Graz University of Technology

Institute for Computer Graphics and Vision

Dissertation

Scene specific object detection and tracking

Sabine Sternig

Graz, Austria, 2013

Thesis supervisors
Prof. Dr. Horst Bischof
Prof. Dr. Björn Ommer

Everything should be as simple as it is, but not simpler.

Albert Einstein

Abstract

Object detection or object tracking are often the first steps towards an automatic video analysis. Numerous applications, such as visual surveillance, industrial applications or sports analysis utilize stationary cameras. Applications for analyzing video data from stationary cameras have to deal with a smaller variability within the data due to restricted environmental conditions. However, they have to be able to deal with changing environmental conditions, like changing lighting conditions, backgrounds or variations in the objects' appearance. Adaptive classifiers (detectors or trackers) adjusting to changing conditions on-the-fly can be used to handle these variations. To avoid human interaction, adaptive approaches have also to incorporate unlabeled information from the scene.

The main challenge with incorporating unlabeled information is to preserve robustness. Thus, the main focus of this thesis is on how to robustly integrate unlabeled information from the scene for object detection and object tracking from single and multiple stationary cameras without losing long-term stability. We propose different approaches able to robustly adapt to changing environmental conditions. The approaches for object detection are based on the idea of classifier grids, where the scene is divided into highly overlapping patches, each of them holding its own classifier. We develop different update strategies for the classifier grids like fixed update strategies, inverse multiple instance learning and classifier co-grids. Even though the object detector is updated by incorporating unlabeled information from the scene over a long period of time (i.e., running over a week), the long-term stability is preserved. We propose a robust on-line learning algorithm (TransientBoost), which allows for combining reliable (labeled) information with unreliable (unlabeled) information within one classifier. The reliable information is kept fixed while the unreliable information is adapted over time. Furthermore, we propose a method for linking off-line and on-line learning that allows for exploiting prior information about the object class. Incorporating scene specific information is not only beneficial for single camera applications. We demonstrate that exploiting this information is also beneficial for networks of stationary cameras, where we propose a multiple object tracking approach which implicitly handles geometric uncertainties within a novel Hough voting scheme.

For all applications we demonstrate that incorporating scene specific information is beneficial. It allows for using less training data and less complex classifiers, which are suited for the actual problem. Therefore, even using less training data, the number of false positive as well as false negative predictions is reduced and thus the performance and accuracy is improved for all video analysis tasks.

Kurzfassung

Das Erkennen und Verfolgen (Tracken) von Objekten stellen oftmals die ersten Schritte in Richtung einer automatisierten Videoanalyse dar. In zahlreichen Bereichen wie Videoüberwachung, industriellen Anwendungen oder Sportanalysen kommen statische Kameras zum Einsatz. Obwohl statische Kameras die Datenvariabilität einschränken, müssen die sich im Laufe der Zeit verändernden Umweltbedingungen berücksichtigt werden. Um mit wechselnden Bedingungen (z.B. Beleuchtungsverhältnisse) umgehen zu können, werden adaptive Klassifikatoren (Detektoren oder Tracker) benötigt. Für adaptive Ansätze ist zudem die selbstständige Verarbeitung von unbekannter Information aus der Szene erforderlich, damit manuelle Eingriffe vermieden werden. Als größte Herausforderung gilt es dabei die Robustheit aufrecht zu erhalten. In Anbetracht dessen liegt der Fokus dieser Arbeit darin, Szenen-Information robust zu integrieren und dadurch das Erkennen und Tracken von Objekten in Einzel-und Multi-Kamera-Systemen zu verbessern. Dafür wurden verschiedene Ansätze entwickelt, die eine Adaption an veränderliche Umweltbedingungen ermöglichen, aber dennoch Langzeitstabilität bieten. Die Ansätze für die Erkennung von Objekten basieren auf der Idee der "Classifier Grids", in welchen die Szene in zahlreiche sich überlappende Bereiche unterteilt wird, wobei jeder dieser Bereiche über einen eigenen Detektor verfügt. In dieser Arbeit werden unterschiedliche Ansätze präsentiert um die Detektoren über die Zeit hinweg an die sich verändernden Bedingungen zu adaptieren: fixe Update-Strategien, inverses Multiple-Instance Lernen und ein auf Co-Training basierender Ansatz. Zusätzlich wurde ein robuster on-line Lernalgorithmus (TransientBoost) entwickelt, welcher die Möglichkeit zur Kombination von zuverlässiger Information (vom Benutzer bestimmt) mit unzuverlässiger Information (aus der Szene entnommen) bietet. Die Aufrechterhaltung der Langzeitstabilität wird experimentell belegt. Des Weiteren wurde ein Verfahren eingeführt, welches die Verknüpfung von on-line und off-line Lernen erlaubt, wodurch vorhandenes Vorwissen eingebracht werden kann. Der positive Effekt von szenenspezifischer Information wird auch für das Tracken von Objekten in Multi-Kamera-Systemen demonstriert. Der vorgestellte Tracking-Ansatz basiert auf Hough-Transformationen und kombiniert Informationen aus den einzelnen Kameras auf einer gemeinsamen Grundebene, wobei Unsicherheiten in der Geometrie implizit gehandhabt werden.

Der Vorteil der Verwendung von szenenspezifischer Information wird für alle Anwendungen aufgezeigt. Diese Information führt sowohl zu einer Reduktion der benötigten Trainingsdaten als auch zu weniger komplexen Klassifikatoren. Die präsentierten adaptiven Ansätze passen sich an die jeweilige Problemsituation an, weisen dadurch eine geringere Fehleranfälligkeit auf und eignen sich daher besonders gut für eine automatische Videoanalyse.

Statutory Declaration

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

Place

Date

Signature

Eidesstattliche Erklärung

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

Ort

Datum

Unterschrift

Acknowledgments

I would like to thank all those people who contributed to this thesis in numerous different ways and made the last years an unforgettable experience for me!

First of all, I would like to thank my supervisor Horst Bischof for providing me the possibility of doing research in the exciting field of computer vision! He supported me throughout my thesis with his patience and knowledge whilst allowing me the room to work in my own way. I am grateful to Björn Ommer for agreeing to serve as the second thesis supervisor on short notice.

I am particularly grateful to Peter Roth, the Boss, for your guidance, mentoring and invaluable feedback on my research! Without your support this thesis would not exist in this form! I would like to thank all co-authors for the fruitful discussions, in particular, Martin Godec, Thomas Mauthner, Michael Donoser and Horst Possegger. Special thanks to Hayko Riemenschneider for your continuous encouragement, countless fruitful (scientific and non-scientific) discussions and the peerless cooperation!

There are numerous different groups which brighten up my time at ICG! Thanks to all current and former members of the Learning, Recognition and Surveillance (LRS) group and the reading group for fruitful discussions and unforgettable social events! Thanks to the coffee break group, Spar group and noodle group for making daily grind stuff to be enjoyable and entertaining. Thanks to the climbing group, especially to the two “slave drivers” Kö and Mani. Thanks to the administrative staff at ICG, in particular to Renate and Andi for their support, but also for all the refreshing conversations.

I am glad to know that there are people I always can count on when times are rough: my friends and my family. I am especially grateful to my parents. Your unconditional love carries me through always! Many thanks to my sister Nici for the great journeys that helped to re-charge my batteries! Thanks to Dagmar, Günter, Verena, Thomas and Emma for the warm support and numerous positive distractions! Last but not least, I would like to thank Gernot, the most important person in my life. Thanks for your love, endless encouragement and for always believing!

Contents

List of Figures

List of Tables

If you want truly to understand something try to change it.

Kurt Lewin

1 Introduction

Today, all over the world there is a sustained growth in the number of closed circuit television (CCTV) surveillance cameras monitoring public areas. The country where CCTV is most popular is the UK, where in 2011 the number of surveillance cameras was estimated at 1.85 million or one for every 32 citizens*. The presence of surveillance cameras alone can already prevent personal and property crime caused by their daunting effect. If publicized well, CCTV may deter crime because the increased risk of detection discourages potential offenders [140]. The effect of CCTV in public areas for crime prevention was evaluated in [140], showing that an overall decrease of 16% in crime in experimental areas with CCTV compared to areas without CCTV could be reached. The largest influence was observed in car parks, where a significant crime reduction of 51% was noticed. Another positive effect of CCTV cameras was observed in public transport areas, where crime was reduced by 23%. Least effective was the use of CCTV cameras in city and town centers or public housing communities, where only a decrease by 7% was shown. This evaluation was performed in the UK and the US. They discovered that schemes evaluated in the UK were more effective than those in the US and other

*`http://www.guardian.co.uk/uk/2011/mar/02/cctv-cameras-watching-surveillance` (accessed at 01/08/2013)

countries, mainly influenced by the extremely positive effect of CCTV cameras in car parks in the UK. If a crime could not be prevented, CCTV could at least help to solve an already committed crime.

Positive effects on crime prevention as well as continuously decreasing hardware costs lead to a continuously increasing number of surveillance cameras. This requires more and more human operators and qualified personal to review all video information. A study on video surveillance at US schools [71] came to the distressing result that the attention of most operators degrades to below an acceptable level after only 20 minutes of watching and evaluating monitor screens. In fact, even for motivated employees monitoring video output is such a boring task that they become non-productive after a short period of time. Besides the risk of lacking concentration human operators incur substantial personal costs.

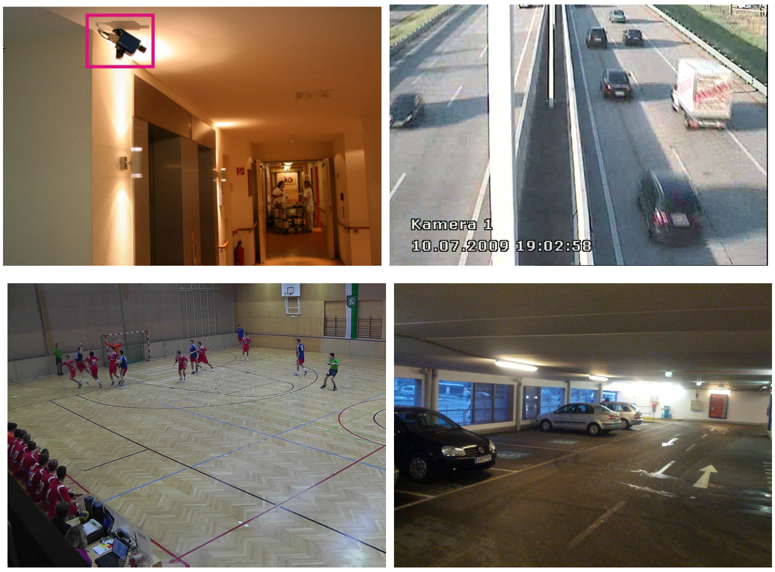


Figure 1.1: Object detection and tracking from stationary cameras is applicable to various different applications in everyday life including unusual event detection for assisted living (image taken from [49]), sports analysis, event detection on highways or monitoring car parks.

However, at the moment manual video surveillance, i.e., a human operator analyzing the content of videos is still common. In order to reduce costs, increase productivity and ease the task for human operators the goal would be to have autonomous surveillance systems notifying human operators automatically in case of event detection. Computer vision algorithms like object detection or tracking can be considered as a first step toward autonomous surveillance systems. Besides visual surveillance, there are various other applications for such vision based approaches, especially in areas like health-care, assisted living and sports analysis as shown in Figure 1.1. The goal of these applications is to autonomously detect and track people, cars or other objects to support us in our day-to-day lives. Various studies showed that the life expectancy is going to increase. Today a large percentage of people older than 60 years live alone in their homes. Autonomous surveillance systems reporting an alarm in case of suspicious behavior would extend the possibility for elderly people to live independently. Another application for object detection and tracking is sports analysis which can be used to reveal performance issues through efficient game analysis or to individualize the physical training plans of athletes. Visual surveillance systems on highways for example can automatically detect events like wrong way drivers or traffic jams and trigger an alarm. Such security systems can also be employed to detect conspicuous persons at car parks or detect unattended luggage at airports.

The main requirement for object detectors or object trackers for applications such as visual surveillance that they have to work under real-world scenarios, requiring to deal with changing environmental conditions, e.g., illumination conditions, changing weather as illustrated in Figure 1.2. To deal with different kinds of environmental conditions one either needs a classifier (detector or tracker) capable for dealing with all different conditions or a classifier that is adapting to changing environmental conditions. Having one single classifier capable for all different conditions requires a very complex detector trained on a huge set of training data containing all possible variations. Collecting such a huge set of training data is very time- and cost-consuming. Additionally, since it is hard to handle all variations within a single classifier, such a generic classifier suffers from false positive detections as well as missed detections. In contrast, an adaptive classifier can incorporate new scene specific information, which allows for having a significantly less complex detector trained on a smaller set of training data. The required number of training data can be drastically decreased, since it is possible to incorporate new information on-the-fly.

However, the main challenge of an adaptive object detector is to incorporate the scene specific information without human intervention to reach the goal of fully autonomous surveillance. This particular problem is addressed within this thesis, where the focus is on developing adaptive approaches for object detection and object tracking from single and multiple stationary cameras that are able to incorporate unlabeled information. One main challenge of incorporating unlabeled information is to preserve the long-term

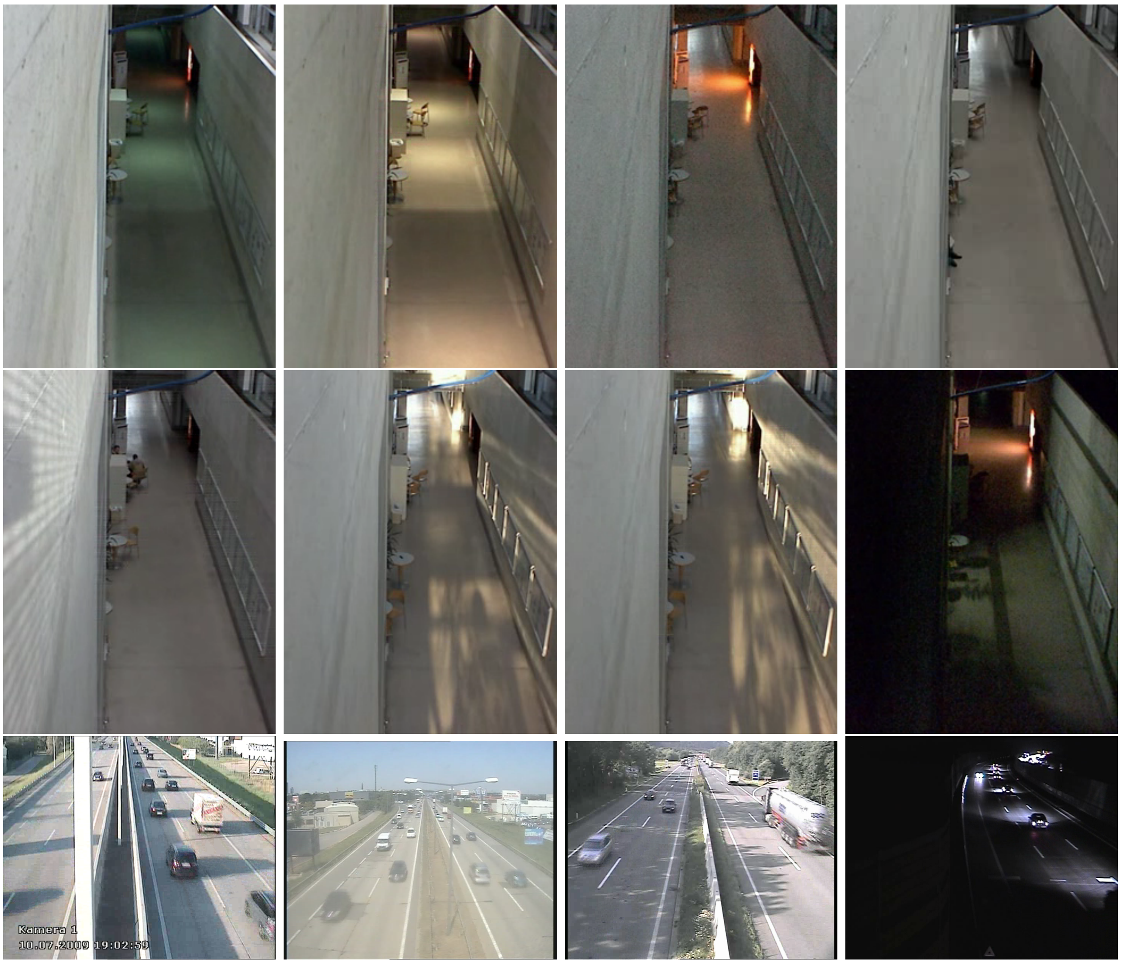


Figure 1.2: Changing lightning conditions in indoor and outdoor scenes.

robustness of object detection or object tracking, which is a major requirement for real-world applications.

Contributions

The main contributions of the thesis tackle the question of how to incorporate prior knowledge or scene specific information in an unsupervised manner. Therefore, we develop different approaches that allow incorporating scene specific information for object detection and tracking in static camera setups. This allows adapting to specific scenes, which is beneficial in both single and multiple camera setups. The content of this thesis is based on the work presented in [116, 127, 126, 125, 129, 128].

- **Linking off-line and on-line learning for boosting for feature selection**
  To exploit prior knowledge we propose an approach to link off-line and on-line learning for boosting for feature selection. On-line boosting for feature selection is an efficient algorithm for object detection and tracking. However, originally it is based on a random feature initialization. Prior information about the object class can be exploited by using a modified off-line boosting for feature selection algorithm in an initial step to initialize the features for the subsequent on-line boosting for feature selection process with features fitting the actual problem.

- **Adaption to changing environmental conditions by using unlabeled information from the scene**
  Adaptive approaches require to robustly incorporate unlabeled information. However, most existing approaches that do not rely on human interaction are prone to drifting. To avoid this effect we develop different update strategies to adapt to a scene by including scene specific background information. This adaption is essential for real-world applications, where e.g., changing illumination conditions need to be handled. The presented approaches provide real-time capabilities.

- **Adaption to changing foreground information**
  Scene specific foreground information might be beneficial, if no training data for a particular subgroup of an object class is available. Incorporating unlabeled foreground information and unlabeled background information requires a robust learning algorithm, which is able to deal with a certain amount of label noise. This has the nice side-effect that the labeling effort for the foreground training data can be reduced. Here we tackle both, the question of what kind of samples from the scene should be used to update the model by a novel update strategy as well as the question of how to incorporate the noisy samples which are generated on-line by proposing a robust learning algorithm (*TransientBoost*).

- **Multi-camera multi-object tracking inherently avoiding problems caused by geometric uncertainties**
  After demonstrating the benefits of incorporating scene specific information for object detection from single stationary cameras, we also exploit this information for multi-camera multi-object tracking. We propose based on Hough voting, which incorporates scene specific information and implicitly handles the often ignored problem of geometric uncertainties within a Hough tracking approach.

Outline

The outline of this thesis is as follows. First, related work and preliminaries on machine learning are described in Chapter 2. The thesis is further divided into two major parts, object detection from a single static camera (Chapter 3) and object tracking in a multiple camera setup (Chapter 4). Both chapters start with an introduction describing the task of interest followed by related work, the proposed approaches, experiments demonstrating the benefits of each individual approach and a summary. The goal of both chapters is to explain how to incorporate scene specific information to improve object detection or object tracking by exploiting the available information without human intervention. Finally, we summarize and conclude with an outlook to future work.

The beautiful thing about learning is nobody can take it away from you.

B. B. King

2 Machine Learning

Contents

Over the last years there has been a tremendous progress in the field of machine learning, enabling to apply machine learning in various different fields of research, like text classification, network intrusion detection, brain computer interfaces or computer vision. In this chapter, we give a brief introduction to machine learning, describe the different categories of learning algorithms, state the difference between off-line and on-line learning and give a summary on the algorithms which are relevant for the rest of this thesis.

2.1 Introduction

The ambitious goal of machine learning is to enable machines to "learn". Machine learning is a subcategory of the field of artificial intelligence. Simon [122] deals with the question "Why should machines learn" and came up with the following definition of learning:

> *Learning denotes changes in the system that are adaptive in the sense that they enable the system to do the same task or tasks drawn from the same population more efficiently and more effectively the next time.*

Thus, learning algorithms aim for training a model f that infers from a set of observed samples (training samples) to unknown samples (test samples). Depending on the training data X we can discriminate between two extreme cases of learning algorithms: *Supervised learning* and *unsupervised learning*. **Supervised learning** requires class labels for all training data $X = \{(x_1, y_1), \cdots, (x_N, y_N)\}$, i.e., training data is given as tuples of feature vectors x and labels y. The task of supervised learning is to find a function f that predicts the correct labels for all test data given the training data. Thus, the function $f : X \rightarrow Y$ maps an input space X to an output space Y. As the goal is to generalize from the training data to all unseen test data, it is also known as induction. Depending on the type of label we can discriminate between classification and regression. Classification is a mapping to a discrete class label $f : X \rightarrow Y \subset \mathbb{N}$, while regression maps to a real valued class label $f : X \rightarrow Y \subseteq \mathbb{R}$. If the number of classes is limited to two the problem is referred to as binary classification, while a larger number of classes is referred to as multi-class classification. The second extreme case of learning algorithms is **unsupervised learning**, where no labels are available for the training data $X = \{x_1, \cdots, x_N\}$ and the goal is to find a natural structure within the training data, e.g., via clustering.

A midway between both approaches is semi-supervised learning. In general, **semi-supervised learning** approaches aim for using labeled data as well as unlabeled data. The goal is to exploit the often available unlabeled data to improve the performance of the learner. More details on semi-supervised approaches and on how to incorporate unlabeled data are discussed in Section 2.5. Besides supervised, unsupervised learning and semi-supervised learning there is the category of **reinforcement learning** algorithms, which uses a teaching feedback about whether the predicted class label is correct or wrong in order to improve the performance. This kind of learning algorithms use the goodness of policies and learn from past good actions.

2.2 Off-line vs. on-line learning

The terms off-line learning, on-line learning, incremental learning or batch learning are often used differently and inconsistently in literature. Especially the terms on-line and incremental learning are often used inconsistently. To avoid this inconsistency in terminology we follow the terminology used in [63, 89]. Both, batch and off-line learning require the whole set of training data to be available in advance. The starting point is an initial model f, which is updated with all the training data until a certain stop criterion is met. The iterations are often referred to epochs. All training samples are stored and can be accessed repeatedly. Storing all training samples has the disadvantage that a large amount of memory is required. Additionally, having all training data available before training is not always possible.

In such situations on-line or incremental learning is required, since they are also applicable if the training data is not available at the beginning. The starting point is again an initial model f, but in contrast to using all training data, incremental or on-line learning uses only one training example at a time and directly performs an update of the model. In general, the model is initialized randomly. The main difference between on-line and incremental learning is that on-line learning discards the training sample after an update is performed while the training sample is kept in incremental learning. Hence, on-line learning can also be applied if the amount of training data is too large to fit into memory at once. On-line learning is always incremental, but incremental learning can be done on-line or off-line.

2.3 Supervised Ensemble Learning Algorithms

Ensemble learning algorithms are learning algorithms that combine a number of learners to one “strong learner” [36]. They are often referred to as meta-learning algorithms. The decisions of each individual classifier are combined to one joint decision. Either a weighted or an unweighted combination can be used to classify new examples. The ensemble of learners can improve the performance compared to each individual learner if each of them performs better than random guessing and if the classifiers are diverse, i.e., they make errors on different samples, which mean that the classifiers are independent. There are various different ensemble learning algorithms, like Bagging [21], Boosting [119], Wagging [17] or Random Forests [22]. An overview on ensemble based classifiers for supervised learning is given in [114]. In the following we discuss Boosting and Random Forests in detail, since they are used within the rest of this thesis.

2.3.1 Boosting and Boosting for feature selection

Boosting is an ensemble learning algorithm, which was first introduced by Schapire [119] in 1990. The initial idea was to improve the performance of a single weak classifier by training two additional weak classifiers on different versions of the input data. The "strength of weak learnability" theorem proves that combined classifiers have an improved performance compared to a single classifier. Later, Freund [52] proposed a "boost by majority" variation which simultaneously combines many weak classifiers and improves the performance compared to [119]. Even though today various different categories of boosting algorithms exists (e.g., MixtBoost [68], SemiBoost [96], etc.), the first boosting algorithm was proposed as supervised off-line learning algorithm. For an overview on different boosting algorithms we refer to [99].

The general idea of boosting is that a combination of T weak classifiers h_t to one "strong classifier"

$$H(x) = \operatorname{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right) \tag{2.1}$$

that performs better than any of the weak classifiers h_t. A weak classifier is a simple classifier. Any learning algorithm can be used, like for example a simple decision stump, which is a one-level decision tree. The only requirement for a weak classifier is to perform better than random guessing. The theorem [53] on the upper bound of the training error states that even if the training error of a weak classifier is only slightly better than random guessing, the training error of the final classifier goes to zero exponentially.

One of the most common boosting algorithms is AdaBoost [54], i.e., adaptive boosting. It was called the "best off-the shelf classifier in the world" [55] by Leo Breiman in 1996. AdaBoost is a supervised learning algorithm, where the set of training data $X = \{(x_1, y_1), \cdots, (x_N, y_N)\}$ consists of tuples (x_i, y_i), where x_i is an arbitrary feature vector and $y_i \in \{-1, +1\}$ is its corresponding class label. Each weak classifier h_t gets assigned a weight α_t, depending on its error ϵ_t. The smaller the error ϵ_t of the weak classifier h_t on the training data, the larger the weight α_t of this weak classifier. The final strong classifier H is the weighted combination of the weak classifiers h_t.

During training, each training sample x_i has a corresponding weight $D(i)$, which are forming a distribution. Initially, the weights of all training samples are equal $D_1(i) = \frac{1}{N}$. In each iteration the weights $D_t(i)$ are adapted such that they focus on hard examples, i.e., the weights of misclassified samples are increased while the weights of correct classified samples are decreased. In general, boosting can be seen as minimization of a convex loss function, where AdaBoost optimizes over an exponential loss function. The exponential loss function causes the subsequent weak classifier to focus on examples which have not been classified correct up to now and has the effect that the final strong classifier consists of complementary weak classifiers. The AdaBoost algorithm is described in Algorithm 1 and depicted in Figure 2.1.

Algorithm 1 AdaBoost algorithm [53]

```
**Require:** Labeled training data $(x_1, y_1), \cdots, (x_N, y_N)$
  Initialize weights $D_1(i) = \frac{1}{N}$
  **for** $t$ = 1 to T **do**
    Train weak classifier $h_t : X \rightarrow Y$ with small error with respect to $D_t$:
        $$\epsilon_t = \sum_{n=1}^{N} D_t(n) \cdot I(h_t(x_n) \neq y_n)$$
    Chose $\alpha_t = \ln\frac{1-\epsilon_t}{\epsilon_t}$
    Update weight distribution:
        $$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \cdot \begin{cases} \exp(-\alpha_t) & h_t(x_i) = y_i \\ \exp(\alpha_t) & h_t(x_i) \neq y_i \end{cases}$$
  **end for**
  Output the final strong classifier:
        $$H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x))$$
```

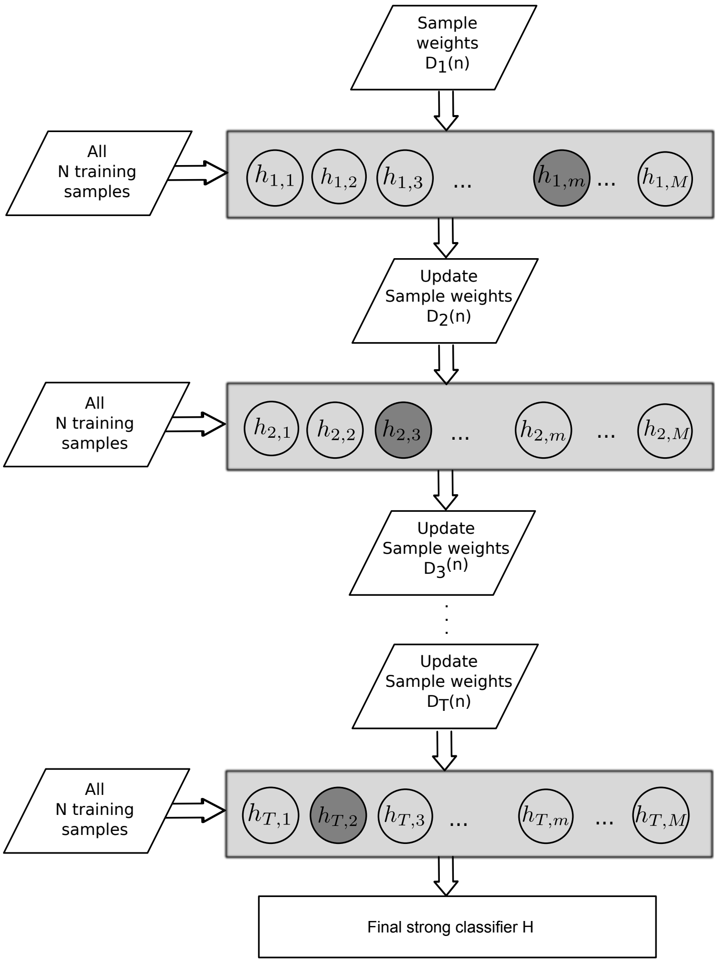

Sample weights $D_1(n)$
All N training samples
$h_{1,1}$ $h_{1,2}$ $h_{1,3}$ ... $h_{1,m}$ ... $h_{1,M}$
Update Sample weights $D_2(n)$
All N training samples
$h_{2,1}$ $h_{2,2}$ $h_{2,3}$ ... $h_{2,m}$ ... $h_{2,M}$
Update Sample weights $D_3(n)$
Update Sample weights $D_T(n)$
All N training samples
$h_{T,1}$ $h_{T,2}$ $h_{T,3}$ ... $h_{T,m}$ ... $h_{T,M}$
Final strong classifier H

Figure 2.1: Illustration of the concept of off-line AdaBoost, in each iteration all training samples are used to update a set of weak classifiers and to select the best weak classifier out of the pool of weak classifiers.

The exponential loss function makes AdaBoost prone to outliers, since a large weight is assigned to misclassified examples. Further, it does not generalize to classification problems with more than two classes, since the expected error of a randomly guessing weak classifier is $1 - 1/k$, where k is the number of classes. If k is larger than two, this requirement is hard to meet [54]. However, the interpretation of boosting as sequential minimization of the exponential loss function [55] enables to use different loss functions for boosting, which paved the way for multi-class extensions of boosting as well as for extensions to regression problems.

In the field of computer vision there are numerous applications for boosting algorithms. Tieu and Viola [131] applied boosting for feature selection, which allows for automatically selecting highly discriminative visual features. Based on 25 simple linear features (e.g., oriented edges, center surround, bar filters ...) a set of more than 46.000 features is generated by three levels of filtering. Boosting can now be applied to select highly discriminative visual features. In particular, each weak classifier corresponds to one feature and the task of boosting is to select those features with a small training error, i.e., discriminative features. Initially, AdaBoost for feature selection was applied to the task of image retrieval, where the idea is based on the assumption that each image consists of a sparse set of visual causes and that similar images share causes. Inspired by this idea, Viola and Jones [133, 134] developed a real-time object detector based on boosting for feature selection, demonstrating excellent performance on the task of face detection. The cascade structure combines classifiers with different complexity in each stage. The first stage contains the least complex classifier, which is used to reject a large number of subwindows out of all possible subwindows from the sliding window procedure. This allows for putting the focus on the most promising regions and to apply the most complex classifiers only to a small number of promising subwindows, which is beneficial for the real-time object detection. The cascade structure in combination with the efficient way for computing the Haar-like features based on an image representation called integral image is responsible for the real-time capability. The integral image *ii* is calculated for each image *i* by summing up the intensity values in the area spanning from the uppermost and leftmost position to the current position in the image by

$$ii(x,y) = \sum_{m=0}^{x} \sum_{n=0}^{y} i(m,n).$$

Thanks to this efficient representation the Haar-like features can be calculated within constant time, as the rectangular structure of the Haar-like features allows to compute them by fast lookups in the integral images. Haar-like features are well suited to describe coarse structures like edges or bars.

All algorithms described so far work off-line, which requires all training data to be available in advance, which is not always possible. Hence, on-line algorithms are required. Oza [107] introduced an on-line version of boosting. Grabner and Bischof introduced an on-line version of AdaBoost for feature selection [64]. They introduced the concept of "selectors" which group a set of weak classifiers and guide the feature selection process. Each selector contains a number J of weak classifiers $h_{t,j}$ and is repre-

Algorithm 2 On-line AdaBoost for feature selection

```
Require: Labeled training sample $(x, y)$
  Initialize all weights to $\lambda^c_{t,j} = \lambda^w_{t,j} = 1$, sample importance weight $\lambda = 1$
  for $t$ = 1 to T do
    for $j$ = 1 to J do
      Retrain weak learner $h_{t,j}$ with example $x$ and label $y$ according to $\lambda$
      if $h_{t,j}(x) = y$ then
        $\lambda^c_{t,j} = \lambda^c_{t,j} + \lambda$
      else
        $\lambda^w_{t,j} = \lambda^w_{t,j} + \lambda$
      end if
      $\epsilon_{t,j} = \dfrac{\lambda^w_{t,j}}{\lambda^c_{t,j} + \lambda^w_{t,j}}$
    end for
    $k = \underset{j}{argmin}(\epsilon_{t,j})$
    $\epsilon_t = \epsilon_{t,k}$
    $h_t = h_{t,k}$
    $\alpha_t = \frac{1}{2} \cdot ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$
    Update sample weights
        $$\lambda = \lambda \cdot \begin{cases} \frac{1}{2\cdot(1-\epsilon_t)} & h_t(x_i) = y_i \\ \frac{1}{2\cdot\epsilon_t} & h_t(x_i) \neq y_i \end{cases}$$
  end for
  Output the final strong classifier:
        $$H(x) = \text{sign}(\sum_{t=1}^{T} \alpha_t h_t(x))$$
```

sented by its best weak classifier h_t. As well established in boosting for feature selection, one weak classifier corresponds to one feature and the feature selection process is performed by selecting the best weak classifier $h_{t,k}$, i.e., the weak classifier with the smallest training error within the selector:

$$k = \underset{j}{argmin}(\epsilon_{t,j}), \tag{2.2}$$

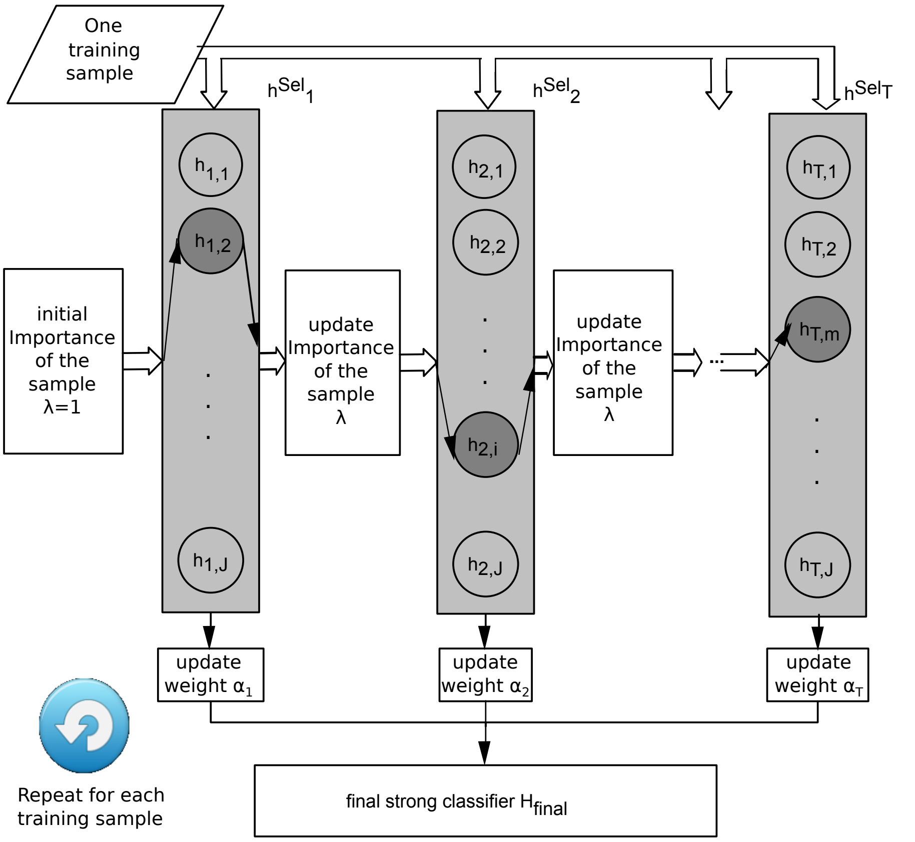

One training sample
hSel1
hSel2
hSelT
h1,1
h1,2
h1,J
h2,1
h2,2
h2,i
h2,J
hT,1
hT,2
hT,m
hT,J
initial Importance of the sample λ=1
update Importance of the sample λ
update Importance of the sample λ
update weight α1
update weight α2
update weight αT
Repeat for each training sample
final strong classifier Hfinal

Figure 2.2: Illustration of the concept of on-line boosting for feature selection, in each iteration one training sample is used to update all classifiers within the selectors, where each selector is represented by its best weak classifier.

where

$$\epsilon_{t,j} = \frac{\lambda^{w}_{t,j}}{\lambda^{c}_{t,j} + \lambda^{w}_{t,j}}. \tag{2.3}$$

Selecting the best weak classifier is equivalent to selecting the best feature for the actual task. A fast on-line adaption, which is required for adapting to changing situations, is realized by switching between different selected weak classifiers (i.e., between different features) within one selector.

