tag{4.14}$$

4.5 Classification

In this section, we discuss our classification procedure for our center-surround contrast features introduced above. First of all, we discuss the size of our feature vector. Given a pedestrian model of 60×120 pixels, considering all the four patterns proposed in Section 4.2.2, ''Center-surround Neighborhood Patterns'', we compare feature sizes under different settings in terms of scale structures, and dimensions of contrast vectors denoted as $D(\overrightarrow{cst})$ in Table 4.1. Apparently, the feature size grows as the number of scales increases. Among all the contrast measurements considered in this chapter, only the signed gradient matrix is two dimensional, while the others are all one dimensional.

Similar to Section 3.2, ''Classification'', we also apply a fast version of AdaBoost [Appel et al., 2013] since it offers a convenient and fast approach to feature selection from a large number of candidate features. We also use decision trees of depth 2 as our weak classifiers since they are efficient to learn, and a multi-round training strategy, which shows to lead



(a) (b)

Figure 4.6: *Illustration of representative center-surround features. (a) Body parts weight map: different colors are used to indicate the accumulative weight of each pixel after boosting. (b) Channel weight bars: accumulative weight of each channel is indicated by one bar.*

to better performance than a simple one round training procedure with the same number of negative samples. But we choose the number of weak classifiers to be 4096 in this chapter, as we observe that smaller numbers cause decrease in performance, while larger numbers do not lead to further gains in performance from experiments.

After boosting, each selected feature is assigned a single weight, indicating its amount of contribution to the final response. In order to observe the locations of the most representative features, we plot an accumulative weight map of the top 1000 features with highest weights from the final strong classifier, as shown in Figure 4.6(a), where different colors indicate different weights. This map is generated by simply adding the weight of each selected feature to the pixels it covers. From the weight map, we can see that accumulative weights for the head-shoulder area are much higher than other body parts, which means this area is more discriminative for pedestrians. This conclusion accords with the biological characteristic of human bodies' special shape structure at head-shoulder.

Moreover, we also observe which channels are more representative by adding the weight of each feature to its channels. As shown in Figure 4.6(b), we use bars to illustrate the accumulative weight of each channel. We find that all the channels we choose contribute rather evenly to the final classifier, indicating no channel redundancy in our approach.

4.6 Experiments

In this section, we show experimental results for different feature schemes, and select the optimal setting after comprehensive comparisons. Finally, we compare our optimal detector with other state-of-the-art detectors.

4.6.1 Comparisons for Different Feature Settings

According to Section 4.2, ''Overview of Feature Extraction'', there are several configurations can be adjusted: statistical descriptors, contrast measurements, scale structures, and numbers of histogram bins where histograms are used. Here, we make comprehensive comparisons on the INRIA data set so as to seek the strongest feature scheme.

First, we define a default setting: three scales of 4×4, 6×6, and 8×8 pixels; 5 histogram bins when histograms are used. This setting is utilized in the following experiments unless otherwise specified.

C_1S_8 pattern vs. C_1S_1 pattern

We propose the C_1S_8 pattern in Section 4.2, ''Overview of Feature Extraction'', in order to incorporate more information regarding local image differences. Here, we show experimental results of both patterns to support the argument that the C_1S_8 pattern is superior. From Figure 4.7, it appears that the C_1S_8 pattern produces significantly better results than C_1S_1 over all descriptor-measurement combinations.

Contrast measurements

We compare the contrast measurements proposed in Section 4.4, ''Contrast Measurements'' for two descriptors respectively. From Figure 4.8, we see that the results of using different contrast measurements do not show big differences for both descriptors. Despite of their stable performance, we observe the best measurements for Gaussian and histogram descriptors are $\mathcal{W}_2$ distance and Hellinger distance, respectively. Therefore, we select Gaussian-$\mathcal{W}_2$and Hist-Hellinger as the two preferable combinations.

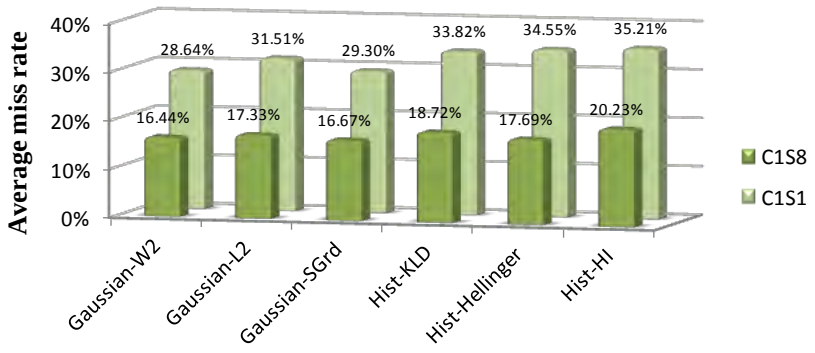
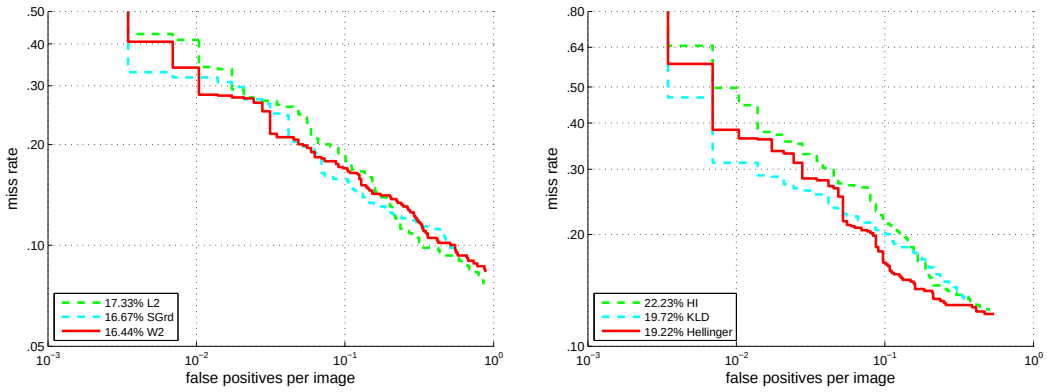


Figure 4.7: *Comparison of two center-surround patterns.*



(a) (b)

Figure 4.8: *Experiments on different contrast measurements for two cell descriptors: (a) Gaussian distributions; (b) Histograms.*

Number of histogram bins

As discussed in Section 4.3.2, ‘‘Histograms’’, the number of bins is an important parameter in practical applications of histograms. Figure 4.9 shows experimental results when using 5, 10, 15 and 20 bins and the Hellinger distance which has been shown to be the best among the three contrast measurements considered for histograms. When we increase the number of histogram bins from 5 to 15, we obtain better results as expected, especially a significant gain from 10 to 15. This is because more histogram bins integrate finer information of the local

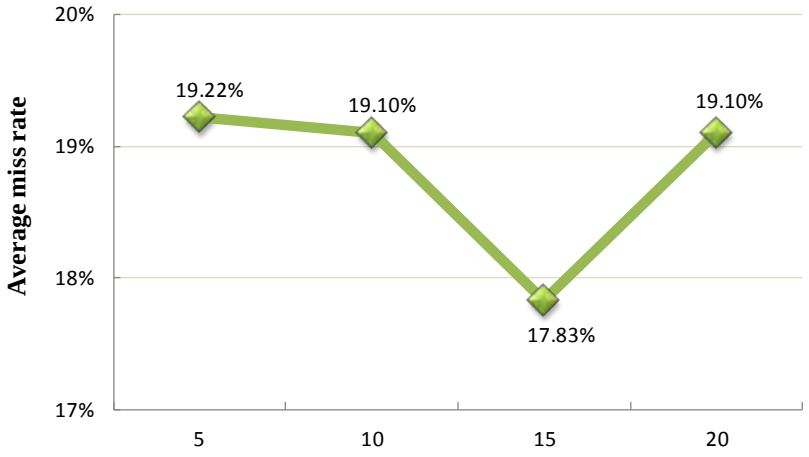


Figure 4.9: *Experiments on different histogram bins using the Hellinger distance.*

image region, thus leading to better performance. However, performance begins to decrease when we consider more than 15 bins. We attribute this phenomenon to the low degree of tolerance to noisy real world data. The curve in Figure 4.9 indicates that the number of bins should be chosen to be neither too small nor too big.

Note that, in the following experiments, we thus use 15-bin histograms instead of the default 5-bin histograms.

Descriptors

After choosing the parameter for histograms, we obtain two optimal combinations: Gaussian-$\mathcal{W}_2$and Hist(15 bins)-Hellinger. Now we can make a comparison of descriptors. From Figure 4.10, we can see that both optimal combinations outperform the baseline detector [ChnFtrs] consistently, which illustrates the effectiveness of our new features. Moreover, Gaussian-$\mathcal{W}_2$obtains better results than Hist(15 bins)-Hellinger and then is selected as the optimal descriptor-measurement combination in our approach.

Scale structures

Most computational visual attention systems compute local difference at multiple scales, so as to incorporate richer information, thus leading to a better performance. In this chapter, different scales are indicated using different cell sizes. Generally, the smaller the cell size, the finer the local difference, and vice versa. We implement experiments under

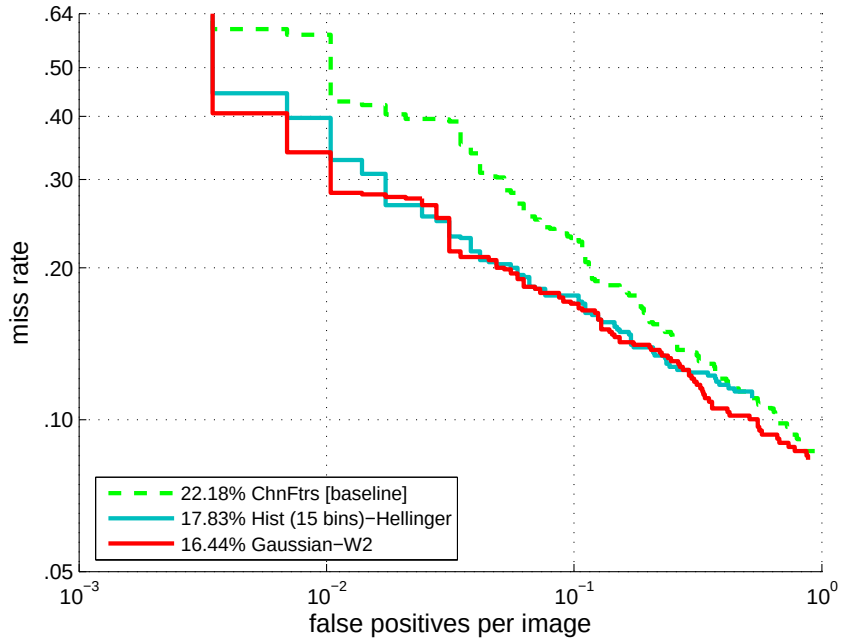


Figure 4.10: *Comparison of two optimal descriptor-measurement combinations and the baseline detector [ChnFtrs].*

three different scale structures: 4-6; 4-6-8; and 4-6-8-10, and show their comparisons in Figure 4.11. Increasing the scales from 4-6 to 4-6-8 brings about a significant improvement of approximately 5% with respect to average miss rates; on the other hand, continuing to increase scales to 4-6-8-10 produces a less prominent performance gain of less than 1% with respect to average miss rates. This implies that further increasing scales will not bring about significant improvements. Therefore, we choose the scale structure of 4-6-8-10 as our optimal choice.

