Computer Vision Winter Workshop 2009, Adrian Ion and Walter G. Kropatsch (eds.)
Eibiswald, Austria, February 4–6
Publisher: PRIP, Vienna University of Technology, Austria

Robust Adaptive Classifier Grids for Object Detection from Static Cameras

Sabine Sternig[1], Peter M. Roth[1], Helmut Grabner[2], and Horst Bischof[1]

[1] Graz University of Technology
Institute for Computer Graphics and Vision
{sternig, pmroth, bischof}@icg.tugraz.at

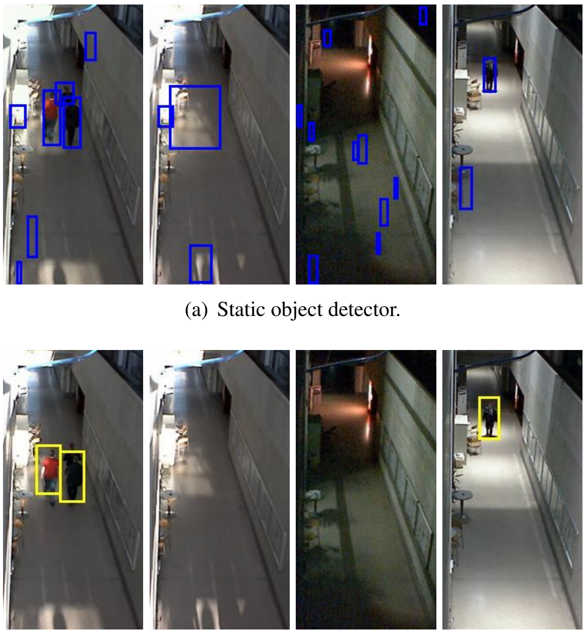
[2] ETH-Zürich
Computer Vision Lab
grabner@vision.ee.ethz.ch

Abstract *In this work we present a robust object detection system for static cameras, which is suitable for real-time applications. Thus, the system has to cope with changes of environmental conditions, which is realized by adaptive on-line learning a scene specific classifier. In particular, we apply the ideas of grid-based classification, where each image patch corresponds to one classifier. Thus, the complexity of the detection task is reduced and a more compact and thus more efficient representation can be applied. The main contribution of this paper is to introduce three learning strategies to improve the performance of grid-based detectors: (a) pre-selecting features to assure a more efficient representation, (b) pre-training the positive representation, and (c) combining off-line and on-line learning. The experimental results on person and car detection show that these strategies significantly improve the overall performance of the detection system. In addition, a long-term experiment demonstrates that the proposed system is stable over time and can thus be applied for real-world tasks.*

1 Introduction

Detecting and tracking of objects are important tasks in computer vision. Especially, for surveillance applications, where the behavior of persons or unusual events should be detected, object detection is the very first task in the processing queue. Typically for object detection a sliding window is applied. Each patch of an image is tested if it is consistent with a previously estimated model or not. Finally, all consistent patches are reported. These models are mostly based on local features in connection with a classifier, (e.g., AdaBoost, Winnow, neural network, support vector machine, or PCA), which is obtained by learning. Hence, when discussing the problem of object detection, implicitly, the problem of visual learning is addressed.

The goal of all such methods is to build a fixed generic object model that is applicable for all possible scenarios and tasks. But even if detectors are trained from a large number of samples they often fail in practice. This is illustrated in Figure 1, where we show the changing illumination conditions over 24 hours. The results shown in the first row are obtained by a fixed generic model. It can be seen that even for a rather simple scenario the precision is quite low (i.e., there are a lot of false alarms). This is the result of an insufficient not-representative training set (i.e., not all variability, especially in the background, can be captured). In contrast, the results in the second row are obtained by an adaptable object detection system. Assuming a static camera the system can adapt to the changing environmental conditions and those variabilities have not to be handled by the learned model. In addition, the complexity of the task is reduced.


(a) Static object detector.