To focus on hard examples and give less attention to examples already described by the actual classifier – similar to off-line AdaBoost – each sample gets assigned a weight. This weight

$$\lambda = \lambda \cdot \begin{cases} \frac{1}{2\cdot(1-\epsilon_t)} & h_t(x_i) = y_i \\ \frac{1}{2\cdot\epsilon_t} & h_t(x_i) \neq y_i \end{cases} \tag{2.4}$$

is adapted on-line depending on the error of the currently selected weak classifier while the sample is propagated through the selectors. Initially, the weight for each sample is set to 1. The sample weight is increased, if the current selector miss-classifies the example or decrease, if the current selector classifies the example correct. The combination of T selectors forms the strong classifier H. In fact, the final strong classifier is again a weighted combination of all selected weak classifiers h_t, where the weight depends on the performance of the selected weak classifier within a selector. A pseudo-code of the algorithm is given in Algorithm 2 and is illustrated in Figure 2.2.

2.3.2 Random Forests

A random forest is an ensemble learning algorithm, where a set of T randomly trained decision trees (weak learner) is combined to one strong learner. Random forests have been introduced by Ho [73] for handwritten digit recognition, where they proposed a random feature selection. Both, training and testing of the individual trees can be performed in parallel, which can be implemented efficiently. The predictions $p_t(c|\mathbf{v})$ gathered of each individual tree during testing can for an example $\mathbf{v}$ be combined by simply averaging them as shown by Breiman [22]:

$$p(c|\mathbf{v}) = \frac{1}{T}\sum_{t=1}^{T} p_t(c|\mathbf{v}).$$

Decision trees are supervised learning algorithms consisting of a set of hierarchically arranged nodes used for making decisions. We follow the terminology in [8] and for simplicity also focus on binary decision trees. Depending on the position within the tree one can distinguish between split nodes and leaf nodes. An example of a decision tree for deciding whether an input image shows an indoor or an outdoor scene is shown in Figure 2.3. A split node j has two outgoing edges and is responsible for splitting the data $\mathbf{v}$ according to a splitting criterion

$$h(\mathbf{v}, \theta_j) \in \{0, 1\}$$

in order to simplify the problem, where 0 indicates false, i.e. such training samples are sent to the left and 1 indicates true, i.e. such training samples are sent to the right. Thus, each split node can be considered as weak learner h with parameters $\theta = (\phi, \psi, \tau)$. ψ defines the data separation primitive. In the example given in Figure 2.3 this is the question "Is top part blue?". τ is the threshold for the binary test and θ defines the filter function for selecting features out of all possible features. During training, the goal is to find the optimal parameters θ^* for a split node that maximize the information gain I_j:

$$\theta^* = \arg\max_{\theta} I_j$$

with

$$I_j = I(\mathcal{S}_j, \mathcal{S}_j^L, \mathcal{S}_j^R, \theta_j),$$

where $\mathcal{S}_j$ is the set of all training samples arriving in node j and $\mathcal{S}_j^L$ and $\mathcal{S}_j^R$ are all training samples sent to the left and right child of node j depending on the parameters θ_j. In this particular example $\mathcal{S}^L$ contains all samples where the top part is not blue while $\mathcal{S}^R$ contains all samples where the top part is blue. Depending on the predefined stopping criterion (e.g. a predefined level of depth D, too low information gain, too small number of training samples reaching a node) the training procedure stops. During testing, the leaf nodes are used for prediction. For classification, the leaf nodes contain the distribution over the classes c of all training data reaching the particular leaf node. The probabilistic leaf node predictor of the t^{th} tree is:

$$p_t(c|\mathbf{v}),$$

where c is the class label and $\mathbf{v}$ is the training sample.

There are two common ways for injecting randomness during the training stage, which are randomly sampling the training data (i.e., bagging) [22] or randomized node optimization [74]. By injecting randomness by randomized node optimization a small subset $\mathcal{T}_j$ of the entire set $\mathcal{T}$ all possible parameters θ is selected, and the optimal parameters θ^* are selected out of this subset $\mathcal{T}_j$.

The excellent generalization capability gained through the injection of the randomness during training makes random forests very popular. As described in [8], the testing accuracy increases monotonically with an increased number of random decision trees in the ensemble, while too deep trees can lead to overfitting. Additionally, random forests

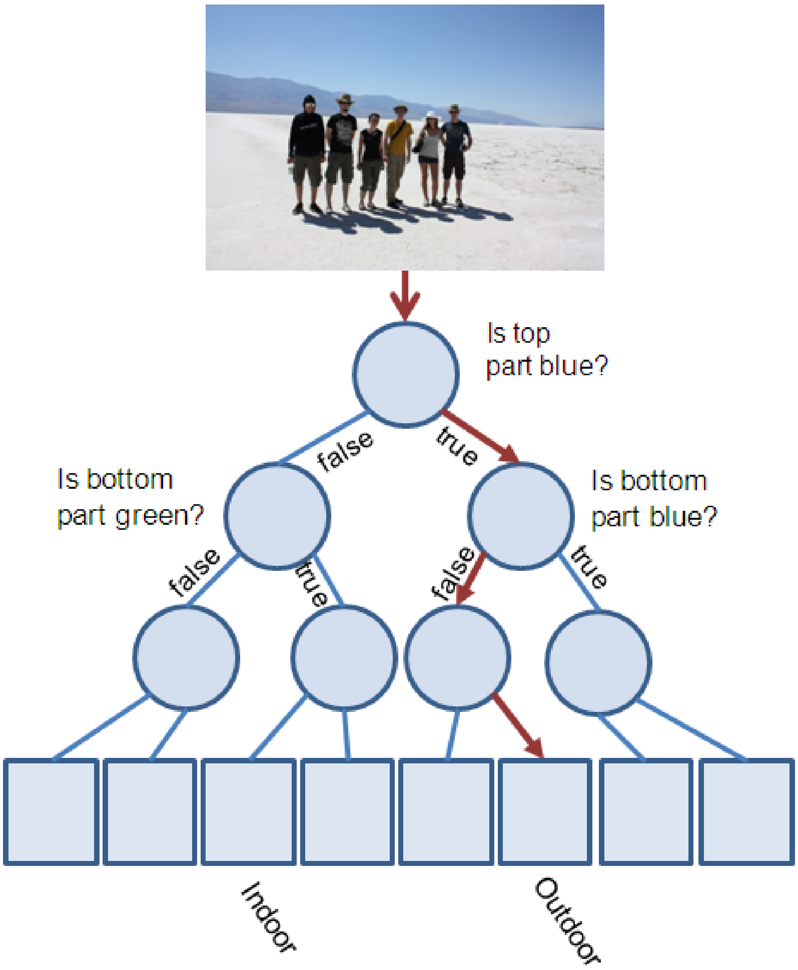


Figure 2.3: Illustration of a decision tree which is used to figure out whether an image shows an indoor or outdoor scene, based on [8].

are less prone to noisy training data. One major advantage of random forests compared to other ensemble learning algorithm like boosting is the inherent multi-class capability. For more details on random forests see the extensive summary given in [8, 30].

2.4 Multiple Instance Learning

Multiple instance learning is a variation of supervised learning. Dietterich et al. [37] introduced the concept of multiple instance learning (MIL) motivated by a drug activity prediction problem. Multiple instance learning is a machine learning paradigm for dealing with ambiguously labeled data. Since there is a huge amount of problems which have to deal with ambiguously labeled data, there has been a considerable interest in multiple instance learning and various different approaches have been proposed.

In contrast to supervised learning algorithms, where each training sample (instance) is provided a label, in multiple instance learning the training samples are assembled to so called bags $B_i \subset \mathbb{R}^d, i = 1, \dots, N$, where each bag B_i consists of an arbitrary number m_i of instances $B_i = \{x_{1i}, x_{2i}, \dots, x_{m_i i}\}$. Negative bags B_i^- are required to solely consist of negative instances, whereas for positive bags B_i^+ it has only to be guaranteed that they contain at least one positive instance. There are no further restrictions to the non-positive instances within the positive bag B_i^+, they might not even belong to the negative class. This can be written formally as

$$y_i = \underset{j}{max}(y_{ij}),$$

where y_{ij} are the instance labels within one bag.

The task of multiple instance learning is now to learn either a bag classifier $f : B \to \{0, 1\}$ or an instance classifier $f : x \to \{0, 1\}$. However, a bag classifier can follow automatically from instance prediction, e.g., by using the *max* operator over posterior probabilities over the instances p_{ij} within the i^{th} bag:

$$p_i = \max_j \{p_{ij}\}.$$

There are various different multiple instance learning algorithms based on popular supervised learning algorithms such as SVM [6] or boosting [135], which are adopted allowing for incorporating the MIL constraints. Babenko et al. [11, 12] proposed the first on-line multiple instance learning algorithm based on MILBoost [136] and on-line AdaBoost [107] for the task of object tracking, i.e. tracking-by-detection. Since we build on this algorithm, we will describe it in more detail.

In general, tracking-by-detection approaches learn a discriminative classifier to discriminate the object from the background. The prediction of the classifier is directly used to update the classifier and select positive (object) and negative (background) training samples. However, if the prediction is not precise enough, the object model is updated with suboptimal samples, which may lead to drifting. This problem has already been addressed by Viola et al. [136] for the task of object detection, where an off-line multiple instance boosting algorithm was proposed. They showed that a weaker labeling in combination with a multiple instance setting outperforms supervised learning. As common for multiple instance learning, the samples are grouped into bags X_i which have a label y_i assigned. For object detection these bags contain a set of positive samples around the object of interest, where only the center of the object is marked and samples around this center are cropped. The algorithm itself handles the ambiguity and figures out which instance is the correct positive sample. The same arguments can be used for object tracking. However, an on-line algorithm is required. The on-line multiple instance learning algorithm is shown in Algorithm 3.

Algorithm 3 On-line MILBoost

```
Require: Labeled training bags $\{X_i, y_i\}_{i=1}^{N}$, where $X_i = \{x_{i1}, x_{i2}, \cdots\}$ and $y_i \in \{0, 1\}$
  Update all $M$ weak classifiers $h_m$ in the pool with all training data $x_{ij}, y_i$
  Initialize $H_{ij} = 0$ for all $i, j$
  for $k$ = 1 to K do
    for $m$ = 1 to M do
      $p_{ij}^m = \sigma(H_{ij} + h_m(x_{ij}))$
      $p_i^m = 1 - \prod_j (1 - p_{ij}^m)$
      $\mathcal{L}^m = \sum_i (y_i \log(p_i^m) + (1 - y_i)\log(1 - p_i^m))$
    end for
    $m^* = \underset{m}{\operatorname{argmax}}(x)$
    $h_k = h_m^*$
  end for
  Output the final strong classifier:
      $H(x) = (\sum_{k=1}^{K} h_k(x))$
  where
```

All instances x_{ij} of a bag X_i are used to update the weak classifiers. The instance probabilities $p(y|x)$ are modeled as

$$p(y|x) = \sigma(H(x)),$$

where

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

is the sigmoid function. The bag probabilities are modeled using the Noisy-OR model

$$p(y_i|X_i) = 1 - \prod_j (1 - p(y_i|x_{ij})).$$

The weights of the weak classifiers are absorbed in the weak classifier. Hence, they return real values. The best weak classifier is selected by maximizing the log likelihood of the bags

$$\mathcal{L} = \sum_i (\log p(y_i|X_i).$$

2.5 How can we incorporate unlabeled data?

Labeling data is always a time-consuming and cost-consuming, whereas countless unlabeled data is cheaply available. Therefore, more and more emphasis in machine learning is placed on how to also exploit the information given by the huge amount of unlabeled data. There are various reasons why labeled data is hard to get. Beside the fact that annotating is an annoying task, often experts are required to provide the labels for the data.

One way to incorporate unlabeled information is to use semi-supervised learning algorithms. Semi-supervised learning approaches use both, labeled and unlabeled data. In general, the considered amount of labeled data is significantly smaller than the amount of unlabeled data. Another possible form of semi-supervised learning is learning by using hints, where one has additional constraints on the unlabeled data (e.g., all of these examples have the same label) [2]. A related concept is transductive learning, initially introduced by Gammerman et al. [59]. Given a set of labeled training data and a set of unlabeled test data, the goal is to learn a prediction for a specific set of test data, whereas the goal of inductive learning is to learn a generic prediction valid for all test sets, as for supervised learning. Gammerman et al. described transduction as a more specific problem than induction, due to the focus on solving a particular problem, i.e., learning a model for one particular test set. Transduction is inference from particular to particular, i.e., the prediction from a particular set of labeled training data to a particular test set. In contrast, induction is an inference from particular to general, i.e., the prediction from a particular set which should generalize for all future test data. Self-training can be seen as the simplest method of semi-supervised learning. It can be considered as a wrapper method or concept, where every learning algorithm can be used as a classifier. Self-training is an iterative process, where a classifier is trained on a set of labeled data and the classifier is directly used to predict the class label of the unlabeled data. In the next iteration, the predicted class labels are used to re-train the classifier. There are various variations of self-training. It is possible to add those examples with the most confident labels, add all labeled data or weight each sample with a certain confidence. The main problem of self-training is that mistakes (i.e., wrong class labels) reinforce themselves and may lead to a completely degenerated classifier. Another method of semi-supervised learning is co-training [20], which is a multi-view algorithm. The main idea is to train two initial classifiers h_1 and h_2 on some labeled data $\mathcal{D}^L$ and then let these classifiers teach each other using the unlabeled data set $\mathcal{D}^U$. Co-training exploits the redundancy of unlabeled input data. An update is performed if one classifier is confident on a sample whereas the other one is not. Since Abney [1] showed that co-training classifiers aim for minimizing the error on the labeled samples while increasing the agreement on the unlabeled data, it is clear that the unlabeled data can help to improve the margin of the classifiers and to decrease the generalization error.

One way for dealing with ambiguously labeled data is multiple instance learning, where training samples are grouped into bags. Instead of requiring a label for each instance as common for supervised learning only one label per bag is required. As shown by Zhou and Xu [146] multiple instance learning can also be considered as special case of semi-supervised learning. The constraint of multiple instance learning that a negative bag solely consist of negative samples entails that all negative samples are labeled, since they have to be negative samples. The constraint on the positive bag, which has to contain at least one positive sample, provides no information about a particular training sample within the bag. Hence, the samples within the positive bag can be considered as unlabeled samples enforced with the positive constraint that at least one sample within the bag is positive. Based on this observation they developed MissSVM, a multiple instance learning by semi-supervised Support Vector Machine, which handles multiple-instance learning problems by exploiting semi-supervised learning techniques.

In computer vision both, self-training as well as co-training are popular. Self-training is very popular for tracking-by-detection (e.g., [65, 12, 10]), where the output of the classifier (i.e., the image patch at the predicted object location) is directly used to update the classifier in order to adapt the tracker. The concept of co-training has been introduced to the field of computer vision by Levin et al. [91] for the task of object detection. They proposed a different way to train scene-specific classifiers by exploiting information of unlabeled data from a scene. In particular, starting with a small number of hand-labeled samples they generated additional labeled examples by applying co-training of two boosted off-line classifiers. One is trained directly from gray-value images whereas the other one is trained from background subtracted images. The additional labels are generated based on confidence-rated predictions. Using the additionally labeled samples the training process is started again from scratch. In this way better classifiers can be obtained. Besides the problem of object detection co-training is applied to a variety of different tasks in the field of computer vision including background modeling [147] or tracking [94]. Since the approach of Levin et al. [91] is based on off-line classifiers, it is not suitable for an adaptive real-world detection system. However, since on-line boosting has become popular for visual learning (e.g., [64, 11, 90]), having an initial classifier of sufficient accuracy the off-line classifier can easily be replaced by an on-line method still preserving the required properties. In fact, Liu et al. [94] give a proof for error bounds for on-line boosting in co-training. Compared to self-training co-training is less sensitive to errors, but often it's hard to find completely independent views. However, the originally strong condition of conditionally independent classifiers was later relaxed by several authors (e.g., [13, 1]). Wang and Zhou [139] provided a PAC-style proof that co-training can converge to good accuracy even if the classifiers are strong and highly uncorrelated. For more information on semi-supervised learning see [28, 148].

If the domains, tasks or distribution of the training data and the test data are different, transfer learning algorithms have to be used. Transfer learning is motivated by human learning behavior and the fact that people can exploit previously learned knowledge to solve new problems.

Nothing is particularly hard if you divide it into small jobs.

Henry Ford

3 Robust Scene Adaption from Single Stationary Cameras

Contents

Very often, the environmental conditions are changing over time. One possible way for object detection in real-world scenarios is to robustly incorporate unlabeled information from the scene.

3.1 Introduction and Problem Statement

With the increasing number of surveillance cameras the need for autonomous visual surveillance systems is increasing tremendously. One of the first steps towards autonomous visual surveillance is object detection. The main focus of this chapter is on object detection from static cameras with specific emphasis on the applicability to real-world environments. To deal with changing environmental conditions which usually occur in real-world environments an adaptive object detector is required. To ensure robust object detection without the need for human intervention we develop different approaches which allow for robustly incorporating scene specific information.

In the following, we first describe the problem of object detection, the main challenges occurring and different concepts for object detection. Related work which deals with scene specific unlabeled information is described in Section 3.2. Then, we describe different approaches to robustly incorporate unlabeled scene specific background as well as foreground information. The robustness of these approaches is demonstrated empirically in Section 3.7. Finally, we summarize our approaches in Section 3.8.

3.1.1 What is object detection and what are the challenges?

Before talking about object detection we have to clarify the term object recognition and place object detection in this context. The goal of object recognition is to learn visual categories and to find new instances of these categories in images. One has to distinguish between two different types of object recognition, the specific case, where one is interested in identifying one specific object (e.g., a specific person like Albert Einstein) and the generic case, where one is interested in identifying one object of a certain category (e.g., cars, pedestrians, ...). While object recognition refers to classification among objects in a particular isolated region of the image, object detection is the more general task of *localizing* the object of interest in an image [5]. In this section, we particularly address the latter case, where the goal is to learn a model for a category of objects and to localize the object within the image.

The task of object detection can be divided into three consecutive steps [69]. The first step is to find a representation and a suitable model to describe the object category. The second step is to find evidence supporting the object model. The third step is to find the objects of interest by suppressing redundant information found by the object model.

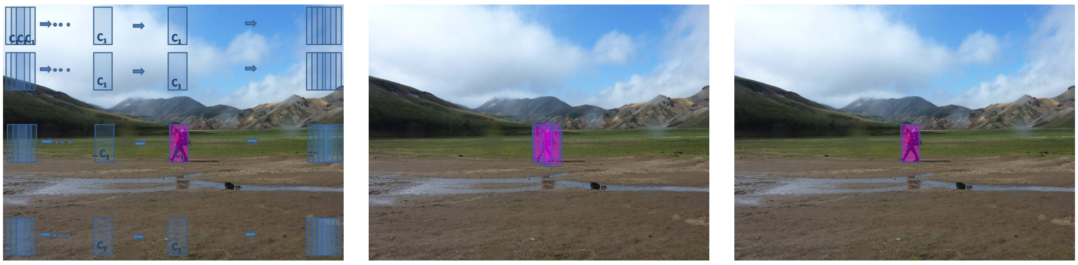
Object detection entails various challenges, like different and/or changing illumination conditions, changing backgrounds, occlusions or cluttered scenes, where it is even hard for a human to detect the object of interest. Beside changing external influences like varying illumination or moving backgrounds, there are challenges caused by deformations of the object itself, like different object poses, object distortions or viewpoint changes. Our main goal is to detect the object of interest in all possible scenarios despite all these challenges. A generic model has to capture all poses, deformations and viewpoint variations of the object as well as all changes of environmental conditions. Imagine the task of face detection, for example identifying all images containing faces from a private photo collection. A common photo collection contains photos from various different activities, including miscellaneous spare time activities, different kinds of events or various holiday trips. Hence, there will be a huge variability within the data. The variability mainly arises from considerably varying background. Very often the variability within the object class is by far less than the variability within anything but the object. Most of the variability within the object class arises from deformations of the object, which is often depending on the class of object. For the class of faces only slight deformations are possible. To be able to deal with this huge variability within the data, general models require a huge set of labeled training data. Describing all possible scenarios usually comes with a tremendous labeling effort. Even though having a large set of training data, such general models suffer from the problem of not being specific enough, which leads to a large number of false alarms. Because of this, instead of building on a generic model throughout this thesis we concentrate on scene specific models, which are able to adapt to the scene by focusing on varying backgrounds as well as to incorporate scene specific object information.

3.1.2 How can object detection be realized?

Object detection approaches can be classified into window-based approaches, which describe the appearance within a local region, and part-based approaches, which use the geometric structure in combination with the appearance of local parts for describing the object of interest [69].

A prominent window-based approach is the sliding window technique illustrated in Figure 3.1(a). Window-based approaches are evaluated a model on highly overlapping patches with different scales on the whole test image. As already mentioned, this model can either describe one specific object (e.g., a specific person) or a class or category of objects (e.g., pedestrians, cars, faces, etc.). Models describing one specific object can for example be subspace-based models like Eigenfaces [132], where a set of "Eigenfaces" (eigenvectors of a set of faces) project characteristics of individual faces and the weights of these eigenface features can be used to recognize a particular face. Models describing a class or category of objects are for example bag-of-words descriptors [32]. The bag-of-words idea was originally used for natural language processing and information retrieval and later on applied to the field of computer vision. Images are represented by so called visual words. Representative visual words for a specific object category can be learned. Other examples for models describing a class of objects are learned classifiers like (e.g. AdaBoost [54], Support Vector Machines [29]) based on local features (e.g. HOG descriptor histograms in combination with Support Vector Machines [33], boosted Haar-cascades [134]).

After evaluating these models at each image location one gets a set of possible object locations. This dense evaluation on a set of highly overlapping patches results in a set of patches located around the actual object position which gives a positive response for the same object of interest as shown in Figure 3.1(b). To avoid multiple detections for one object of interest non-maxima suppression has to be applied. Even though a detailed discussion on non-maximum suppression is often lacking and sentences like "a standard non-maximum suppression is applied" are still common, this crucial point has recently become of more and more interest [16, 35, 113]. After performing the non-maximum suppression one gets the locations of the objects of interest as shown in Figure 3.1(c).



(a) Sliding window technique: one model is tested at each location within the image, often an evaluation with different scales is necessary.

(b) Dense evaluation results in a large number of possible locations for the object of interest.

(c) Result after non-maxima suppression: one single bounding box for object of interest.

Figure 3.1: Sliding window for object detection.

In contrast to window-based models, which aim for describing an object by the whole image patch, part-based models describe the object by an assembly of parts with relations to each other. A very popular part-based model is the pictorial structures model introduced by Fischler and Elschlager [48]. The pictorial structures model for a face is visualized in Figure 3.2. In contrast to bag-of-words models, which in general discard the spatial information, pictorial structures models combine appearance information with spatial information. Burl et al. [26] introduced constellation models, where an object is represented by a constellation of local features. However, this model is not well suited for articulated objects, since all parts are constrained with respect to a central coordinate system, which cannot capture multiple articulations. Pictorial structures models are well-established for detecting people in images as well as for pose estimation (e.g., [46, 7, 39, 149]). Felzenszwalb et al. [46] propose the deformable part model (DPM), which uses a star-structured part-based model, where one filter is defined as root filter and additionally a set of part filters with associated deformation models are defined. The model parameters are learned by a latent SVM. An extension to the deformable part model as a cascaded classifier with a significant speedup was proposed in [47]. Andriluka et al. [7] propose a framework based on pictorial structures, where the appearance of each part is described by shape context descriptors and an AdaBoost classifier is used to train discriminative part classifiers. The deformations are described by a kinematic tree prior. Eichner and Ferrari [39] propose a framework for multi-person detection based on pictorial structures, where interactions between people are modeled as occlusions in a multi-person pictorial structures model which allows for a better pose estimation in images with groups of people. Recently, Zuffi et al. [149] propose a deformable structures model as an extension to the pictorial structures model by representing each part as a deformable contour instead of a rigid template as proposed in the pictorial structures model. The contours are learned using Principal Component Analysis (PCA) in a low-dimensional linear subspace.

3.2 Related Work

Since the work described within this section deals with the topic of incorporating unlabeled information within the field of object detection, we give an overview on related work in this context with the focus on how to adapt to a specific scene and how to incorporate unlabeled information.

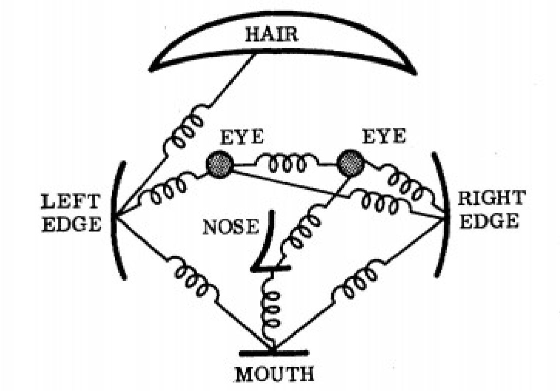


Figure 3.2: Pictorial structures model for a face (taken from [48])

Recently, a number of surveys on pedestrian detection appeared [60] [38] [40]. Both, Geróandnimo et al. [60] and Enzweiler and Gavrila [40] focused on pedestrian detection on a data set captured on-board a vehicle driving through urban environment. In contrast, Dollár et al. [38] performed an extensive evaluation of 16 different pedestrian detectors, analyzing the size, position and the level of occlusion within different datasets. They showed that the performance of state-of-the-art generic pedestrian detectors significantly drops when they are applied to video sequences of different scenes, which clearly indicates the need of adaptive or scene specific detectors instead of generic detectors.

Stalder et al. [124] proposed an approach for incorporating scene specific information into an object detection and tracking framework by introducing a cascaded confidence filter, which exploits constraints like the size of the objects, the background of the scene as well as the smoothness of the trajectories. Sharma et al. [120] proposed an unsupervised incremental multiple instance learning approach for incorporating unlabeled scene specific information. They incorporate a MIL loss function to Real AdaBoost in order to incorporate noise samples and collected samples from the scene by focusing on missed detections and false alarms by exploiting the tracking information. In contrast to [124], [120] does not require static cameras since they solely rely on tracking.

Recently, Wang et al. [138] proposed a transfer learning framework for an automatic scene adaption for the task of pedestrian detection. Starting with the histogram of oriented gradients (HOG) based detector of Dalal and Triggs [33] a set of positive training samples is extracted from the scene by evaluating the generic detector on the video sequence. This gives the data from the target domain. The training data from the source domain is re-weighted according to the similarity to the data from the target domain by using a graphical representation which increases the weight of samples from the source domain, which are more similar to the target domain since they are taken under similar view points, lighting conditions or resolutions or negative samples which contain the same objects, like trees, streets, etc.. Additionally, contextual information like motion, scene structure and scene geometry (similar to [137]) are incorporated to compute confidence scores of target samples in the transfer learning setup. The objective function encoded in the proposed Confidence-Encoded SVM aims for assigning similar weights to samples with similar appearance, small weights to samples where the context information contradicts the appearance information and includes a term for regularizing the contextual information. The training step is repeated until convergence, which is claimed to be reached after a small number of iterations.

Another work targeting the disparities between the distributions of training data and test data was introduced by Pang et al. [108]. The goal was to transfer a generic boosted detector towards different viewpoints and scenes. As transfer learning algorithm they use Covariate Boost (CovBoost) which is incorporated into the cascaded detector of Viola and Jones [134]. used to select features appropriate for the task of interest. To adapt to different viewpoint a new feature pool is generated by using feature shift, where CovBoost is used to select features suitable for the new viewpoint. To transfer the generic detector to a particular scene without viewpoint changes new classifiers trained by CovBoost on training data from the target domain are appended to the initially trained classifier.

Joachims [80] introduced transductive support vector machines (TSVM) for text classification, which allow for incorporating unlabeled data to the training process. Another work on incorporating unlabeled information and on exploiting information from the target domain is based on a combination of Expectation-Maximization with a naive Bayes classifier [102]. They showed that using unlabeled data, which can be acquired very cheap, significantly improves the classification results.

Pishchulin et al. [110] tackled the problem of acquiring labeled data by proposing to use 3D shape models from computer graphics to generate virtual examples of training data that can be used for object detection.

Li et al. [93] proposed to use scene specific knowledge by exploiting the information of the camera orientation to perform object detection in the 3D world space instead of the 2D image space.

3.3 Classifier grids

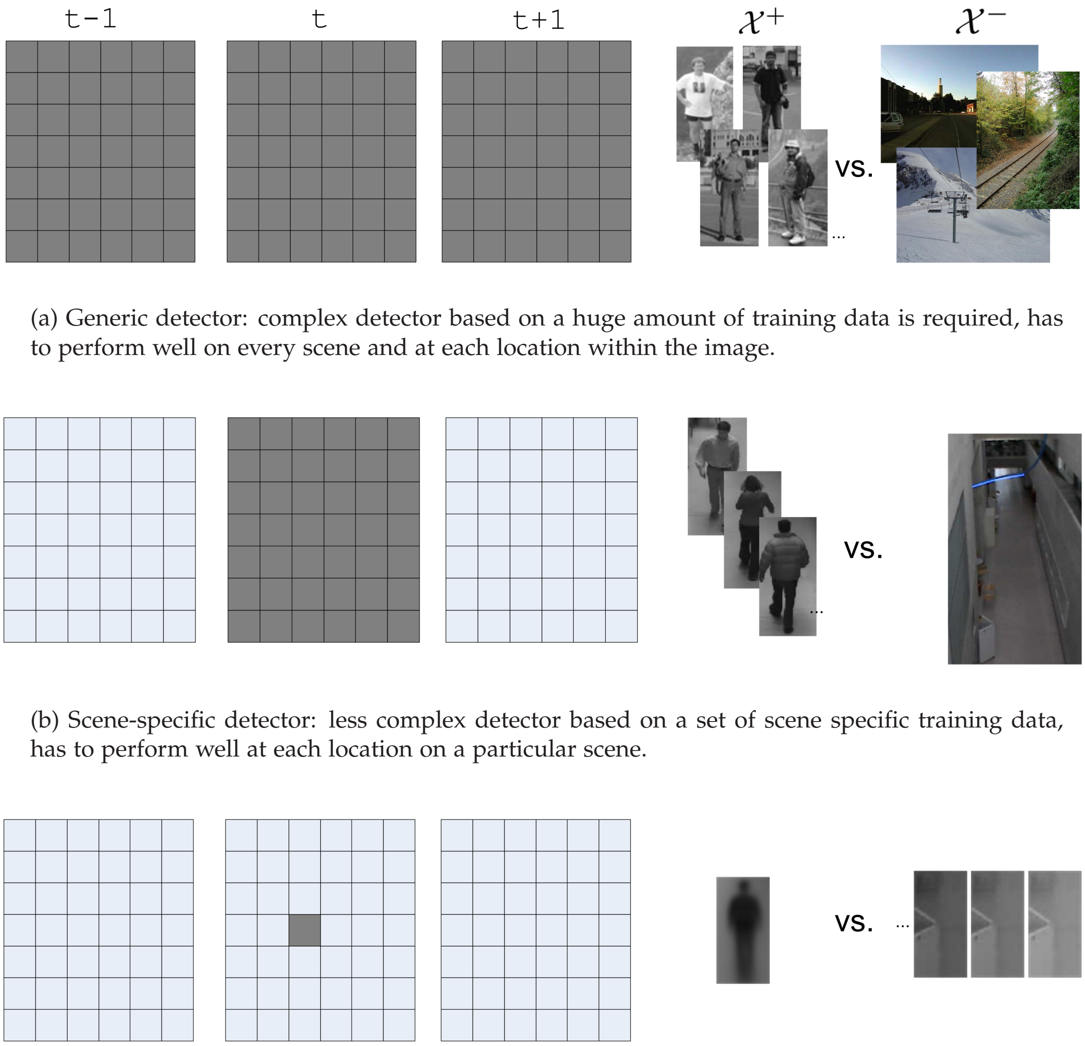
The main challenge of adaptive object detectors is to incorporate scene specific unlabeled information, which allows for adapting the detector to new environmental conditions in a robust manner without human intervention.

3.3.1 What are classifier grids?

A common way for object detection is to apply the sliding window technique (see Section 3.1.2), where a generic classifier is evaluated at each position within the image, as illustrated in Figure 3.3(a). A huge amount of training data is necessary to train such a generic classifier suitable for dealing with all possible variations within the object class as well as all possible variations within the background class. The expensive labeling makes it is hard to get a huge set of labeled training data. Besides huge efforts for acquiring labeled data the major problem of generic classifiers is that they are often not specific enough, resulting in false alarms (false positives) as well as missed detections. This is illustrated in the first row of Figure 3.5, where the results of a well-established generic object detector (i.e., the histograms of oriented gradients based human detector [33]) are illustrated. This detector is applied in a sliding window manner on a typical surveillance sequence, monitoring an indoor scenario of a corridor in a public building.