To summarize, the optimal feature setting is to use the descriptor-measurement combination of Gaussian-$\mathcal{W}_2$ and scale structure of 4-6-8-10. We use this configuration in the following experiments.

4.6.2 Computational complexity

We investigate the computational complexity of different feature settings. Our normal-distribution as well as histogram based features are computed from local averages of certain values. Such local features can be computed in $O(n)$ time with n denoting the number of image pixels using *moving averages* or *integral image* techniques. They only differ in the

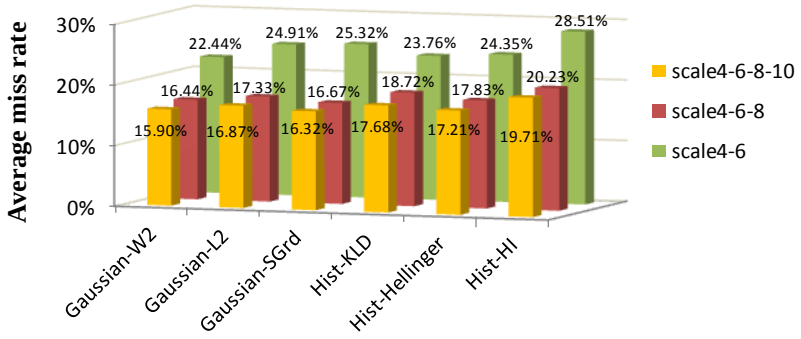


Figure 4.11: *Comparison of three scale structures. In this experiment, 15-bin histograms are used.*

number of layers needed (one for each distribution parameter or bin) which amounts to a constant factor. Looking into details of the diverse distance functions implemented for different feature settings, we can see the very same effect: the time complexity is constant per pixel (linear growing with image size), so the overall complexity for each setting is still $O(n)$. We have to note that the constant factor for normal-distributions is 2 per input channel, while histograms require $b \geq 2$ (*e. g.* 15) number of histogram bins.

The computational complexity of our baseline detector [ChnFtrs] is also $O(n)$, because each pixel is visited once per channel for computing local sums. Therefore, the computational complexity of our features at different settings is on par with [ChnFtrs].

4.6.3 Comparisons with State-of-the-art Detectors

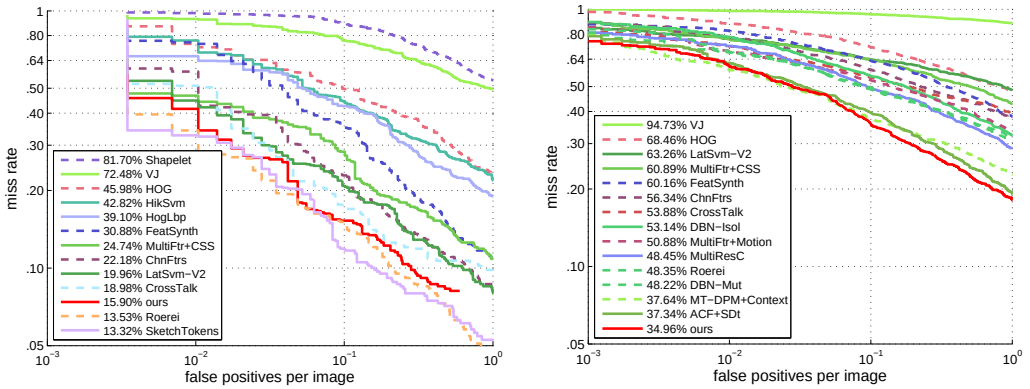
In order to evaluate our approach, in this section, we further make comparisons with other state-of-the-art detectors whose results are publicly available[1] using the evaluation protocols explained in Section 2.3, ‘‘Experiment Settings’’.

We choose the detector [ChnFtrs] as our baseline detector because it also considers multiple channels, but uses the channel values directly as feature values. In contrast, we interpret local difference by emulating human visual systems. From Figure 4.12, we see that our detector outperforms [ChnFtrs] on the INRIA and Caltech pedestrian data set by 6% and 15% with respect to average miss rates, respectively. These significant improvements indicate that our modifications to [ChnFtrs] are effective. Moreover, comparing with other state-of-the-art

[1] `http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/`

Detector	Average miss rate	
	INRIA	Caltech
VJ[Viola and Jones, 2004]	72.48%	94.73%
HOG[Dalal and Triggs, 2005]	45.98%	68.46%
Shapelet[Sabzmeydani and Mori, 2007]	81.70%	91.37%
MultiFtr [Wojek and Schiele, 2008]	36.50%	68.62%
MultiFtr+CSS [Walk et al., 2010]	24.74%	60.89%
MultiFtr+Motion [Walk et al., 2010]	/	50.88%
HikSvm [Maji et al., 2008]	42.82%	73.39%
HogLbp [Wang and Han, 2009]	39.10%	67.77%
LatSvm-V1 [Felzenszwalb et al., 2008]	43.83%	79.78%
LatSvm-V2 [Felzenszwalb et al., 2010]	19.96%	63.26%
ChnFtrs [Dollár et al., 2009a]	22.18%	56.34%
FeatSynth [Bar-Hillel et al., 2010]	30.88%	60.16%
MultiResC [Park et al., 2010]	/	48.45%
CrossTalk [Dollár et al., 2012]	18.98%	53.88%
VeryFast [Benenson et al., 2012]	15.96%	/
SketchTokens [Lim et al., 2013]	13.32%★	/
Roerei [Benenson et al., 2013]	13.53%★	48.35%
AFS+Geo [Levi et al., 2013]	/	66.76%
MT-DPM+Context [Yan et al., 2013]	/	37.64%★
DBN-Isol [Ouyang and Wang, 2012]	/	53.14%
DBN-Mut [Ouyang et al., 2013]	/	48.22%
ACF+SDt [Park et al., 2013]	/	37.34%★
ours	15.90%★	34.96%★

Table 4.2: *Performance comparisons to state-of-the-art pedestrian detectors. Each row in this table displays the corresponding average performance in terms of average miss rates. The approach proposed in this chapter yields state-of-the-art performance on the INRIA data set and consistently better results than previously reported methods on the Caltech data set. We indicate the top three detectors for each data set using a symbol of ★.*



(a) (b)

Figure 4.12: *Overall results of different detectors on the (a) INRIA and (b) Caltech data sets, under standard evaluation settings.*

detectors, our detector reaches state-of-the-art performance on the INRIA data set and even yields the overall best performance on the Caltech date set. More extensive comparisons are shown in Table 4.2.

4.7 Summary

In this chapter, we proposed to mimic early human visual processing by designing local center-surround contrast features and boosting them to respond to the appearance of pedestrians. In this way, our pedestrian detector realized a computational top-down saliency system. Seeking the strongest local contrast scheme, we evaluated two different cell descriptors and three contrast measurements for each accordingly. Moreover, we tested four different neighborhood patterns and three scale structures to optimize our feature extraction. Our features are very efficient to compute by means of combining a fast integral method for local averaging and a clever arrangement of additional image layers for fast maximum likelihood estimation of parameters of normal distributions.

We implemented extensive experiments on two standard benchmarks: the INRIA and Caltecch pedestrian data sets. Our detector achieved state-of-the-art performance on the INRIA data set, and outperformed all the other previously proposed detectors on the Caltech data set.

Future Work. Given these results, it appears promising to further explore feature design driven by human visual mechanisms. Immediate extensions of the approach presented in this

chapter consist of incorporating information from additional channels, such as motion and depth. Furthermore, it is feasible to apply the proposed techniques for more general object detection.

Chapter 5

Fast Moving Pedestrian Detection Based on Motion Analysis

In the previous chapters, we investigated and proposed methods only employing spatial information, for example, colors and gradients, for effective pedestrian detection. In this chapter, we concentrate on exploiting motion information as supplementary to spatial information, and focus on moving pedestrians.

We have several reasons to focus on moving pedestrian detection in this chapter. First of all, moving pedestrians are more interesting for some applications, for example, ADASs, because they are more probable to cause collision than those who are just standing along the street. Second, for the task of detection, moving pedestrians are actually more challenging, due to the significant variations in appearance caused by movements. In addition, through observing a large number of motion maps of moving and static pedestrians, we find that moving pedestrians show to be more distinguishable against the background even when they are far away from the camera; while static pedestrians are easily submerged in the background due to no relative motion.

To further prove the significance of moving pedestrian detection, we find an early attempt on this topic [Curio et al., 2000], which was restricted to detecting pedestrians walking across the street. By contrast, we clarify that the definition of moving pedestrians in this chapter includes those who walk across and also along the street.

In our approach, we use an optical flow estimation to represent motion information, that is to say, our motion analysis is implemented on optical flow maps. To specify, motion information is used in the following two ways:

This work has been partially published in [Zhang et al., 2013].

(a) (b) (c)

Figure 5.1: *An example of using motion segmentation for ROIs selection when the pedestrian is of small scale and low contrast. (a) Original frame. The red bounding box indicates the ground truth annotation. (b) Motion segmentation results. (c) Final detection results. The yellow bounding box denotes the final detection results, and the number above the box indicates the detection score output from our classification method.*

1. To select *regions of interest* (ROIs). It is very time consuming for sliding window detection methods to execute exhaustive search over the whole image. To avoid it, we propose to select interesting blobs, which probably contain moving pedestrians, via a motion segmentation procedure.
2. To design novel motion based features. Observing optical flow maps for moving pedestrians and other objects, for example, buildings, vehicles, bicycles, we find that moving pedestrians exhibit rather distinct motion patterns. We make use of this characteristic and design *motion self difference* features.