(b) Adaptive object detector.

Figure 1: Changing environmental conditions (e.g., lighting changes or changes of objects in the background) arise the need of a system to be adaptive to changes: (a) detection result obtained by a fixed detector and (b) detection results obtained by an adaptive detector.

The main problem of adaptive systems is drifting. An object detection system starts drifting, if it is adapted in a wrong way, which leads the system to detect something different than the object of interest. Hence the main goal in this paper is to develop an adaptive but still robust object detection system that runs over a long period of time without drifting. We address this problem by using the ideas of grid-based object classification (e.g., [8, 7]). In contrast, to sliding window approaches the main idea is to apply a separate classifier on each image location (grid element). Thus, the complexity of the classification task that a single classifier has to handle can drastically be reduced. In addition, to avoid the drifting problem a fixed update strategy can be applied [7]. But such update schemes have two main shortcomings. First, they tend to be over-adaptive and second due to the simple positive update strategy either a huge amount of data has to be stored in memory or the variability for more complex objects can not be handled. In this paper, we propose two strategies that overcome these problems.

As a third contribution, we introduce a pre-training step that improves the quality of selected features during online learning. These benefits are demonstrated in the experiments, where we extensively compare the proposed approach to state-of-the art methods for person and car detection. In addition, in a long-term experiment we show that our system is stable over time, even if 150,000 updates were performed.

The paper is organized as follows. In Section 2 we review the main ideas, which build the basis of our work. Next, in Section 3 we introduce and discuss our new robust adaptive grid-based detector. The benefits of the proposed approach are demonstrated in Section 4. Finally, we summarize and conclude the paper in Section 5.

2 Preliminaries

2.1 Classifier Grid

The main idea of classifier grids (e.g., [8, 7]) is to sample an input image by using a fixed highly overlapping grid, where each grid element $i = 1, \dots, N$ corresponds to one classifier C_i. This is illustrated in Figure 2. Thus, the classification task that has to be handled by one classifier C_i is reduced to discriminate the background of the specific grid element from the object-of-interest. Due to this simplification less complex classifiers, which can be evaluated and updated more efficiently, can be applied.

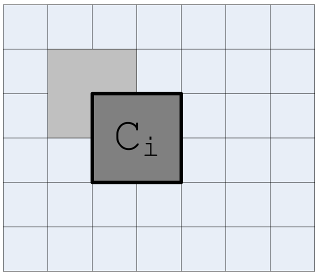
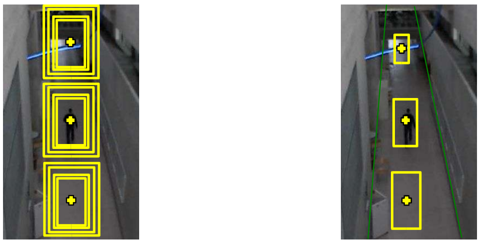


Figure 2: Concept of grid-based classification: a highly overlapping grid is placed over the image, where each grid element corresponds to a single classifier.

In addition, we can take advantage of knowing the scene calibration (i.e., we know the ground-plane), which is illustrated in Figure 3. Typically, as illustrated in Figure 3(a), a single classifier is evaluated on different positions and different scales. In contrast, if the approximative size of an object is known, as illustrated in Figure 3(b), the search space can be reduced.



(a) General approach for object detection, where each position in the image is evaluated on different scales.

(b) Scale information (if available) can be used to determine the size of a grid element, which increases the performance.

Figure 3: Including scene information reduces the search space.

2.2 Fixed Update Rules

Having a classifier grid such as defined in the previous section a compact classifier can be trained using an on-line learning method. But on-line systems have one main disadvantage: new unlabeled data has to be robustly included into an already built model. More formally, at time t given a classifier C_{t-1} and an unlabeled example $\mathbf{x}_t \in \mathbb{R}^m$, the classifier predicts a label $y_t \in \{+1, -1\}$ for $\mathbf{x}_t$, which can further be used to generate the label $\hat{y}_t$, which is then used to update the classifier: $C_t = \text{update}(C_{t-1}, \langle \mathbf{x}_t, \hat{y} \rangle)$.