The number of false positives as well as the number of required training samples can be significantly reduced, if the environment is known in advance. Considering a typical surveillance scenario with static cameras, this information is available beforehand. Having this prior knowledge allows for training a scene specific model. This reduces the complexity of the problem, since the structure of the scene as well as the structure of the object of interest is given in advance, as shown in Figure 3.3(b). The reduced complexity of the problem allows for training a less complex model (i.e., a smaller and more efficient classifier), still able to solve the task. The results of the scene specific detector are illustrated in the second row of Figure 3.5. It can clearly be seen that by using scene specific information, such as the available scale information, the number of false positives can be reduced significantly.

The goal of classifier grids is to further reduce the complexity of the task by training a separate classifier for each position within the image, as illustrated in Figure 3.3(c). Using a separate classifier for each position within the image significantly simplifies the problem. In this way, classifier grids follow the, in computer science well-established, di-



(a) Generic detector: complex detector based on a huge amount of training data is required, has to perform well on every scene and at each location within the image.

(b) Scene-specific detector: less complex detector based on a set of scene specific training data, has to perform well at each location on a particular scene.

(c) Classifier grid detector: small and compact detectors, has to perform well at one particular scene and one particular location within the image.

Figure 3.3: Overview of different concepts for object detection from static cameras and the corresponding training sets: (a) generic detector, (b) scene specific detector, and (c) classifier grid detector. The gray blocks highlight the regions in both, time and location, where the classifier has to perform well.

vide and conquer paradigm, where the problem is broken down until the sub-problems become simple enough. Afterwards, the solutions to the sub-problems are combined to solve the original problem. Classifier grids divide each input image into a highly overlapping set of grid elements (regions), where each of the grid elements corresponds to one sub-problem of the whole object detection problem which is solved by a separate classifier. This is visualized in Figure 3.4. The classifiers within the classifier grid can profit from simplifying the problem to discriminate between the object of interest and the background at one specific location within the image. The reduces variability at one specific location within the image allows for using less complex and compact on-line classifiers, which can be evaluated and updated efficiently and further reduces the number of false alarms. This is illustrated in the last row of Figure 3.5. Stationary cameras allow for incorporating known scene specific information like scale information. Hence, evaluating different scales at one particular location within the image is not necessary. The simplification of the problem gives superior results compared to generic models.

For a static camera the scene structure as well as the extent of the object of interest at each position within the image is known. There are approaches like [23], which automatically estimate the structure of the scene. Hence, in contrast to standard sliding window approaches which have to evaluate different scales at every position in the image, the use of scale information can significantly reduce the number of classifiers within the classifier grid. The number of classifiers within the classifier grid can be defined by an overlap parameter. There is always a trade-off between run-time and performance of the classifier grid object detector. The effect of the overlap parameter is shown evaluated in Section 3.7.2.

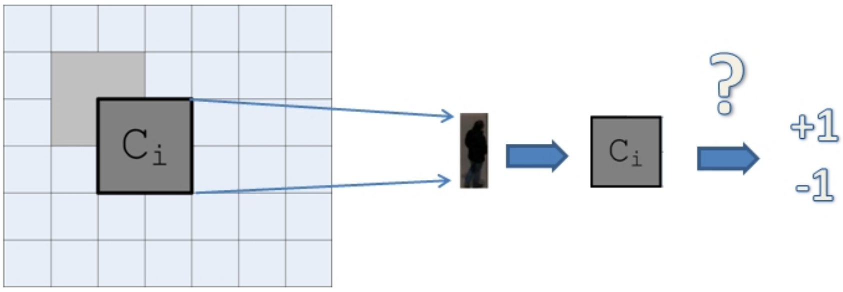
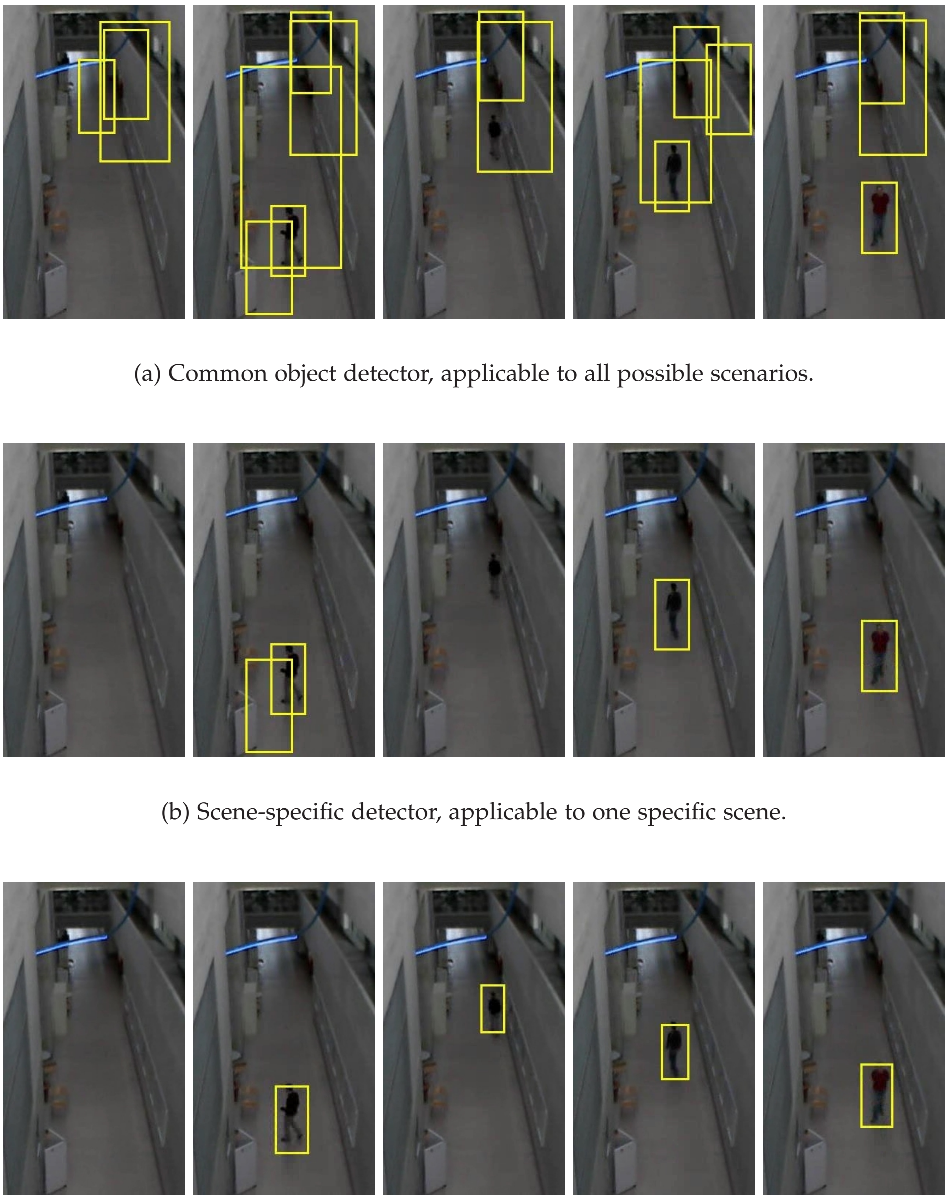


Figure 3.4: The main idea of classifier grids follows the divide and conquer principle. The image is divided into highly overlapping grid elements (regions), where each grid element has its own classifier.

3.3.2 Fixed Update Strategies

For handling changing environmental conditions, the object detector has to be adapted over time. The main challenge of an adaptive object detector is to guarantee robustness over time while incorporating new, scene specific information. In general, scene specific



(a) Common object detector, applicable to all possible scenarios.

(b) Scene-specific detector, applicable to one specific scene.

(c) Adaptive object detector, adapting to changing situation.

Figure 3.5: Since changing environmental conditions (e.g., lightning changes or changes in the background of the scene) cannot be handled by a fixed model an adaptive/scene specific system is beneficial.

information is only available in terms of unlabeled data. Hence, we need to incorporate the unlabeled information robustly. Therefore, two requirements have to be fulfilled for the classifier grid approach. First, classifiers, which are able to incorporate the new information on-the-fly, are needed. Since off-line classifiers demand all training data beforehand, on-line classifiers are required to incorporate the information during training. Originally, each classifier within the classifier grid is initialized by the same classifier, but steadily updated over time, which allows for adapting each classifier to the actual problem as well as to deal with changing environmental conditions. Second, unlabeled data has to be robustly incorporated. Typical update schemes for incorporating such unsupervised information to supervised classifiers are self-training (e.g., [115, 92]) and co-training (e.g., [20, 91]). Both of them may suffer from the drifting problem, which means that wrong class label information used to update the classifier leads to arbitrarily wrong results. To overcome the drifting problem, the initial idea of classifier grids was to use a fixed update strategy [67]. These fixed updates are based on a fixed set of hand labeled positive examples $\mathcal{X}^+$ describing the object class and an adaptive set of negative examples $\mathcal{X}^-$ describing the background class. The set of negative examples is extracted directly from the scene. Using a fixed set of hand labeled examples

$$\langle \mathbf{x}, +1 \rangle, \quad \mathbf{x} \in \mathcal{X}^+ \tag{3.1}$$

for the positive updates of the classifier, these updates are correct by definition. For updating the negative class (i.e., the background) the unlabeled information of the scene has to be exploited. This is necessary to adapt to changes within the scene. By using the images of the scene directly without any label information, one cannot guarantee correct updates of the background class. However, the probability of having an object present at one specific location within the image (i.e., at one patch of the grid $\mathbf{x}_i$) at one specific point in time is small. It can be calculated as

$$P(\mathbf{x}_i = \text{object}) = \frac{\#p_i}{\Delta t} , \tag{3.2}$$

where $\#p_i$ is the number of objects entirely present in a particular patch within the time interval Δt. Thus, negative updates with the current patch

$$\langle \mathbf{x}_{i,t}, -1 \rangle \tag{3.3}$$

are correct most of the time (wrong with probability $P(\mathbf{x}_i = \text{object})$). Hence, the classifiers need only to handle a small amount of label noise. The long-term robustness is given by the low probability of wrong updates for the background class in combination with the per definition correct updates of the object class. Even if an object is standing at a particular location for a while and the background class is degraded (short-term drifting), correct updates after the object is moving again regenerate the background class, which ensures the long-term robustness.

3.3.3 Discussion

The fixed update strategies cause three main problems. First, using a fixed pool of labeled training data leads the positive distribution of the weak classifier to converge to a fixed distribution after a large number of updates. Even if the positive information is kept fixed and the distribution is not modified over time, positive updates are required which hurts the run-time performance. Second, still wrong negative updates might arise leading to short-term drifting. This situation might emerge, if an object of interest is not moving over a larger number of frames, which cause wrong updates for the particular classifier and entails this classifier to drift until the object moves again. Third, by using a fixed pool of labeled data for the object class, no new scene specific object information can be acquired.

The first problem will be addressed in Section 3.4.2, where the main idea is to further increase the stability and to speed up the computation by a combination of two generative models in parallel: an off-line trained model for the positive (object) class and an adaptive on-line trained model for the negative (background) class. In particular, we introduce a method to link off-line and on-line boosting for feature selection. By using off-line boosting for feature selection the classifier is initialized by features well suited for the task of interest in contrast to a random initialization of an on-line classifier. Since well suited features are selected within the classifier, the classifier size can be further reduced compared to randomly initialized classifiers, where a larger classifier size is required for a good classification result. The strong positive prior inhibits fast temporal drifting, while the negative updates during run-time ensure the required adaptivity. Moreover, since the positive (object) model is kept fix, the number of required updates is reduced.

The second problem results from too many wrong updates of the background class, which may cause a foreground object to grow into the background and finally leads the detector to fail (i.e., it generates a miss). Even though the classifier recovers quickly – within a few frames (short time drifting) – this problem should be avoided. In particular, we address this problem in Section 3.5.2 by introducing the idea of Inverse Multiple Instance Learning.

The third problem is addressed in Section 3.6.1, where we propose to use a co-training approach (*Classifier Co-Grids*) in combination with a novel robust on-line learner. The robust on-line learner keeps two separate models for the positive class as well as two separate models for the negative class. For both, the positive and the negative class, one model is off-line trained and kept fixed during run-time, while the other one is adapted over time. This combination of an off-line pre-trained model with an on-line adapted model within a single learner allows for incorporating scene specific positive information (i.e., the recall can be increased), while still preserving the accuracy.

3.4 Linking Off-line and On-line Learning

In the section we describe how to combine off-line boosting for feature selection with on-line boosting for feature selection to allow for combining prior information from labeled data with new information, which is not available at off-line training time.

3.4.1 Linking off-line and on-line AdaBoost for feature selection

There are many problems in computer vision, where not all training data is available before training. All these problems require on-line learning algorithms. An on-line algorithm showing good results for object detection and object tracking is on-line boosting for feature selection [64]. The feature selection process allows for handling changing situations efficiently by switching between different features and choosing the features most suitable for the actual task. In general, on-line classifiers like on-line boosting for features selection are initialized randomly from the set of all possible weak classifiers (i.e., randomly drawn from the set of all possible features).

However, if the problem is known in advance it is possible to use a suitable representation describing the actual problem, i.e., features that are suitable for the particular task. Using this prior information, which is often available, can improve the results. Originally, on-line boosting for feature selection initializes the selectors with random features. In order to exploit the often available prior information, we propose to link off-line and on-line boosting for feature selection. Off-line boosting for feature selection allows for initializing the classifier with features suitable for the actual task. Therefore, off-line boosting for feature selection needs to be modified. Originally, in each iteration off-line boosting for feature selection selects one weak classifier, i.e., one particular feature. To allow a subsequent on-line boosting for feature selection, a set of J weak classifiers has to be selected in each boosting iteration, where J is the number of weak classifiers within one selector. This allows for incorporating prior information about the object class. The huge pool of randomly initialized weak classifiers may contain very similar features. In general, similar features give a similar training error, which is the criterion for selecting the weak classifiers. This does not influence off-line boosting for feature selection, since only the best weak classifier is selected in each iteration. However, to allow for a subsequent on-line boosting for feature selection, instead of a single best weak classifier we select the best J weak classifiers in each boosting iteration. To avoid having too similar features within one selector, an additional selection criterion measuring the similarity between features has to be introduced. Too similar features within one selector would hinder the adaptivity of the classifier during on-line updates. Hence, we have to introduce a similarity criterion based on the overlap between the features and the feature types. The overlap criterion considering the spatial position and extend of features within the patch. Features with an overlap larger than a specified threshold are only allowed if they have distinctive feature types, i.e., horizontal vs. vertical or diagonal feature types. The avoidance of too similar features within a selector ensures the adaptivity of the strong classifier, which is required for an on-line adaption. Using off-line boosting for feature selection to select features appropriate for a specific task allows for using less complex classifiers to solve the same problem, since the features within a classifier are well suited for the particular problem. The modified off-line boosting for feature selection algorithm is described in Algorithm 4.

To keep the information from the off-line boosting for feature selection step we have to slightly modify the on-line boosting for feature selection algorithm as shown in Algorithm 5. Here, the error calculation has to be modified. We now have to calculate a combined error based on the off-line error as well as the on-line error. The linkage between off-line and on-line boosting for feature selection is illustrated in Figure 3.6.

3.4.2 Application of Linked off-line and on-line learning to classifier grids

Based on the promising results of on-line boosting for feature selection on various object detection and object tracking tasks [64, 66] and the fast adaptivity to changing situations we choose on-line boosting for feature selection as learner for the classifier grid approach. The fixed update strategies described in Section 3.3.2 are based on a fixed set of hand-labeled samples used for the positive updates to describe the appearance of the object of interest and on samples extracted on the fly from the scene to model the back-

Algorithm 4 Modified off-line AdaBoost for features selection algorithm for linking off-line and on-line AdaBoost for feature selection.

```
Require: Labeled training data $(x_1, y_1), \cdots, (x_N, y_N)$
  Initialize weights $D_1(i) = \frac{1}{N}$
  for $t$ = 1 to T do
    For each feature $j$ train one weak classifier $h_j : X \rightarrow Y$ with error with respect to $D_t$:
      $$\epsilon_j^{\text{off-line}} = \sum_{n=1}^{N} D_t(n) \cdot I(h_j(x_n) \neq y_n)$$
    Select best $J$ weak classifiers to initialize selector $t$ with proper features
    Chose $\alpha_t = \ln \frac{1 - \epsilon_t^{\text{off-line}}}{\epsilon_t^{\text{off-line}}}$
    Update weight distribution:
      $$D_{t+1}(n) = \frac{D_t(n)}{Z_t} \times \begin{cases} \exp(-\alpha_t) & h_j(x_n) = y_n \\ \exp(\alpha_t) & h_j(x_n) \neq y_n \end{cases} \quad (3.4)$$
  end for
  Output the final strong classifier:
      $$H(x) = \text{sign}(\sum_{t=1}^{T} \alpha_t h_t(x))$$
```

ground. The positive updates are correct by definition, since they are taken from a finite set X^+ of hand-labeled positive samples. Thus, for each feature $f_j \in \mathcal{F}$, where $\mathcal{F}$ is the full feature space, a generative model D_j^+ can be estimated. By drawing from a fixed set of hand-labeled samples X^+ for the positive updates, the positive distributions D_j^+ are not changing over time and can be calculated in an off-line manner, since all information is given in advance. This allows for neglecting these updates during the on-line scene adaption and results in a fixed distribution for the object class (positive class) D_j^+. If this step is performed by our modified off-line boosting for feature selection algorithm (Algorithm 4), we can exploit the advantage of having a classifier which consists only

Algorithm 5 Modified on-line AdaBoost for features selection algorithm for linking off-line and on-line AdaBoost for feature selection.

```
Require: Labeled training sample $(x, y)$
  Initialize all weights to $\lambda^c_{t,j} = \lambda^w_{t,j} = 1$, sample importance weight $\lambda = 1$
  for $t$ = 1 to T do
    for $j$ = 1 to J do
      Retrain weak learner $h_{t,j}$ with example $x$ and label $y$ according to $\lambda$
      if $h_{t,j}(x) = y$ then
        $\lambda^c_{t,j} = \lambda^c_{t,j} + \lambda$
      else
        $\lambda^w_{t,j} = \lambda^w_{t,j} + \lambda$
      end if
      $\epsilon^{\text{on-line}}_{t,j} = \frac{\lambda^w_{t,j}}{\lambda^c_{t,j} + \lambda^w_{t,j}}$
      $\epsilon_{t,j} = \frac{1}{2}(\epsilon^{\text{off-line}}_{t,j} + \epsilon^{\text{on-line}}_{t,j})$
    end for
    $k = \underset{j}{argmin}(\epsilon_{t,j})$
    $\epsilon_t = \epsilon_{t,k}$
    $h_t = h_{t,k}$
    $\alpha_t = \frac{1}{2} \cdot ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$
    Update sample weights
      $$\lambda = \lambda \times \begin{cases} \frac{1}{2\cdot(1-\epsilon_t)} & h_t(x_i) = y_i \\ \frac{1}{2\cdot\epsilon_t} & h_t(x_i) \neq y_i \end{cases}$$
  end for
  Output the final strong classifier:
      $$H(x) = \text{sign}(\sum_{t=1}^{T} \alpha_t h_t(x))$$
```

of features suitable for the task of interest, i.e., the features selected during this off-line training stage are well suited to describe the object class. Additionally, the number of updates can be reduced to the half, since positive updates are no longer required because the distribution of the positive information is calculated during off-line training.

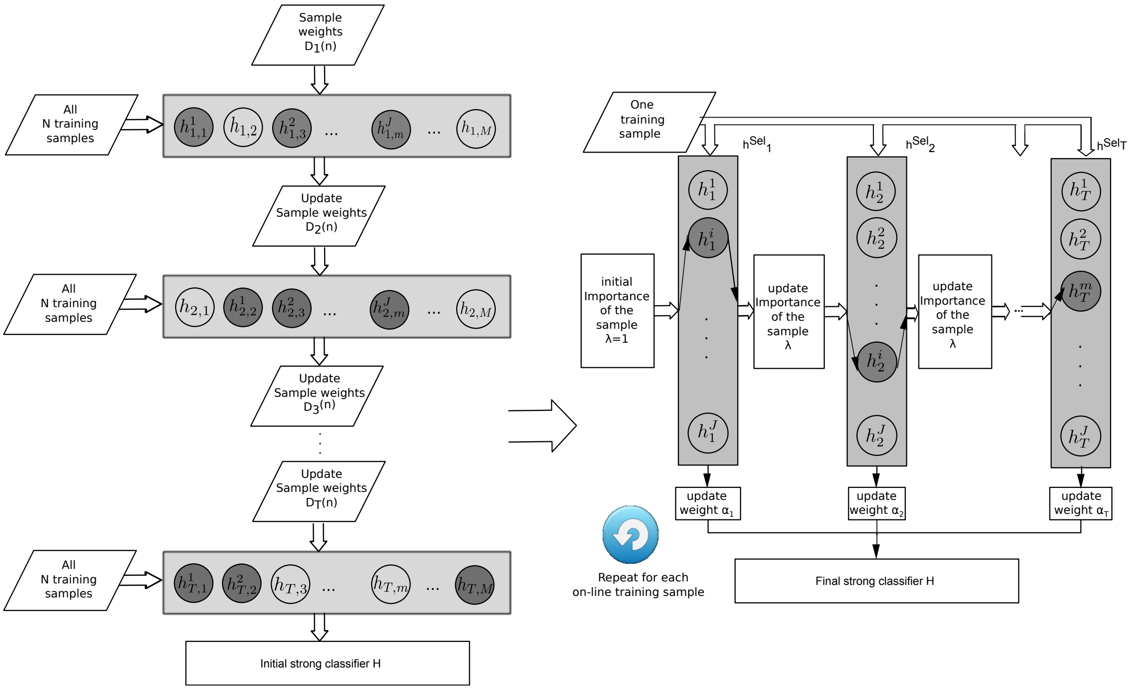

Sample weights D1(n)
All N training samples
Update Sample weights D2(n)
Update Sample weights D3(n)
Update Sample weights DT(n)
Initial strong classifier H
One training sample
hSel1
hSel2
hSelT
initial Importance of the sample λ=1
update Importance of the sample λ
update weight α1
update weight α2
update weight αT
Final strong classifier H
Repeat for each on-line training sample

Figure 3.6: Off-line on-line linkage: the off-line boosting for feature selection algorithm has to be adapted to select a set of J weak classifiers within each boosting iteration.

In order to adapt to changing environmental conditions, the negative distributions D_j^- have to be updated all the time. Therefore, the input images are directly used to perform updates of corresponding grid elements. Based on Equation (3.2) we assume that these updates are correct most of the time. Finally, in the particular case of classifier grids the discriminative classifier can be estimated by combining the two generative models D_j^+ (can be calculated off-line) and D_j^- (has to be calculated on-line) at feature level. This combination can be efficiently realized by using on-line boosting for feature selection. The overall idea is illustrated in Figure 3.7.

Since during the on-line stage of our classifier grid approach only negative updates are performed, the error of the positive samples $\epsilon_+^{\text{off-line}}$ stemming from the off-line training is kept fixed during the on-line stage while the error of the negative samples is adapted all the time. By using solely the error calculated during on-line learning, only the error for negative samples, i.e., the false positive rate can be estimated. However, the fixed distributions of the object class D_j^+ were estimated off-line. Thus, instead of Equation (2.3) we can use the combined error

$$\epsilon = \frac{1}{2}(\epsilon_+^{\text{off-line}} + \epsilon_-^{\text{on-line}}) \tag{3.5}$$

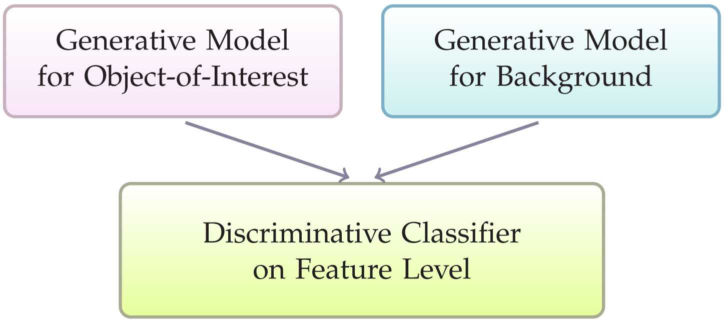

Generative Model for Object-of-Interest
Generative Model for Background
Discriminative Classifier on Feature Level

Figure 3.7: The classifier grid detector can be interpreted as a combination of two generative models, one describing the background and one describing the object of interest, which are combined to a discriminative model at feature level by linking off-line and on-line boosting.

to select the best weak classifier within the selector. Yet, this linkage of off-line and on-line learning is ideally suited for the classifier grid approach, since prior information can be exploited in the off-line stage while information available at run-time can be considered for on-line adaption.

3.5 Exploiting temporal information for ambiguously labeled samples

To avoid drifting in classifier grids a fixed update strategy was proposed as described in Section 3.3.2. The linkage of off-line and on-line learning for classifier grids described in Section 3.4.2 further allows for solely performing negative updates for a classifier, whereas the positive representation was trained off-line in advance and kept fix. These update strategies ensure "long-term" stability, i.e., the classifier cannot get totally degenerated, as shown experimentally in Section 3.7.3.3, where the classifier grids are updated over a whole week, performing 580.000 updates for each classifier in the classifier grid.

Even though the updates generated by the fixed update strategy are correct most of the time, they might be wrong causing the classifier to drift within a certain time interval, which we will refer to as "short-term" drifting. This might happen when an object stays at the same position over a long period of time. In particular, when an object is not moving over a long period of time, foreground information (the object of interest) is used to perform negative updates of the background class. Thus, the background information is temporally unlearned. Since this can be seen in the context of ambiguously labeled samples, multiple instance learning could help to deal with this problem. Hence, we address the problem of short-term drifting by incorporating temporal information and using a Multiple Instance Learning (MIL)-based approach instead of the fixed update strategy. Multiple instance learning [37] inherently copes with the problem of unreliably labeled samples. In particular, the single instances are organized in constrained bags, where a positive bag has to contain at least one positive sample and a negative bag solely consists of negative samples. This can for example be used to solve the problem of inaccurately aligned samples typically occurring in object tracking. For more details on Multiple Instance learning see Section 2.4.

We introduce temporal bags for each classifier grid element, containing patches of background models operating on different time scales. As background models we used common approximated median background model [98], which are updated in different time intervals to ensure both, adaptivity to changing situations as well as stability over time to avoid foreground objects growing into the background model. We assume that for each grid element the bag consists of at least one correctly labeled sample. Since in our case the positive samples are well defined and the ambiguity arises from the negative samples, we have to adapt the original MIL concept for our purpose. Thus, in the following we introduce the idea of inverse Multiple Instance Learning.

3.5.1 On-line Inverse MILBoost (IMIL)

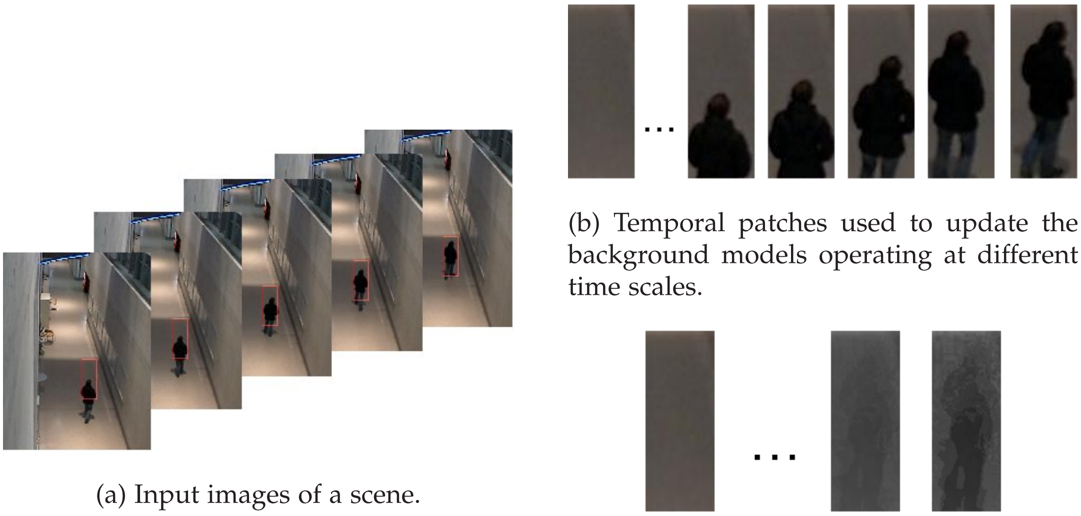
We build on the on-line Multiple Instance Boosting algorithm of Babenko et al. [11] described in Section 2.4. As common for boosting algorithms, a strong classifier is a linear combination of N weak classifier $h_j(x)$. The bag labels are binary $y_i \in \{0, 1\}$.

In general, multiple instance learning is used for dealing with ambiguity within the positive samples. However, since in the classifier grid scenario the positive samples are well defined, i.e., the positive samples are hand labeled and thereby correct by definition, the ambiguity concerns only the negative samples, i.e., the examples coming directly from the scene without any labeling, the original MIL idea has to be adapted. Thus, the negative bags B_i^- would need to contain at least one negative example whereas the positive bags B_i^+ solely consist of positive examples:

$$\forall x_{ij}^+ \in B_i^+ : y(x_{ij}^+) = 1 \tag{3.6}$$

$$\exists x_{ij}^- \in B_i^- : y(x_{ij}^-) = 0 \; . \tag{3.7}$$

In order to correctly calculate the loss $\mathcal{L}$ by inverting the problem, we have to switch the labels between the positive and the negative class; hence we term it *inverse MIL*. This causes to focus on examples that are more likely to be correct negative examples, which directly fits to our problem.



(a) Input images of a scene.

(b) Temporal patches used to update the background models operating at different time scales.

(c) Collected temporal bags, operating at different time scales, for one classifier grid element.

Figure 3.8: Input image of a scene with corresponding patches used to update the background models within the temporal bags of each classifier grid.

3.5.2 Application of IMIL to Classifier Grids

As already described in Section 3.4.2 the model describing the object class (the positive class) is fixed and can be calculated off-line while subsequently only negative updates are performed. However, to cope with ambiguously labeled negative samples, in particular non-moving foreground objects grown into the background, we apply the Inverse MILBoost as described in Section 3.5.1 to perform negative updates. Each grid element has its own negative bag, which is generated by collecting a stack of input images from the image sequence over time, which we refer to as "temporal bag". This is visualized in Figure 3.8. Having a large stack assures that the assumption for the negative bag containing at least one negative sample is mostly valid. In contrast, the probability that an object stays at one specific location over a longer period of time is very low (see Equation (3.2)). Since collecting a large stack of input images is adversarial for both, run-time as well as memory requirements, the temporal stack consists of a small number of background images operating at different time scales. By background images operating at different time scales we refer to different background models that are updated at different time intervals. Hence, they capture different information from an actual image of the current scene (which is updated using each input image) to an out-of-date image, which is updated only every few hundred frames. The adaptivity to changing illumination conditions is given by the background models which are updated in small time intervals. To avoid that objects staying at the same position for a while become part of the background, we use background models updated on long time intervals. Any kind of background model can be used. In particular, we apply the approximated median background model [98]. The different time intervals for updating the background models within the temporal bag ensure that they fulfill the MIL constraints. Hence, the multiple instance learning property of inherently dealing with ambiguity in data can be exploited for improving the classifier grid approach and avoiding short-term drifting.

3.6 Incorporating unreliable object information from the scene

By incorporating object information from the scene without any label information the fixed update strategy described in Section 3.3.2 cannot be applied any more, since the long-term stability is not guaranteed. Up to now the long-term stability is given by the fixed model for the object class. To preserve the off-line trained models and to combine them with on-line trained models we introduce a new binary boosting algorithm which builds on an internal multi-class representation.

3.6.1 On-line TransientBoost

Existing methods to include new (unreliably labeled) samples are either too firm hindering to acquire new information or too adaptive tending to drift. Moreover, even by using a strong prior, more sophisticated semi-supervised methods can fail if false positives (fitting to the prior) are used for updating the classifiers.

In contrast, we propose to combine reliable knowledge (gathered from labeled data) with unreliable information (acquired on-line, without any labeled information). The main idea is to model reliable and unreliable samples within different classes in a multi-class representation, while still preserving binary update and evaluation strategies. Since the unreliable data can be considered as transient information which may change over time, we refer to the method as *On-line TransientBoost*. The transient information comes directly from scene and implies foreground as well as background information which may change over time. This information might be relevant only within a certain time interval. We can assure robustness (i.e., avoid long-term drifting), but in contrast to existing approaches, we are able to include new (completely orthogonal) information, especially increasing the recall. The whole idea is visualized in Figure 3.9. TransientBoost builds on on-line GradientBoost [90]. Inspired by the on-line multi-class boosting algorithm of Saffari et al. [118] we introduce an on-line binary boosting algorithm with an internal multi-class representation, which gives the capability to cope with reliable and unreliable (transient) information in parallel.

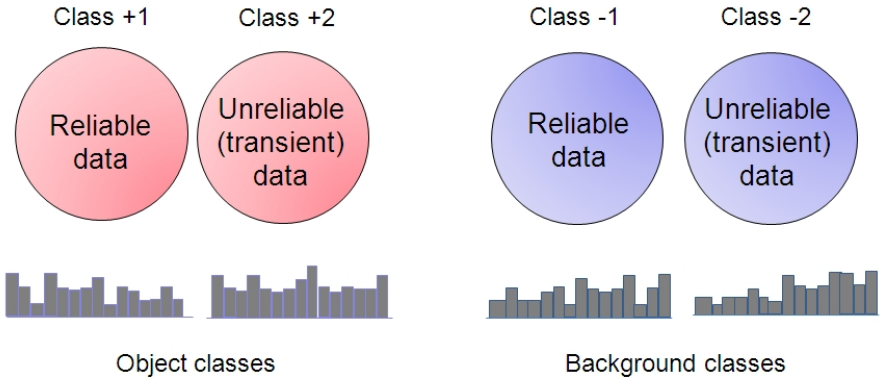


Figure 3.9: The main idea of TransientBoost is to exploit multi-class boosting algorithm for a binary classification problem in a way that combines reliable and unreliable information for the foreground class as well as for the background class.

Given a loss function $\ell(\cdot)$ and labeled training data, $X = \{(x_1, y_1), \cdots, (x_N, y_N)\}, x_n \in \mathbb{R}^D, y_n \in \{-1, +1\}$, the goal of GradientBoost is to estimate a strong classifier $H(x)$ as a linear combination of N weak learners $h_n(x)$ minimizing the loss. Hence, at stage t, we are searching a classifier h_t which maximizes the correlation with negative direction of the loss function:

$$h_t(x) = \arg\max_{h(x)} \; -\nabla\mathcal{L}^T h(x), \tag{3.8}$$

where $\nabla\mathcal{L}$ is the gradient vector of the loss at $H_{t-1}(x) = \sum_{m=1}^{t-1} h_m(x)$. This can be simplified to

$$h_t(x) = \underset{h(x)}{\arg\max} \; - \sum_{n=1}^{N} y_n \underbrace{\ell'(h_n H_{t-1}(x_n))}_{-w_n} h(x_n), \tag{3.9}$$

where $\ell'(\cdot)$ are the derivatives of the loss with respect to H_{t-1} and w_n are the sample's weights. Optimizing Equation (3.9) is independent of the applied loss function.

This formulation can simply be adopted for the on-line domain by using selectors as introduced in [64], where each selector $s_m(x)$ consists of N weak classifiers $\{h_{m,1}(x), \cdots, h_{m,N}(x)\}$ and is represented by its best weak classifier $h_{m,k}(x)$. The optimization step in Equation (3.9) is then performed iteratively by propagating the samples through the selectors and updating the weight estimate w_n according to the negative derivative of the loss function.