An obvious disadvantage of sliding window detection approaches is to analyze a large number of windows at different locations and of different scales all over the image. Therefore, it is of remarkable importance to extract ROIs as a pre-step [Geronimo et al., 2010], for the purpose of discarding those regions, unlikely to consist of pedestrians, such as the uniform sky and ground plane [Leibe et al., 2007], thus resulting in a significant reduction of the number of candidate detection windows that have to be examined by the classifier. From the literature, many efforts have been made for ROIs selection. [Itti et al., 1998] solved the problem from a perspective of biologically inspired attentional system. They computed a saliency map based on color, intensity, and gradient orientation of pixels and then selected those regions with higher saliency values as candidate regions. Some researchers chose to use weak constraints on symmetry and pedestrian size to select ROIs [Broggi et al., 2003] [Bertozzi et al., 2003] [Bertozzi et al., 2004]. [Gualdi et al., 2010] proposed to estimate likelihood of belonging to the ROIs for each pixel based on Monte Carlo sampling. Besides spatial cues, motion information also plays an important part in this task and can be integrated with spatial or even stereo information to obtain better performance. [Franke and Heinrich, 2002] proposed to merge motion analysis and stereo processing to select candidate regions containing moving objects. [Elzein et al., 2003] and [Lim and Kim, 2013] selected moving object regions by analyzing the temporal difference maps computed between successive

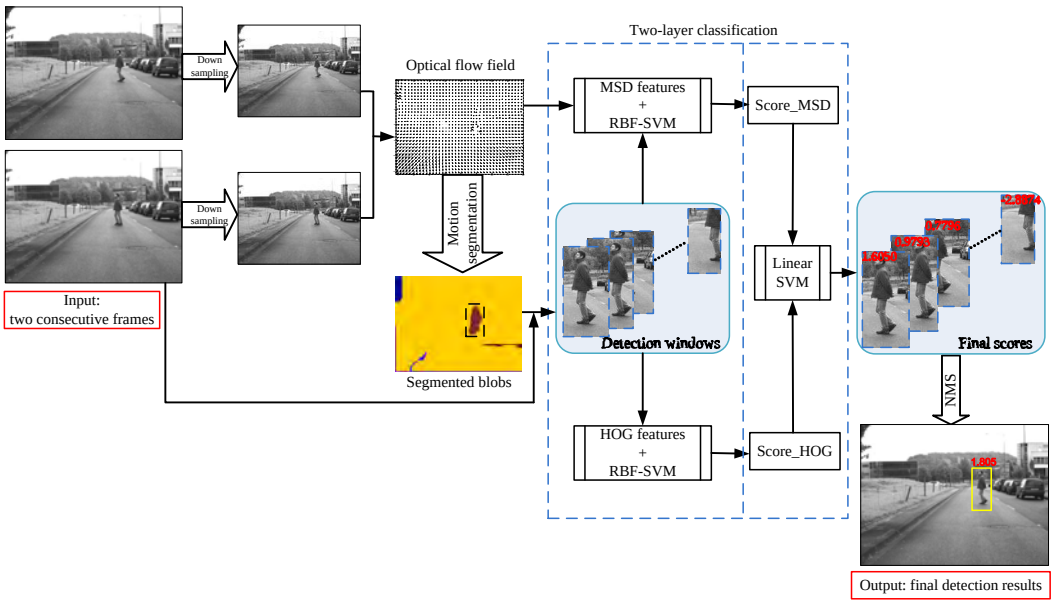


Figure 5.2: *The flow chart of motion based moving pedestrian detection. See more details in Section 5.1, ''Overview on Our Approach''.*

frames. [Enzweiler et al., 2008] proposed motion parallax features and applied Bayes' rule to estimate the posterior probability for the presence of pedestrians in a certain image region. [Kamijo et al., 2010] applied a spatio-temporal Markov Random Field (MRF) model [Kamijo et al., 2000] for foreground objects extraction from background.

All the above approaches employ very complex models and are too time consuming, alternatively, we propose a simple method which still obtains reliable results.

5.1 Overview on Our Approach

We show the flow chart of our approach in Figure 5.2. The whole procedure can be divided into four main parts:

Motion segmentation for ROIs selection. In this part, ROIs are defined as those regions which probably contain moving objects, including walking pedestrians, driving cars and so on. In order to select ROIs, we apply an efficient graph-based motion segmentation algorithm as a pre-processing on the optical flow field, which is computed from two down-sampled, consecutive input frames. After segmentation, the whole image is divided into several blobs.

Only those blobs which satisfy some constraints based on prior knowledge are selected as interesting blobs.

The motivation of implementing segmentation on motion fields comes from the observation of a large number of optical flow maps, computed from video frames captured by a moving camera mounted on a driving vehicle. We find that the optical flow fields of moving objects are usually rather distinguishable from other static objects, including trees and buildings from the background. The differences between moving and static objects appear in terms of optical flow magnitude as well as orientation. We also notice that motion field sensitively responds to different objects in some demanding scenes, where small scale ($\sim$ 50 pixels in image height) and low contrast pedestrians appear, see Figure 5.1 for an example.

Detection window generation. Detection windows are generated for each interesting blob using a height-prior principle. In traditional sliding window detection methods, windows are generated under multiple scales at a single location. However, this strategy is no more necessary in our approach. For each segmented blob, its upper and lower boundaries usually fits the head and feet area for each person. By contrast, the left and right boundaries do not always respond to the sides of human bodies. An illustration is shown in Figure 5.4, where we can see wide segments consisting of several pedestrians walking side by side, while it is very unlikely to see one pedestrian appears on top of another. Therefore, the height of each blob is an useful clue for determining scales. Next, we define a line of detection windows based on the height-prior principle for each blob. We slide windows along the width direction and generate one window at each position, whose height is determined by the upper and lower boundaries and width is computed given a fixed aspect ratio.

Feature computation. We extract two kinds of features: HOG features [Dalal and Triggs, 2005] and *Motion Self Difference* (MSD) features, for each detection window. HOG features have been widely used as effective spatial features for pedestrian detection. MSD features are novel motion features proposed for moving pedestrians in this chapter. We show exemplary flow magnitude maps for various objects in Figure 5.5. One can easily notice that all the static objects and non-articular moving objects, for example, driving cars, exhibit flow magnitude maps reflecting their fixed outer shapes; by contrast, moving pedestrians show non-uniform structures due to inter-body motion. We find that this characteristic motion pattern of walking pedestrians is a very important property and can be used to design motion based features. The basic idea of MSD features is to compute the local difference between neighboring rectangular regions all over the human body with respect to optical flow vectors.

Classification. To accomplish the classification task based on two different kinds of features, a two-layer scheme is introduced. On the first layer, one Support Vector Machine with

Radial Basis Function kernel (RBF-SVM) is trained for each kind of features individually. On the second layer, a linear SVM is trained on two primary scores obtained from two RBF-SVMs on the first layer. The final decision score for each detection window is the output from the linear SVM. After classification, the final detection results for each image are obtained by a *non-maximum suppression* (NMS) approach [Felzenszwalb et al., 2008] to suppress less confident windows nearby.

5.2 Graph-based Motion Segmentation for Detection Window Generation

In this section, we describe how we apply motion segmentation as a pre-processing step to select ROIs, and how we generate detection windows based on the segmented blobs.

Although we are only interested in those regions where moving objects appear, sometimes static objects may also be included in selected segments. Typically, the magnitudes of optical flow vectors are related to the distance between camera and subjects. Let us define a parameter, describing the relative distance between objects and the background:

$$R_z = \frac{\text{distance}(bg, cam)}{\text{distance}(object, cam)} \,. \tag{5.1}$$

When the value of R_z satisfies the following constraint:

$$R_z > \xi, \tag{5.2}$$

static objects are also distinguishable from the background. Since optical flow magnitudes are inversely proportional to the scale of distance, objects at different distances show different magnitude values on the optical flow map. In this way, in order to observe minor difference of distance, we need an accurate flow map. Intuitively, the more accurate optical flow is, the smaller the value of ξ can be chosen.

Therefore, although we focus on moving pedestrian detection in this chapter, we demonstrate that our approach for ROIs selection can be extended for static pedestrian detection given an accurate optical flow estimation.

The possibility for extensions raises an additional problem: what if some static pedestrians close to the camera are also selected as ROIs but we are only interested in moving pedestrians? This problem can be solved by employing MSD features proposed in Section 5.3, ‘‘Motion Self Difference Features’’, which are specifically designed for moving pedestrians. In this way, static pedestrians will be automatically discarded during the classification procedure, since they produce relatively low confidence scores.

Table 5.1: *Effects of down-sampling the input frames during optical flow estimation on performance.*

Down-sampling ratio	1.0	1/4	1/8	1/16	1/32	1/64
Average miss rate	38%	40%	42%	43%	68%	90%

5.2.1 Optical Flow Estimation

Prior to motion segmentation, we first estimate the optical flow fields. There are many algorithms yielding dense optical flow fields, see comparisons at `http://vision.middlebury.edu/flow/eval/results/results-e1.php`. The most classic one was proposed by [Horn and Schunck., 1981], which heavily relied on assumptions of brightness constancy and spatial smoothness, and solved a quadratic formulation. Many spatially-discrete formulations derived approaches [Brox et al., 2004, Lempitsky et al., 2008] have been proposed subsequently. This kind of methods are not reliable when too many outliers are involved. Another kind of methods are formulated with an L_1 robust penalty and are often coupled with specialized total variation (TV) optimization methods [Zach et al., 2007].

Among recently proposed methods, most of them require color images as input, and some are too slow for real-time applications. We chose a robust algorithm proposed by [Liu, 2009], which produces a layered flow field from two consecutive gray scale frames I_t and I_{t+1}, at a tolerable speed.

Because optical flow computation is the most time consuming part over the whole approach, we decide to implement it on downsampled frames instead of on the full resolution frames. The input frames are resized to 1/16, preserving the original aspect ratio. The effects of down-sampling on performance is shown in Table 5.1, where we can see that the performance decreases moderately when the down-sampling ratio is above 1/16, but begins to drop significantly from 1/32. Thus, we choose the down-sampling ratio to be 1/16 in our approach, which only causes a performance decrease of 5% in terms of average miss rate, but reduces the consuming time by an order of magnitude.

5.2.2 Graph-based Motion Segmentation

The computed optical flow field is denoted as a flow image $I_f(t)$ in the following procedure. $I_f(t)$ is considered as a two-channel image, and its two channels f_x and f_y denote flow vector elements along horizontal and vertical directions, respectively. In order to reduce noise from optical flow estimation, which may cause too many small segments, it is necessary to implement a smoothing before segmentation. We employ a gentle Gaussian smoothing

($\sigma = 0.8$) on $I_f(t)$, so as not to largely inhibit the local difference, which is essential for the following segmentation.

Next, we employ an efficient graph based algorithm proposed by [Felzenszwalb and Huttenlocher, 2004] for segmentation on the flow image $I_f(t)$. This segmentation algorithm is firstly ported to work on flow image by us. In the following, we describe the implementation procedure in more detail.

First, we represent each pixel p_i with a 4D vector:

$$\boldsymbol{v}(i) = (x(i), y(i), f_x(i), f_y(i))^{\mathsf{T}} , \tag{5.3}$$

where $x(i)$ and $y(i)$ are the coordinates of p_i in flow image $I_f(t)$; $f_x(i)$ and $f_y(i)$ are its two flow elements. The segmentation is not implemented directly on the flow image, but on a 4D feature space by mapping all the pixels using Equation 5.3.