Typical update schemes (i.e., label generators) for online learning are self-training (e.g., [15, 12]) and co-training (e.g., [2, 11]). But these update strategies suffer from the drifting problem. A classifier that was trained using many incorrect updates would yield many false positives and/or the detection rate would decrease. Further on, since the classifier response is used for labeling new samples, this would result in a self-fulfilling prophecy. In fact, self-training or co-training, which rely on a direct feedback of the current classifier, must be avoided.

Considering the constraints of the grid-based classifier structure, the labels $\hat{y}_t$ can be generated without a feedback of $C_{i,t-1}$ by using the following fixed update rules:

Positive updates: Given a set of positive (hand) labeled examples $\mathcal{X}^+$. Then, using

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

to update the classifier is a correct positive update. The set can by quite small; in the extremal case it contains only one positive sample. The only assumption is that $\mathcal{X}^+$ is a representative set. Roughly speaking, each possible appearance should be captured by this subset.

Negative updates: The probability that an object is present on patch $\mathbf{x}_i$ is given by

$$P(\mathbf{x}_i = \text{object}) = \frac{\#p_i}{\Delta t} \ , \tag{2}$$

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

$$\langle \mathbf{x}_{i,t}, -1 \rangle \quad (3)$$

is correct most of the time (wrong with probability $P(\mathbf{x}_i = \text{object})$). The probability of a wrong update for this particular image patch is indeed very low.

2.3 On-line Learning

Since the positive updates are always correct per definition the remaining problem is that occasionally false negative updates might be carried out. Hence, the applied on-line learning method (a) must cope with some (low) label noise and (b) must have fading memory (forgetting). In general, any learning method that fulfills these requirements can be applied, but in this work we use on-line boosting for feature selection [6].

In general, Boosting[1] forms a strong classifier

$$H(\mathbf{x}) = \sum_{n=1}^{N} \alpha_n h_n(\mathbf{x}) \quad (4)$$

by a linear combination of N weak classifiers $h_n(\mathbf{x})$, which have only to perform better than random guessing. These weak classifiers are trained by re-weighing the training samples, i.e., more emphasis is given to still misclassified examples. In order to do feature selection, where each weak classifier h_j corresponds to a feature f_j, for on-line boosting [14] selectors were introduced. Each selector is represented by its best weak hypothesis according to the estimated training error $\hat{e} = \frac{\lambda_{wrong}}{\lambda_{wrong}+\lambda_{corr}}$, where λ_{corr} and λ_{wrong} are the importance weights of the samples seen so far that were classified correctly and incorrectly, respectively. The actual boosting step can then be performed on these selectors. In this way, the classifier selects a good subset of simple image features from a large pool.

The weak classifier h_j (which corresponds to one feature j) is built based on two distributions, i.e., the estimated response of the feature D_j^+ and D_j^- for negative and positive images, respectively. Based on these a simple decision stump is calculated. In order to select a feature, the reweighted error is continuously updated and the feature with the lowest estimated error is selected by each selector. Thus, finally, at each time a strong classifier (subset of features) is available.

3 Adaptive Grid-based Detector

Using the ideas discussed in Section 2 similar to [7] we can define a grid-based object detection system. In particular, we apply on-line boosting for feature selection to train the classifier by using a fixed update strategy. The main concept - consisting of an evaluation and an update stage - is depicted in Figure 4.