On-line GradientBoost was designed for a binary classification problem. However, by introducing weak learners that are able to handle more than two classes, it can be simply extend to the multi-class domain. In general, any weak learner providing confidence rated responses can be applied. We use histogram-based classifiers, based on the idea of Friedman et al. [55], who used symmetric multiple logistic transformation as weak learner for a J-class problem

$$h_j(x) = \log \; p_j(x) - \frac{1}{J} \sum_{l=1}^{J} \log p_l(x) \; , \tag{3.10}$$

where $p_j(x) = P(y_j = 1|x)$. In particular, they showed that if the sum over the weak classifier responses over all classes is normalized to zero, i.e., $\sum_{j=1}^{J} h_j(x) = 0$, the probability $p_j(x)$ can be estimated by using histograms. Beside this, using histograms gives the advantage that they are highly appropriate for on-line learning since they can easily be updated.

This multi-class formulation allows for modeling reliable and unreliable data by using different classes, i.e., $y = [+1, -1]$ for the reliable data and $y = [+2, -2]$ for the unreliable data. Thus, during an update the classifier is provided a sample x_t and a label $y_t \in \{-2, -1, +1, +2\}$ and depending on the label the corresponding histograms are updated. Moreover, for the reliable samples the histogram updates are performed incrementally whereas for the unreliable transient samples an iir-like filtering of the histogram bins is applied (i.e., the knowledge is scaled down according to its age). In this way the reliable information is accumulated whereas the unreliable information allows higher adaptivity, but is fading out quickly (depending on the forgetting rate f), thus, avoiding drifting.

The next, crucial step is to include the uncertainty of the sample $\langle x_t, y_t \rangle$ into the feature selection procedure. In each update step, similar to the binary case, the best weak classifier $h_{m,k}$ within a selector s_m is estimated according to its error. The error is updated depending on the weight of the correct classified samples $\lambda^c_{m,n}$ and the misclassified samples $\lambda^c_{m,n}$ within each weak classifier. However, the error calculation has to be adapted due to the multi-class formulation. If the prediction was correct, i.e., the signum of the classifier response $h_{m,n}$ equals the signum of class label used to update the classifier y_t ($\mathrm{sign}(h_{m,n}(x)) = \mathrm{sign}(y_t)$), the weight is updated as follows:

$$\begin{cases} \lambda^c_{m,n} = \lambda^c_{m,n} + w_n, & \text{if } \mathrm{sign}(h_{m,n}(x)) = \mathrm{sign}(y_t) \\ \lambda^w_{m,n} = \lambda^w_{m,n} + w_n, & \text{otherwise} \end{cases} \tag{3.11}$$

where w_n is the actual weight estimate of the current sample. In the original GradientBoost algorithm any differentiable loss function ℓ can be used to update the weight by $w_n = -\ell'(y_t H_m(x_t))$, where $H_m(x) = \sum_{t=1}^{m} s_t(x)$ is the combination of the first m weak classifiers and y_t is the label of the current sample. In our case, however, we have to re-formulate the weight update according to our multi-class model. Otherwise the classifier would try to distinguish between the reliable and the unreliable classes and would penalize samples that are already classified correctly. Hence, since we are interested in discrimination of positive and negative classes, we have to change the weight update to

$$w_n = -\ell' \left(sign(y_t) F_m(x) \right) \ . \tag{3.12}$$

The derived update procedure for TransientBoost is summarized more formally in Algorithm 6. To finally obtain a binary classification result, during evaluation a sample is classified based on the signum of the classifier's prediction.

3.6.2 Applications of TransientBoost to Classifier Grids

TransientBoost is applicable to every problem where reliable and unreliable information needs to be combined within one classifier. Hence, it is ideally suited for the classifier grid approach to incorporate unlabeled scene specific information. The approaches presented in Section 3.4.2 and Section 3.5.2 allow for a robust adaption of the classi-

Algorithm 6 On-line TransientBoost Update

```
Require: sample $x_t$, label $y_t \in \{\pm 1, \pm 2\}$, model $H^{t-1}$
Output: updated model $H^t$
  Set initial weight $w_0 = -\ell'(0)$
  for $m = 1$ to $M$ do
    for $n = 1$ to $N$ do
      Train multi-class weak learner $h_{m,n}(x)$ with sample $(x_t, y_t, w_n)$
      if $sign(h_{m,n}(x_t)) = sign(y_t)$ then
        $\lambda^c_{m,n} = \lambda^c_{m,n} + w_n$
      else
        $\lambda^w_{m,n} = \lambda^w_{m,n} + w_n$
      end if
    end for
    Find best weak learner: $k = \arg\min_n \frac{\lambda^w_{m,n}}{\lambda^c_{m,n} + \lambda^w_{m,n}}$
    Set $s_m(x_t) = h_{m,k}(x_t)$
    Set $H^t_m(x_t) = H^t_{m-1}(x_t) + s_m(x_t)$
    Set $w_n = -\ell'(sign(y_t)H^t_m(x_t))$
  end for
```

fier grids to changing backgrounds. However, scene specific object information, which would increase the recall, could not be incorporated without losing long-term stability. In order to increase the recall but keep the classifiers' accuracy on the given level we introduce classifier co-grids, an extended update scheme for the classifier grid approach. Classifier co-grids are related to the well-established semi-supervised learning concept of co-training and allow for both, a higher recall due to the scene specific object information and a preserved stability due to TransientBoost algorithm.

The key idea is to use an independent "orthogonal" information source instead of the fixed update strategy to provide positive and negative updates from the scene. To get such an "orthogonal" information source we adopted the visual co-training approach of Levin et al. [91]. In fact, we apply background subtraction (i.e., approximated median background model [98]) to exploit the information given by this additional view on the data. In contrast to [91], co-training is performed only in an initial phase. Later on this classifier is kept fix and used as an oracle for two reasons: First, not all situations can be handled robustly by co-training. Hence, if too many wrong updates are performed the co-trained classifier would start to degenerate (i.e., the classifier start to drift) and finally fails. Second, as illustrated in Figure 3.10, most environmental changes are eliminated by the background subtraction and the variability in the positive class vanishes. Thus, no further information can be gained.

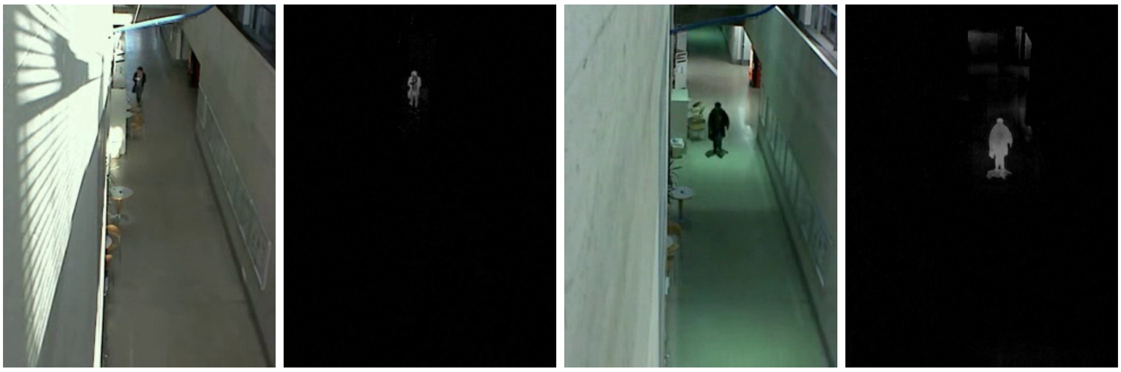


Figure 3.10: Different illumination conditions with the corresponding background subtracted images. Even in case of totally different illumination conditions and differently appearing objects the background subtracted image gives a similar representation of the object.

3.6.2.1 Co-Training Stage

During the initial stage our system is trained in a co-training manner as shown in Figure 3.11. Given n grid classifiers G_j operating on gray level image patches $\mathbf{X}_j$ and one compact classifier C operating in a sliding window manner on background subtracted images $\mathbf{B}$. To start co-training, the classifiers G_j as well as the classifier C are initialized with the same off-line trained classifier (see Algorithm 4). The classifiers within the classifier grid G_j and the classifier C operating on the background subtracted images co-train each other. A confident classification (no matter if positive or negative) of a classifier G_j is used to update the classifier C with the background subtracted representation at position j. Vice versa, a confident classification of classifier C at position j generates an update for classifier G_j. The off-line trained prior information already capturing the generic information causes a small number of updates to be sufficient to adapt the classifiers to a new scene. The update procedure during the initialization for a specific grid element j is summarized in Algorithm 7.

3.6.2.2 Detection Stage

After the initial stage, as described above, the classifier C operating on the background subtracted images is no longer updated and is applied as an oracle to generate new pos-

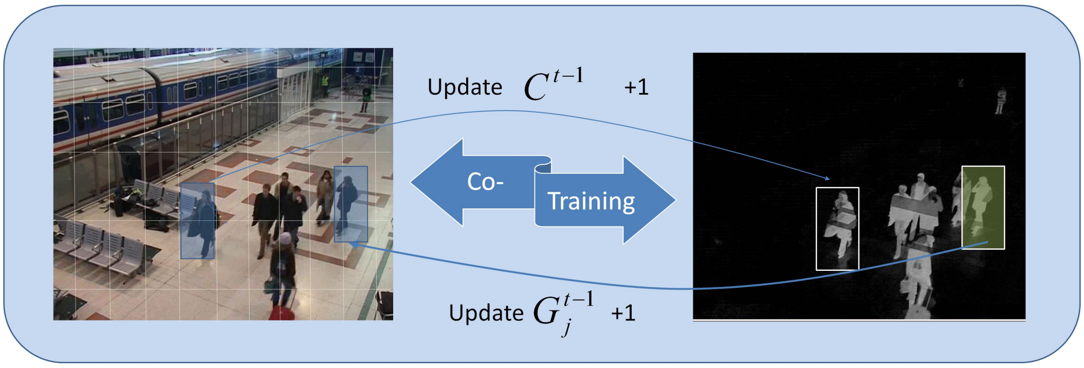


Figure 3.11: Co-grid initialization stage: the grid classifiers on the left side are co-trained with an independent classifier operating on the background subtracted image on the right side.

Algorithm 7 Co-Grid Initialization

```
Input: grid-classifier $G_j^{t-1}$
Input: co-trained classifier $C^{t-1}$
Input: patch corresponding to grid-element $\mathbf{X}_j$
Input: background subtracted patch $\mathbf{B}_j$

  if $C^{t-1}(\mathbf{B}_j) > \theta$ then
    $update\left(G_j^{t-1}, \mathbf{X}_j, +\right)$
  else if $C^{t-1}(\mathbf{B}_j) < -\theta$ then
    $update\left(G_j^{t-1}, \mathbf{X}_j, -\right)$
  end if

  if $G_j^{t-1}(\mathbf{X}_j) > \theta$ then
    $update\left(C^{t-1}, \mathbf{B}_j, +\right)$
  else if $G_j^{t-1}(\mathbf{X}_j) < -\theta$ then
    $update\left(C_{t-1}, \mathbf{B}_j, -\right)$
  end if

Output: grid-classifier $G_j^t$, classifier $C^t$
```

itive and negative samples as illustrated in Figure 3.12. In combination with our robust learning algorithm this oracle can now be to replace the fixed update rules described in Section 3.3.2. Moreover, we perform negative updates for the classifiers G_j only if they are necessary, i.e., if the scene is changing. Even if the oracle classifier C has a low recall, the precision is very high. Thus, only very valuable patches are used to update the classifier G_j, which leads to an increasing performance of the classifiers within the classifier grid. In particular, a confident positive classification result of classifier C at position j generates an update for all classifier G_i, $i = 1, \ldots, n$ in the classifier grid. In this way new scene specific positive samples are disseminated over the whole classifier grid. Negative updates are performed for classifiers G_j if there is no corresponding detection reported at this position for classifier C. The update procedure during the detection phase for a specific grid element j is summarized in Algorithm 8.

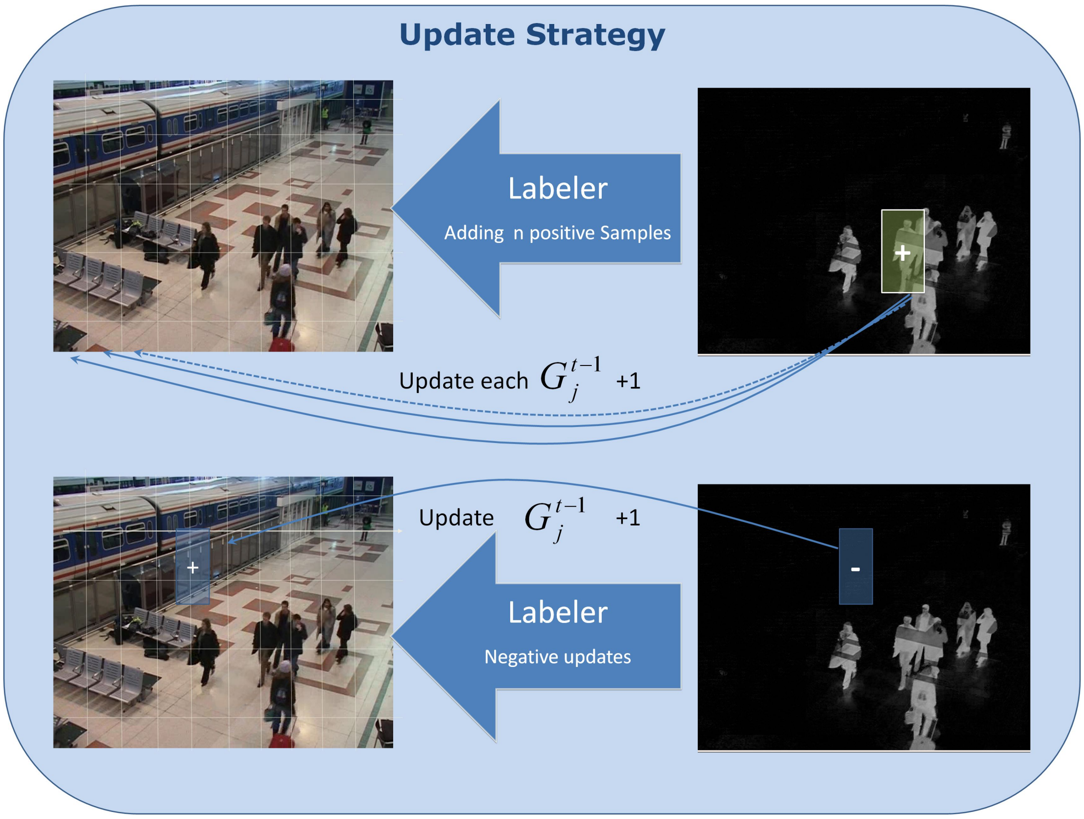


Figure 3.12: Co-grid detection stage: the classifier C is used as an oracle to perform positive as well as negative updates of the classifiers within the classifier grid. Positive updates are spread to all classifiers in the grid whereas negative updates are performed for a particular classifiers in grid.

3.6.2.3 Implementation details

In general, any on-line learner can be applied within the co-grid approach. However, to ensure robustness and long-term stability, we use TransientBoost as learner for the co-grid approach. In particular, we define a 3-class problem, capturing two positive and one negative class as illustrated in Figure 3.13. The class $+1$ is trained from labeled samples and is kept fixed whereas the class $+2$ and -1 are updated using the samples

```
Algorithm 8 Co-Grid Update
Input: grid-classifier $G_j^{t-1}$
Input: co-trained classifier $C$
Input: patch corresponding to grid-element $\mathbf{X}_j$
Input: background subtracted patch $\mathbf{B}_j$
if $C(\mathbf{B}_j) > \theta$ then
    $\forall i : update\left(G_i^{t-1}, \mathbf{X}_j, +\right)$
end if
if $C(\mathbf{B}_j) < -\theta$ then
    $update\left(C_j^{t-1}, \mathbf{X}_j, -\right)$
end if
Output: grid-classifier $G_j^t$
```

labeled by the co-grid (on-line). In this way, since the prior information is kept fixed and the scene specific information is transient, i.e., is fading out over time, in a long-term range at least the initial performance can be ensured due to the fixed pre-trained object class while allowing for highly adaptive updates. Thus, robustly new positive information can be gained especially increasing the recall but preserving the accuracy. We neglect a fixed off-line trained background class -2 since high generic background information does not provide valuable information for a particular scene due to large variations in the background class.

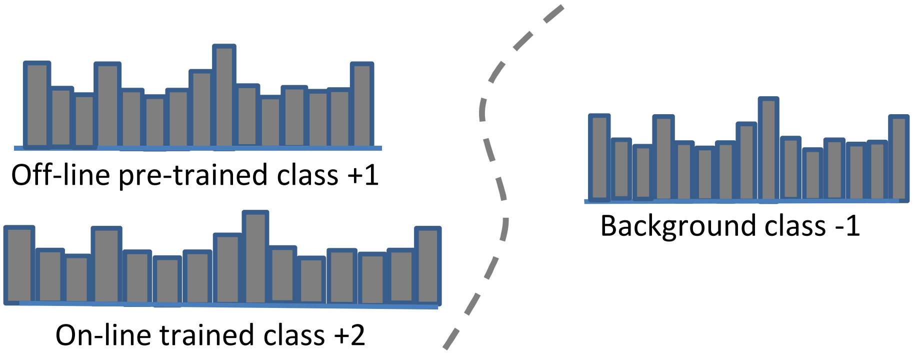


Figure 3.13: TransientBoost used for this particular problem consists of three models, two describing the object class and one describing the background class.

Moreover, we can benefit from properties inherited from the on-line GradientBoost algorithm, which allows for using different loss functions. The exponential loss function makes AdaBoost highly susceptible to label noise, which arises from samples with a wrong class label. This is caused by AdaBoost's focus on misclassified samples, if a weak classifier classifies the sample correct but the class label is wrong, the weight of the particular sample is increased, which can injure the learner. Different loss functions are visualized in Figure 3.14, where $yF(x)$ is the classification margin of a sample, where a negative margin indicates a mis-classification. In order to increase the robustness to label noise for the new positive examples we use the logistic loss function. For the negative updates, however, we use an exponential loss function to ensure that false positives in the background are fading out quickly. By using the co-trained classifier operating on the background subtraction we are less sensitive to wrong negative updates, which avoid the effect of short-term drifting.

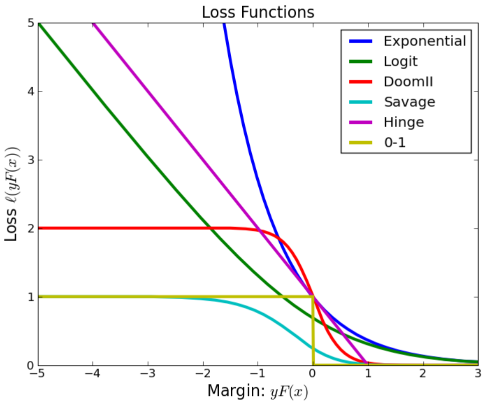


Figure 3.14: Different loss functions used in boosting, taken from [90].

To get the background subtracted (BGS) image we apply a simple approximated median background model [98]. The classifiers used for the initialization are off-line trained. As features we use simple Haar-like features.

3.6.2.4 Discussion

Related work in this field also dealt with incorporating scene specific object information by context-based classifier grids [123]. This context information is gained through three different ways: a fixed detector, a tracker and 3D-context information. The authors showed that the recall can be increased significantly, but on the expense of the precision. In contrast, our Co-Grid approach allows for incorporating unreliable information without manual labeling effort within an on-line learning algorithm which is highly adaptive but still robust. Using an on-line multi-class algorithm for combining different representations would cause the problem, that if the initial model becomes too similar to the on-line model of the same class the sample might be classified wrong, which implies a large weight and hence increases the focus on such samples, even though they are already classified correct. This is not the case in our "pseudo-multi-class" algorithm, since different representations for one class can be arbitrarily similar. Classifying the sample as different model within the same class (same signum) is still a correct classification for the proposed TransientBoost algorithm. Further on, TransientBoost allows unreliable data to fade out over time, resulting in a highly adaptive learning algorithm. In contrast, the reliable data is preserved completely and not harmed by wrong updates.

3.7 Experimental evaluation

In this section we demonstrate the benefits of the proposed approaches and compare to different state-of-the-art approaches. We first describe the experimental setup used throughout all experiments in Section 3.7.1. Then, we show the effect of the overlap parameter for the grid elements in Section 3.7.2, which is of interest for all proposed approaches. The subsequent sections contain an evaluation of the proposed approaches on various different datasets. Section 3.7.3 demonstrates the benefits of off-line on-line linkage for the tasks of pedestrian detection and car detection. Furthermore, a long-term experiment demonstrates the robustness of our adaptive approach. Section 3.7.4 demonstrates the effect of inverse multiple instance learning for classifier grids, which avoids the short-term drifting problem. In Section 3.7.5 we show the advantages of the proposed classifier co-grid approach in combination with the robust TransientBoost algorithm and again show the long-term stability. Finally, we compare all approaches on a common publicly available benchmark and on a challenging sequence corridor sequence, where besides pedestrians objects like chairs or a yellow ball are moved through the scene in Section 3.7.6.

3.7.1 Experimental Setup

If not specified otherwise all experiments within this section are performed and evaluated as described below. To generate the classifier grid the approximate size of the object-of-interest in the scene is needed. For reasons of simplicity we estimated the ground plane for our experiments manually. However, this could also be done automatically (e.g., [109, 23]). Based on this estimate the classifier grid is initialized using an overlap of at least 85%. Slightly varying overlaps are caused by the required memory consumption depending on the image size. Hence, we state the overlap parameter separately for each experiment. More details on the effect of the overlap parameter for the classifier grid approach are discussed in Section 3.7.2.

For a quantitative evaluation, we use the well-established recall-precision curves (RPC) [3]. Therefore, we have to estimate the precision $Pr = TP/(TP + FP)$ and the recall $R = TP/P$. TP is the number of true positives, i.e., the number of predictions that coincides with the ground truth, FP is the number of false positives, i.e., the number of predictions that do not coincide with the ground truth, and P is the number of positives in the test data represented by the given ground truth. In particular, a detection is accepted as true positive, if it fulfills the strict PASCAL bounding box evaluation criterion [44], with a minimal overlap of 50%. The overlap is calculated as

$$o = \frac{area(B_{det} \cap B_{gt})}{area(B_{det} \cup B_{gt})}, \tag{3.13}$$

where B_{det} is the predicted bounding box (detections with a confidence above a certain threshold) and B_{gt} is the ground truth bounding box. $B_{det} \cap B_{gt}$ denotes the intersection of the predicted and the ground truth bounding box, while $B_{det} \cup B_{gt}$ denotes the union of these bounding boxes. The allowed shift between the bounding box and the predicted detection is visualized by the green overlay in Figure 3.15, the green rectangle is the ground truth bounding box.

Once we have estimated these parameters we can plot recall R against $1 - Pr$. Additionally, we report the $F - measure$ [3], which is the harmonic mean between *recall* and *precision* and is defined by $FM = (2 \cdot R \cdot Pr)/(R + Pr)$. In particular, the characteristics on the recall and precision given in the tables within this chapter were generated such that the $F - measure$ was maximized.

We performed experiments on two different tasks, namely pedestrian detection and car detection. For pedestrian detection we compared our approaches to two state-of-the-art generic object detectors, publicly available, i.e., the Histograms of Oriented Gradients based pedestrian detector of Dalal and Triggs (HOG-DT) [33] [*] and the deformable part model object detector of Felzenszwalb et al. (DPM-FS) [45] [†]. For the task of car detection we again compare to the deformable part model and additionally to the

[*] http://pascal.inrialpes.fr/soft/olt

[†] http://people.cs.uchicago.edu/~pff/latent

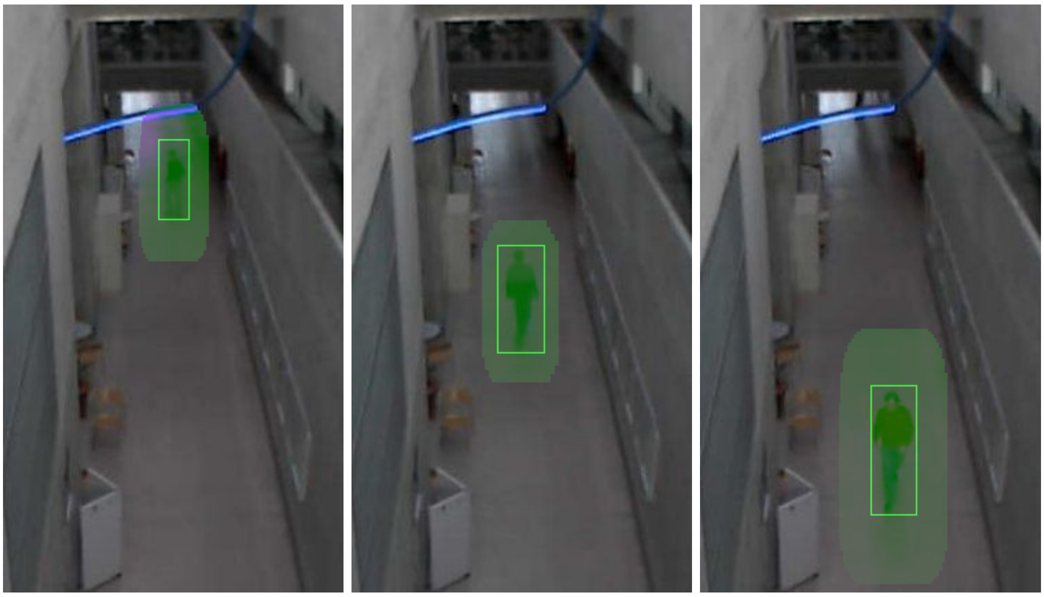


Figure 3.15: The overlaid area shows the allowed shift for a 0.5 overlap between the predicted bounding box and the ground truth using the PASCAL bounding box evaluation criterion.

Implicit Shape Models (ISM) of Leibe et al. [88] [‡]. All these approaches are generic object detectors trained on a large set of training data and do not use any scene specific prior knowledge. Hence, to allow for a fair comparison, in a post-processing step we removed all detections of the generic approaches that do not fit to the estimated ground plane. In fact, a detection was removed if the scale was smaller than 75% or greater than 125% of the expected patch-height at this specific location. This post-processing does not reduce the recall since these detections would be counted as false positives, it solely improves the precision. We remove these false positive detections since the other approaches make use of this scene specific information by evaluating with the correct scale at each position in the image. In order to perform a fair evaluation we resized the input images for the two generic detectors to double size if the object of interest is too small for the trained model.

Unless otherwise stated, we use on-line boosted classifiers within our classifier grid which are pre-trained off-line. As features we use Haar-like features, caused by good results on various different tasks and the fast evaluation. The used feature types are shown in Figure 3.16. In general, any kind of weak classifier can be used. Unless otherwise stated, we use simple decision stumps as shown in Figure 3.17 throughout our experi-

[‡] http://www.vision.ee.ethz.ch/~bleibe/code/ism.html

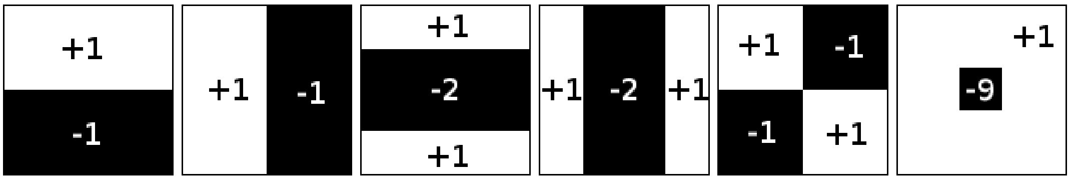


Figure 3.16: Haar-like features used throughout all experiments.

ments. Two generative models describing the two classes (the positive distribution D_j^+ describes the object class, the negative distribution D_j^- describes the background class) are combined to one discriminative weak classifier. These two generative distributions are estimated from the feature responses of the positive and negative training samples. By assuming these distributions to be Gaussian, they can be easily updated, e.g., by using a Kalman filtering technique. This results in a simple decision stump

$$h_j(\mathbf{x}) = p_j \cdot \operatorname{sign}(f_j(\mathbf{x}) - \theta_j), \tag{3.14}$$

where the threshold θ_j and the parity p_j are calculated using the Bayesian rule with respect to D_j^+ and D_j^-. The number of weak classifiers per selector as well as the number of selector slightly varies throughout the experiments depending on the complexity of the task and is hence stated separately for each experiment.

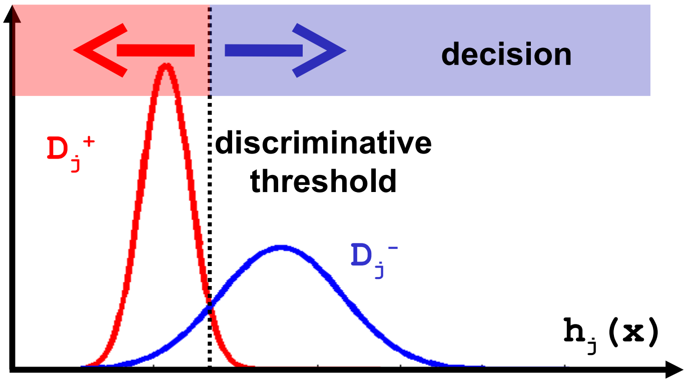


Figure 3.17: Simple decision stump.

3.7.2 Overlap between grid elements

Within this section we perform an evaluation on the effect of the overlap parameter which affects the overlap between the individual classifiers in the classifier grid. We evaluate on the task of pedestrian detection. The selected sequence monitors a corridor in our lab building, consisting of 324 frames with an image size of 248 × 428 pixels. Illustrative examples are shown in Figure 3.18. The sequence contains 215 occurrences of pedestrians, which in general do not occlude each other. Hence, we will refer to this sequence as *Simple Corridor* sequence.

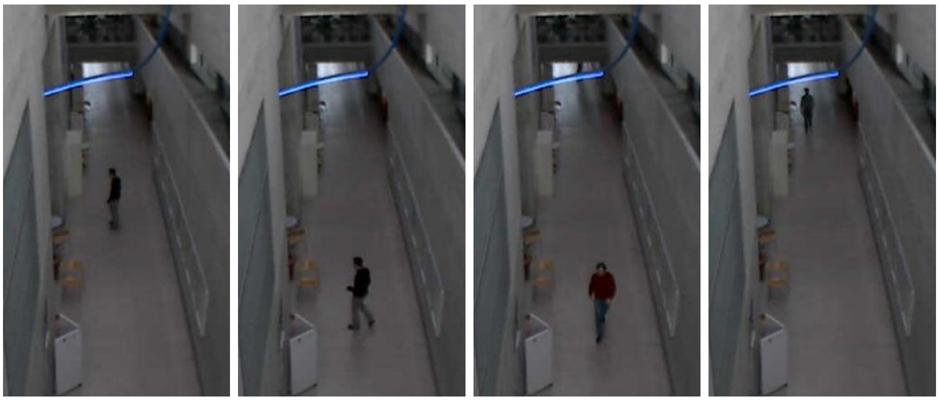


Figure 3.18: Illustrative frames of the *Simple Corridor* Sequence.

The overlap between the grid elements determines the number of grid elements and along with the number of classifiers. Hence, one has to find a trade-off between runtime, memory consumption and detection performance. For a region of interest of a size of 248 × 368 pixels and a known function describing the scale of the object of interest dependent on the position in the image, the number of grid elements depending on the overlap parameter is visualized in Figure 3.19. One can see that the number of grid elements and along with the memory consumption rises exponentially fast.

For this experiment we used an off-line trained classifier suitable for on-line updates with 50 selectors, each of it containing 10 weak classifiers for the feature selection process. The performance depending on the overlap parameter is shown in Figure 3.20. One can clearly see that the performance is increasing with an increasing number of grid elements. Depending on the image resolution we use an overlap between 70% and 92% to find a trade-off between a good performance and acceptable memory consumption.

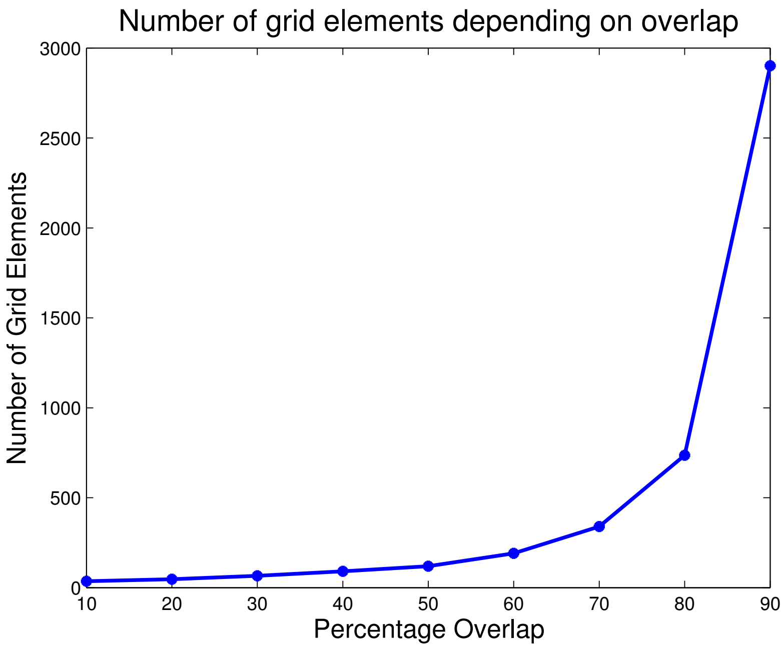


Figure 3.19: Number of patches for *Simple Corridor* Sequence depending on the overlap.

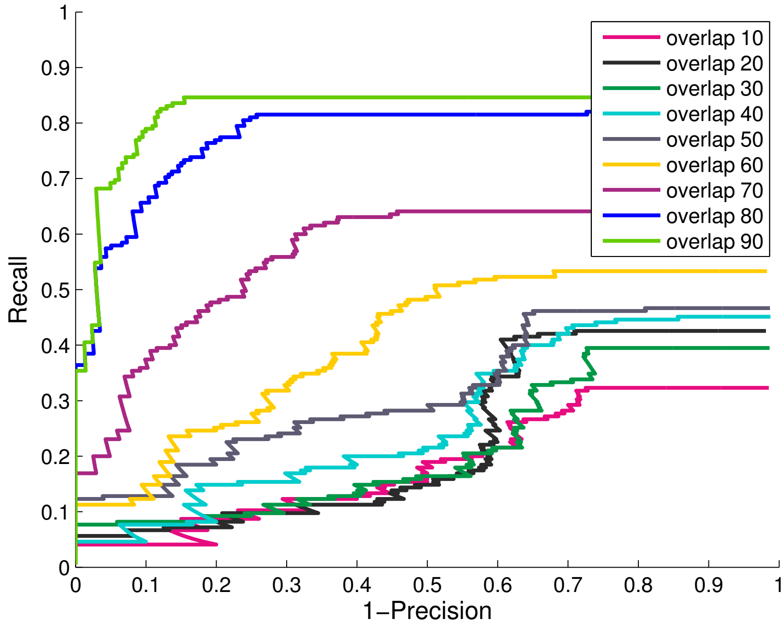


Figure 3.20: RPC for *Simple Corridor* Sequence with a varying overlap between the grid elements.