Second, in this 4D space, each feature vector is treated as a vertex. A graph $G = (V, E)$ is constructed by connecting each vertex to its 8 nearest neighbors, in the 4D feature space. Let V denote the set of vertices:

$$V = \{\boldsymbol{v}_i | 1 \leq i \leq n_v\}, \tag{5.4}$$

where n_v indicates the total number of vertices in the 4D space, equaling to the number of pixels in the input image.

Let E denote the set of edges between two vertices:

$$E = \{\boldsymbol{e}_{i,j} | 1 \leq i \leq n_v, 1 \leq j \leq n_v, i \neq j\}. \tag{5.5}$$

The weight of edge $e_{i,j}$ is represented by the L_2 (Euclidean) distance between the two corresponding vertices in the 4D feature space:

$$\mathrm{w}(\boldsymbol{e}_{i,j}) = \sqrt{\sum_{k=1}^{4} (\boldsymbol{v}_i^k - \boldsymbol{v}_j^k)^2}. \tag{5.6}$$

After that, an iteration is run on the edge set E, to combine elements of low weight values $\mathrm{w}(\boldsymbol{e}_{i,j})$, so that those vertices of high similarities are included in one segment.

Finally, the iteration stops when a segmentation S which satisfies

$$\mathrm{w}(\boldsymbol{e}_{i,j}) > \mathrm{MInt}(C_p, C_q), \quad \boldsymbol{e}_{i,j} \in E, \quad i \neq j \tag{5.7}$$

has been found. Here, C_p and C_q indicate the components which v_i and v_j belong to, and MInt is a function of the minimum internal difference, defined as:

$$\begin{aligned} \mathrm{MInt}(C_p, C_q) = \min(&\max_{i,j \in C_p} \mathrm{w}(\boldsymbol{e}_{i,j}) + \tau(C_p), \\ &\max_{i,j \in C_q} \mathrm{w}(\boldsymbol{e}_{i,j}) + \tau(C_q)). \end{aligned} \tag{5.8}$$

The threshold function τ affects the sizes of segmented components.

5.2.3 Analysis of Interesting Blobs

After segmentation, we obtain n segments, and each of them may contain different objects, for example, the ground plane, the sky, buildings, cars, or pedestrians. In order to avoid examining a large number of windows belonging to the background, we discard some blobs that are not likely to contain pedestrians and only reserve those blobs probably containing moving pedestrians.

By observing the segmented blobs, we find that typically there is at least one large blob, which has a huge span from left to right. Such kind of blobs usually belong to the ground plane or the sky and should be considered as outliers. Besides, those blobs whose sizes are beyond the lower limit of pedestrian size we want to detect (cf. Section 2.3, ''Experiment Settings'') should also be discarded. Thus, the weak constraints on the size of blob i, comprising a set of $s(i)$ unique pixels

$$\mathrm{S}_i = \{(x_1, y_1), (x_2, y_2), ..., (x_{s(i)}, y_{s(i)})\}_i, \tag{5.9}$$

are defined as follows:

$$w_{\min} \leq \max_{a,b \in [1,s(i)]} \{|x_a - x_b|\}_i \leq w_{\max}, \tag{5.10}$$

$$h_{\min} \leq \max_{a,b \in [1,s(i)]} \{|y_a - y_b|\}_i \leq h_{\max}. \tag{5.11}$$

Another constraint is on the locations of blobs. Assuming pedestrians are always standing on the ground, the lower boundary of each interesting blob should go below the vanishing point. In the theory of graphical perspective, a vanishing point is a point in the picture plane π that is the intersection of all the horizontal parallel lines in 3D space onto the picture plane. In urban traffic scenes, all the parallel lines from the ground plane intersect at the vanishing point, which means, the ground plane is always below the vanishing point. In this way, the lower boundary, consisting of the human feet, should go below the vanishing point on the image plane:

$$\max_{k \in [1,s(i)]} \{y_k\}_i \geq \xi y_v, \tag{5.12}$$

Algorithm 2 Generating detection windows from selected interesting blobs.

initialize the list of detection windows: $T_{\text{wnd}} \leftarrow \emptyset$;
for each interesting blob $i \in [1, m]$ **do**
  set $x_{\min} \leftarrow \min_{k \in [1, s(i)]} \{x_k\}$ and $x_{\max} \leftarrow \max_{k \in [1, s(i)]} \{x_k\}$;
  for $u = x_{\min}$ to $x_{\max}$ **do**
    $j \leftarrow u - x_{\min} + 1$;
    upper boundary: $y_t \leftarrow \min_{x_k = u, k \in [1, s(i)]} \{y_k\}$;
    lower boundary: $y_b \leftarrow \max_{x_k = u, k \in [1, s(i)]} \{y_k\}$;
    height: $h \leftarrow |y_b - y_t|(1 + 2m_b)$;
    width: $w \leftarrow h \cdot r_{\text{wh}}$;
    center point: $(x_c, y_c) \leftarrow (u, \frac{y_t + y_b}{2})$;
    append (h, w, x_c, y_c) to T_{wnd};
  end for
end for
return T_{wnd}

where y_v indicates the vertical coordinate of the vanishing point, which can be computed given camera parameters; $\xi (\xi < 1)$ is a tolerance parameter.

Those blobs which satisfy both the size constraints in Equation 5.11 and an additional location constraint in Equation 5.12, are selected as interesting blobs and undergo further examinations for moving pedestrian detection.

5.2.4 Detection Window Generation

In this subsection, our task is to generate detection windows from each selected interesting blob. Need to mention, we only obtain the window coordinates from blobs in a flow map $I_f(t)$, but crop the contents from the original image frame I_t. Since we downsample the input image before optical flow computation in Section 5.2.2, ''Graph-based Motion Segmentation'', coordinate transformation is necessary to reverse this effect.

We propose a height-prior strategy to define the size of detection windows at different positions of each blob. From left to right, at each x-coordinate, the height of the detection window is determined by the upper and lower boundaries of the blob at the according position; the width is computed using a constant aspect ratio to this height value. More details about this procedure is illustrated in Algorithm 2, and two examples of detection windows generated at different x-coordinates are shown in Figure 5.3.

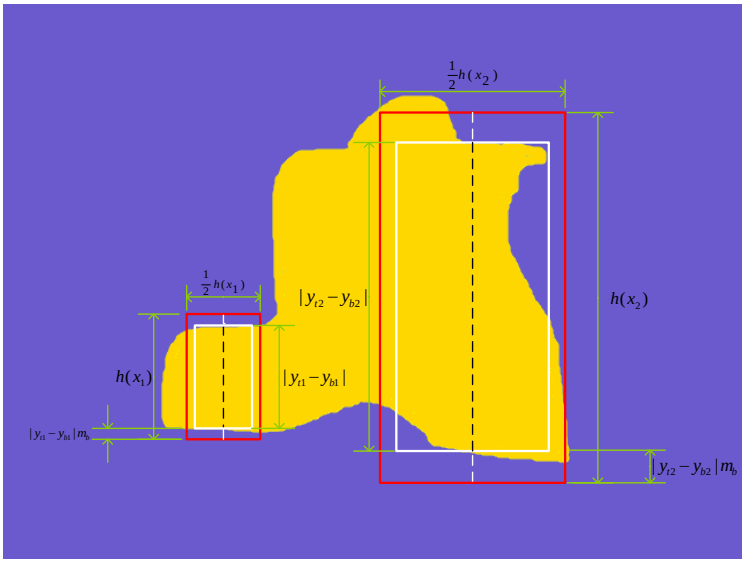
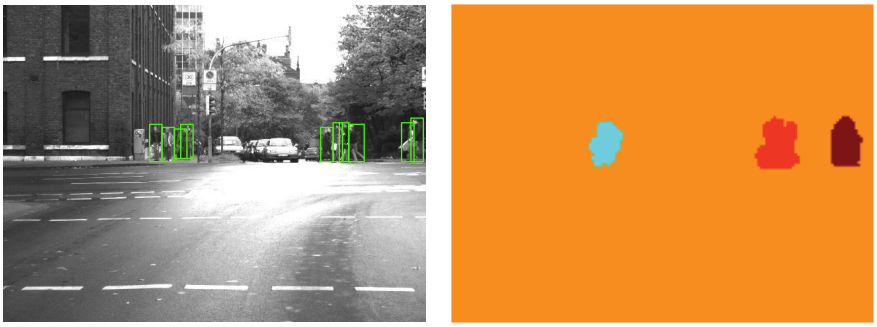


Figure 5.3: *Generation of detection windows from a blob: two examples are drawn at different horizontal positions. The black dashed lines link the upper and lower boundaries at different x-coordinates, and we obtain the detection windows (red) by adding borders around white candidate rectangles.*

The motivation of this height-prior strategy comes from our observation on the segmented blobs. The upper and lower boundaries of the blobs usually correspond to the head-to-sky and feet-to-ground areas; by contrast, the left and right boundaries are not guaranteed to correspond to the arms-to-background areas. The reason is that it is very often to see pedestrians walking shoulder by shoulder, but rarely to see one above another. We show an example in Figure 5.4, where horizontally connected regions are formed by a group of pedestrians. In order to separate several pedestrians included in one blob, we have to determine the height value priorly to the width at each location.

As illustrated in Algorithm 2, we point out two details which turned out to be crucial for the final performance.

1. Aspect ratio. This parameter denoted as r_{wh} should be set consistent to the definition of the training samples and classification model. In our approach, it is set to be 0.5 ($64/128$).
2. Context. Gradient information between the human bodies and the background is very important for the further feature extraction procedure, thus, it is necessary to include some surrounding context for gradient computation. Here, we employ a fixed proportion of m_b applied to the height value of each detection window.



(a) (b)

Figure 5.4: *An example of blobs each containing multiple people walking side by side. (a) Multiple pedestrians. The green bounding boxes indicate ground truth annotations. (b) Segmented blobs.*

(a)

(b)

Figure 5.5: *Several examples of flow magnitudes for different categories of objects in a typical urban traffic environment. Each blue bounding box contains a pair of the original image and its corresponding flow magnitude map. (a) Moving pedestrians. (b) Other objects: non-pedestrian objects (including buildings and cars) and static pedestrians.*

5.3 Motion Self Difference Features

In a typical urban traffic environment, we may see various objects, for example, buildings, cars, traffic signs and pedestrians. Since our task is to detect moving pedestrians in this chapter, we categorize all the objects into: moving pedestrians; and the others, including non-pedestrian objects and static pedestrians.