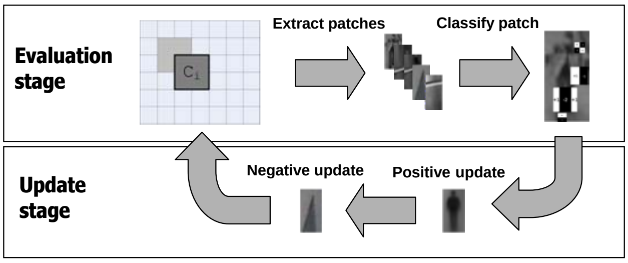


Figure 4: Overview of the grid-based approach. The grid elements are highly overlapping and have a fixed size, depending on the scene calibration. Each grid element is an independent classifier, which discriminates between objects and the background. In order to be adaptive to changes in the scene, each classifier is updated by a fixed update strategy.

First of all, the classifiers are initialized randomly. Then, at each time step in the evaluation stage a particular patch (grid-element) is analyzed and classified by the corresponding classifier. Independent of the obtained classification result the classifier is then updated using the fixed rule discussed in Section 2.2. In particular, a positive update is performed using a representative sample (e.g., the mean image of the training samples) and a negative update is performed using the current patch.

Considering this learning strategy three main problems arise: (a) if the variability of the object's appearance is too large a single sample would not sufficiently represent the appearance (even not if the complexity of the classification task is dramatically reduced). (b) Since the classifiers are updated for each arising image, (non-moving) objects may be included into the negative representation. (c) Due to the random initialization of the features a sub-optimal initial feature set would be selected that might not be sufficient to solve the required task. To overcome these problems, in the following we propose three strategies to improve the grid-based classification.

3.1 Efficient Fixed Update Strategy

In order to increase the robustness of the grid-based object detector, we slightly modified the fixed update strategy. Instead of iteratively updating the classifier using a single sample or a small set of labeled samples, which have to be kept in memory, we pre-train the positive distribution D_j^+ for each feature j. Therefore, the system is updated until the distribution of the positives values converge. Since only the thus obtained distribution is required later on, even a huge training set representing a wide range of variability can be used in this pre-training step.

Afterwards the distribution of the positive feature values stays fixed and only the negative distribution is adapted over time (i.e., the patch corresponding to the classifier is used for the negative update). Since no positive updates are required as an additional advantage the performance is increased. In fact, in the update stage only negative updates are necessary. This is illustrated in Figure 5 for one specific weak classifier.

3.2 Feature Pre-selection

For training our classifiers we apply on-line AdaBoost for feature selection [6], which initializes each weak classifier with randomly selected features. As a result the final strong

[1] In this paper, we focus on the discrete AdaBoost algorithm for binary classification problems [5].

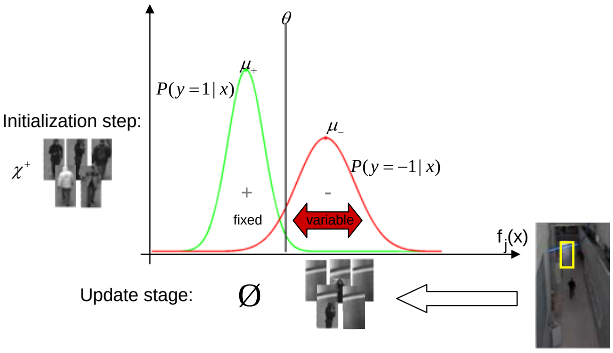


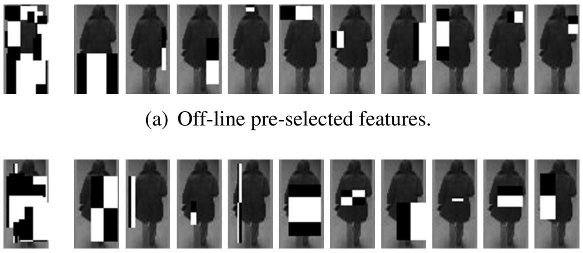
Figure 5: The two distributions describe the feature values for the positive and negative examples of one weak classifier. The green Gaussian distribution describes the distribution of the feature values of all positive examples, which is pre-trained and fixed. The red Gaussian distribution models the distribution of the feature values for all negative examples. The standard deviation and the displacement of the mean value depend on the images in the scene.