3.7.3 Linking Off-line and On-line Boosting

We evaluate the classifier grid approach with linked off-line and on-line boosting (CG-OOL) on different scenarios to demonstrate the benefits compared to generic state-of-the-art approaches. Besides evaluations on two different pedestrian datasets and one car dataset we perform a long-term experiment to demonstrate the stability over time. The long-term experiment ran for one week, processing 580.000 frames, where each classifier within the classifier grid is evaluated and updated whenever a new frame arises.

3.7.3.1 Pedestrian Detection

For all pedestrian detection evaluations within this section we use an off-line trained boosted classifier with a size of 20 × 10 weak classifiers, which means that we have 20 selectors, each of it containing 10 weak classifiers for the feature selection process. We use an overlap between the grid elements of 90%. For the CG-OOL approach we first perform updates with each frame within the first 50 frames. After we adapted our classifiers in the initial stage (first 50 frames) we reduce the number of updates by using only every 10-th frame to improve the run-time performance.

Caviar Dataset

We evaluate the off-line on-line linkage approach on the publicly available *Caviar* dataset [§] and compare it to the two generic detectors, which might be considered a fair baseline. The *Caviar* dataset was provided by the CAVIAR (Context Aware Vision using Image-based Active Recognition) project[¶]. This dataset contains a large number of clips. We selected clip *ShopAssistant2cor* since it contains a large number of pedestrians (i.e., 1265). The sequence consists of 370 frames with an image size of 384 × 288 pixels and shows a Corridor of a public shopping mall. Two persons are entering a shop, crossing the corridor while a number of persons are walking along this corridor or standing at the end of the corridor. The results of the *Caviar* sequence are shown in Figure 3.21 and Table 3.1. Again it can be seen that the adaptive classifier grid detector (CG-OOL) outperforms the generic detectors (HOG-DT and DPM-FS), especially, in terms of recall. Illustrative detection results are shown in Figure 3.22.

[§] http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1

[¶] http://homepages.inf.ed.ac.uk/rbf/CAVIAR/

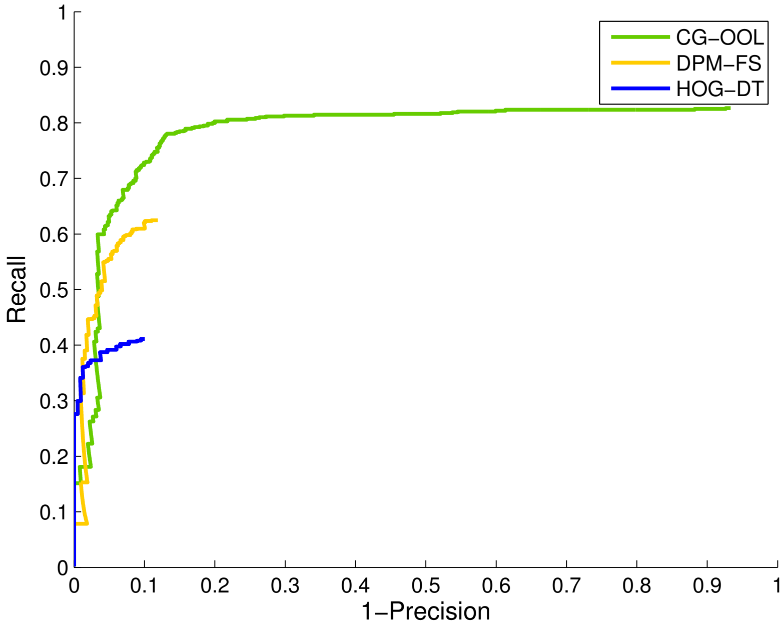


Figure 3.21: RPC for the *Caviar Sequence*.

	R	Pr	FM
CG-OOL	0.78	0.87	0.82
DPM-FS	0.62	0.90	0.74
HOG-DT	0.41	0.91	0.57

Table 3.1: Detection characteristics of the *Caviar* Sequence for different methods sorted by the F-measure.

3.7.3.2 Car Detection

To show that the proposed approach is not limited to detecting persons, we additionally demonstrate results for the task of car detection. We compare our method to existing established methods, namely the Implicit Shape Models (ISM) of Leibe et al. [88] ‖ and the deformable part model detector (DPM-FS). The methods were evaluated on a sequence showing one direction of a public highway. In the following we refer to this dataset as “highway dataset. The whole scene consists of 1000 frames and contains a total number of 1952 cars from the rear view. For the ISM method and the DPM-FS detector the original images (384 × 324) were resized to the double size. In order to obtain a sufficient number of detections from the DPM-FS detector the detection threshold was set to −0.5. The overlap between the grid elements was set to 92%.

‖http://www.vision.ee.ethz.ch/~bleibe/code/ism.html

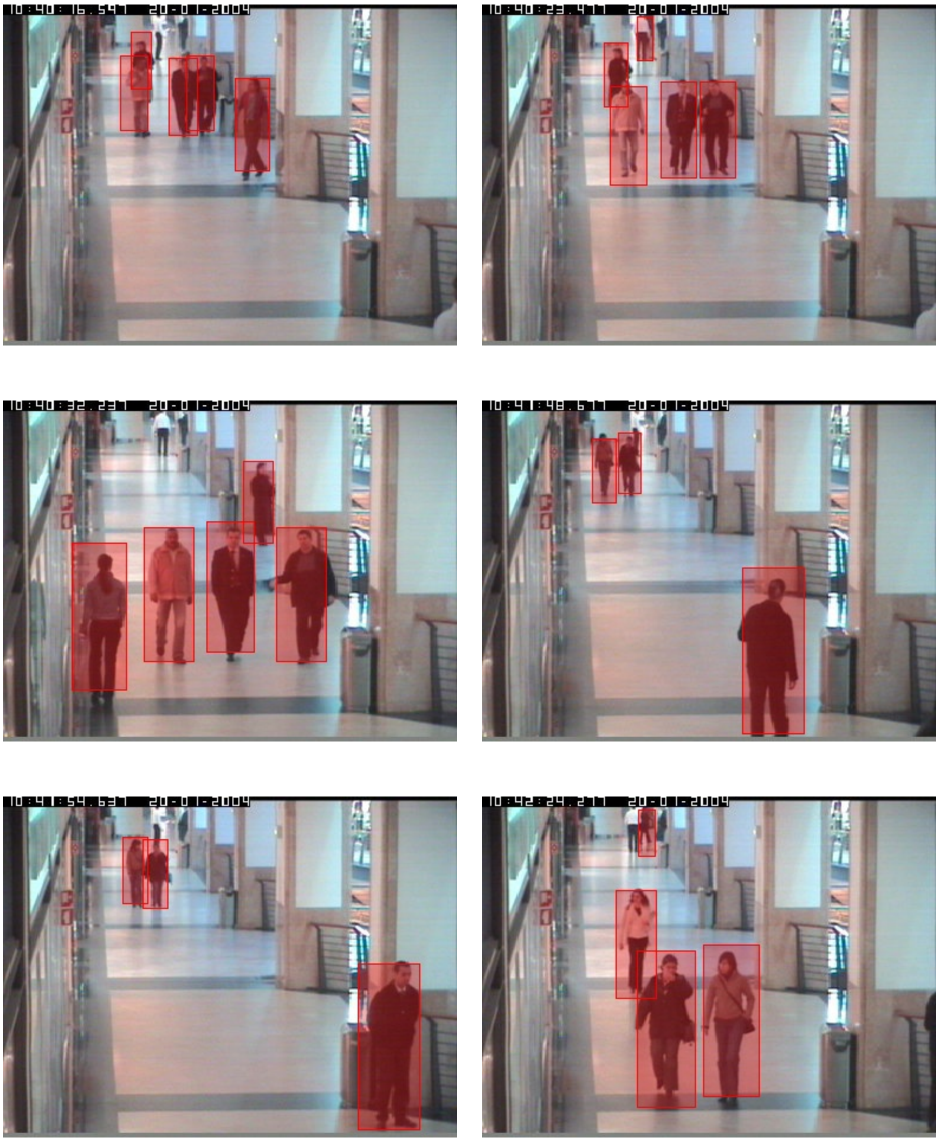


Figure 3.22: Illustrative detection results of the grid-based person detector for the *Caviar Sequence*.

From the results shown in Figure 3.23 it can be seen that the proposed method clearly outperforms the generic car detectors (ISM and DPM-FS). Illustrative detection results obtained by the proposed approach are shown in Figure 3.24. The detection characteristics are summarized in Table 3.2.

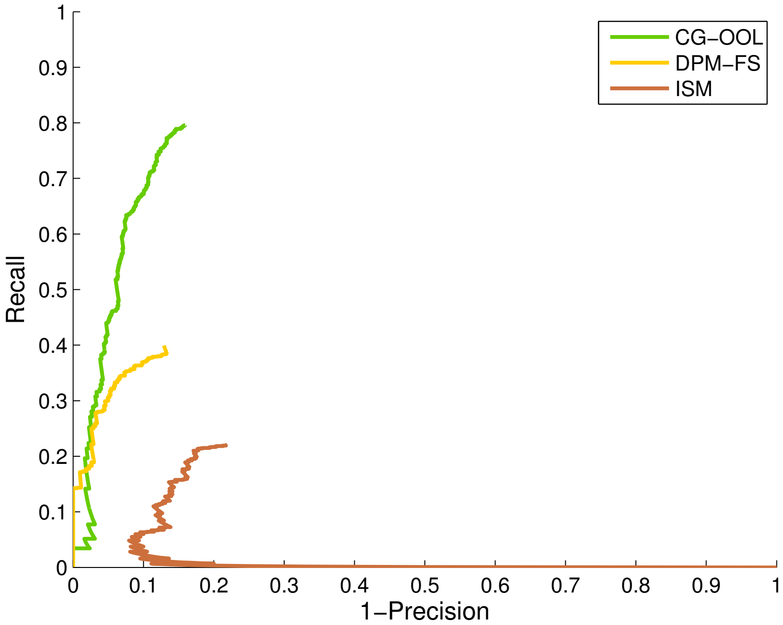


Figure 3.23: RPC for the *Highway sequence*.

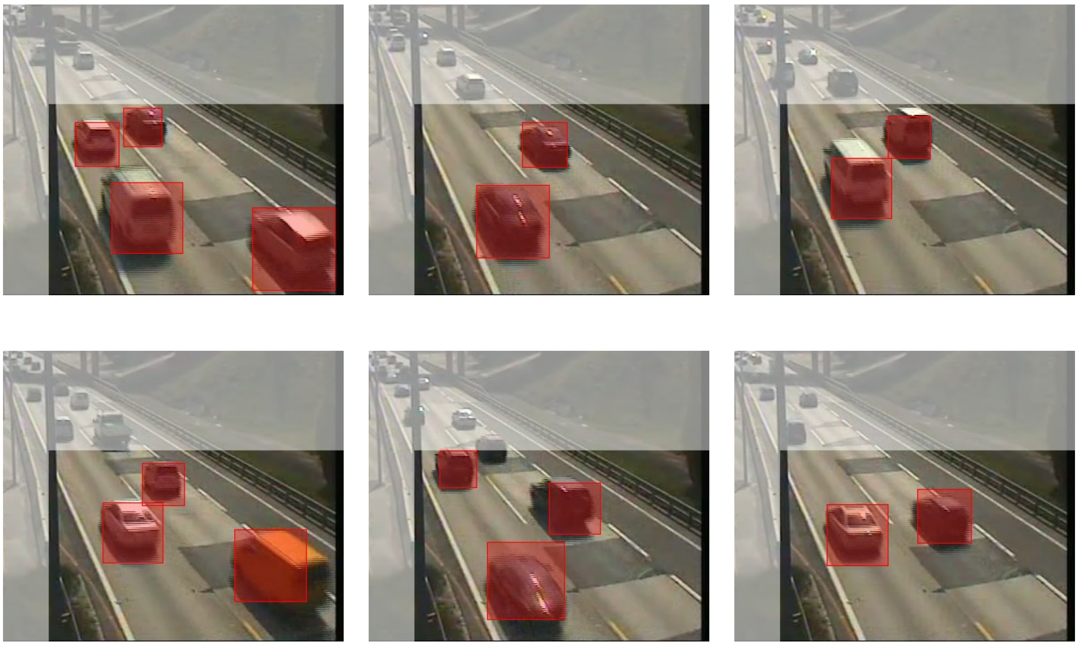


Figure 3.24: Illustrative detection results of the grid-based person detector for the *Highway sequence*.

	R	Pr	FM
CG-OOL	0.79	0.85	0.82
DPM-FS	0.40	0.87	0.55
ISM	0.22	0.78	0.35

Table 3.2: Detection characteristics of the *Highway* Sequence for different methods sorted by the F-measure.

3.7.3.3 Long-term behavior

Since the main goal of classifier grids is to develop an adaptive but still robust system that is learning 24 hours a day and 7 days a week, in the following we demonstrate the long-term behavior of the proposed method. We captured a sequence of a corridor in our building with $1fps$ during 7 days, resulting in 580.000 frames. Every single frame was used to perform an update of the classifiers within the grid. To show that the performance is unchanged over time, we annotated $2,500$ frames at four different points in time (which corresponds to approx. 40 minutes of video data). The sequences are selected at first day, the third day, the sixth day and the last day. The number of pedestrians visible as well as the number of updates performed before the sequence starts are summarized in Table 3.3 for all four sequences.

	# updates yet performed	# persons
1st day	3,390	475
3rd day	179,412	546
6th day	484,891	950
7th day	577,500	1005

Table 3.3: Description of the selected sequences of the long-term experiment.

From the results shown in Figure 3.25 and in Table 3.4 it can be seen that the method is stable over time. The slightly variations in the curves can be explained by the different levels of complexity for the four sequences (i.e., number of persons, density of persons, etc.). As can be seen from Table 3.4 the F-measure is unchanged over time.

Finally, in Figure 3.26 we illustrate the significantly changing conditions we had to deal with during these 7 days (i.e., natural light, artificial lighting, inadequate lighting, etc.). These drastically changing conditions arise the need for adaptive approaches, like the classifier grid approach.

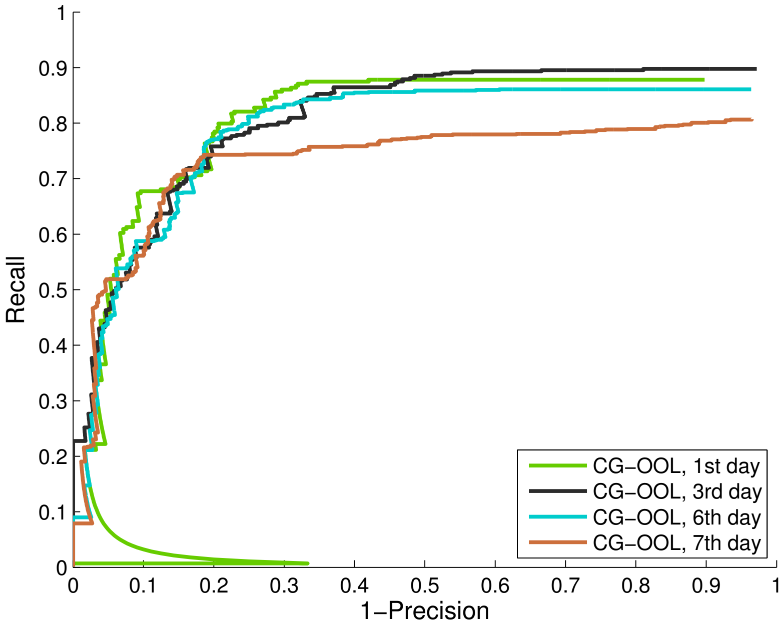


Figure 3.25: RPC for the *long-term experiment*

	R	Pr	FM
CG-OOL, 1st day	0.80	0.79	0.80
CG-OOL, 3rd day	0.76	0.80	0.78
CG-OOL, 6th day	0.78	0.79	0.79
CG-OOL, 7th day	0.74	0.81	0.78

Table 3.4: Comparison of Recall and Precision for the best F-Measure value for at different points in time for the long-term experiment.

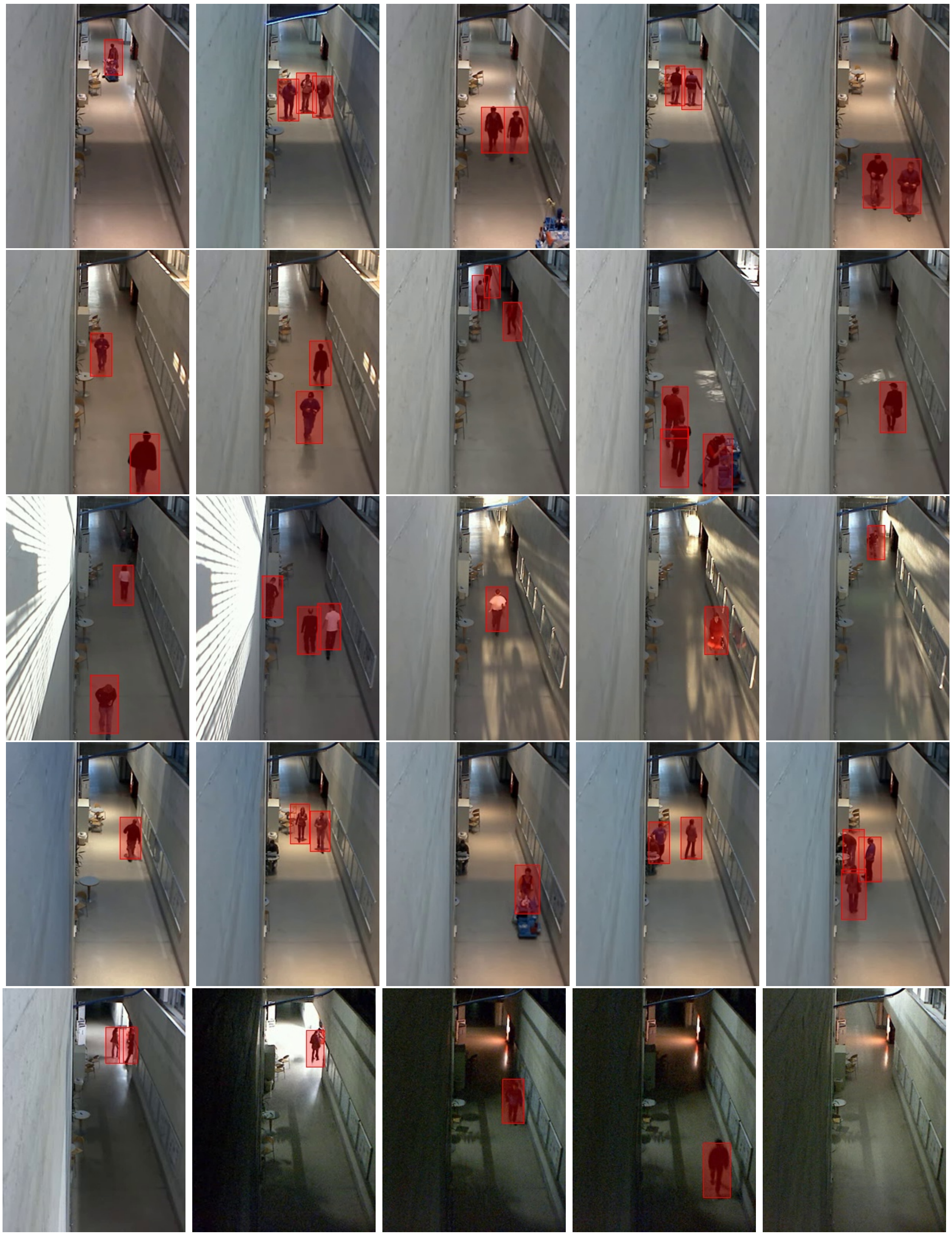


Figure 3.26: Illustrative detection results of the classifier grid approach (CG-OOL) obtained during to long-term experiment, where each row corresponds to one time of the day, morning, noon, afternoon, evening, and night respectively.

3.7.4 Inverse multiple instance learning for classifier grid

To demonstrate the benefits of the idea of inverse multiple instance learning for classifier grids, we run three different experiments particularly addressing the problem of short-term drifting. Since the performance of the CG-OOL approach compared to state-of-the-art approaches has already been demonstrated in the previous section and the short-term drifting problem does not occur for generic detectors which are not updated over time, we solely compare to the CG-OOL approach within this section. We first give an illustrative comparison between the CG-OOL approach and the inverse multiple instance learning approach (CG-IMIL). Then, we selected two datasets (pedestrians and cars) containing objects which are not moving over a long period of time. This causes short-term drifting for the CG-OOL approach caused by the fixed update strategy which can be avoided by using inverse multiple instance learning. From all experiments the benefits of the proposed methods are clearly visible.

For all experiments on pedestrian detection we use classifiers consisting of 30 selectors, where each selector consists of a set of 30 weak classifiers. For the car detection experiment we use classifiers consisting of 50 selectors, each of them containing 30 weak learners. To increase the robustness of the negative updates, we collect a stack of four background images, operating on four different timescales, which are updated every second frame, every 50-th frame, every 100-th frame, and every 150-th frame. As background model we applied a simple approximated median background model [98].

For practical applications, it is not necessary to update the system with every input frame (typically there is a trade-off between run-time and adaptivity to changing environments). However, to demonstrate the benefits and the robustness of our approach, i.e., the avoidance of temporal drifting, we update each classifier within the classifier grid with every single input frame. The overlap of the grid elements within the classifier grid is set to 70% for the pedestrian sequences and to 85% for the car sequence.

3.7.4.1 IMIL Behavior Analysis

To illustrate the benefits of the IMIL (CG-IMIL) approach compared to the CG-OOL we selected the particular case that an object (i.e., a person) is not moving for a long period of time. We picked out a sub-sequence of the *Stillstanding* sequence (see Section 3.7.4.2), where one person is standing at the same position over 450 frames, which results in short-term drifting and analyze the classifier responses (confidences) at one specific position in the image. Frames 550 to 1000 of this sequence are illustrated in Figure 3.27.

The confidence for both, the IMIL approach and the OOL approach as well as the ground truth are shown in Figure 3.28. One can clearly see that the confidence for the IMIL approach stays the same due to the correct updates with the inverse multiple instance learning strategy while the wrong updates with the current input image of the OOL approach lead to decreasing confidence over time. Moreover, it can be seen that for the negative class, i.e., the background the confidence is significantly lower.

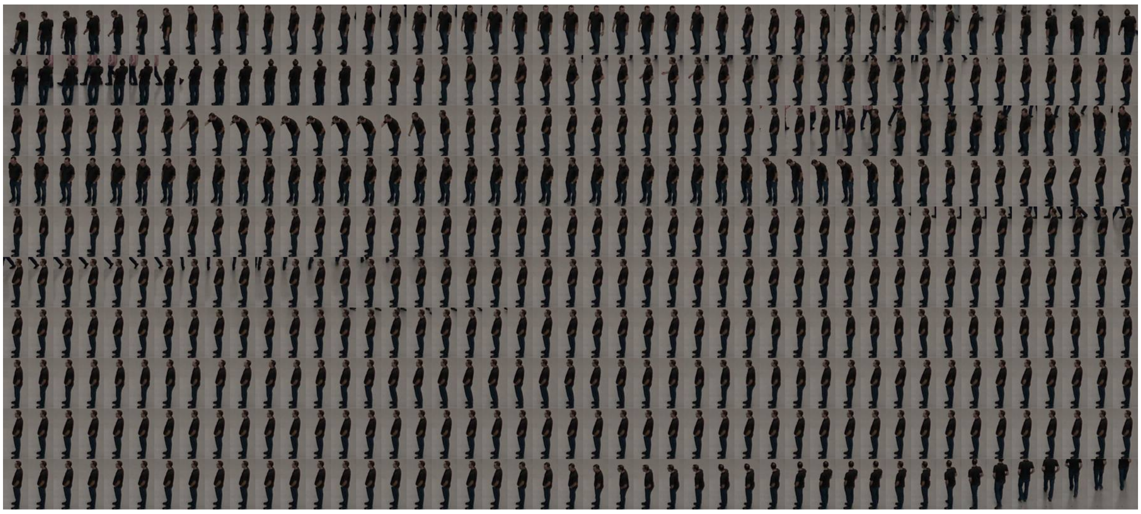


Figure 3.27: One patch of the corridor sequence, showing frames 550 to 1000, where one Person standing at the same position over 450 frames.

3.7.4.2 Stillstanding Sequence

To demonstrate the benefits of the IMIL approach in presence of non-moving objects compared to the classifier grid approach (CG-OOL), we generated a test sequence showing exactly this problem: *Stillstanding* sequence. The sequence showing a corridor in a public building consists of 900 frames (640x480) containing 2491 persons, which are staying at the same position over a long period of time. The results obtained by the CG-IMIL approach and the CG-OOL method are shown in Figure 3.29. We set the overlap between the classifier grid elements to 87%. The classifiers consist of 30 selectors, each of it containing 30 weak classifiers.

Due to the IMIL formulation we get rid of short-term-drifting, the recall can be significantly improved at a reasonable precision level. This is also illustrated in Figure 3.30, where the first row shows detection results of the CG-OOL approach, whereas the second row shows detection results using the CG-IMIL approach. It can clearly be seen that

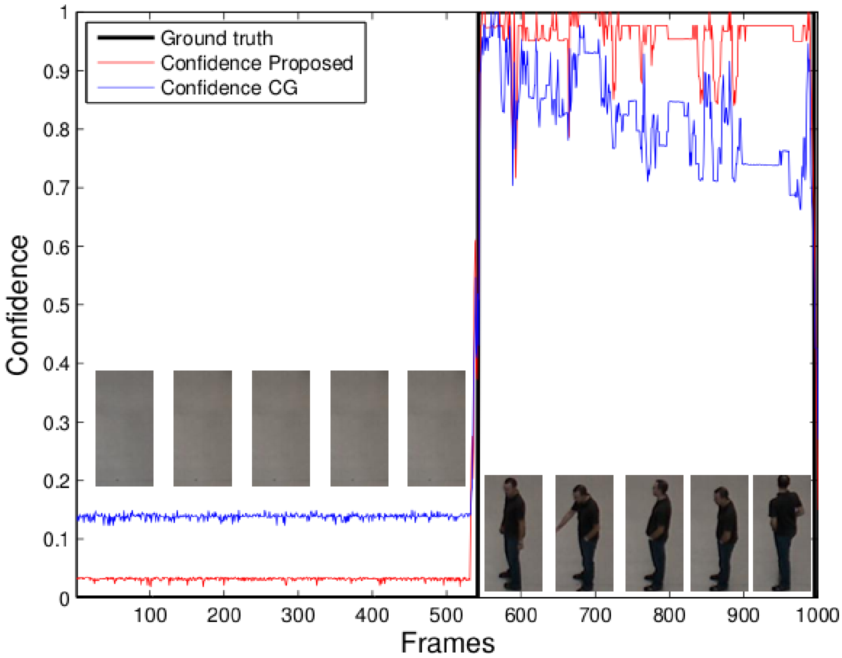


Figure 3.28: Confidence values for the CG-IMIL approach and the CG-OOL approach for a typical scenario: left - background, right - person standing on the same position for a longer period of time.

the person on the right side, standing at the same position over 175 frames, is detected by the CG-IMIL approach whereas it is not in the other case. In addition, Table 3.5 shows the recall and precision for the best F-Measure value.

	R	Pr	FM
CG-IMIL	0.87	0.95	0.91
CG-OOL	0.60	0.59	0.60

Table 3.5: Comparison of Recall and Precision for the best F-Measure value for the CG-OOL and the CG-IMIL approach on the *Corridor Sequence*.

3.7.4.3 Vehicle Sequence

Additionally, we additionally evaluate it on a sequence showing vehicles on a highway: *Vehicle broken* sequence. This sequence consists of 500 frames (720x576), containing 2375 cars. One car broke down within this sequence and is standing at the same position for 400 frames, which would cause short-term drifting for the classifier grid approach with

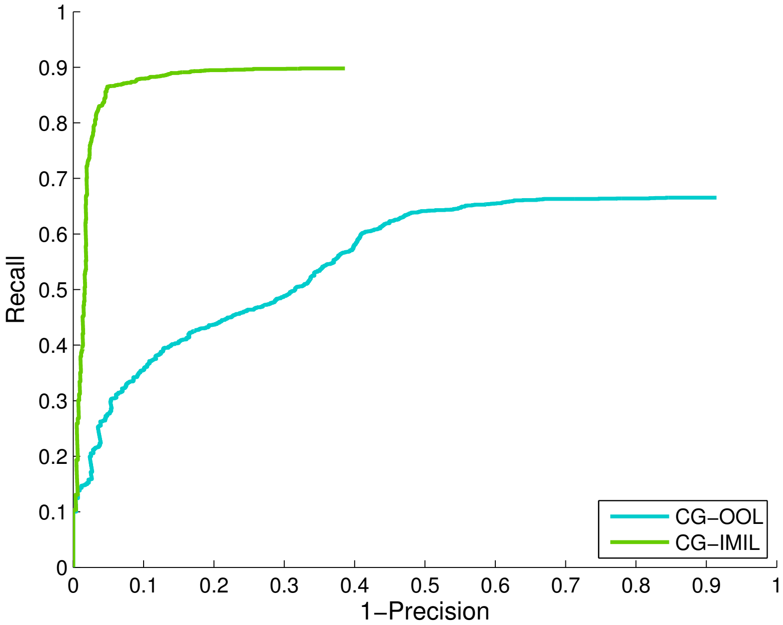


Figure 3.29: Recall-Precision Curves for the *Stillstanding* sequence for the original Classifier Grid and the proposed approach.

fixed negative updates from the scene (CG-OOL). The overlap between the classifier in the grid was set to 87%.

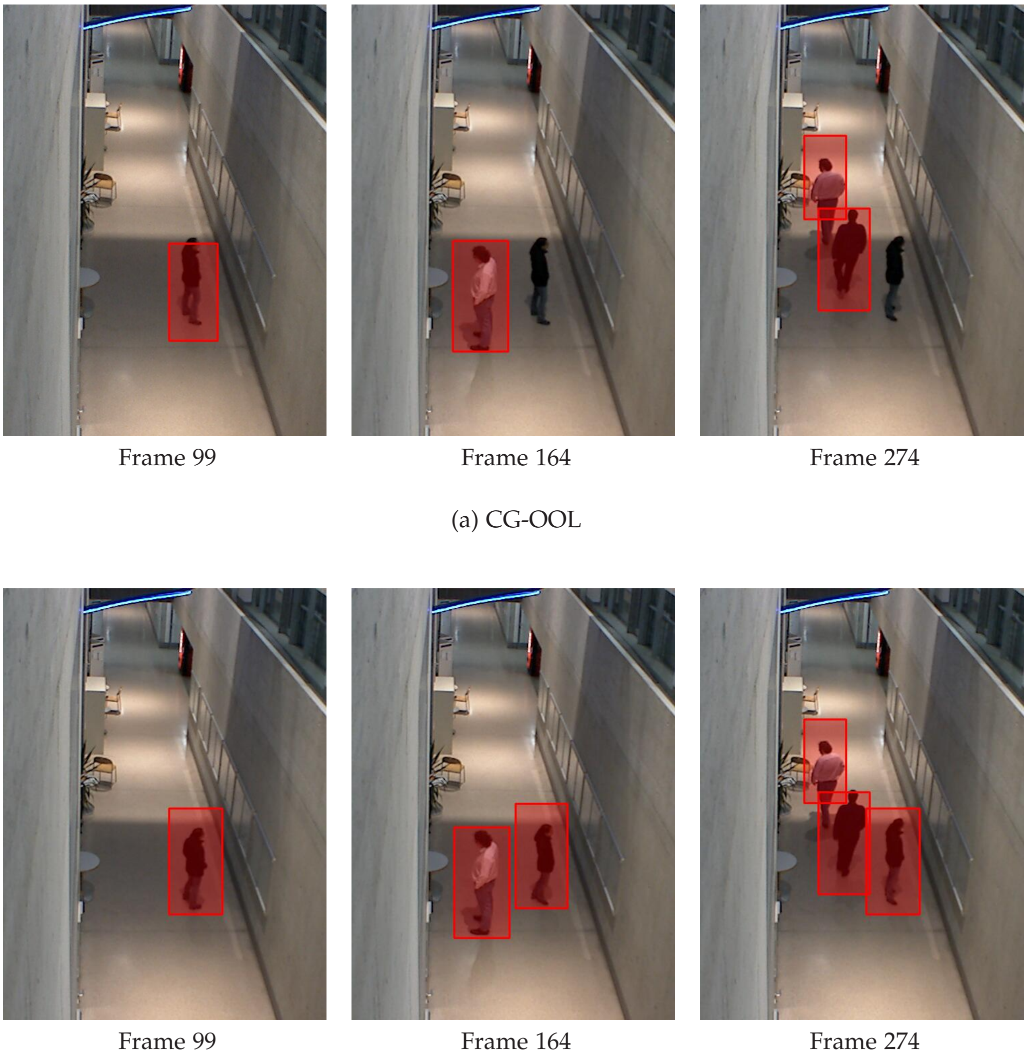
The Recall-Precision curves are shown in Figure 3.31. Again it can be seen that compared to the baseline (CG-OOL) the detection performance can be noticeable improved by introducing inverse multiple instance learning. Illustrative detection results for this scenario are shown in Figure 3.32. Table 3.6 shows the recall and precision for the best F-Measure value.

	R	Pr	FM
CG-IMIL	0.93	0.90	0.91
CG-OOL	0.56	0.90	0.69

Table 3.6: Comparison of recall and precision for the best F-Measure value for the CG-OOL and the CG-IMIL approach on *Vehicle broken* sequence.

3.7.5 Classifier Co-Grids

From the results presented so far it can be seen that classifier grids, in general, provide a considerable alternative to typical sliding window approaches when run on static cameras. The only remaining issue is the question how to incorporate scene specific object



(a) CG-OOL

(b) CG-IMIL

Figure 3.30: Temporal information incorporation by MIL avoids short-term drifting. The original classifier grid approach (first row) temporary drifts after about 60 frames whereas the proposed approach (second row) avoids temporal drifting even after more than 170 frames.

information to further increase the recall while preserving the precision. This is now possible due to the combination of a robust learner (TransientBoost) with a co-training setup. To demonstrate the performance of the classifier Co-Grid approach (CG-CoT) with TransientBoost as a classifier within the classifier grid, we performed two different experiments. The first experiment shows that a comparable performance to the CG-

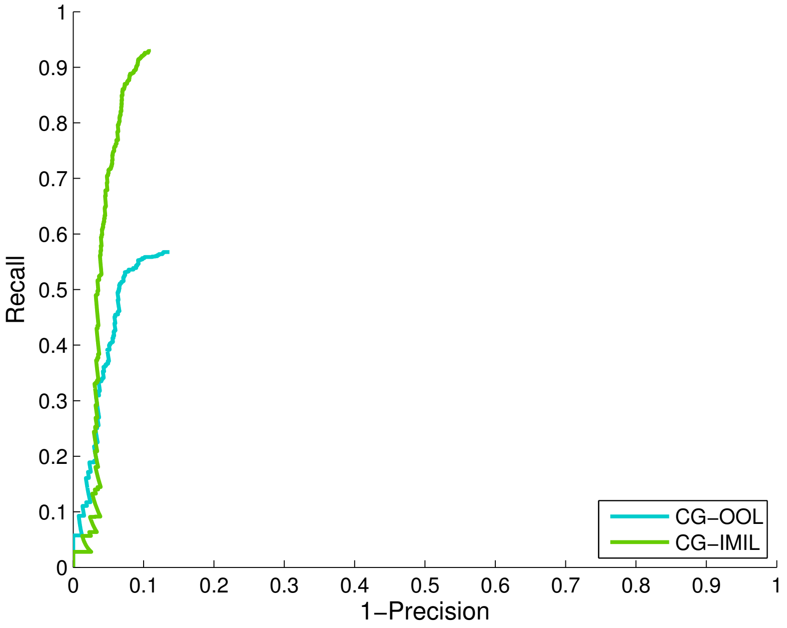


Figure 3.31: Recall-Precision Curves for the *Vehicle* sequence, containing objects that are not moving over a long period of time.