Although the purpose of motion segmentation in Section 5.2.2, ''Graph-based Motion Segmentation'' is to select those blobs containing moving objects, it is inevitable to include some static scene parts due to high R_z (cf. Equation 5.1), and errors from the previous procedures. To ensure robust classification, we should design some features which are discriminative for moving pedestrians against all the other possible objects. Therefore, we observe the optical flow magnitude maps of moving pedestrians in Figure 5.5(a) and other objects in Figure 5.5(b) as two groups respectively.

From Figure 5.5, we find that moving pedestrians show rather different flow patterns from other objects and summarize the characteristics of two categories as follows:

- Moving pedestrians. The motion inside moving pedestrians' bodies varies over different body parts:
  - Upper body. For the upper body, the magnitudes reflect the body shape clearly.
  - Lower body. The contrast between the left and right legs are rather high. This is because only one leg moves at a time, resulting in higher magnitude values for one leg. We call this inter-body motion, caused by the non-rigidity of human body.
- Other objects. The flow magnitude maps of static objects and rigid moving objects clearly show the silhouette of the objects.

Based on the above findings, we propose MSD features to describe the special inter-body *relative* motion pattern for moving pedestrians. The feature extraction procedure is listed in Algorithm 3.

First, each detection window is resized to the size of our pedestrian model: $wd_m \times ht_m$ pixels, and then divided into $s \times s$ pixel square-sized regions, which we call *cells* in the following. We denote the number of cells along the horizontal and vertical directions as nc_x and nc_y, respectively:

$$nc_x = \frac{ht_m}{s}, \tag{5.13}$$

$$nc_y = \frac{wd_m}{s}. \tag{5.14}$$

Second, inside each cell, two histograms are computed using trilinear interpolation [Dalal and Triggs, 2005] in terms of two elements of flow vectors f_x and f_y respectively. For the whole detection window, two histogram sets are obtained and can be denoted as:

$$H_x = \{h_x(i, j) | 1 \leq i \leq nc_x, 1 \leq j \leq nc_y\}, \tag{5.15}$$

and

$$H_y = \{h_y(i, j) | 1 \leq i \leq nc_x, 1 \leq j \leq nc_y\}, \tag{5.16}$$

where i and j indicate the indexes of cells along the horizontal and vertical directions.

Third, we compute the feature values, which are represented by the difference between histograms in H_x or H_y. The difference measurement we choose is histogram intersection (HI), which has been proven to be effective and of low computational costs. Given two histograms h_p and h_q of n bins, histogram intersection is defined as

$$\mathrm{HI}(h_p, h_q) = \frac{\sum_{k=1}^{n} \min(h_p(k), h_q(k))}{\sum_{k=1}^{n} h_p(k)}. \tag{5.17}$$

For each cell, we consider all of its eight surrounding cells and compute the difference between the central cell and each surrounding cell using Equation 5.17.

However, if the above iteration goes over all cells, redundancy will emerge since almost all the cell pairs are considered for twice. To solve this problem, we employ a blacklist, which records the indexes of cell pairs that have been visited already. During the iteration, those cell pairs which have been visited are appended to the blacklist and a second visit is thus prohibited.

In our approach, we choose the size of pedestrian model to be 64×128 pixels, and the cell size of s to be 8 pixels, thus, the final MSD feature vector for one detection window is of 1072 dimensions. Notably, those cells along detection window borders do not have eight surrounding cells.

5.4 Two-layer Classification

Besides the MSD features introduced in Section 5.3, ‘‘Motion Self Difference Features’’, we also extract HOG features for each detection window. That is to say, we use two different kinds of features: appearance based HOG features and motion based MSD features. For classification with multiple features, there are generally two different strategies:

Algorithm 3 Extracting motion self difference features.

compute histogram sets H_x and H_y for the detection window;
initialize feature set: $F_{\text{MSD}} \leftarrow \emptyset$;
initialize cell pair tabu set: $P \leftarrow \emptyset$;
initialize cell pair index set used for each cell:
$dx = \{0, 1, 1, 1, 0, -1, -1, -1\}$, $dy = \{-1, -1, 0, 1, 1, 1, 0, -1\}$;
for $x_c = 1$ to nc_h **do**
  for $y_c = 1$ to nc_v **do**
    for $i = 1$ to 8 **do**
      $x_n = x_c + dx[i]$,$y_n = y_c + dy[i]$;
      if $x_n \notin [1, nc_h]$ || $y_n \notin [1, nc_v]$ **then**
        break;
      else if $[(x_c, y_c), (x_n, y_n)] \notin P$ AND $[(x_n, y_n), (x_c, y_c)] \notin P$ **then**
        compute HI$(h_x(x_c, y_c), h_x(x_n, y_n))$ using Eq. 5.17 and append it to F_{MSD};
        compute HI$(h_y(x_c, y_c), h_y(x_n, y_n))$ using Eq. 5.17 and append it to F_{MSD};
        append $[(x_c, y_c), (x_n, y_n)]$ to the tabu set P;
      end if
    end for
  end for
end for
return F_{MSD}

1. To combine features. This is rather simple to implement. In those approaches, all the features are concatenated into a long feature vector and only one classifier is trained as a whole. The disadvantages of those approaches are twofold: first, it is difficult to train one classifier when the dimension of the feature vector is too high; moreover, different kinds of features may have rather different classification boundaries, while the concatenating strategy prohibits the characteristic of individual features and thus have negative effects on the performance.
2. To combine classifiers. These methods are more complex. One base classifier is trained for each set of features individually, and then the final confidence score is from the combination of the outputs of all these base classifiers. Obviously, these methods consist of two layers, and are more adaptable to high variations among different kinds of features.

Based on the above discussion, we therefore apply a two-layer classification scheme in our approach, see an illustration in Figure 5.2.

It is an important issue for the two-layer classification scheme to choose a proper base classifier. In our approach, we choose SVMs, which have been very frequently used in the

field of pedestrian detection [Dalal and Triggs, 2005, Felzenszwalb et al., 2008, Maji et al., 2008]. More details regarding the implementation are explained in the following.

On the first layer, one individual classifier should be trained for each kind of features. We employ RBF-SVMs for both HOG and MSD features. In traditional methods, a linear kernel is most often chosen due to its efficiency but not the quality of results. However, we choose the RBF kernel instead which yields more reasonable results. Although the RBF-SVMs are more time consuming than linSVMs, the speed is still acceptable in our approach because we examine much less detection windows than previous methods did. For training of HOG features, we also use the INRIA pedestrian data set as in [Dalal and Triggs, 2005]; for training of MSD features, the INRIA pedestrian data set is not applicable because it only consists of single images other than image sequences, then we choose the TUD-Brussel data set [Wojek et al., 2009], which provides motion pair annotations. A multi-round training strategy is used for both features, so that the final model is trained on an augmented set, consisting of initial training data and hard negative samples, which are accumulatively selected over the negative images using the classifier trained in the previous round.

On the second classification layer, we have to output the final confidence score by combining the decision scores obtained from the first layer. There are multiple possible ways to accomplish this task. *Winner-take-all* is a simple and well-known framework, which chooses only one decision score among several scores from the base classifiers and treats the selected classifier as the winner. But this framework is not applicable in our case, where each kind of features is considered as supplementary to each other. Alternatively, we choose a linear kernel SVM for a more reasonable combination. A linear kernel is employed here because the feature space on the second layer only consists of two decision scores from the first layer classifiers. The final decision score is output from the trained linear kernel SVM.

After receiving final scores from our two-layer classification procedure, all the detection windows should go through a non-max suppression method [Felzenszwalb et al., 2008], which discards the less confident one with a lower score of every pair of detection windows that overlap sufficiently according to Equation 2.1.

5.5 Experiments

To run our experiments, we should choose some proper data set, which provides annotations for moving pedestrians. Unfortunately, to our best knowledge, among all the currently public pedestrian data sets, none of them satisfies such a requirement. Therefore, we decide to manually add ‘‘moving’’ annotations for some data set.

First, we chose the *Daimler mono pedestrian detection* benchmark [Enzweiler and Gavrila, 2009] as our basis data set, which was captured by a monochrome camera mounted on

Table 5.2: *Configuration comparison of HOG detector and our detectors.*

Detector	ROIs selection	Features	Classifier
HOG [Dalal and Triggs, 2005]	×	HOG	linSVM
MoSeg+HOG	√	HOG	RBF-SVM
MoSeg+HOG+MSD	√	HOG+MSD	RBF-SVM + linSVM

a vehicle driving through urban environment. This data set consists of a large number of pedestrians moving across or along the street. See more description of this data set in Section 2.2.4, ''Daimler Mono''.

Next, we manually label each pedestrian in the ground truth data to be moving or static. The status is determined through observing multiple consecutive video frames before and after the current time point. The additional label of ''moving'' is set to ''1'' in the ground truth data if the current pedestrian is considered moving, and ''0'' for static pedestrians.

In the following part of this section, we analyze the performance gains of our detector by ROIs selection and MSD features in Section 5.5.1, ''Comparisons of Our Detectors Under Different Configurations'', and make comparisons against several other state-of-the-art detectors in terms of performance in Section 5.5.2, ''Comparisons Against State-of-the-Art Detectors'', and also runtimes in Section 5.5.3, ''Runtime Analysis''.

5.5.1 Comparisons of Our Detectors Under Different Configurations

In this chapter, we made two modifications to the classic HOG detector: to implement motion segmentation as a preprocessing for ROIs selection; and to design novel motion based features as supplementary to appearance based HOG features. In order to examine the improvements brought by each modification individually, we define two detectors: MoSeg+HOG and MoSeg+HOG+MSD with different configurations. A comparison of our detectors and the classic HOG detector is shown in Table 5.2.

From Figure 5.6, we can see that both of our detectors obtain significant improvements to the baseline HOG detector. To specify, the modification of supplementing motion segmentation for ROIs selection brings about a considerable decrease of 12% in terms of average miss rate; furthermore, the integration of our novel, motion based MSD features results in an additional performance improvement of 7% with respect to average miss rate, compared to the above detector MoSeg+HOG. Therefore, our optimal detector is MoSeg+HOG+MSD, which includes two modifications and yields an overall improvement of 19% with respect

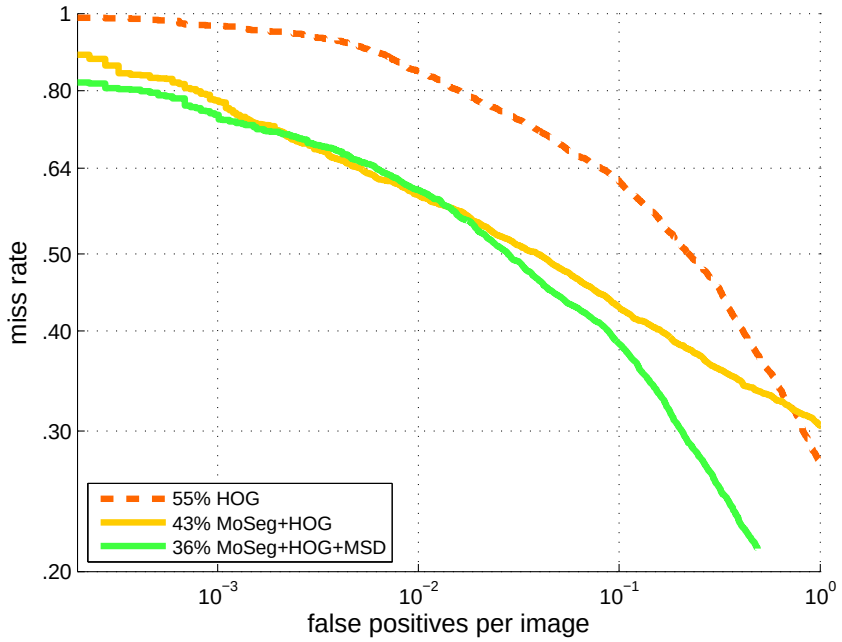


Figure 5.6: *Performance comparison of our detectors under different configurations on Daimler monocular pedestrian data set.*

to average miss rate to the baseline HOG detector. Notably, we use the configuration of MoSeg+HOG+MSD in the following experiments.