classifier is highly dependent on this random initialization! The random initialization of the classifier has an even larger impact, if the number of selectors and the number of weak classifiers are small. The number of selectors that is necessary to solve the classification problem depends on the complexity of the problem. Considering for instance a quite simple classification problem, where 10 selectors, each containing 20 weak classifiers are sufficient to solve the task. In this case each selector chooses one weak classifier out of 20 randomly initialized weak classifiers. If the 20 features out of these 20 randomly selected weak classifiers are not very accurate to solve the classification problem, this selector would give poor results. But this can be avoided by initializing the feature pool in a more sophisticated way than random selection.

In order to overcome this problem we train an off-line classifier by using off-line AdaBoost for feature selection at a pre-training stage. The thus selected and pre-trained features are used in the on-line training later on. We adapted the off-line AdaBoost for feature selection algorithm [16]. Instead of selecting the best feature out of a large feature pool we selected the best n features, where n is the number of weak classifiers in one selector of the on-line algorithm. In each iteration of the off-line version one selector is initialized. The exactly calculated error of the off-line algorithm can be used to initialize the two variables λ_{wrong} and $\lambda_{correct}$, which are used for estimating the error of the on-line algorithm. The off-line pre-trained classifier ensures that instead of choosing random features the strong classifier consists of features that are more appropriate for this task. The off-line trained classifier can either be trained for a specific scene or for a generic task. An illustrative example of thus initialized features sets are given in Figure 6. It clearly can be seen that the features shown in Figure 6(a) describe a person considerably better than those shown in Figure 6(b).

3.3 Including Prior Knowledge

Since we are very often dealing with problems where prior knowledge is available it might be advantageous to use this information in order to further improve the performance of an object detection system. In particular, for an adaptive system, which should be as adaptive as possible, this prior information can limit the problem of over-adaptivity.



(a) Off-line pre-selected features.

(b) Randomly selected features.

Figure 6: Sophisticated (a) vs. random (b) feature initialization.

We take advantage of prior knowledge in form of an off-line trained classifier that is directly included into the response of the strong classifier:

$$H(x) = \text{sign}(p \cdot \text{conf}_{H_{\text{off}}}(x) + (1-p) \cdot \text{conf}_{H_{\text{on}}}(x)), \quad (5)$$

where H_{off} is the prior knowledge and H_{on} is the on-line classifier. The required confidence of a classifier is calculated as

$$\text{conf}(x) = \frac{\sum_{t=1}^{T} \alpha_t h_t(x)}{\sum_{t=1}^{t} \alpha_t}, \quad (6)$$

where α_t is the voting weight of the classifier h_t of the ensemble.

We use an off-line classifier trained by the off-line AdaBoost for feature selection algorithm to describe the prior knowledge. Depending on the training data used for this off-line classifier either a common off-line object detector (e.g., a pedestrian detector) or an off-line classifier for a specific scene can be trained. Training the off-line classifier on the specific scene might further improve the results. This single off-line classifier is used as a decision support for all classifiers in the classifier grid.

Depending on the performance of the off-line classifier the influence p of the off-line classifier on the whole classification result can be varied. Instead of just using the class label of the on-line and the off-line classifier we used the confidence values of both classifiers and combined them.

4 Experimental Results

In the following, we demonstrate the benefits of the presented approach by three different experiments. First, we give a detailed evaluation on a publicly available pedestrian detection benchmark dataset in Section 4.1. Second, to demonstrate that the method is not limited to pedestrian detection it is applied for car detection in Section 4.2. Third, since the goal was to develop a system that is stable over time, in Section 4.3 we show the stability of the proposed approach in a long-term experiment, where we performed approximately 150,000 updates.