OOL approach can be reached for car detection. The second experiment demonstrates the long-term stability on the long-term dataset, where the detector is updated over more than 580.000 frames. Superior results compared to the CG-OOL approach are reported in Section 3.7.6.

The background subtracted (BGS) images for the co-trained classifier are gained by subtracting from a simple approximated median background model [98]. For the initial classifier operating on the BGS images we use a compact classifier consisting of 20 selectors, each of it containing 10 weak classifiers.

3.7.5.1 AVSS 2007

The *AVSS 2007* dataset is from the "i-LIDS Bag and Vehicle Detection Challenge (AVSS 2007)" [**], where we evaluated on the first 500 frames (720x576 pixels) of the vehicle detection sequence AVSS_PV_Hard, containing 673 cars. This sequence shows a street in a residential area. The complexity (variability in the positive class) requires a larger classifier consisting of 80 selectors, each of them containing 10 weak classifiers.

[**] http://www.eecs.qmul.ac.uk/~andrea/avss2007_ss_challenge.html

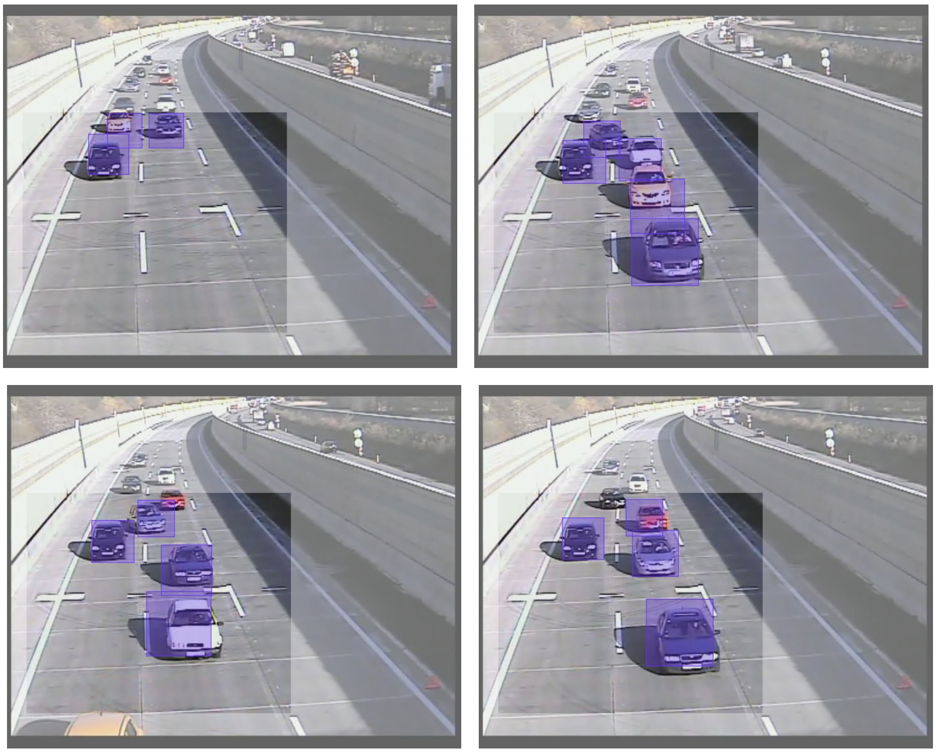


Figure 3.32: Illustrative detection results on the *Vehicle broken* sequence.

The RPCs for the CG-CoT approach, the CG-OOL approach, and DPM-FS are shown in Figure 3.33. It can be seen that the DPM-FS approach can be significantly outperformed. The classifier co-grid approach yields comparable results to the CG-OOL, even given an excellent baseline. Illustrative detection results of the co-grid approach are given in Figure 3.34.

3.7.5.2 Long-term dataset

To show the robustness over time, we run experiments on our publicly available long-term dataset. For evaluation purposes we selected two sequences, one right at the first day, starting after frame 3.390, right at the beginning of the sequence and one at the end of the dataset on the last day with frame 575.000. Both sequences contain 2.500 annotated frames, the first one contains 201 pedestrians, the second one contains 316 pedestrians. To demonstrate the long-term behavior of the different methods all online methods are updated throughout all 580.000 frames. The thus obtained results are presented in form of recall-precision curves (RPC) in Figures 3.35 and 3.36.

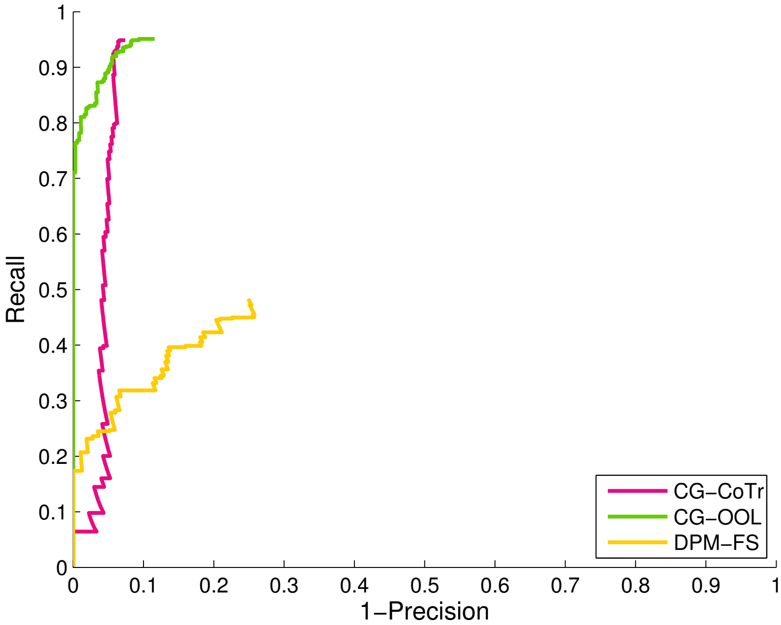


Figure 3.33: RPCs for the *AVSS 2007 Sequence*.

We used classifiers of a size of 20 selectors, each of it containing 10 weak classifiers to initialize the classifiers within the grid. The overlap between the grid elements was set to 80%.

From Figures 3.35 and 3.36 it can be seen that TransientBoost provides more or less the same performance as the CG-OOL approach, which does not use any positive updates. Even though the recall is not increased (this can be explained by the complexity of the scenes) the precision is not decreased by adding unlabeled scene specific positive information. It can be further seen that TransientBoost clearly outperforms GradientBoost, especially in terms of precision. Moreover, on both sequences the static detectors can be outperformed.

3.7.6 Comparison between the proposed approaches

We compare the proposed approaches on two different datasets: the first is the public PETS 2006 dataset showing pedestrians at a train station. The second is dataset is the Yellow ball sequence, which we created on our own, showing a corridor of a public building, where besides pedestrians different other objects are moved through the scene.

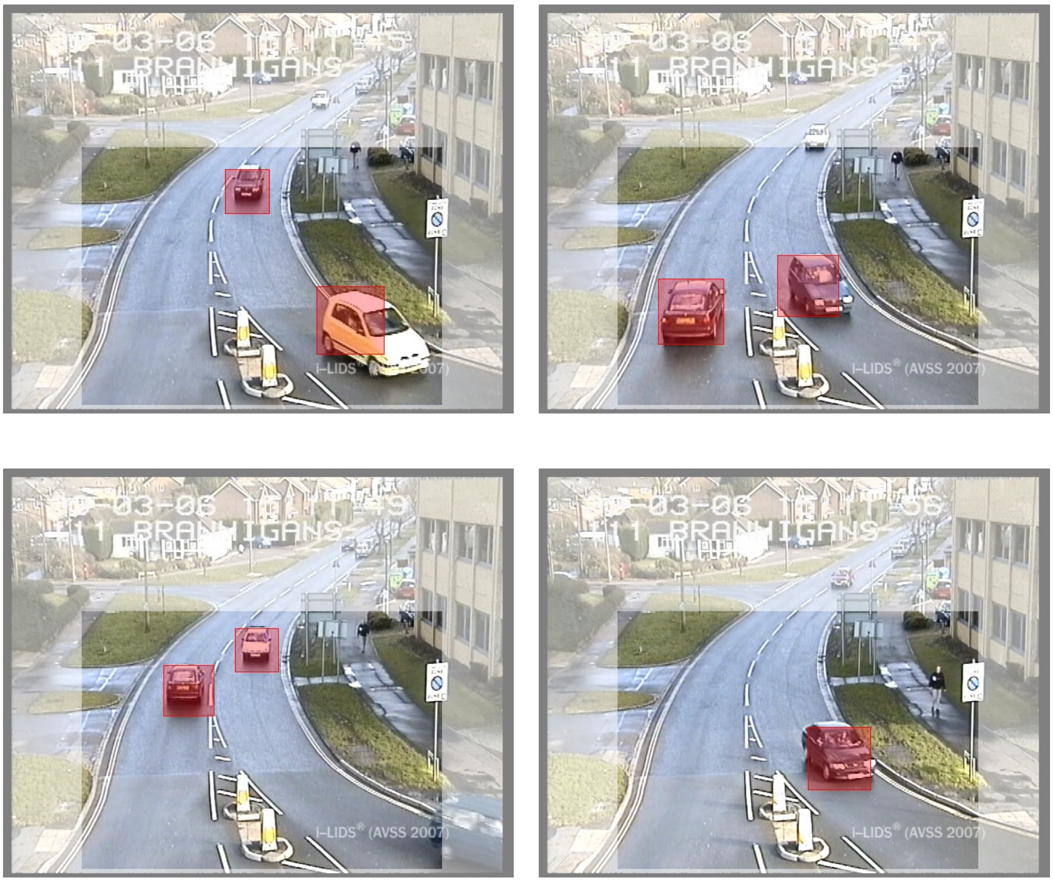


Figure 3.34: Illustrative detection results of our approach for the *AVSS 2007* sequence (Detection results within the fully colored region).

PETS 2006

We compare the different proposed approaches on a publicly available dataset for pedestrian detection, namely the *PETS 2006* dataset [††]. This dataset was released for the PETS 2006 Workshop at IEEE Conference on Computer Vision and Pattern Recognition. It contains left-luggage scenarios at a train station. We evaluate on Dataset S5, Take 1-G, a scenario where one person leaves a ski equipment. The sequence consists of 308 frames with a resolution of 720 × 576 pixels containing 1265 pedestrians.

We compare the classifier grid approach with the off-line on-line linkage (CG-OOL) to the classifier grid approach with the inverse multiple instance learning (CG-IMIL) and the classifier grid approach, where the fixed update strategies are replaced by a

[††] http://www.cvg.rdg.ac.uk/PETS2006/data.html

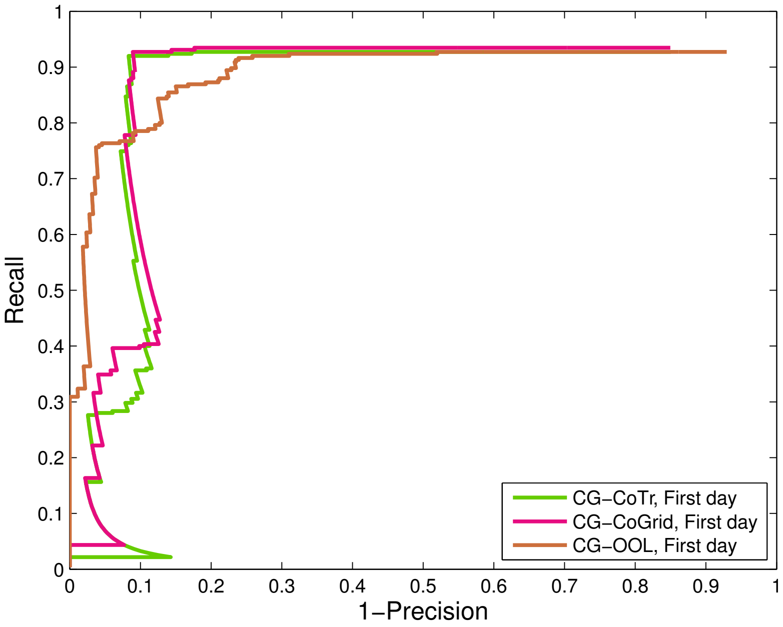


Figure 3.35: RPCs for the first day of the long-term dataset.

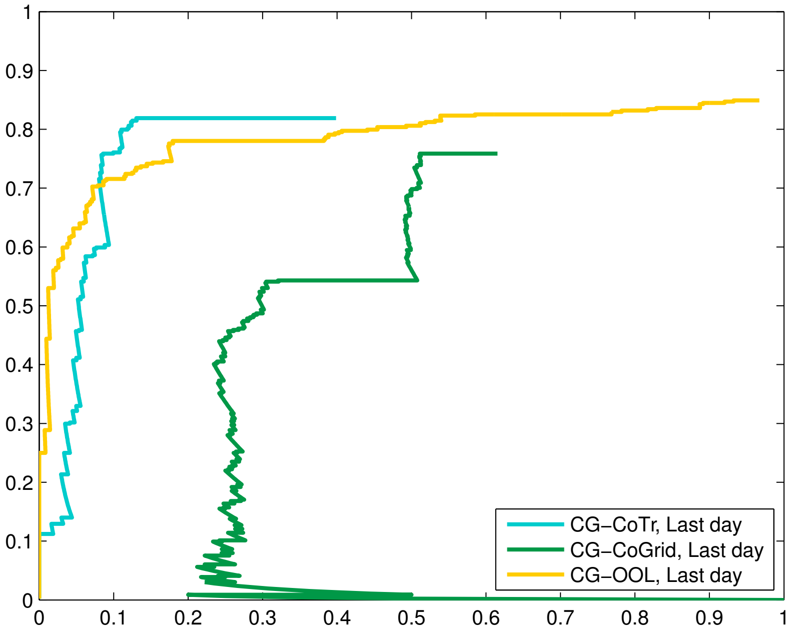


Figure 3.36: RPCs for the last day of the long-term dataset.

co-training approach in combination with the proposed robust TransientBoost learning algorithm (CG-CoT). We performed an additional evaluation, where we skipped the updates from the current scene after an initialization stage of 50 frames for the CG-OOL approach, which means that we don't perform negative updates using the current frame (CG-OOL, only initial updates). One can see that this leads to a significant increase in recall compared to the CG-OOL approach, where after the initial updates we use only every 10-th frame to update our classifier grid approach. In this case, a major problem of the fixed update strategies within the classifier grid occurs, namely "temporal drifting". Temporal drifting occurs, if an object is not moving, i.e., occupying the same position over a while. Within this sequence we have a number of objects standing at the same position for a while, causing a punch of wrong background updates of the classifier. The CG-IMIL approach can be used to guarantee adaptivity (which is not given if the updates are neglected as it is the case for CG-OOL, only initial updates) but avoid temporal drifting. The results of the CG-IMIL approach clearly show that even though we are updating using every single frame, objects are still detected, which is not the case for the CG-OOL approach, caused by temporal drifting. The CG-IMIL approach is able to handle these problems and to increase the recall compared to the CG-OOL approach. The results can be further improved by using co-training and a robust learner instead of the fixed update strategies (CG-CoTr). The co-training approach which performs negative updates on-demand, triggered by the co-trained classifier operating on the background subtracted images avoids short term drifting while the scene specific updates of the object class further increase the performance, resulting in an increased recall compared to all other classifier grid approaches. Additionally, we again compared to the two generic approaches (HOG-DT and DPM-FS), which can be clearly outperformed. The recall and precision for the best F-measure value is shown in Table 3.7. For all classifier grid approaches we set the overlap between the grid elements to 87%.

	R	Pr	FM
CG-CoTr	0.82	0.93	0.87
CG-IMIL	0.75	0.86	0.81
CG-OOL, only initial updates	0.74	0.87	0.80
CG-OOL	0.60	0.77	0.67
DPM-FS	0.50	0.83	0.62
HOG-DT	0.46	0.89	0.61

Table 3.7: Comparison of recall and precision for the best F-Measure value on *PETS 2006* sequence.

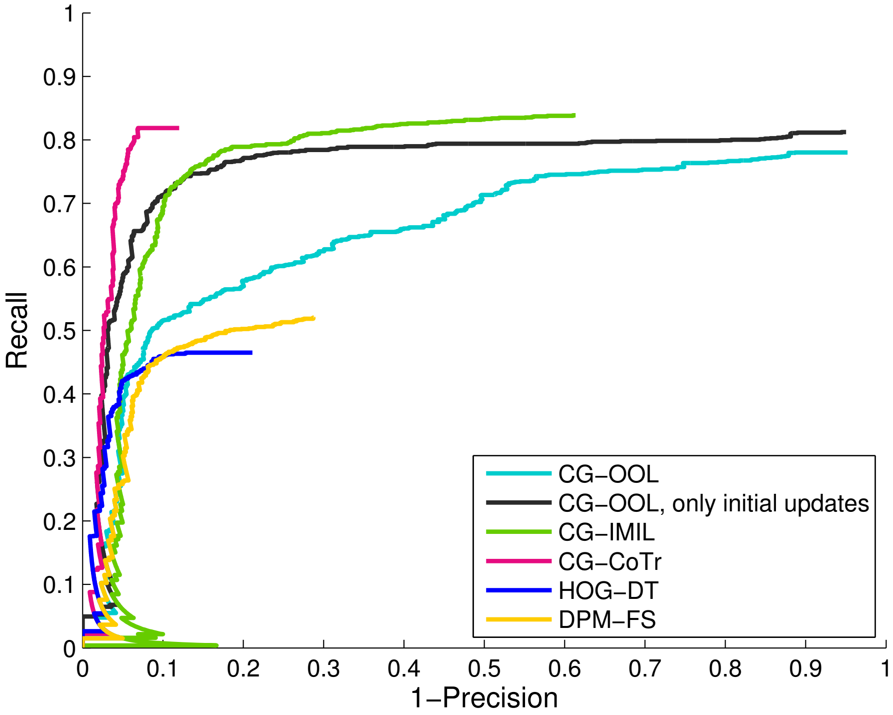


Figure 3.37: Recall-Precision Curves for *PETS 2006* sequence for different state-of-the-art detectors compared to the proposed approaches.

Yellow Ball Sequence

Additionally to the publicly available PETS 2006 dataset we created a challenging test dataset showing a corridor of a public building consisting of 300 frames, which contains 532 persons. The sequence, which was taken at a resolution of 320 × 240, shows various challenges such as different moving objects, like a yellow ball moved through the scene, moved chairs and people passing by carrying an umbrella. We refer to this sequence as *yellow ball sequence*. This scene further contains a large number of self-occlusions of the persons caused by the viewpoint of the camera.

In addition to the two state-of-the-art generic approaches for pedestrian detection (HOG-DT and DPM-FS) we compared the proposed off-line on-line linkage approach (CG-OOL) to the initially proposed classifier grid approach with fixed update strategies without off-line training (CG-FUS) [67]. Additionally, we compare to the proposed inverse multiple instance learning approach (CG-IMIL) and to the proposed classifier co-grid approach (CG-CoTr). To ensure satisfactory results for the HOG-DT detector and the DPM-FS detector, we resized the input images to 640 × 480 for these two approaches to better fit the learned object size. Additionally, we compared to low level methods, i.e., a simple approximate median background model (BGM) [98], template matching (TM), and a combination of both (TM+BGM), which might be considered a simple pendent to the CG-OOL method, combining one model describing the object of interest with one model describing the background.

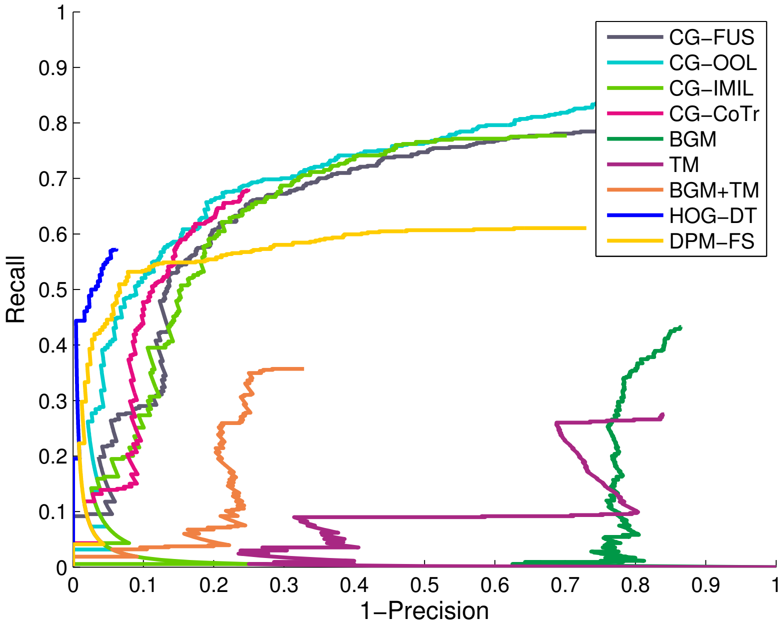


Figure 3.38: RPC for the *yellow ball sequence*.

	R	Pr	FM
CG-OOL	0.67	0.79	0.73
CG-CoTr	0.66	0.79	0.72
HOG-DT	0.57	0.94	0.71
CG-FUS	0.66	0.75	0.70
CG-IMIL	0.65	0.75	0.70
DPM-FS	0.53	0.92	0.67
BGM+TM	0.36	0.73	0.48
TM	0.26	0.31	0.28
BGM	0.34	0.22	0.27

Table 3.8: Detection characteristics of the *yellow ball sequence* for different methods sorted by decreasing F-measure.

The results are summarized in Figure 3.38 and in Table 3.8, where we show the recall-precision curves for all methods and the corresponding detection characteristics.

From these results it can be seen that the generic detectors show an excellent precision, but that the recall is lower compared to the classifier grid approaches. For the classifier grid approach with fixed update strategy (CG-FUS) a larger classifier with 100 selectors, each of them containing 30 weak classifier was required. In contrast, for the off-line trained classifier (CG-OOL) a size of 20 selectors, each of it containing 10 weak classifiers was enough to reach this performance, because during the off-line training suitable features for the task of pedestrian detection have been selected. Besides a run-time performance gain the amount of required memory can be reduced significantly. The low level cues totally fail, i.e., for the background model (BM) and the template matching (TM) both, the recall and the precision are very poor, such that even a combination of both (TM+BGM) yields insufficient results. The continuous poor precision for the background model rises from a few false positives with a high confidence. The results of all proposed classifier grid approaches for this sequence are comparable for two reasons: First, this sequence does not contain any objects standing at the same position for a while. Hence, short-term drifting does not occur. Second, the appearance of the objects is already captured well by the classifiers. Hence, new scene specific object information does not increase the recall for this particular sequence. Typical results of the CG-OOL approach are depicted in Figure 3.39.

3.8 Summary

Within this section we presented a number of approaches for object detection from stationary cameras. All approaches aim for transferring an initially generic detector to a specific scene and to constantly adapt to changing conditions. This can be seen in the light of transfer learning. All approaches have in common that in the first step a generic boosted classifier is trained, this generic classifier is replicated in a way that at each position where a classifier is evaluated in the image a separate classifier is located. This classifier has then to discriminate between the object of interest and the background at one specific location. By using different update strategies and different learning algorithms these classifiers are modified over time to ensure scene adaptation of our object detection framework. Even though the updates are completely unsupervised, empirical results on long-term evaluations demonstrated that all these approaches are robust over time.

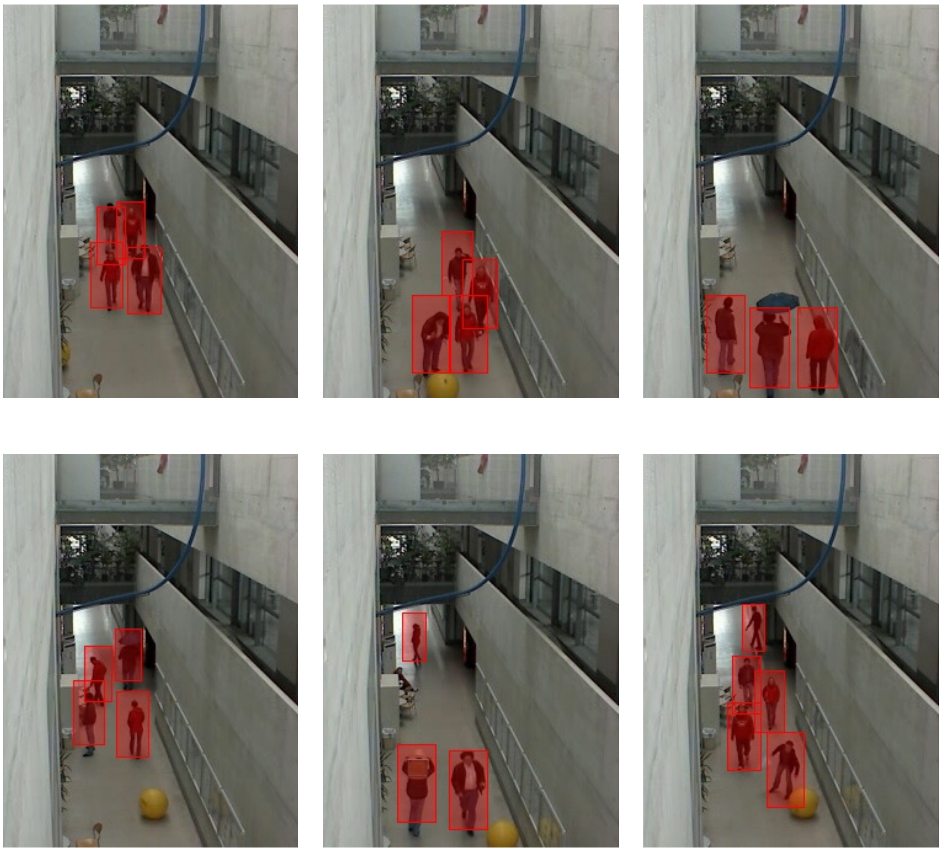


Figure 3.39: Illustrative detection results of the grid-based person detector for the *yellow ball sequence*.

Out of clutter, find simplicity.

Albert Einstein

4

Robust Scene Adaption within a Multiple Camera Setup

Contents

The approaches presented so far are robust and well suited for real-world conditions. However, spatial-temporal trajectories have not been exploited. In order to interpret complex behavior of individuals and to make one step further towards fully autonomous video surveillance we investigate in object tracking to get spatial-temporal trajectories, which are an important cue for autonomous video surveillance.

4.1 Introduction and Problem Statement

Motivated by numerous applications, such as visual surveillance, sports analysis or industrial applications, considerable research activity has been made in the area of object tracking from video sequences. Spatial-temporal trajectories can be used to interpret complex behavior of individuals and to help autonomous systems to interact with complex environments. Thus, there is still a high scientific interest and various successful methods have been proposed (e.g., [25, 12, 10, 62]). The task of object tracking becomes even more challenging, if the scene contains multiple interacting objects which occlude each other. This especially applies for person tracking, when a large number of persons are occluding each other and the positions of single instances cannot be robustly identified (e.g.,[24, 27, 43]). Tracking an individual is fairly easy as long as the individual is isolated, i.e., it is not occluded by any other individual or any other distractor in the scene. An increasing density of interacting persons significantly increases the complexity of the problem. One way to overcome this problem is to take advantage of multiple cameras. A multiple camera setup gives the advantage of resolving occlusions which occur in one view by exploiting the information of another view, as visualized in Figure 4.1.

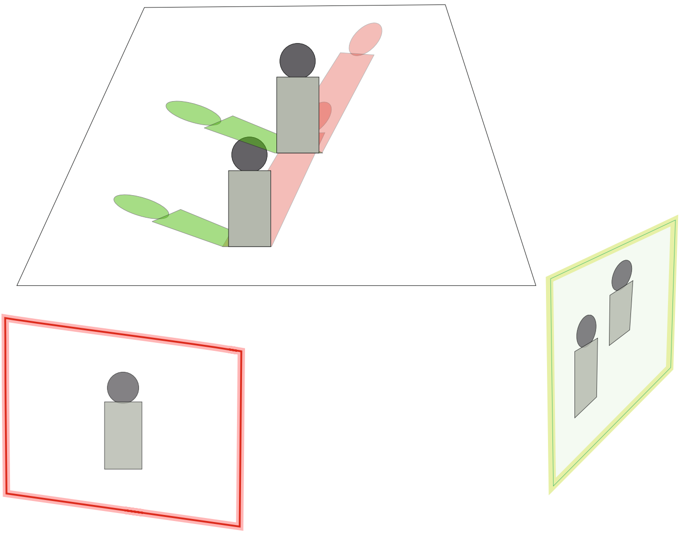


Figure 4.1: Multi-camera tracking exploits the fact that occlusions occurring in one view might be not present in another view.

4.2 Related work

In the following, we first describe approaches for multi-object tracking from single stationary cameras. Second, we briefly introduce the concept of homographies and describe homography based multi-camera multi-object tracking approaches.

4.2.1 Monocular Multi-object Tracking

Multi-object tracking is a challenging problem, especially if the objects are interacting with each other. There are numerous works dealing with the challenging problem of multi-object tracking from static cameras [86, 75, 95, 145, 83, 141, 78] as well as from moving cameras (e.g., [95, 101, 51]. However, in this work we focus on static cameras.

Common approaches for multi-person tracking from a single, static camera base the occlusion reasoning on the appearance of the individual objects. Numerous methods are based on Bayesian inference to deal with occlusions [141, 83, 100, 78]. Isard and MacCormick [78] proposed a particle filtering implementation of a Bayesian multi-object tracker. The main problems of this tracker are identity switches caused by a single foreground model for all objects. Zhao and Nevatia [145] also formulated the problem of object detection and tracking as Bayesian inference. They proposed a Markov Chain Monte Carlo (MCMC)-based method for multi-object tracking from a stationary camera. A color-based joint likelihood in combination with an MCMC-based approach enables to perform detection and tracking of multiple objects and to compute an optimal solution. Another approach based on MCMC sampling was proposed by Khan et al. [83], where MCMC was incorporated into a particle filter framework to replace the importance sampling step of the particle filter.

Recently Kuo et al. [86, 87] proposed to on-line learn a discriminative appearance model for each individual person. In [87] training samples are collected on-line by using spatio-temporal constraints. An AdaBoost classifier is trained based on the on-line collected training samples to discriminate between individual persons. Instead of learning global models for all tracklets as described in [87], a target-specific appearance model is learned in [86]. They incorporate person identity recognition to their tracking framework to discriminate between different persons by on-line learning the appearance of each individual object. Based on detections they generate short tracklets and build query tracklets and gallery tracklets, where for each gallery tracklet an appearance model is learned. Finally, the gallery and query tracklets are merged within a hierarchical association framework. Based on [86] Yang and Nevatia [143] learn non-linear motion maps, entry/exit maps, and additionally identify moving groups. In [144] on-line part-based appearance models are trained to explicitly deal with occlusions. The part-based appearance models are used to provide detections which deliver tracklets. To avoid to solely rely on detections and to reduce the influence of missed detections, a category free tracking is additionally used. A global appearance model for the tracklet association is learned.

Hu et al. [75] explicitly defined occlusion relationships and integrated them into the tracking framework by a particle filtering approach. Foreground regions are identified by a background subtraction and compared to motion regions in the previous frame. If objects are occluding each other they are tracked as a single object as long as the motion region is divided again. An appearance model based on a simple shape, color and motion model is used to re-identify the objects.

Another way to incorporate appearance and motion information for multi-object tracking is to use generalized minimum clique graphs [4], where a global approach is used for the data association. The whole video sequence is divided into subparts which are analyzed separately. The tracker aims at associating given detections by building a graph, where the nodes are the detections and edges exist to all detections in previous and subsequent frames, i.e., there are no edges between detections of the same frame. The weights are calculated based on the appearance of the detections. The association is done by solving a generalized minimum clique graph.

There are various other approaches that aim at performing data association between detections, which cause problems in case of occlusions and may lead to missed detections. To avoid these problems, Fragkiadaki et al. [51] introduced two levels of tracking granularities, where detections are used if the objects are mostly visible while additional point trajectories are used during partial occlusions. Tracking and detection is also combined in [142], where within a max-margin framework detection confidence, appearance affinity, geometric information as well as motion information is combined to decide whether to detect or to track the objects. The tracker and the detector are considered independently.

4.2.2 Homographies and Multiple camera multi-person tracking

Having only a single camera raises lots of problems in the field of multi-person tracking, since a lot of information is simply not available from a single viewpoint as shown in

Figure 4.1, which makes it hard to resolve occlusions. By having a multi-camera setup advantages like geometric constraints available in real-world problems can be exploited. Typical multi-camera setups assume overlapping cameras observing the same 3D scene, which allows for exploiting so called closed-world assumptions [76] such as people are moving on a common ground plane (e.g., [50, 42, 82, 84, 121, 18]).

A general multi-camera tracking setup consists of n overlapping cameras, where each of them is observing the same 3D scene. Each camera view v has its own local image coordinate system $\{x_v, y_v\}$, which can be mapped to the common world coordinate system $\{X, Y, W\}$, requiring a fully calibrated setup. By exploiting the closed-world assumption that the objects-of-interest are mainly moving on a common ground plane, which is valid for most real-world tracking applications, a simple plane-to-plane homography can be used. Thus, the mapping of an image point $\mathbf{x}$ from a camera v onto a corresponding world point $\mathbf{X}$ on the ground plane can be realized by a simple plane-to-plane homography:

$$\mathbf{X} = \mathbf{H}_v \mathbf{x} \, , \tag{4.1}$$

where $\mathbf{H}_v$ is a homogeneous 3×3 matrix. Both, world and image coordinates, are given as homogeneous 3×1 vectors $\mathbf{X} = (X, Y, W)^\top$ and $\mathbf{x} = (x, y, 1)^\top$, respectively. The plane-to-plane homography defines the transformation up to scale. Hence the matrices $\mathbf{H}_v$ have only 8 degrees of freedom. A normalization can be used to compute real-word positions [72]

$$\widetilde{\mathbf{X}} = (\widetilde{X}, \widetilde{Y})^\top = (X/W, Y/W)^\top . \tag{4.2}$$

There are various approaches, which exploit the multi-view information by selecting one specific view out of all available views or by grouping the views into pairs [141, 100, 85]. Wu et al. [141] proposed an approach based on a dynamic Bayesian network with multiple hidden Markov chains. Their method is applicable to single and multi-view scenarios. The multi-view information is exploited by switching between the views. Mittal and Davis [100] propose a multi-camera approach for detection, segmentation, and tracking of multiple people in cluttered scenes. Instead of using simple background subtraction, color models for objects and background are used to get the foreground maps for each view. Multiple views are grouped into pairs of views and evidence for an object is gathered from pairs of cameras. Each pedestrian is modeled by color models at different heights and presence probabilities along the horizontal direction at different heights. To model the color distribution at different heights a non-parametric Gaussian kernel estimation technique is used. As presence probability they use a model depending on the width and height of a person. The idea of grouping views into pairs has already been proposed in [85], where background subtraction is used to identify blobs and color histograms are created to identify the individual persons.