5.5.2 Comparisons Against State-of-the-Art Detectors

Besides classic HOG detector, we also compare our proposed detector MoSeg+HOG+MSD to several other state-of-the-art detectors, selected from a recent survey [Dollár et al., 2011] in the field of pedestrian detection. We exclude those detectors who use color information, since the data set only contains gray scale images. Moreover, we do not discuss those detectors whose runtimes are on minute-level per frame (640×480 pixels) reported by [Dollár et al., 2011], as they are far from real-time requirements for intelligent vehicle applications. We list the selected detectors as follows:

1. HOG [Dalal and Triggs, 2005]
2. VJ [Viola and Jones, 2004]
3. Shapelet [Sabzmeydani and Mori, 2007]
4. MultiFtr [Wojek and Schiele, 2008]
5. HikSvm [Maji et al., 2008]

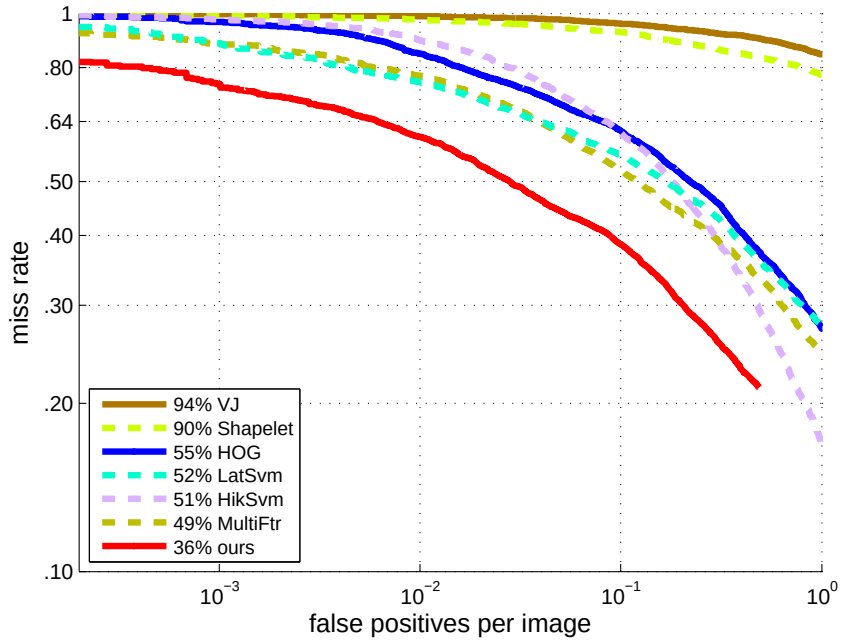


Figure 5.7: *Performance comparison of state-of-the-art detectors on Daimler monocular pedestrian data set.*

6. LatSvm [Felzenszwalb et al., 2008]

The original detection results of the selected detectors are available at `http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians`. The evaluations are implemented using the evaluation protocol explained earlier in Section 2.3, ‘‘Experiment Settings’’.

We plot miss rate vs. FPPI curves for all the above detectors, and indicate the *log-average miss rate* values in the left bottom rectangle as shown in Figure 5.7. Generally, our approach obtains a *log-average miss rate* of 36%, lower than other state-of-the-art detectors considered here, and especially our miss rate is consistently lower than the others in the whole FPPI range.

Moreover, we show some exemplary results under various scenarios in Figure 5.8, demonstrating the robustness of our approach against complex background, small scales and occlusions.

5.5.3 Runtime Analysis

In this subsection, we compare the runtimes for all the detectors considered in this chapter.

Table 5.3: *Runtimes (in seconds) of different detectors by normalizing to the rate of a single machine.*

Detector	ROIs selection	Features	Classifier	Runtime
VJ	×	Haar-like	AdaBoost	11.97
Shapelet	×	Gradients	AdaBoost	106.43
HOG	×	HOG	linSVM	19.72
LatSvm	×	HOG	latent SVM	10.86
HikSvm	×	HOG	HIK-SVM	29.58
MultiFtr	×	HOG+Haar	AdaBoost	62.63
ours	√	HOG+MSD	RBF-SVM+linSVM	1.93

At first, we report that the runtime of our detector is 1.93 seconds for a 640×480 single frame. Our detector is implemented on a modern computer with Intel Core-i5 CPU (2.4GHz), and our source code is written in Matlab. This value is computed by averaging the whole runtime for the data set of $21,790$ frames.

The coming problem is how to obtain the runtimes of other detectors. One solution is to re-implement every other detector and rerun them on our computer, which requires a large amount of work and is nearly infeasible to accomplish. Inspired by the normalization method first proposed in [Dollár et al., 2011], we decide to normalize their data to the rate of our machine so that we can make a fair comparison. In the following, we explain how the normalization is implemented in our case. First, we downloaded the source code of one detector in [Dollár et al., 2011] and ran it on our computer. After obtaining its runtime on our computer, we compare it with the data reported by [Dollár et al., 2011], thus we get a relative speed ratio between our computer and the computer used in [Dollár et al., 2011]. Then we use this ratio to normalize the runtimes of other detectors to the rate of our computer.

We list the runtimes of all the detectors considered in this chapter in Table 5.3. From this table, our detector is the fastest, and is around 10 times faster than the second fastest detector LatSvm. We also notice that our detector is the only one which applies ROIs selection. This demonstrates that ROIs selection is very important for enhancing efficiency.

5.6 Summary

In this chapter, we proposed a new approach exploiting motion information, for moving pedestrian detection. The main contributions are as follows:

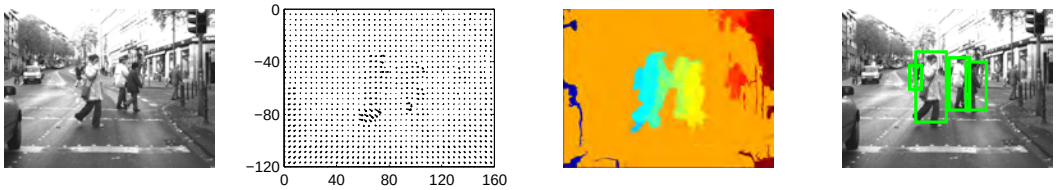
- ROIs selection by motion segmentation. The purpose of ROIs selection is to reduce the number of detection windows that should be examined by the classifiers, so as to enhance efficiency. From optical flow maps, moving objects are distinguishable from

the background and other static objects. This characteristic enables us to implement efficient graph-based motion segmentation and thus select those interesting blobs which probably contain moving pedestrians.
- Novel motion self difference features. By observing optical flow maps for various object categories, we find that moving pedestrians exhibit rather different flow patterns than other objects, due to the non-rigid inter-body motion. By making use of this property, we design new motion features, which describe the local difference in terms of flow histograms between neighboring square regions inside human bodies.
- Two-layer classification integrating motion and appearance cues. HOG features and MSD features are employed as an integration of motion and appearance information. Two base classifiers are trained individually for both features, whose decision scores are used to output the final confidence score by a joint decision.

Experimental results on the standard *Daimler mono pedestrian detection* benchmark showed that our approach obtained significant improvements compared to the baseline HOG detector. Moreover, we also outperformed several other state-of-the-art detectors. At the same time, we improved the efficiency by an order of magnitude.

The improvements in terms of both accuracy and efficiency were attributed to the utilization of motion information. In our approach, the number of examined detection windows was significantly reduced by explicitly generating pedestrian hypotheses inside selected motion field blobs (ROIs) after an efficient motion segmentation. This pre-processing not only reduced runtime, but also avoided many false positives which may be generated from the huge area of background. On the other hand, the novel motion features MSD enhanced the robustness by integrating with HOG features in a two-layer classification scheme.

Future work. The success achieved by utilizing motion information in our approach inspires us to further exploit it for more general pedestrian detection. If more accurate optical flow estimation is given, then our approach for ROIs selection can be also adapted to static pedestrians. Further more, our MSD features can be used to determine the moving status of detected pedestrians.



(f)

Figure 5.8: *Some exemplary results from our approach in different scenarios. In each row, the original frame, optical flow field, motion segmentation blobs and final detection results are shown from left to right.*

Chapter 6

Conclusions

This dissertation has covered three different approaches for pedestrian detection in urban traffic environments. On-board video sequences comprise abundant information about the surrounding environments, including spatial and temporal cues. It is thus an important yet difficult task to exploit useful information for recognizing pedestrians in a dynamic scene with complex backgrounds. The motivation of our work comes from the observation of a large number of image sequences consisting of pedestrians in outdoor environments. We aimed to exploit uniform characteristics patterns for pedestrians in huge data and design effective algorithms accordingly. Our main contributions and insights on efficient pedestrian detection can be recapitulated as follows.

First, we designed informed Haar-like features based on prior shape. We observed that pedestrians exhibit a common visual appearance from statistical data. Accordingly, we built a pedestrian shape model. This uniform geometry inspired us to employ Haar-like features, which have gained remarkable success for detecting objects of low intra-class variability, such as human faces. Considering the specific shape structure of pedestrians, we designed Haar-like templates tailored to our shape model. As supplementary to traditional binary modality, which only carries two different weights, we proposed a ternary modality, so as to better describe the complex geometry at corners. Furthermore, we applied each template on multiple channel values, in terms of colors and gradients, in order to integrate more information on local difference. The informed Haar-like features reached and surpassed state-of-the-art performance on standard benchmarks, at low computational costs. Additionally, our features are robust to occlusions since features in the head-shoulder area were automatically selected as more representative ones.

Second, we designed center-surround contrast features, motivated by human visual systems, which are always able to locate the interesting objects in an accurate and efficient way.

We emulated a top-down saliency system, and introduced a successful center-surround mechanism for feature design. To this end, we computed the difference between a central region and its surrounding regions as feature values. Various region descriptors and contrast measurements were evaluated, so as to select the optimal combination. Unlike the above informed Haar-like features, these center-surround contrast features did not rely on any prior knowledge, nevertheless they also reached state-of-the-art performance.

Third, we considered utilizing motion information for moving pedestrians. When the task is narrowed down to detecting moving pedestrians, which are actually more interesting in many applications, motion can be used in pre-processing for ROIs selection, because moving objects are more distinguishable on the motion maps and can be extracted by segmentation. Motivated by the inter-body motion caused by the non-rigidity of human body, we designed motion self difference features as supplementary to appearance features, to enable more robust recognition. In order to evaluate the effects of ROIs selection and novel motion features for moving pedestrian detection, we added additional moving annotations to the Daimler mono pedestrian data set. Experimental results showed that both modifications significantly improved the original HOG detector.

Based on the presented results, there are multiple promising directions for future research, to further enhance the accuracy and speed of the proposed approaches, or to generalize for other categories of objects.

Utilization of depth information. In this thesis, we have investigated how to employ spatial and temporal information for pedestrian detection, but we expect further improvement by integration with depth information. Thanks to the development of hardware, stereo cameras are becoming more and more popular in recent research. For indoor applications, Kinect is widely used due to its low price and real-time dense depth maps. For outdoor scenes, TYZX provides stereo cameras for different ranges. Depth information can be used in different detection stages. First, in pre-processing, depth can be used to select ROIs according to the interesting detection distance range. There is usually a limited distance range for a specific task. For example, for a typical driver assistance system, the interesting distance range is less than 50 meters. Consequently, regions that correspond to a bigger depth value can be ignored in further examination procedure. Second, depth can also be used to discard many false positives in post-processing. A set of pixels with rather diverse depth values unlikely belong to one single object.

GPU or hardware acceleration. Although our approaches are more efficient than many state-of-the-art detectors, they still need to be accelerated for real-time applications. One possibility is the usage of concurrency offered by general-purpose computing on graphics processing units (GPGPU). In our current implementation, the most time consuming parts are

inherently parallelizable: each feature value can be computed independently to others. An alternative way is to re-implement our approaches on an ARM processor (formerly known as Advanced RISC Machine or Acorn RISC Machine). This option may be more interesting for industry, due to its high performance and low price. The above two solutions should empower our approaches for real-time applications after appropriate re-implementation.

Combination with tracking. All the approaches proposed in this thesis are for the purpose of detection only. However, in real world applications, it is not necessary to do full frame search for every single frame, since each object usually changes only slightly in terms of appearance or location during a short time slot. Therefore, tracking is a useful strategy to estimate the current status from the previous ones. It can not only increase the speed, but also enhance the robustness. A simple yet reliable way for integration of detection and tracking is to run the detection algorithm every $N(N > 1)$ frames, and the objects can be traced during the time slot. Generally, the larger the value of N, the faster the whole system. But it is risky to choose a too large value for N, which may cause severe deviation after a long time period. In other words, there is a trade off between accuracy and speed.

Extension to more general object detection. Although we focused on pedestrian detection in this thesis, the proposed approaches can be also extended for more general object detection. The success obtained by informed Haar-like features implies that customized templates may be used for detecting other objects of relatively uniform shape structure. Even for more variable objects, it is possible to design a multiple template pool, each for one shape model built for one single posture or viewpoint. A similar extension may be applied to center-surround contrast features. The motivation behind center-surround contrast features is the mechanism of human visual systems, which is applicable for all kinds of objects. The only modification we need is alternative training samples. To train the center-surround contrast features extracted from a certain object class would enable the corresponding detection.

List of Figures

List of Tables

Bibliography

Agarwal, S., Awan, A., and Roth, D. (2004). Learning to detect objects in images via a sparse, part-based representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 26(11):1475--1490.

Aharon, M., Elad, M., and Bruckstein, A. (2006). K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *Signal Processing, IEEE Transactions on*, 54(11):4311--4322.

Alonso, I., Llorca, D., Sotelo, M., Bergasa, L., Revenga de Toro, P., Nuevo, J., Ocana, M., and Garrido, M. (2007). Combination of feature extraction methods for svm pedestrian detection. *IEEE Transactions on Intelligent Transportation Systems (TITS)*, 8(2):292--307.

Andriluka, M., Roth, S., and Schiele, B. (2008). People-tracking-by-detection and people-detection-by-tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1--8.

Appel, R., Fuchs, T., Dollár, P., and Perona, P. (2013). Quickly boosting decision trees-pruning underachieving features early. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 594--602.

Bar-Hillel, A., Levi, D., Krupka, E., and Goldberg, C. (2010). Part-based feature synthesis for human detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 127--142.

Behley, J. (2013). *Three-dimensional Laser-based Classification in Outdoor Environments*. Phd thesis, University of Bonn.

Benenson, R., Mathias, M., Timofte, R., and Gool, L. V. (2012). Pedestrian detection at 100 frames per second. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2903--2910.

Benenson, R., Mathias, M., Tuytelaars, T., and Gool, L. V. (2013). Seeking the strongest rigid detector. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3666--3673.

Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(8):1798--1828.

Bertozzi, M., Broggi, A., Chapuis, R., Chausse, F., Fascioli, A., and Tibaldi, A. (2003). Shape-based pedestrian detection and localization. In *Proceedings of the IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 328--333.

Bertozzi, M., Broggi, A., Fascioli, A., Tibaldi, A., Chapuis, R., and Chausse, F. (2004). Pedestrian localization and tracking system with kalman filtering. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 584--589.

Bourdev, L., Maji, S., Brox, T., and Malik, J. (2010). Detecting people using mutually consistent poselet activations. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 168--181.

Bourdev, L. and Malik, J. (2009). Poselets: Body part detectors trained using 3d human pose annotations. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 1365--1372.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5--32.

Broggi, A., Bertozzi, M., Fascioli, A., and Sechi, M. (2000). Shape-based pedestrian detection. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 215--220.

Broggi, A., Fascioli, A., Fedriga, I., Tibaldi, A., and Rose, M. (2003). Stereo-based preprocessing for human shape localization in unstructured environments. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 410--415.

Brox, T., Bruhn, A., Papenberg, N., and Weickert, J. (2004). High accuracy optical flow estimation based on a theory for warping. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 494--499.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3):273--297.

Cover, T. M. and Thomas, J. A. (2006). *Elements of Information Theory*. Wiley, 2nd edition.

Curio, C., Edelbrunner, J., Kalinke, T., Tzomakas, C., and Von Seelen, W. (2000). Walking pedestrian recognition. *IEEE Transactions on Intelligent Transportation Systems*, 1(3):155--163.

Dalal, N. (2006). *Finding people in images and videos*. Phd thesis, Institut National Polytechnique de Grenoble / INRIA Rhâne-Alpes.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 886--893.

Dalal, N., Triggs, B., and Schmid, C. (2006). Human detection using oriented histograms of flow and appearance. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 428--441.

Dollár, P., Appel, R., and Kienzle, W. (2012). Crosstalk cascades for frame-rate pedestrian detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 645--659.

Dollár, P. and Perona, P. (2010). The fastest pedestrian detector in the west. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 68.1--68.11.

Dollár, P., Tu, Z., Perona, P., and Belongie, S. (2009a). Integral channel features. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 91.1--91.11.

Dollár, P., Wojek, C., Schiele, B., and Perona, P. (2009b). Pedestrian detection: A benchmark. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 304--311.

Dollár, P., Wojek, C., Schiele, B., and Perona, P. (2011). Pedestrian detection: an evaluation of the state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34(4):743--761.

Elzein, H., Lakshmanan, S., and Watta, P. (2003). A motion and shape-based pedestrian detection algorithm. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 500--504.

Enzweiler, M. and Gavrila, D. M. (2009). Monocular pedestrian detection: survey and experiments. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 31(12):2179--2195.

Enzweiler, M., Kanter, P., and Gavrila, D. M. (2008). Monocular pedestrian recognition using motion parallax. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 792--797.

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

Felzenszwalb, P. F. and Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. *International Journal of Computer Vision (IJCV)*, 59(2):167--181.

Felzenszwalb, P. F., McAllester, D., and Ramanan, D. (2008). A discriminatively trained, multiscale, deformable part model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1--8.

Franke, U. and Heinrich, S. (2002). Fast obstacle detection for urban traffic situations. *IEEE Transactions on Intelligent Transportation Systems*, 3(3):173--181.

Freund, Y. and Schapire, R. V. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55:119--139.

Frintrop, S., Rome, E., and I., C. H. (2010). Computational Visual Attention Systems and their Cognitive Foundation: A Survey. *ACM Transactions on Applied Perception*, 7(1).

Gandhi, T. and Trivedi, M. (2007). Pedestrian protection systems: Issues, survey, and challenges. *IEEE Transactions on Intelligent Transportation Systems (TITS)*, 8(3):413--430.

Gavrila, D. (2000). Pedestrian detection from a moving vehicle. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 37--49.

Gavrila, D. M., Giebel, J., and Munder, S. (2004). Vision-based pedestrian detection: The protector system. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 13--18.

Gavrila, D. M. and Munder, S. (2007). Multi-cue pedestrian detection and tracking from a moving vehicle. *International Journal of Computer Vision (IJCV)*, 73(1):41--59.

Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3354--3361.

Geronimo, D., Lopez, A., Sappa, A., and Graf, T. (2010). Survey of pedestrian detection for advanced driver assistance systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 32(7):1239--1258.

Givens, C. and Shortt, R. (1984). A class of wasserstein metrics for probability distributions. *Michigan Mathematical Journal*, 2(31).

Gualdi, G., Prati, A., and Cucchiara, R. (2010). Multi-stage sampling with boosting cascades for pedestrian detection in images and videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 196--209.

Haddad, R. (1971). A class of orthogonal nonrecursive binomial filter. *IEEE Transactions on Audio and Electroacoustics*, 19(3):296--304.

Hellinger, E. (1909). Neue begrürndung der theorie quadratischer formen von unendlichvielen verärnderlichen. *Journal für die reine und angewandte Mathematik*, 136:210--271.

Hinterstoisser, S., Lepetit, V., Ilic, S., Fua, P., and Navab, N. (2010). Dominant orientation templates for real-time detection of texture-less objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2257--2264.

Hong, X., Chang, H., Chen, X., and Gao, W. (2010). Boosted Sigma Set for Pedestrian Detection. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pages 3017--3020.

Horn, B. and Schunck., B. (1981). Determining optical flow. *Artificial Intelligence*, 17(1-3):185–203.

Horn, B. K. P. and Schunck, B. G. (1980). Determining optical flow. *Artif. Intell.*, 17(1-3):185--203.