Another way to combine information from multiple views is to exploit real-world constraints like a common ground plane. In general, multi-camera multi-object tracking approaches first apply change detection to get binary foreground/background masks [50, 82, 84, 100] or a fixed pre-trained classifier [19, 18] to estimate the foreground likelihood of specific pixels. Then, this information is fused exploiting the common ground plane by either estimating a score map [50, 82, 19] or by estimating axes intersections [84]. Since homographies can easily be estimated (e.g., by extracting SIFT points and running RANSAC), there has been a considerable interest for applying homography-based techniques for multi-camera tracking. Fleuret et al. [50] start with a simplified background subtraction and then build a generative model describing persons as rectangles. This is used to estimate a joint probability of occupancy for each frame and position on the ground plane which is referred to as probability occupancy map (POM). By additionally using a color and a motion model the trajectories of multiple persons can be estimated. To avoid that the score map is polluted by other moving objects, [19] applied a detector instead of a simple background subtraction. A different way for linking detections is by using K-shortest path optimization [18]. To reduce the number of identity switches appearance information has been incorporated [121].

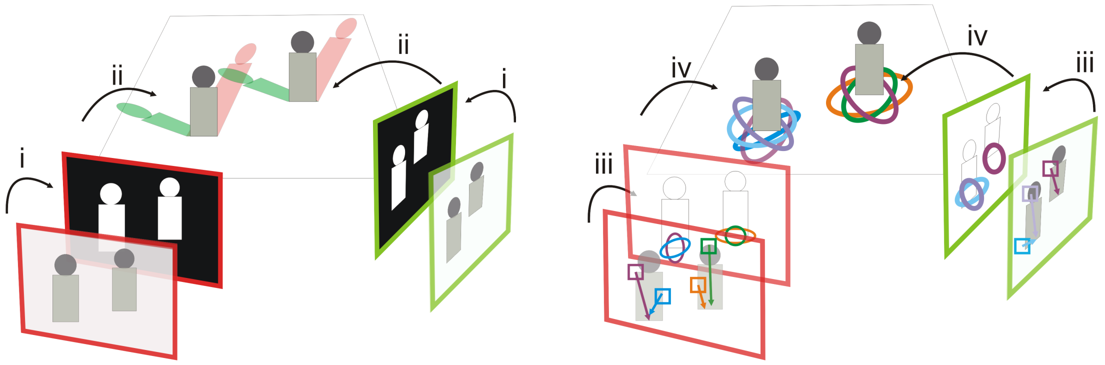
Khan and Shah [81] first obtain foreground likelihood maps for each view by applying a mixture of Gaussian model. These likelihood maps are then projected onto the ground plane using the given homographies and are accumulated into a synergy map. The synergy map is thresholded yielding the approximate feet positions of the persons, which are then back-projected into each view. The actual tracking is then performed using a look-ahead technique on the previously estimated foot-point positions at the ground plane. To make the tracking more robust, they slightly extended this approach by sweeping over multiple planes parallel to the ground plane, to handle inaccurate projections [82].

Using the ground plane assumption for multi-camera tracking has two main disadvantages. First, the foot-points are often not visible due to occlusions and second, in different camera views the foot-points are not well defined (e.g., frontal vs. side view).

Thus, Eshel and Moses [41, 42] track the heads of persons. Moreover, similar to [82] they also sweep over different planes to capture persons of different heights. The main drawbacks of these approaches are that the heads must be visible in all views, that several planes have to be calibrated in parallel, and that the camera positions are limited to deep viewing angles. To overcome the problem of inaccurately estimated foreground maps, Kim and Davis [84] proposed to use the intersection of vertical axes estimated from the foreground blobs to obtain a common localization in the top view. Starting with background subtraction they iteratively run a color segmentation step to get an improved foreground map. As the thus obtained foot-point might be inaccurate due to segmentation errors, the intersection of vertical axes of the estimated blobs is used to obtain a common localization in the top view. The actual tracking is then performed on the top view by applying a particle filter framework.

These methods, however, ignore several important issues hampering their applicability. First, detection and segmentation errors in the original views are projected onto the ground plane and have to be handled in the common view. Second, in general simple geometric transformations are only valid for a single ground plane, which results in wrong projections for points not lying on the ground plane. Third, using a pixel-wise projection ignores imperfect localization in the different views and (minor) uncertainties in the homography. Altogether, this results in an inaccurate localization, making it hard to estimate an adequate back-projection into the single view or could cause ghost projections (i.e., a detection coming from the intersection of two unreliable projections) as shown in Figure 4.2(a).

To overcome these limitations, we introduce a new multiple camera tracking approach, which extends the ideas of generalized Hough voting [14] and implicitly deals with often ignored uncertainties in the projection. Therefore, we introduce a new Hough voting scheme which relates all foreground probabilities to a position on the ground plane. In this way the geometry information is preserved and the voting results can be fused over multiple cameras implicitly considering uncertainties in the projection and still preserving the beneficial properties such as robustness to occlusions. Additionally we improve the detection results for each individual camera view by an on-line scene adaption using a geometric verification and back-projection between views, which further improves the tracking results over time. On top of the fused vote map we use a particle filtering approach exploiting geometric information to avoid overlapping particles on a top view map of the common ground plane.



(a) Common approach to multi-camera tracking: based on background subtraction (i) the foreground pixels are projected to the common ground plane (ii), which may cause ghost detections and requires complex reasoning.

(b) Multi-camera tracking by joint Hough votes: foot-point voting to the ground plane generates a Hough map of each camera (iii), which are projected onto a common ground plane (iv) implicitly considering the geometric uncertainties.

Figure 4.2: Comparison between common approaches to multi-camera tracking based on background subtraction (a) and our proposed approach based on joint Hough voting (b) onto a common ground plane.

4.3 Multi-Camera Hough based tracking

The good performance of Hough voting approaches like Hough forests [56, 104] for object detection and object tracking motivated us to propose a multi-camera Hough voting approach for multi-object tracking. Very often, multi-camera approaches rely on homography projections, where however, uncertainties within the projections are often not considered. The proposed Hough voting approach allows for implicitly considering uncertainties of the homography projections. In general, Hough voting approaches vote to the centroid of the object of interest. Considering the task of multi-camera pedestrian detection, different homographies depending on the height of the person would be required. However, exploiting the closed-world assumption that pedestrians are moving on a common ground plane, voting to the foot-point of a pedestrian only requires for one plane-to-plane homography per camera for the common ground plane to fuse the results from different views on one common Hough voting map.

This common Hough voting map can be interpreted as continuous confidence map and can be used for particle filter based tracking. Since objects cannot overlap each other on one position at the top view map (Hough voting map), this closed world assumption can be exploited within our particle filtering approach.

As already shown for object detection from single stationary cameras including scene specific information is beneficial for simplifying the problem. Therefore, we can exploit geometric information in combination with the reliable tracking results given by our particle filtering approach for view-specific updates of the Hough forests. This can be used to reduce noisy votes from the individual views and to incorporate scene specific information.

4.3.1 Uncertainties in Homography Projections

When projecting image points **x** from perspective images to world points **X** onto the ground plane using a plane-to-plane homography $\mathbf{H}_v$ one has always to deal with uncertainty of these measures [31]. This uncertainty is influenced by two possible error sources, namely the uncertainty of the homography Σ_{H_v} and the uncertainty of the image point Σ_I resulting from the uncertainty in detection of the image point **x** in the image. This is illustrated in Figure 4.3.

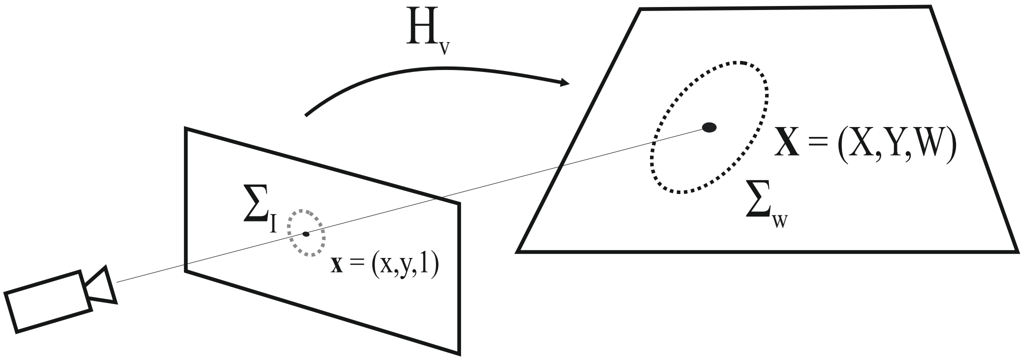


Figure 4.3: Uncertainty in homography projections: Since not all projected points lie on the ground plane and the geometry is often unreliable in practice uncertainty must be considered when projecting points onto the ground plane.

Following [72] the uncertainty Σ_X of a world coordinate **X**, computed by the projection of an image point **x** using homography $\mathbf{H}_v$ is analytically given by

$$\Sigma_X = J_{H_v}\Sigma_{H_v}J_{H_v}^\top + J_I\Sigma_I J_I^\top \ , \tag{4.3}$$

where

$$J_I = \frac{\partial \mathbf{X}}{\partial \mathbf{x}} = \frac{1}{W}\begin{bmatrix} \boldsymbol{h}_1^\top & -X\mathbf{h}_3^\top \\ \boldsymbol{h}_2^\top & -Y\mathbf{h}_3^\top \end{bmatrix} \tag{4.4}$$

and

$$J_{H_v} = \frac{\partial \mathbf{X}}{\partial \mathbf{h}} = \frac{1}{W} \begin{bmatrix} \mathbf{x}^\top & 0 & -X\mathbf{x}^\top \\ 0 & \mathbf{x}^\top & -Y\mathbf{x}^\top \end{bmatrix} \tag{4.5}$$

are the Jacobian matrices, and $\mathbf{h}_i^\top$ is the i-th row of $\mathbf{H}_i$. Assuming that the correspondences for computation of the homography were accurately chosen, the uncertainty in Σ_{H_v} can be neglected. Thus, the uncertainty Σ_X at the ground plane position simplifies to

$$\Sigma_X = J_I \Sigma_I J_I^\top \,, \tag{4.6}$$

where the uncertainties in image coordinates

$$\Sigma_I = \begin{bmatrix} \sigma_{x^2} & \sigma_{xy} & 0 \\ \sigma_{xy} & \sigma_{y^2} & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.7}$$

are derived from the inaccuracy in the image points.

However, these uncertainties are often ignored by existing multi-camera approaches (e.g., [50, 82, 84]). In the following, we introduce a multi-camera tracking approach building on the idea of generalized Hough voting [56, 104] that implicitly copes with these problems.

Recently, several approaches based on generalized Hough transform (e.g., [88, 106, 104, 58, 113, 105]) have shown excellent detection performance. After evaluating the Hough transform based detector on each individual view we use the camera-to-ground plane homographies to map the obtained votes onto the common top view map. This principle is depicted in Figure 4.2(b). Then, we use a multi-object particle filtering approach, which exploits the closed-world assumption that one position on the ground plane can only be occupied by a single object. The tracking results can then be used to improve the combined vote maps by a novel view specific update scheme exploiting the geometric information to reduce the voting noise. In the following, we will use the terms top view and ground plane interchangeable.

4.3.2 Multi-camera Hough Voting

We assume calibrated and synchronized static cameras monitoring one common area, which is a typical surveillance setup. To cope with projection errors we implicitly formulate the uncertainty in the world points Σ_X via Hough voting maps. Recently, Hough forests [56, 104] show excellent performance on object detection. Hence, we use them as a starting point. Hough forests are random forests (see Section 2.3.2) adapted to perform a generalized Hough transform. Each tree directly optimizes the Hough voting performance. They learn a mapping from image features onto a Hough space, where the votes are accumulated. One advantage of Hough transform approaches is their robustness to partial occlusions caused by its additive nature. Even if not all parts of an object are visible, there is still a peak visible in the Hough space. Each Hough Forest $\mathcal{F}$ consists of a set of trees $\mathcal{T}$, where each tree $\mathcal{T}$ is constructed based on a set of patches $\mathcal{P}_i = (\mathcal{I}_i, c_i, \mathbf{d}_i)$; $\mathcal{I}_i$ is the appearance of the patch, c_i is the class label of the patch, and $\mathbf{d}_i$ is the offset vector of the patch with respect to the object's centroid.

During training, two different kinds of uncertainties are optimized: the class label uncertainty $U_1(\mathcal{P}) = |\mathcal{P}| \cdot Entropy(c_i)$ and the offset uncertainty $U_2(\mathcal{P}) = \sum_{i:c_i=1} (\mathbf{d}_i - \mathbf{d}_{\mathcal{P}})^2$ are optimized. The class label uncertainty enforces binary tests used as split criteria during the tree construction to consider the impurity of the class labels c_i, whereas the offset uncertainty enforces to group patches coming from a local environment. Finally, in a leaf node L the vote vectors $D_L = \{\mathbf{d}_i\}$ of the object patches and the foreground probability C_L are stored. Each leaf node in the tree can be considered as a discriminative codebook. During testing for a patch at position $\mathbf{y}$ the probability $p(E(\mathbf{x})|L(\mathbf{y}))$ is estimated, where $E(\mathbf{x})$ indicates whether an object is present at location $\mathbf{x}$ and $L(\mathbf{y})$ is the corresponding leaf node where the patch sampled at position $\mathbf{y}$ ends up. For each tree $\mathcal{T}$ the probability can be estimated as

$$p(E(\mathbf{x})|L(y)) = p(\mathbf{d}(\mathbf{y}) = \mathbf{y} - \mathbf{x}|c(\mathbf{y}) = 1, L(\mathbf{y})) \cdot p(c(\mathbf{y}) = 1|L(\mathbf{y})). \tag{4.8}$$

The first term can be approximated by a Parzen window based on the offset vectors. The second term is the proportion of object patches C_L in a leaf node L at training time. The probabilities for each location $\mathbf{y}$ within the image are accumulated into a Hough map $\mathbf{V}$ over all trees $\mathcal{T}$ within the Hough forest. The actual detection task is finally performed by mode seeking in the thus obtained Hough map.

This idea can be extended for multiple camera views. In fact, we can generate a common Hough map, where the single view maps are accumulated by projecting them onto a common plane using the homographies. However, as described above such projections are prone to uncertainties Σ_X of a world point X resulting from the uncertainties of the corresponding image point $\mathbf{x}$. In our case an image point $\mathbf{x}$ is associated with a patch representation $\mathcal{I}_i(\mathbf{x})$ and the uncertainty results from inaccurately estimated endpoints of the vote vectors D_L, where L is the leaf node where $\mathcal{I}_i(\mathbf{x})$ ends up. The uncertainty Equation (4.7) could be calculated over the endpoints of the vote vectors D_L within each leaf node. Alternatively, the uncertainty of the statistical distribution can be approximated by Monte-Carlo simulation [72]. For our Hough voting scheme, however, this is already estimated within each of the leaf nodes L by the offset vectors D_L. Thus, we can implicitly handle the uncertainty in the projection to the common top view map.

The approach described so far builds on voting to the centroid of an object, as it is common in Hough voting approaches. Considering our intended application, i.e., multi-camera object tracking, the centroid voting would require a large calibration effort. In fact, depending on the vote center (and therefore depending on the height of the person) different homographies would be required. Assuming that objects are moving on a common ground plane this large calibration effort can be avoided. Therefore, we modify the voting scheme: Instead of voting for the objects centroid we propose to *vote for the foot-point* of the object. Since we know the plane-to-plane homography we can estimate the extent of the objects at all positions within the image. This information can be exploited to avoid a large evaluation effort by using the appropriate scale at different positions within the image as illustrated in Figure 4.4.

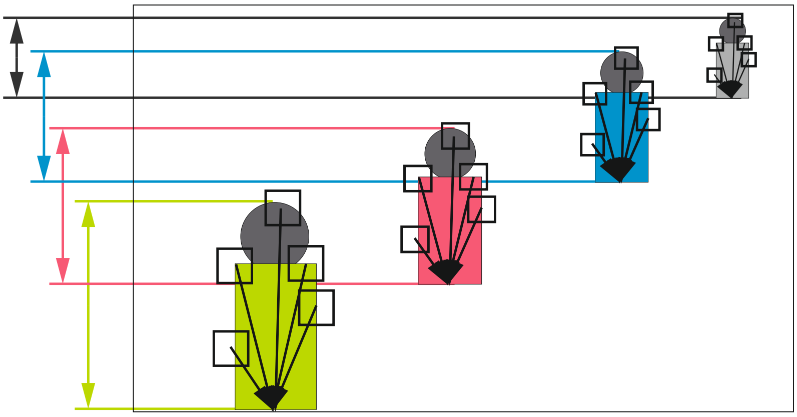


Figure 4.4: To avoid too large evaluation effort the known calibration can be used to evaluate with different scales depending on the position within the image.

Besides the implicitly handled uncertainties the joint multi-camera Hough voting enables the late fusion of detection information. Thus, all information is kept for tracking and we do not have to discard possibly useful information at a too early stage. Compared to approaches solely relying on background subtracted images (as illustrated in Figure 4.2(a)) our detection based approach has further the advantage of not relying on changes compared to the background image. We only consider the objects-of-interest while other moving objects are ignored.

4.3.3 Multi-Camera Tracking

The common top view voting map **V**, visualized in Figure 4.5, can now be used for multi-object tracking. Following the Hough voting scheme, we retrieve high votes on ground plane positions where an object-of-interest is localized. Although **V** does not express probabilities, it can be seen as a continuous confidence map. In contrast to existing single view approaches, the proposed ground plane Hough voting ensures non-overlapping local maximum for each possible detection, since on the top view objects cannot overlap each other.

In particular, we use a particle filtering approach [77], which is widely used for tracking and provides a probabilistic framework for maintaining multiple hypotheses of the current object state. Particle filtering can be used to estimate the state of a system based on noisy measurements **z** by using a set of S weighted particles $\mathbf{x}^i_{1:k}, w^i_{1:k}$. In our case, given the set of S weighted particles $\{x^i_t, w^i_t\}$, $i = 0, ..., S$, at time step t we can estimate the probability distribution of the hidden target state $\mathbf{x}_t$ of the tracked object by $\mathbf{x}_t = [x, y, v_x, v_y]'$, where (x, y) are the center coordinates of the particles rectangle window and (v_x, v_y) are the velocities. The velocities are described by a Gaussian distribution with zero mean and motion dependent standard deviation. Each particle x^i_t simulates the real hidden state of the object. Using the dynamic model $p(x^i_t | x^i_{t-1})$ and the observation likelihood $p(z^i_t | x^i_t)$, the posterior distribution $p(\mathbf{x}_t | \mathbf{z}_t)$ is approximated by the finite set of particles $p(\mathbf{x}_t | \mathbf{z}_{1:t}) \approx \sum_{i=1}^{S} w^i_t x^i_t$.

The weights are updated according to

$$w^i_t \propto w^i_{t-1} \frac{p(z^i_t \mid x^i_t) p(x^i_t \mid x^i_{t-1})}{q(x^i_t \mid x^i_{t-1}, z^i_t)} \ , \tag{4.9}$$

where $\sum_{i=1}^{N_p} w^i_t = 1$ and $q(x^i_t \mid x^i_{t-1}, z^i_t)$ is the proposal distribution to draw particles from.

Using an auto-regression model, the transition probability $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ is represented by $\mathbf{x}_{t+1} = A\mathbf{x}_t + \mathbf{v}_t$. Applying the state transition model $p(x_t^i|x_{t-1}^i)$ as proposal distribution leads to the bootstrap filter, where the weights are directly proportional to the observation model $p(z_t^i|x_t^i)$, which can be calculated from the voting map V within a local neighborhood x_t^i.

Although the voting map cannot be seen as a probabilistic map, the particle filter is still working by seeking for the strongest local mode. Finally, the posterior density $p(\mathbf{x}_t|\mathbf{z}_{1:t})$ is approximated by the weighted mean over the particle distribution, as given in Equation (4.9). To avoid the degeneracy of the particle set, the re-sampling of the weights is performed after each frame. For more details on particle filtering we refer to [9].

The foot-point voting on the centralized top view map enables us to exploit the knowledge that objects cannot overlap each other on the top view (see Figure 4.2(b)) and to incorporate this to the particle filter framework. So far each object is tracked by an individual particle filter, without any knowledge about surrounding objects. Each object o has its own particles $\mathbf{x}_t^{i,o}$, with $i = 1, ...S_o$, which are re-weighted. In general, this leads to hijacked particles, where several trackers are following the same voting maximum, which is often called the "error merge" problem.

After re-weighting the particles for each individual object o, we perform a joint re-weighting, where particles of different objects $\mathbf{x}_t^{i,o1}$ and $\mathbf{x}_t^{j,o2}$ are penalized if they are overlapping [97]. Penalizing such overlapping particles avoids that particles belonging to different objects merge to one maximum within the common vote map. This can be seen related to the magnetic-inertia potential model [111], which proposes to model a gravitation and magnetic repulsion scheme. But, in contrast to [111] the non-overlapping assumption is directly assured in our concept as a result of the ground plane projections described in Section 4.3.2, since one position on the top view map can only be occupied by one single object based on the closed-world assumption.

4.3.4 View-specific Hough Voting

Random forests (and therefore also Hough forests) are a perfect choice for learning generic classifiers, since they allow for training from huge data sets and can cope with multi-modal data. However, for specific camera views not all information is needed and the large variability in the data would cause some noise in the Hough votes increasing the uncertainty of the world points Σ_X. One way to overcome these problems would be to train a separate Hough forest for each camera view. However, training a separate classifier for every scene and every viewpoint requires a massive labeling effort, which should be avoided.

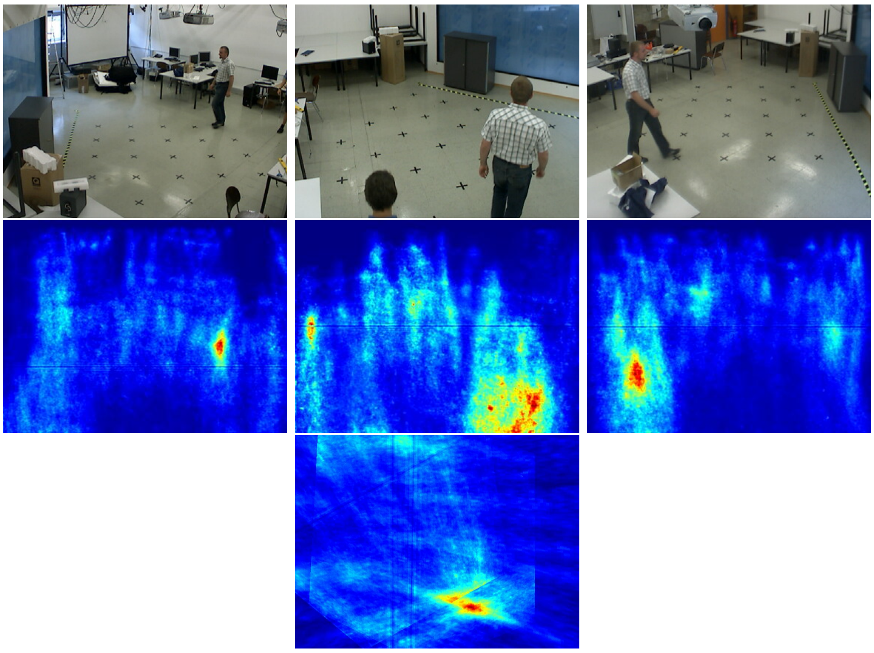
In contrast, we exploit the geometric constraints by using the back-projection of the tracking results on the top view map to each of the camera views to perform camera specific updates. In this way, we can adapt a general off-line trained classifier to each individual camera in order to reduce the amount of noise. We introduce an additional view specific term $p(P_v|c = 1, L(\mathbf{y}))$ for each vote vector $\mathbf{d}_i$ in the leaf node L, where P_v considers only object patches that vote correctly within this specific view. To reduce the importance of a vote vector $\mathbf{d}_i$ which is not suitable for a specific view v and hence introduces noise, we are now interested in $p(E_v(\mathbf{x})|L(\mathbf{y}))$, where $E_v(\mathbf{x})$ is the evidence of an object at location $\mathbf{x}$ in camera view v. By approximating $p(\mathbf{d}(\mathbf{y}) = \mathbf{y} - \mathbf{x}|c(\mathbf{y}) = 1, L(\mathbf{y}))$ by a sum of Dirac measures $\delta_{\mathbf{d}_i}(\mathbf{y} - \mathbf{x})$, as shown in [57], the view specific probability can now be calculated as

$$p(E_V(\mathbf{x})|L(y)) = \frac{1}{|\mathcal{P}_{L(y)}|} \cdot \sum_{P_v \in \mathcal{P}_{L(y)}} p(P_v|c = 1, L(\mathbf{y})) \cdot p(c = 1|L(\mathbf{y})) \cdot \delta_{\mathbf{d}_i}(\mathbf{y} - \mathbf{x}). \tag{4.10}$$

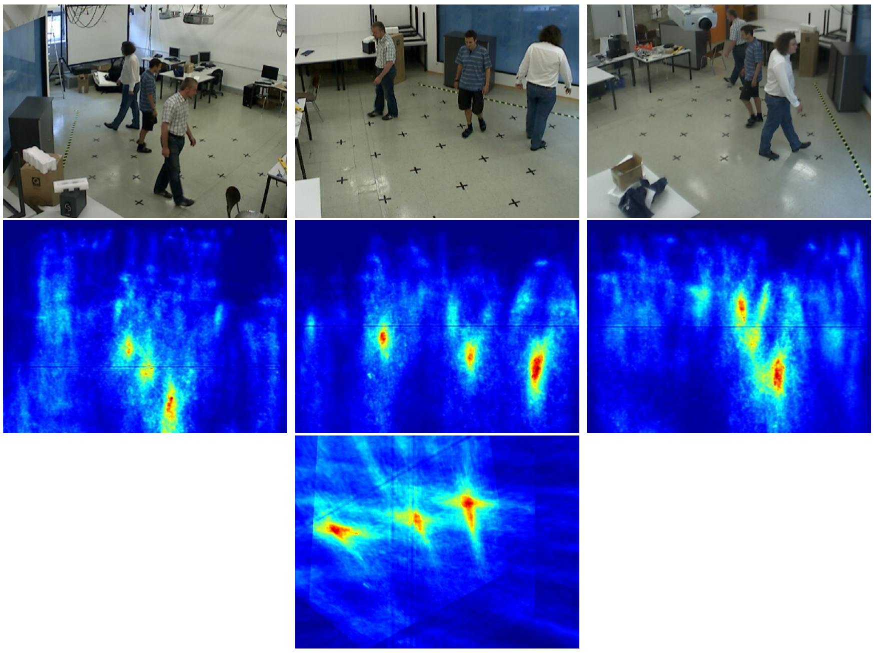
The view specific term $p(P_v|c = 1, L(\mathbf{y}))$ is updated over time by using the back-projection of the tracking results on the common ground plane. Therefore, we count how often a specific vote $\mathbf{d}_i$ votes into a correct position (given by the back-projection of the tracking results coming from the particle filter) and denote this number by $n^+_{\mathbf{d}_i}$. In addition, we count how often this vote casts to a wrong location, i.e., where no object-of-interest is present: $n^-_{\mathbf{d}_i}$. The view specific term can now be calculated by

$$p(P_i \in V|c = 1, L(\mathbf{y})) = \begin{cases} 0.5 & \text{if } sum_n = 0 \\ \dfrac{n^+_{\mathbf{d}_i}}{n^+_{\mathbf{d}_i} + n^-_{\mathbf{d}_i}} & \text{otherwise ,} \end{cases} \tag{4.11}$$

where $sum_n = n^+_{\mathbf{d}_i} + n^-_{\mathbf{d}_i}$. The benefits of the view-specific updates are illustrated in Figure 4.5, where we show the evolution of the vote maps for each camera over time. Figure 4.5(a) shows the vote maps for each of the three camera views for frame 160, where it can be seen that a lot of noisy votes are reported for the background. In contrast, for frame 2300 shown in Figure 4.5(b), the level of noise is decreased in the view specific vote maps as well as in the combined top view vote map, which demonstrates the effect of view-specific updates.



(a) Frame 160: First row shows the input images, second row shows the corresponding Hough maps of each camera view and the third row shows the projected common Hough map of the top view.



(b) Frame 2300: First row shows the input images, second row shows the corresponding Hough maps of each camera view and the third row shows the projected common Hough map of the top view.

Figure 4.5: View-specific Hough votes: Input images, Hough maps of individual views and common top view Hough map. The evolution of Hough maps over time – showing the maps for frames 160 and frames 2300 demonstrates the effect of the updates. The single votes contain less noise resulting in much better combined top-view maps.

4.4 Experimental evaluation

In the following, we demonstrate our approach on different publicly available datasets for multi-camera object tracking. We first describe our experimental setup and evaluation methods used and then evaluate our approach on two different datasets.

4.4.1 Experimental Setup

For the experiments, we trained a Hough forest [56] [*] voting to the foot-point on the VIPeR pedestrian data set [70]. To reduce the run-time complexity, we limited the number of trees within the forest to three. Each of the trees has a maximum depth of 15. We compare our approach to three different baselines. First, to a background subtraction (BGS) based approach, where similar to the foot-point voting we use the scale information available for each foot-point location and calculate the ratio between the foreground and background pixels within a bounding box. This ratio is then projected onto the common ground plane to obtain a summed common foreground pixel probability top view map. Second, we compare to a single camera approach, which builds on the same Hough maps as the proposed method (HV Cam1, HV Cam2 and HV Cam3). Third, we compare to the approach of Berclaz et al. [18] [†] (POM + KSP), where K-Shortest Path (KSP) is used to link the detections coming from the probabilistic occupancy maps (POM). For all particle filter based approaches we used a set of 300 particles described in Section 4.3.3, where the tracking is initialized manually. For both datasets we have given the ground truth annotation for every tenth frame. In each camera view where the object of interest is visible it is annotated by a bounding box. To reduce the effect of slightly varying annotations, the foot-point of each camera is projected onto the top view. For a quantitative evaluation we calculate the pixel error on the ground plane as the minimum distance to the annotations of each single camera.

4.4.2 Medium Dataset

The first experiment we run on the *Set 1* sequence [‡] of the publicly available Medium Dataset [117]. The dataset, showing an indoor lab environment, captured three people walking around occluding each other from three views and contains about 2500 frames per view with a resolution of 384x288 pixels. The pixel error on the ground

[*] http://www.vision.ee.ethz.ch/~gallju/projects/houghforest/index.html

[†] http://cvlab.epfl.ch/research/body/surv/

[‡] http://lrs.icg.tugraz.at/download.php

plane is shown in Figure 4.6. It can be seen that at the beginning all multi-camera approaches yield a comparable performance, but after the persons move too close to each other (which arises around frame 600) the quality of the background subtraction based approach is decreasing. The same occurs for the POM+KSP approach, where around frame 1000 an identity switch occurs. The same can also be recognized for the single view trackers, however, here the tracking accuracy is degraded much earlier. In particular, these methods are suffering from the "error merge" problem as well as the "labeling" problem. The first one, which especially applies for the single view trackers, describes the problem that the tracker loses its specific instance and falsely coalesces with others. The second one means that identities of the objects are mixed up by the trackers. However, it could be seen that the proposed methods avoids both problems and thus yield much more stable tracking results. Additionally, we give the averaged pixel error for all approaches in Table 4.1 and show illustrative results in Figure 4.7.

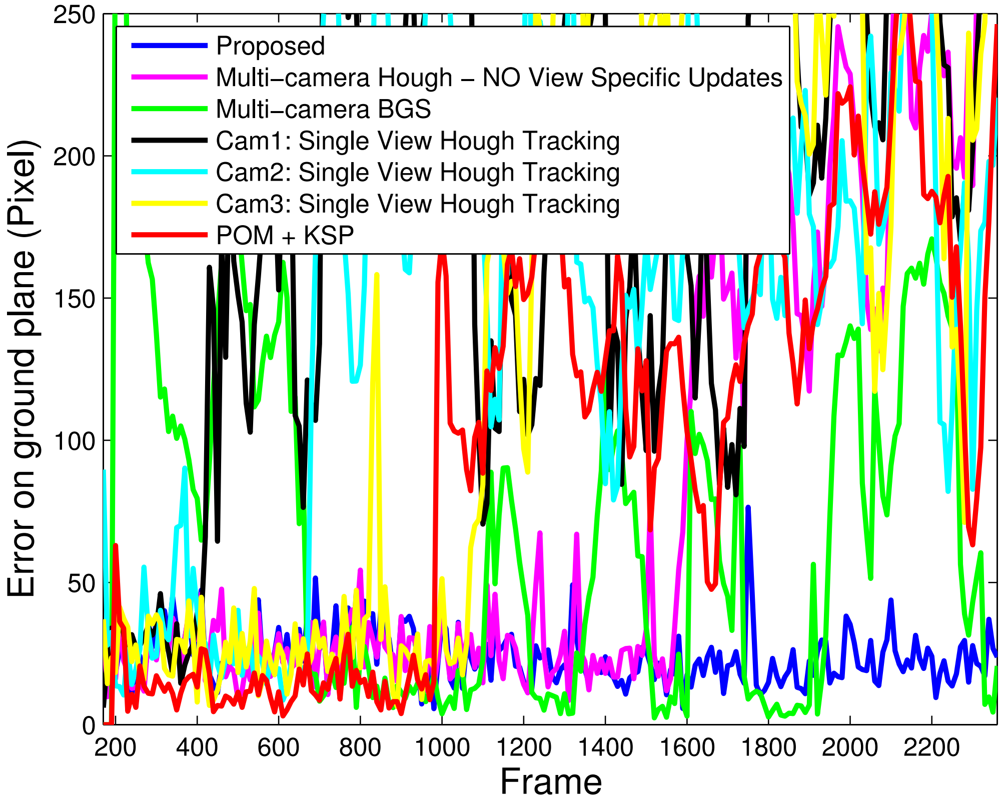


Figure 4.6: Error in pixel on the ground plane stays the same over time for the proposed approach, but increases for all other approaches caused by the labeling and/or the error merge problem.

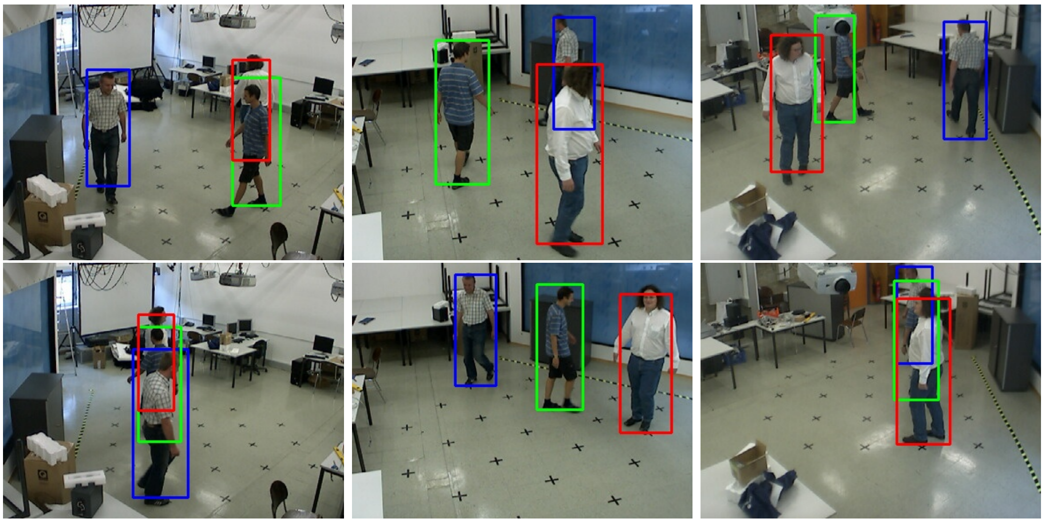


Figure 4.7: Illustrative results on the *Set 1* sequence.