Hussain, S., Triggs, B., and Kuntzmann, L. J. (2010). Feature Sets and Dimensionality Reduction for Visual Object Detection. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 112.1--112.10.

Itti, L., Koch, C., and Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 20(11):1254--1259.

Jiang, H. (2012). Finding people using scale, rotation and articulation invariant matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 388--401.

Jones, J. P. and Palmer, L. A. (1987). An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex. *Journal of Neurophysiology*, 58(6):1233--1258.

Jones, M. J. and Snow, D. (2008). Pedestrian detection using boosted features over many frames. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pages 1--4.

Kamijo, S., Fujimura, K., and Shibayama, Y. (2010). Pedestrian detection algorithm for on-board cameras of multi view angles. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 973--980.

Kamijo, S., Matsushita, Y., Ikeuchi, K., and Sakauchi, M. (2000). Occlusion robust tracking utilizing spatio-temporal markov random field model. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pages 140--144.

Kao, Y.-F., Chan, Y.-M., Fu, L.-C., Hsiao, P.-Y., Huang, S.-S., Wu, C.-E., and Luo, M.-F. (2012). Comparison of granules features for pedestrian detection. In *Proceedings of the IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 1777--1782.

Klein, D. and Frintrop, S. (2011). Center-surround divergence of feature statistics for salient object detection. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 2214--2219.

Klein, D. and Frintrop, S. (2012). Salient Pattern Detection using W2 on Multivariate Normal Distributions. In *Proceedings of the Symposium of the German Association for Pattern Recognition (DAGM)*, pages 246--255.

Kullback, S. and Leibler, R. (1951). On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79--86.

Lawrence, N. D. and Moore, A. J. (2007). Hierarchical gaussian process latent variable models. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 481--488.

Lee, B. B., Martin, P. R., and Grünert, U. (2010). Retinal connectivity and primate vision. *Progress in Retinal and Eye Research*, 29(6):622--639.

Leibe, B., Cornelis, N., Cornelis, K., and Van Gool, L. (2007). Dynamic 3d scene analysis from a moving vehicle. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1--8.

Leibe, B., Seemann, E., and Schiele, B. (2005). Pedestrian Detection in Crowded Scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 878--885.

Lempitsky, V. S., Roth, S., and Rother, C. (2008). Fusionflow: Discrete-continuous optimization for optical flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1--8.

Levi, D., Silberstein, S., and Bar-Hillel, A. (2013). Fast multiple-part based object detection using kd-ferns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 947--954.

Li, Z., Sun, Y., Liu, F., and Shi, W. (2008). An effective and robust pedestrians detecting algorithm & symposia. In *Proceedings of the IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 545--549.

Lim, J. and Kim, W. (2013). Detecting and tracking of multiple pedestrians using motion, color information and the adaboost algorithm. *Multimedia Tools and Applications*, 65(1):161--179.

Lim, J. J., Zitnick, C. L., and Dollár, P. (2013). Sketch tokens: a learned mid-level representation for contour and object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3158--3165.

Lin, Z. and Davis, L. S. (2008). A pose-invariant descriptor for human detection and segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 423--436.

Liu, C. (2009). *Beyond pixels: exploring new representations and applications for motion analysis*. PhD thesis, Massachusetts Institute of Technology.

Liu, Y., Shan, S., Zhang, W., Chen, X., and Gao, W. (2009). Granularity- tunable gradients partition descriptors for human detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1255--1262.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91--110.

Ma, W., He, P., Huang, L., and Liu, C. (2010). Context Inspired Pedestrian Detection in Far-Field Videos. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pages 3009--3012.

Ma, Y., Deng, L., Chen, X., and Guo, N. (2013). Integrating orientation cue with eoh-olbp-based multilevel features for human detection. *Circuits and Systems for Video Technology, IEEE Transactions on*, 23(10):1755--1766.

Maji, S., Berg, A. C., and Malik, J. (2008). Classification using intersection kernel support vector machines is efficient. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1--8.

Mock, C., Quansah, R., Krishnan, R., Arreola-Risa, C., and Rivara, F. (2004). Strengthening the prevention and care of injuries worldwide. *Lancet*, 26(363):2172--2179.

Montabone, S. and Soto, A. (2010). Human detection using a mobile platform and novel features derived from a visual saliency mechanism. *Image and Vision Computing (IVC)*, 28(3):391--402.

Mu, Y., Yan, S., Liu, Y., Huang, T., and Zhou, B. (2008). Discriminative Local Binary Patterns for Human Detection in Personal Album. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1--8.

Munder, S. and Gavrila, D. M. (2006). An experimental study on pedestrian classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 28(11):1863--1868.

Ojala, T., Pietikäinen, M., and Harwood, D. (1996). A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51--59.

Oren, M., Papageorgiou, C., Sinha, P., Osuna, E., and Poggio, T. (1997). Pedestrian detection using wavelet templates. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 193--199.

Ouyang, W. and Wang, X. (2012). A discriminative deep model for pedestrian detection with occlusion handling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3258--3265.

Ouyang, W., Zeng, X., and Wang, X. (2013). Modeling mutual visibility relationship with a deep model in pedestrian detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3222--3229.

Papageorgiou, C. and Poggio, T. (2000). A trainable system for object detection. *International Journal of Computer Vision (IJCV)*, 38(1):15--33.

Parikh, D. and Zitnick, C. (2011). Finding the weakest link in person detectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1425--1432.

Park, D., Ramanan, D., and Fowlkes, C. (2010). Multiresolution models for object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 241--254.

Park, D., Zitnick, C. L., Ramanan, D., and Dollár, P. (2013). Exploring weak stabilization for motion feature extraction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2882--2889.

Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(11):559--572.

Peden, M., Scurfield, R., Sleet, D., Mohan, D., Hyder, A. A., Jarawan, E., and Mathers, C. (2004). World report on road traffic injury prevention. Eds., Geneva, Switzerland:World Health Organization.

Pishchulin, L., Jain, A., Wojek, C., Thormaehlen, T., and Schiele, B. (2011). In good shape: Robust people detection based on appearance and shape. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 5.1--5.12.

Porikli, F. (2005). Integral histogram: A fast way to extract histograms in Cartesian spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 829--836.

Ren, X. and Ramanan, D. (2013). Histograms of sparse codes for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3246--3253.

Rodieck, R. W. (1965). Quantitative analysis of cat retinal ganglion cell response to visual stimuli. *Vision Research*, 5(12):583--601.

Sabzmeydani, P. and Mori, G. (2007). Detecting pedestrians by learning shapelet features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1--8.

Schwartz, W., Kembhavi, A., Harwood, D., and Davis, L. (2009). Human detection using partial least squares analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24--31.

Shannon, C. (1949). Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10--21.

Shashua, A., Gdalyahu, Y., and Hayun, G. (2004). Pedestrian detection for driving assistance systems: single-frame classification and system level performance. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 1--6.

Tang, D., Liu, Y., and kyun Kim, T. (2012). Fast pedestrian detection by cascaded random forest with dominant orientation templates. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 58.1--58.11.

Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 511--518.

Viola, P. and Jones, M. J. (2004). Robust real-time face detection. *International Journal of Computer Vision (IJCV)*, 57(2):137--154.

Viola, P., Jones, M. J., and Snow, D. (2005). Detecting pedestrians using patterns of motion and appearance. *International Journal of Computer Vision (IJCV)*, 63(2):153--161.

Walk, S., Majer, N., Schindler, K., and Schiele, B. (2010). New features and insights for pedestrian detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1030--1037.

Wang, M., Li, W., and Wang, X. (2012). Transferring a generic pedestrian detector towards specific scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3274--3281.

Wang, M. and Wang, X. (2011). Automatic adaptation of a generic pedestrian detector to a specific traffic scene. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3401--3408.

Wang, X. and Han, T. X. (2009). An HOG-LBP human detector with partial occlusion handling. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 32--39.

Wojek, C. and Schiele, B. (2008). A performance evaluation of single and multi-feature people detection. In *Proceedings of the Symposium of the German Association for Pattern Recognition (DAGM)*, pages 82--91.

Wojek, C., Walk, S., and Schiele, B. (2009). Multi-cue onboard pedestrian detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 794--801.

Wold, H. (1985). *Partial least squares*, pages 581--591. Wiley.

Wu, B. and Nevatia, R. (2007). Detection and Tracking of Multiple, Partially Occluded Humans by Bayesian Combination of Edgelet based Part Detectors. *International Journal of Computer Vision (IJCV)*, 75(2):247--266.

Yamauchi, Y., Fujiyoshi, H., Hwang, B.-W., and Kanade, T. (2008). People detection based on co-occurrence of appearance and spatiotemporal features. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pages 1--4.

Yan, J., Zhang, X., Lei, Z., Liao, S., and Li, S. Z. (2013). Robust multi-resolution pedestrian detection in traffic scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3033--3040.

Ye, Q., Jiao, J., and Zhang, B. (2010). Fast pedestrian detection with multi-scale orientation features and two-stage classifiers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 881--884.

Zach, C., Pock, T., and Bischof, H. (2007). A duality based approach for realtime tv-l1 optical flow. In *Proceedings of the Symposium of the German Association for Pattern Recognition (DAGM)*, pages 23--45.

Zhang, S., Bauckhage, C., and Cremers, A. B. (2014a). Efficient pedestrian detection via rectangular features based on a statistical shape model. *IEEE Transactions on Intelligent Transportation Systems*, pages 1--13.

Zhang, S., Bauckhage, C., and Cremers, A. B. (2014b). Informed haar-like features improve pedestrian detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 947--954.

Zhang, S., Bauckhage, C., Klein, D., and Cremers, A. (2013). Moving pedestrian detection based on motion segmentation. In *Proceedings of the IEEE Workshop on Robot Vision (WoRV)*, pages 102--107.

Zhang, S., Bauckhage, C., Klein, D. A., and Cremers, A. B. (2015). Exploring human vision driven features for pedestrian detection. *arXiv*.

Zhang, S., Klein, D. A., Bauckhage, C., and Cremers, A. B. (2014c). Center-surround contrast features for pedestrian detection. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pages 2293--2298.

Zheng, Y., Shen, C., Hartley, R., and Huang, X. (2011). Pyramid center-symmetric local binary/trinary patterns for effective pedestrian detection. In *Proceedings of Asian Conference of Computer Vision (ACCV)*, pages 281--292.

Zhu, Q., Avidan, S., Yeh, M.-C., and Cheng, K.-T. (2006). Fast Human Detection Using a Cascade of Histograms of Oriented Gradients. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1491--1498.

Zini, L. and Odone, F. (2011). Efficient pedestrian detection with group lasso. In *Proceedings of the International Conference on Computer Vision (ICCV) Workshops*, pages 1777--1784.