4.4.3 Campus Sequence 2

Additionally, we evaluate on the publicly available *Campus Sequence 2* [§] [19] consisting of 5884 images from three cameras with a resolution of 360x288 pixels showing an outdoor sequence with three moving persons. The obtained error rates averaged over the whole sequence are listed in Table 4.1. In general, considering the error rates it is revealed that this scenario is a little bit simpler than the other one – a larger area is observed and the number of occlusions is smaller. This also explains the rather good results for the simple background subtraction based approach. However, as for the previous setup, it can be seen that using the combined multi-camera approach the tracking results can significantly be improved. Even though we do not use an instance specific tracking approach the view-specific updates, which reduce the noise within the common vote map, in combination with the non-overlapping constraint of the particle filter enables our approach to track both sequence without any error merge or labeling problems. Finally, illustrative results for this data set are shown in Figure 4.8.

4.5 Summary

Most multiple camera approaches for multiple object tracking rely on background subtraction and project all foreground pixels onto a common ground plane. Hence, hurting the geometric constraints large projection errors are introduced resulting in ghost de-

[§] http://cvlab.epfl.ch/data/pom/

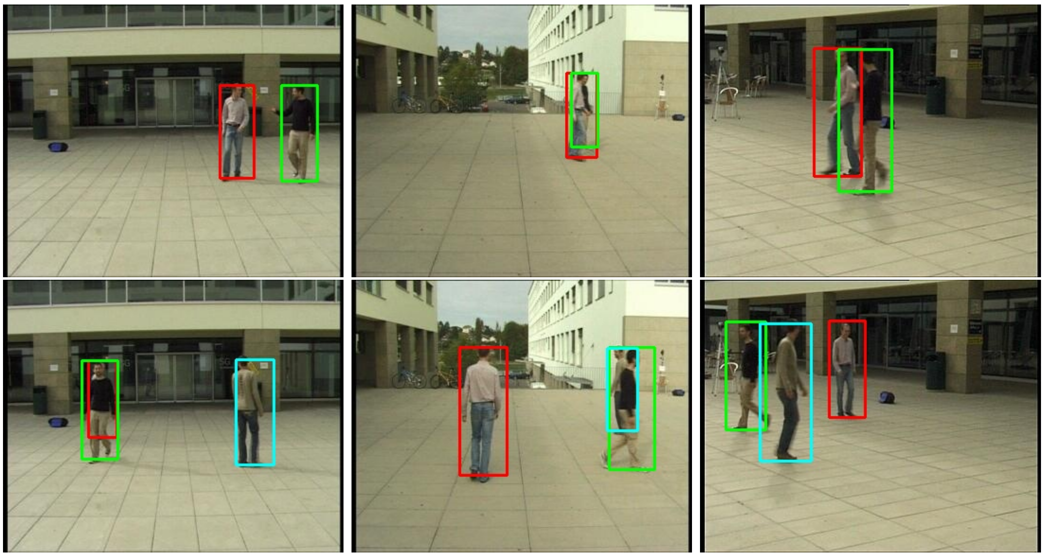


Figure 4.8: Illustrative results on the *Campus Sequence 2*.

Approach	Set 1	Campus 2
Proposed	**23.9**	**8.0**
BGS Based	75.7	27.5
POM+KSP	106.3	73.52
HV Cam1	186.8	79.8
HV Cam2	153.8	78.7
HV Cam3	152.6	137.3

Table 4.1: Comparison of mean error in pixels on the top view map for *Set 1* and *Campus 2* sequences.

tections. To overcome these limitations, we proposed a novel multi-camera tracking approach. We introduce a multi-camera Hough voting scheme, where the key idea is to direct the votes to the foot-points instead of to the centroid. In this way, we can exploit geometric constraints and map the single camera votes onto a common ground plane. Thereby we can implicitly handle geometric uncertainties. By facilitating an extended more robust particle filtering approach we can exploit the reliable tracking results and back-project the tracking results onto each individual view to identify instable votes. This further allows us to incorporate scene specific information and to perform view specific updates, reducing the noise within each individual Hough map. Overall, we get a robust multiple object tracking approach, which avoids the error merge and labeling problem, even though no instance specific information is used.

Science never solves a problem without creating ten more.

George Bernard Shaw

5
Summary and Conclusion

In this thesis we introduced different methods for object detection and object tracking from single and multiple stationary cameras. Both, object detection and object tracking are important steps toward a fully automated visual surveillance. By analyzing the trajectories of individuals it is for example possible to identify suspicious persons. This can help to prevent car crime in public car parking environments. Analyzing the trajectories can also help to identify wrong way drivers on highways. Besides visual surveillance, object detection and object tracking are an important cue for various applications, like industrial applications, sports analysis or assisted living.

However, to be able to use object detection and object tracking algorithms for various different applications it is not enough to demonstrate good performance under lab conditions or for one particular scenario. They have also to be applicable to different kinds of real-world scenarios without the need for manually training separate detectors or trackers for every single scenario. In real-world scenarios one has to deal with challenging situations, like changing lighting conditions, changing backgrounds, variations in the appearance of the objects, etc., but also with various different types of scenes, such as indoor scenes under controlled environments often available in sports analysis or different types of outdoor scenes that could, in addition to illumination changes, be affected by adverse weather conditions. Handling all these changing situations within one classifier requires a tremendous amount of training data containing all possible variations. It would further result in a very complex generic classifier delivering false positive as well as false negative predictions. To avoid such complex classifiers one can draw upon adaptive classifiers which are able to incorporate scene specific information in an on-line fashion. To avoid human interaction, these adaptive approaches have to deal with unlabeled information in an unsupervised manner.

The main challenge in dealing with unlabeled information is to preserve robustness. One possible way to include unlabeled information is using semi-supervised learning approaches, like self-training. However, self-training is prone to drifting, since wrong predictions have a direct influence on the classifier and reinforces wrong classifications. Therefore, such approaches are not applicable for fully unsupervised updates caused by the lacking robustness. In contrast, in this thesis we presented different approaches that allow for robustly incorporating unlabeled information for object detection from stationary cameras. This requires incorporating scene specific object information as well as scene specific background information. We developed different approaches to incorporate this information based on the idea of classifier grids. Classifier grids divide the input image into highly overlapping grid elements, where each grid element contains its own classifier. Based on the idea of classifier grids fixed update strategies (as described in Section 3.3.2) or different learning strategies, like an inverse multiple instance learning strategy (as described in Section 3.5.2) can be used to robustly incorporate scene specific background information. To allow for incorporating both, scene specific object information as well as scene specific background information we propose a co-training related approach for the classifier grids, i.e., classifier co-grid, (described in Section 3.6.2). In combination with our robust learning algorithm (TransientBoost) this allows for incorporating unlabeled information from the scene but still preserving the reliable labeled information. All these approaches aim to preserve long-term stability, which is given by either using specific update strategies or by using our robust learning algorithm (TransientBoost). We demonstrate the long-term stability of the proposed approaches empirically by evaluating them on a real-world surveillance scenario, where a corridor in a public building is monitored over one week and object detection is performed. Even though the whole approach is updated without supervision, the robustness is preserved.

TransientBoost allows for combining reliable and unreliable information within one classifier. The binary classification problem is adapted in a way that each class consists of two models. One model contains reliable information based on labeled data whereas the other one describes the unlabeled information. The reliable information keeps the classifier from drifting if wrong information is incorporated. Additionally, the unreliable information is transient, i.e., it fades out over time. Hence, wrong updates do not harm the classifier over time. TransientBoost cannot only be applied in the context of classifier grids; it can also be used for object tracking, where the reliable information keeps the classifier from drifting, as shown in [61], where we demonstrated that this approach is also suited for long-term tracking.

Furthermore, we presented an approach to incorporate prior knowledge to on-line boosting for feature selection. By developing a modified version of off-line boosting for feature selection the originally randomly initialized features of the on-line boosting for feature selection algorithm can be initialized by features appropriate for the particular problem, e.g., pedestrian detection. The off-line on-line linkage is not restricted to the application within classifier grids as presented here. It can also be used to improve the performance of other problems in computer vision, such as object tracking, if the object of interest is known in advance.

Motivated by the promising results of incorporating scene specific information for object detection from single stationary cameras, we presented a novel approach for object tracking from multiple stationary cameras, suitable for incorporating scene specific information for every single camera. This approach is based on a Hough voting scheme, where geometric uncertainties can be handled implicitly. To avoid projection errors we modify the Hough voting scheme to vote to the foot point location and use a image-to-ground plane homography to project the votes of each individual view and combine them on a common top view. A particle filter exploiting the physical constraints that one ground plane position cannot be occupied by more than one object is used to perform tracking. The reliable tracking results in combination with the geometric information can then be used to adapt the Hough votes of a single camera to the particular scene.

To further improve object detection, in particular in the presence of multiple occluding objects, it would be interesting to investigate in non-maximum suppression. Window based approaches, such as the classifier grid approach, evaluate the input image at highly overlapping subwindows, resulting in a number of detections (maxima) around the object of interest. Identifying the maxima within these detections is not always a trivial task, especially if the objects are (partially) occluding each other. Even though this problem was often neglected in the scientific community, today there are more and more people focusing on developing approaches that are able to implicitly tackle the problem of non-maxima suppression [15, 34, 113, 130].

Another interesting direction for future work is to expand this idea to single images and to perform an image specific domain adaption. Various machine learning problems tackle the problem of domain adaption, where the source domain used to train a classifier significantly differs from the target domain on which the classifier is evaluated. Large variations between both, the object class as well as the background class, entail this problem in almost all problems in computer vision. One possible way to overcome this problem is to train a separate detector for every single input image. However, this requires training data for every single input image, which is intractable. Instead of training a separate detector for every input image the detector can be adapted to every single input image, as shown by Jain and Learned-Miller [79] for the problem of face detection.

If you go as far as you can see,
you will then see enough to go even farther.

John Wooden

6
Publications

In the following, publications written during my research work are listed.

- Peter M. Roth, Sabine Sternig, Helmut Grabner, and Horst Bischof (2009). Classifier grids for robust adaptive object detection. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Miami, USA.

- Hayko Riemenschneider, Sabine Sternig, Michael Donoser, Peter M. Roth, and Horst Bischof (2012). Hough regions for joining instance localization and segmentation. In *Proc. European Conf. on Computer Vision*, Florence, Italy.

- Sabine Sternig, Peter M. Roth, and Horst Bischof (2011). On-line inverse multiple instance boosting for classifier grids. *Pattern Recognition Letters*

- Sabine Sternig, Peter M. Roth, and Horst Bischof (2010). Inverse multiple instance learning for classifier grids. In *Proc. IEEE Int'l Conf. on Pattern Recognition*, Istanbul, Turkey (*Best Scientific Paper award*).

- Sabine Sternig, Thomas Mauthner, Arnold Irschara, Peter M. Roth, and Horst Bischof (2011). Multi-camera multi-object tracking by robust hough-based homography projections. In *Proc. 11th IEEE Workshop on Visual Surveillance (ICCV)*, Barcelona, Spain.

- Sabine Sternig, Martin Godec, Peter M. Roth, and Horst Bischof (2010). Transientboost: On-line boosting with transient data. In *Proc. IEEE Online Learning for Computer Vision Workshop (CVPR)*, San Francisco, USA.

- Martin Godec, Sabine Sternig, Peter M. Roth, and Horst Bischof (2010). Context-driven clustering by multi-class classification in an active learning framework. In *Proc. Workshop on Use of Context in Video Processing (CVPR)*, San Francisco, USA.

- Sabine Sternig, Peter M. Roth, and Horst Bischof. Learning of scene-specific object detectors by classifier co-grids (2010). In *Proc. IEEE Int'l Conf. on Advanced Video and Signal-Based Surveillance*, Boston, USA.

- Sabine Sternig, Hayko Riemenschneider, Peter M. Roth, Michael Donoser, and Horst Bischof (2010). Robust object detection by classifier cubes and local verification. In *Proc. Workshop of the Austrian Association for Pattern Recognition*, Zwettl, Austria.

- Sabine Sternig, Peter M. Roth, Helmut Grabner, and Horst Bischof (2009). Robust adaptive classifier grids for object detection from static cameras. In *Proc. Computer Vision Winter Workshop*, Eibiswald, Austria.

- Horst Possegger, Matthias Rüther, Sabine Sternig, Thomas Mauthner, Manfred Klopschitz, Peter M. Roth, and Horst Bischof (2012). Unsupervised calibration of camera networks and virtual PTZ cameras. In *Proc. Computer Vision Winter Workshop*, Mala Nedelja, Slovenia.

- Andreas Wendel, Sabine Sternig, and Martin Godec (2011), editors. *Proc. 16th Computer Vision Winter Workshop*. Verlag der Technische Universitaet Graz.

Bibliography

[1] Steven Abney. Bootstrapping. In *Proc. Annual Meeting of the Association for Computational Linguistics*, pages 360–367, 2002.

[2] Yaser S. Abu-Mostafa. Machines that learn from hints. *Scientific American*, 272: 64–69, 1995.

[3] Shivani Agarwal, Aatif Awan, and Dan Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(11):1475–1490, 2004.

[4] Afshin Dehghan Amir Roshan Zamir and Mubarak Shah. Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. In *Proc. European Conf. on Computer Vision*, 2012.

[5] Yali Amit. *2D Object Detection and Recognition: Models, Algorithms, and Networks*. The MIT Press, 2002.

[6] Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann. Support vector machines for multiple-instance learning. In *Advances in Neural Information Processing Systems*, pages 561–568, 2003.

[7] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2009.

[8] Jamie Shotton Antonio Criminisi and Ender Konukoglu. Decision forests for classification, regression, density estimation, manifold learning and semi-supervised learning. Technical Report TR-2011-114, Microsoft Research Cambridge, 2011.

[9] Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174 –188, 2002.

[10] Shai Avidan. Ensemble tracking. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume II, pages 494–501, 2005.

[11] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie. Visual tracking with online mulitple instance learning. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2009.

[12] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie. Robust object tracking with online multiple instance learning. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 33(8):1619–1632, 2011.

[13] Maria-Florina Balcan, Avrim Blum, and Ke Yang. Co-training and expansion: Towards bridging theory and practice. In *Advances in Neural Information Processing Systems*, pages 89–96, 2004.

[14] Dana H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.

[15] Olga Barinova, Victor Lempitsky, and Pushmeet Kohli. On detection of multiple object instances using hough transform. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2010.

[16] Olga Barinova, Victor S. Lempitsky, and Pushmeet Kohli. On detection of multiple object instances using hough transforms. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 34(9):1773–1784, 2012.

[17] Eric Bauer and Ron Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, 36(1-2):105–139, July 1999. ISSN 0885-6125.

[18] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 33(9):1806–1819, 2011.

[19] Jérôme Berclaz, François Fleuret, and Pascal Fua. Principled detection-by-classification from multiple views. In *Proc. Int'l Conf. on Computer Vision Theory and Applications*, pages 375–382, 2008.

[20] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proc. Conf. on Computational Learning Theory*, pages 92–100, 1998.

[21] Leo Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996.

[22] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[23] Michael D. Breitenstein, Eric Sommerlade, Bastian Leibe, Luc van Gool, and Ian Reid. Probabilistic parameter selection for learning scene structure from video. In *Proc. British Machine Vision Conf.*, 2008.

[24] Michael D. Breitenstein, Fabian Reichlin, Bastian Leibe, Esther Koller-Meier, and Luc van Gool. Online multi-person tracking-by-detection from a single, uncalibrated camera. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 33(9):1820–1833, 2010.

[25] William Brendel, Mohamed Amer, and Sinisa Todorovic. Multiobject tracking as maximum weight independent set. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2011.

[26] Michael C. Burl, Thomas K. Leung, and Pietro Perona. Face localization via shape statistics. In *Int'l. Workshop Face and Gesture Recognition*, 1995.

[27] Yizheng Cai, Nando de Freitas, and James J. Little. Robust visual tracking for multiple targets. In *Proc. European Conf. on Computer Vision*, 2006.

[28] Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, editors. *Semi-Supervised Learning (Adaptive Computation and Machine Learning series)*. The MIT Press, 2006.

[29] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[30] Antonio Criminisi. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends® in Computer Graphics and Vision*, 7(2-3):81–227, 2011.

[31] Antonio Criminisi, Ian Reid, and Andrew Zisserman. A plane measuring device. In *Proc. British Machine Vision Conf.*, 1997.

[32] Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in Computer Vision, ECCV*, 2004.

[33] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume I, pages 886–893, 2005.

[34] Chaitanya Desai, Deva Ramanan, and Charless Fowlkes. Discriminative models for multi-class object layout. In *Proc. IEEE Int'l Conf. on Computer Vision*, 2009.

[35] Chaitanya Desai, Deva Ramanan, and Charless Fowlkes. Discriminative models for multi-class object layout. *Int'l Journal of Computer Vision*, 95(1):1–12, 2011.

[36] Thomas G. Dietterich. Ensemble methods in machine learning. In *Proc. Int. Workshop on Multiple Classifier Systems*, 2000.

[37] Thomas G. Dietterich, Richard H. Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89 (1–2):31–71, 1997.

[38] Piotr Dollár, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 34(4):743–761, 2012.

[39] Marcin Eichner and Vittorio Ferrari. We are family: Joint pose estimation of multiple persons. In *Proc. European Conf. on Computer Vision*, 2010.

[40] Markus Enzweiler and Dariu M. Gavrila. Monocular pedestrian detection: Survey and experiments. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31(12): 2179–2195, 2009.

[41] Ran Eshel and Yael Moses. Homography based multiple camera detection and tracking of people in a dense crowd. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[42] Ran Eshel and Yael Moses. Tracking in a dense crowd using multiple cameras. *Int'l Journal of Computer Vision*, 2010. (online first).

[43] Andreas Ess, Bastian Leibe, Konrad Schindler, and Luc van Gool. On-line adaption of class-specific codebooks for instance tracking. In *Proc. British Machine Vision Conf.*, 2010.

[44] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *Int'l Journal of Computer Vision*, 88(2):303–338, 2010.

[45] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2008.

[46] Pedro Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.

[47] Pedro F. Felzenszwalb, Ross B. Girshick, and David A. McAllester. Cascade object detection with deformable part models. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2010.

[48] Martin A. Fischler and Robert A. Elschlager. The representation and matching of pictorial structures. *IEEE Transaction on Computers*, 100(1):67–92, 1973.

[49] Sven Fleck and Wolfgang Strasser. Smart camera based monitoring system and its application to assisted living. *Proceedings of the IEEE*, 96(10):1698 –1714, oct. 2008.

[50] François Fleuret, Jérôme Berclaz, Richard Lengagne, and Pascal Fua. Multi-camera people tracking with a probabilistic occupancy map. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(2):267–282, 2008.

[51] Katerina Fragkiadaki, Weiyu Zhang, Geng Zhang, and Jianbo Shi. Two-granularity tracking: Mediating trajectory and detection graphs for tracking under occlusions. In *Proc. European Conf. on Computer Vision*, 2012.

[52] Yoav Freund. Boosting a weak learning algorithm by majority. In *Computational Learning Theory*, 1995.

[53] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of online learning and an application to boosting. In *Proc. European Conf. on Computational Learning Theory*, pages 23–37, 1995.

[54] Yoav Freund and Robert E. Shapire. Experiments with a new boosting algorithm. In *Proc. Int'l Conf. on Machine Learning*, pages 148–156, 1996.

[55] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 28(2):337–407, 2000.

[56] Juergen Gall and Victor Lempitsky. Class-specific hough forests for object detection. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2009.

[57] Juergen Gall, Nima Razavi, and Luc J. van Gool. On-line adaption of class-specific codebooks for instance tracking. In *Proc. British Machine Vision Conf.*, 2010.

[58] Juergen Gall, Angela Yao, Nima Razavi, Luc J. van Gool, and Victor Lempitsky. Hough forests for object detection, tracking, and action recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(11):2188–2202, 2011.

[59] Alexander Gammerman, Katy S. Azoury, and Vladimir Vapnik. Learning by transduction. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, University of Wisconsin Business School, Madison, Wisconsin, USA, July 24-26, 1998*, 1998.

[60] David Geróandnimo, Antonio M. Lóandpez, Angel D. Sappa, and Thorsten Graf. Survey of pedestrian detection for advanced driver assistance systems. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(7):1239–1258, 2010.

[61] Martin Godec, Sabine Sternig, Peter M. Roth, and Horst Bischof. Context-driven clustering by multi-class classification in an active learning framework. In *Proc. Workshop on Use of Context in Video Processing*, 2010. (in conj. CVPR).

[62] Martin Godec, Peter M. Roth, and Horst Bischof. Hough-based tracking of non-rigid objects. In *Proc. IEEE Int'l Conf. on Computer Vision*, 2011.

[63] Helmut Grabner. *On-line Boosting and Vision*. PhD thesis, Graz University of Technology, 2008.

[64] Helmut Grabner and Horst Bischof. On-line boosting and vision. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume I, pages 260–267, 2006.

[65] Helmut Grabner, Michael Grabner, and Horst Bischof. Real-time tracking via on-line boosting. In *Proc. British Machine Vision Conf.*, volume I, pages 47–56, 2006.

[66] Helmut Grabner, Peter M. Roth, Michael Grabner, and Horst Bischof. Autonomous learning of a robust background model for change detection. In *Proc. IEEE Workshop on Performance Evaluation of Tracking and Surveillance*, pages 39–46, 2006. (in conj. CVPR).

[67] Helmut Grabner, Peter M. Roth, and Horst Bischof. Is pedestrian detection really a hard task? In *Proc. IEEE Workshop on Performance Evaluation of Tracking and Surveillance*, 2007. (in conj. ICCV).

[68] Yves Grandvalet, Florence d'AlchÃ© Buc, and Christophe Ambroise. Boosting mixture models for semi-supervised learning. In *International Conference on Artificial Neural Networks (ICANN)*, 2001.

[69] Kristen Grauman and Bastian Leibe. *Visual Object Recognition*. Morgan & Claypool Publishers, 2011.

[70] Douglas Gray, S. Brennan, and H. Tao. Evaluating appearance models for recognition, reacquisition, and tracking. In *Proc. IEEE Workshop on Performance Evaluation of Tracking and Surveillance*, 2007.

[71] Mary W. Green. The appropriate and effective use of security technologies in u.s. schools. Technical report, Sandia National Laboratories, 1999.

[72] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2003.

[73] Tin Kam Ho. Random decision forests. *Document Analysis and Recognition, International Conference on*, 1:278, 1995.

[74] Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(8):832–844, August 1998.

[75] Weiming Hu, Xue Zhou, Min Hu, and Steve Maybank. Occlusion reasoning for tracking multiple people. *Circuits and Systems for Video Technology, IEEE Transactions on*, 19(1):114–121, 2009.

[76] Stephen S. Intille and Aaron F. Bobick. Visual tracking using closed-worlds. 1995.

[77] Michael Isard and Andrew Blake. Contour tracking by stochastic propagation of conditional density. In *Proc. European Conf. on Computer Vision*, 1996.

[78] Michael Isard and John MacCormick. Bramble: a bayesian multiple-blob tracker. In *Proc. IEEE Int'l Conf. on Computer Vision*, 2001.

[79] Vidit Jain and Erik G. Learned-Miller. Online domain adaptation of a pre-trained cascade of classifiers.

[80] Thorsten Joachims. Transductive inference for text classification using support vector machines. In *International Conference on Machine Learning*, 1999.

[81] Saad M. Khan and Mubarak Shah. A multiview approach to tracking people in crowded scenes using a planar homography constraint. In *Proc. European Conf. on Computer Vision*, volume IV, pages 133–146, 2006.

[82] Saad M. Khan and Mubarak Shah. Tracking multiple occluding people by localizing on multiple scene planes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31(3):505–519, 2009.

[83] Zia Khan, Tucker Balch, and Frank Dellaert. Mcmc-based particle filtering for tracking a variable number of interacting targets. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(11):1805–1819, 2005.

[84] Kyungnam Kim and Larry Davis. Multi-camera tracking and segmentation of occluded people on ground plane using search-guided particle filtering. In *Proc. European Conf. on Computer Vision*, volume III, pages 98–109, 2006.

[85] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer. Multi-camera multi-person tracking for easyliving. In *Proceedings of Third IEEE International Workshop on*, 2000.

[86] Cheng-Hao Kuo and R. Nevatia. How does person identity recognition help multi-person tracking? 2011.

[87] Cheng-Hao Kuo, Chang Huang, and R. Nevatia. Multi-target tracking by on-line learned discriminative appearance models. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2010.

[88] Bastian Leibe, Alesš Leonardis, and Bernt Schiele. Robust object detection with interleaved categorization and segmentation. *Int'l Journal of Computer Vision*, 77 (1–3):259–289, 2008.

[89] Christian Leistner. *Semi-Supervised Ensemble Methods for Computer Vision*. PhD thesis, Graz University of Technology, 2010.

[90] Christian Leistner, Amir R. Saffari A. A., Peter M. Roth, and Horst Bischof. On robustness of on-line boosting - a competitive study. In *Proc. IEEE On-line Learning for Computer Vision Workshop*, 2009.

[91] Anat Levin, Paul Viola, and Yoav Freund. Unsupervised improvement of visual detectors using co-training. In *Proc. IEEE Int'l Conf. on Computer Vision*, volume I, pages 626–633, 2003.

[92] Li-Jia Li, Gang Wang, and Li Fei-Fei. Optimol: automatic online picture collection via incremental model learning. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[93] Yuan Li, Bo Wu, and Ram Nevatia. Human detection by searching in 3d space using camera and scene knowledge. In *Proc. Int'l Conf. on Pattern Recognition*, 2008.

[94] Rong Liu, Jian Cheng, and Hanqing Lu. A robust boosting tracker with minimum error bound in a co-training framework. In *Proc. IEEE Int'l Conf. on Computer Vision*, 2009.

[95] Wei-Lwun Lu, Jo-Anne Ting, Kevin P. Murphy, and James J. Little. Identifying players in broadcast sports videos using conditional random fields. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2011.

[96] Pavan Kumar Mallapragada, Rong Jin, Anil K. Jain, and Yi Liu. Semiboost: Boosting for semi-supervised learning. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31(11):2000–2014, 2009. ISSN 0162-8828.

[97] Thomas Mauthner and Horst Bischof. A robust multiple object tracking for sport applications. In *Proc. Workshop of the Austrian Association for Pattern Recognition*, 2007.

[98] Nigel J. B. McFarlane and C. Paddy Schofield. Segmentation and tracking of piglets. *Machine Vision and Applications*, 8(3):187–193, 1995.

[99] Ron Meir and Gunnar Rätsch. An introduction to boosting and leveraging. pages 118–183, 2003.

[100] Anurag Mittal and Larry S. Davis. M2tracker: A multi-view approach to segmenting and tracking people in a cluttered scene. *Int'l Journal of Computer Vision*, 51(3): 189–203, 2003.

[101] Dennis Mitzel and Bastian Leibe. Taking mobile multi-object tracking to the next level: People, unknown objects, and carried items. In *Proc. European Conf. on Computer Vision*, 2012.

[102] Kamal Nigam, Andrew McCallum, Sebastian Thrun, and Tom Mitchell. Learning to classify text from labeled and unlabeled documents. In *Artificial intelligence/Innovative applications of artificial intelligence*, 1998.

[103] Clive Norris and Michael McCahill. Cctv: Beyond penal modernism? *British Journal of Criminology*, 46(1), January 2006.

[104] Ryuzo Okada. Discriminative generalized hough transform for object dectection. In *Proc. IEEE Int'l Conf. on Computer Vision*, 2009.

[105] Björn Ommer and Jitendra Malik. Multi-scale object detection by clustering lines. In *Proc. IEEE Int'l Conf. on Computer Vision*, 2009.

[106] Andreas Opelt and Andrew Zisserman Axel Pinz. A boundary-fragment-model for object detection. In *Proc. European Conf. on Computer Vision*, volume II, pages 575–588, 2006.

[107] Nikunj C. Oza. *Online Ensemble Learning*. PhD thesis, University of California, Berkeley, 2001.

[108] Junbiao Pang, Qingming Huang, Shuicheng Yan, Shuqiang Jiang, and Lei Qin. Transferring boosted detectors towards viewpoint and scene adaptiveness. *IEEE Transaction on Image Processing*, 20(5):1388–1400, 2011.

[109] R. Pflugfelder and H. Bischof. Online auto-calibration in man-made worlds. In *Digital Image Computing: Technqiues and Applications*, 2005.

[110] Leonid Pishchulin, Arjun Jain, Christian Wojek, Mykhaylo Andriluka, Thorsten Thormaehlen, and Bernt Schiele. Learning people detection models from few training samples. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2011.

[111] Wei Qu, Dan Schonfeld, and Magdi Mohamed. Real-time interactively distributed multi-object tracking using a magnetic-inertia potential model. In *Proc. IEEE Int'l Conf. on Computer Vision*, 2005.

[112] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[113] Hayko Riemenschneider, Sabine Sternig, Michael Donoser, Peter M. Roth, and Horst Bischof. Hough regions for joining instance localization and segmentation. In *Proc. European Conf. on Computer Vision*, 2012.

[114] Lior Rokach. Ensemble-based classifiers. *Artificial Intelligence Rev.*, 33(1-2):1–39, 2010.

[115] Charles Rosenberg, Martial Hebert, and Henry Schneiderman. Semi-supervised self-training of object detection models. In *IEEE Workshop on Applications of Computer Vision*, pages 29–36, 2005.

[116] Peter M. Roth, Sabine Sternig, Helmut Grabner, and Horst Bischof. Classifier grids for robust adaptive object detection. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2009.

[117] Peter M. Roth, Christian Leistner, Armin Berger, and Horst Bischof. Multiple instance learning from multiple cameras. In *Proc. IEEE Workshop on Camera Networks*, 2010. (in conj. CVPR).

[118] Amir Saffari, Martin Godec, Thomas Pock, Christian Leistner, and Horst Bischof. Online Multi-Class LPBoost. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2010.

[119] Robert E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2): 197–227, 1990.

[120] Pramod Sharma, Chang Huang, and Ram Nevatia. Unsupervised incremental learning for improved object detection in a video. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2012.

[121] H. Ben Shitrit, J. Berclaz, F. Fleuret, , and P. Fua. Tracking multiple people under global appearance constraints. *Proc. IEEE Int'l Conf. on Computer Vision*, 2011.

[122] Herbert A. Simon. Why should machines learn? In *Machine Learning: An Artificial Intelligence Approach*. Springer, 1984.

[123] Severin Stalder, Helmut Grabner, and Luc van Gool. Exploring context to learn scene specific object detectors. In *Proc. IEEE Workshop on Performance Evaluation of Tracking and Surveillance*, 2009.

[124] Severin Stalder, Helmut Grabner, and Luc J. Van Gool. Cascaded confidence filtering for improved tracking-by-detection. In *Proc. European Conf. on Computer Vision*, 2010.

[125] Sabine Sternig, Martin Godec, Peter M. Roth, and Horst Bischof. Transientboost: On-line boosting with transient data. In *Proc. IEEE Online Learning for Computer Vision Workshop*, 2010. (in conj. CVPR).

[126] Sabine Sternig, Peter M. Roth, and Horst Bischof. Learning of scene-specific object detectors by classifier co-grids. In *Proc. IEEE Intern. Conf. on Advanced Video and Signal-Based Surveillance*, 2010.

[127] Sabine Sternig, Peter M. Roth, and Horst Bischof. Learning of scene-specific object detectors by classifier co-grids. In *IEEE International Conference on Advanced Video and Signal-Based Surveillance*, 2010.

[128] Sabine Sternig, Thomas Mauthner, Arnold Irschara, Peter M. Roth, and Horst Bischof. Multi-camera multi-object tracking by robust hough-based homography projections. In *Proc. 11th IEEE Workshop on Visual Surveillance*, 2011. (in conj. CVPR).

[129] Sabine Sternig, Peter M. Roth, and Horst Bischof. On-line inverse multiple instance boosting for classifier grids. *Pattern Recognition Letters*, 33(7):890 – 897, 2011.

[130] Siyu Tang, Mykhaylo Andriluka, and Bernt Schiele. Detection and tracking of occluded people. In *Proc. British Machine Vision Conf.*, 2012.

[131] Kinh Tieu and Paul Viola. Boosting image retrieval. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume I, pages 228–235, 2000.

[132] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[133] Paul Viola and Michael J. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume I, pages 511–518, 2001.

[134] Paul Viola and Michael J. Jones. Robust real-time face detection. *Int'l Journal of Computer Vision*, 57(2):137–154, 2004.

[135] Paul Viola, John C. Platt, and Cha Zhang. Multiple instance boosting for object detection. In *Advances in Neural Information Processing Systems*, pages 1417–1426, 2005.

[136] Paul A. Viola, John C. Platt, and Cha Zhang. Multiple instance boosting for object detection. In *Advances in Neural Information Processing Systems*, 2005.

[137] Meng Wang and Xiaogang Wang. Automatic adaptation of a generic pedestrian detector to a specific traffic scene. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2011.

[138] Meng Wang, Wei Li, and Xiaogang Wang. Transferring a generic pedestrian detector towards specific scenes. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2012.

[139] Wei Wang and Zhi-Hua Zhou. Analyzing co-training style algorithms. In *Proc. European Conf. on Machine Learning*, pages 454–465, 2007.

[140] Brandon C. Welsh and David P. Farrington. Public area cctv and crime prevention: An updated systematic review and meta–analysis. *Justice Quarterly*, 26(4), 2009.

[141] Ying Wu, Ting Yu, and Gang Hua. Tracking appearances with occlusions. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2003.

[142] Xu Yan, Xuqing Wu, Ioannis A. Kakadiaris, and Shishir K. Shah. To track or to detect? an ensemble framework for optimal selection. In *Proc. European Conf. on Computer Vision*, 2012.

[143] Bo Yang and Ram Nevatia. Multi-target tracking by online learning of non-linear motion patterns and robust appearance models. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2012.

[144] Bo Yang and Ram Nevatia. Online learned discriminative part-based appearance models for multi-human tracking. In *Proc. European Conf. on Computer Vision*, 2012.

[145] Tao Zhao and Ram Nevatia. Tracking multiple humans in crowded environment. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2004.

[146] Zhi-Hua Zhou and Jun-Ming Xu. On the relation between multi-instance learning and semi-supervised learning. In *Proceedings of the 24th international conference on Machine learning*, ICML '07, pages 1167–1174. ACM, 2007.

[147] Qiang Zhu, Shai Avidan, and Kwang-Ting Cheng. Learning a sparse, corner-based representation for background modelling. In *Proc. IEEE Int'l Conf. on Computer Vision*, volume I, pages 678–685, 2005.

[148] Xiaojin Zhu and Andrew B. Goldberg. Introduction to Semi-Supervised learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 3(1):1–130, 2009.

[149] Silvia Zuffi, Oren Freifeld, and Michael J. Black. From pictorial structures to deformable structures. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2012.