For a quantitative evaluation, we use recall-precision curves (RPC) [1]. Therefore, the number of true positives TP and the number of false positives FP are computed based on the given ground-truth. A detection is accepted as true positive if it fulfills the overlap as well as the relative distance criterion where for both criteria the parameters (minimal overlap, maximal relative distance) are set to 50%. The precision rate PR describing the accuracy of the detections is calculated by

$$PR = \frac{TP}{TP + FP} \tag{7}$$

whereas the recall-rate RR describing the number of positive samples that were correctly classified is given by

$$RR = \frac{TP}{P} \ , \tag{8}$$

where P is the total number of objects in the ground-truth. Finally, to evaluate the detection results we plot the recall-rate RR against $1 - PR$.

4.1 Pedestrian Detection

First of all, we give a detailed evaluation of the proposed method on a challenging publicly available pedestrian detection benchmark dataset, i.e., "Central Pedestrian Crossing" sequence from Leibe et. al. [10]. The dataset consists of three different outdoor scenes, but in particular, the results shown here were obtained from the first of these three sequences. The sequence contains 101 frames of a resolution of 640 × 480. This specific scene was chosen, since it contains both, frontal views and side views of pedestrians. Thus, it represents a quite realistic scenario.

In addition, a ground-truth is available (i.e., every fourth frame is annotated). Thus, the updates for the proposed approach were performed in each frame but the evaluations were estimated only from the annotated frames. In particular, for the grid-based detector the parameters summarized in Table 1 were used:

Parameter	Value
Patch size	32×64
Overlap of grid elements	83%
Number of classifiers	1.262
Number of selectors per classifier	10
Number of weak classifiers per selector	20

Table 1: Parameters for the pedestrian detection experiment.

The small and compact classifiers used enable real-time detection even for this large number of 1.262 classifiers.

In addition, the proposed approach was compared to the following methods including low level cues such as background subtraction and state-of-the-art generic person detectors:

Background Model (BGM): The simplest method for object detection is a foreground/background segmentation using a(n) (adaptive) background model. In particular, we apply the approximated median background model [13]. The pixel intensity of the background model $B(x, y)$ is increased by a constant value c if $B(x, y) < I_n(x, y)$, where $I_n(x, y)$ is the intensity of the current image, and is decreased otherwise.

Template Matching (TM): In contrast to BGM Template Matching is an appearance-based approach. By using a sliding window technique each patch of an image is tested if it is consistent with a template or not. Typically, for that purpose the similarity between the intensity values of the template and the analyzed patch are estimated by applying the "normalized cross-correlation" as similarity measure.

BGM+TM: The combination of a background model and template matching can be seen as a very simple pendant to the grid-based approach. In fact, for a given patch the appearance and the background is described within a single model.

Viola & Jones: The object detector of Viola and Jones [17] is based on a cascade of classifiers trained by the off-line AdaBoost for feature selection. For our experiments we used the full body classifier attached to the OpenCV[2] [9].

Dalal & Triggs: The pedestrian detector of Dalal and Triggs [3] is based on histogram of oriented gradient descriptors. For our comparison we used the code available on the Internet[3].

Felzenszwalb et al.: The object detector of Felzenszwalb et al. [4] is trained using a latent SVM (to cope with articulated parts) for multiple objects (e.g., persons, cars, buses). In particular, for this experiment the person detector trained for the VOC'06 challenge was used. The MATLAB code as well as the classifiers can be downloaded from the Internet[4].

The thus obtained results sorted by the F-measure are summarized in Table 2. To ensure a fair comparison, in a post-processing step we discarded all detections with an inappropriate scale. In fact, a detection was removed if the scale was smaller than 75% or greater than 150% of the expected patch-size (defined by the corresponding grid-element). Please note, this post-processing does not reduce the recall since these detections would be counted as false positives otherwise.

From Table 2 it can be seen that the highest recall is obtained by using the background model (BGM) and template matching (TM), but in both cases the precision is far below a sufficient level in practice. But by combining these two approaches "competitive" results can be obtained. In fact, state-of-the-art methods such as Viola & Jones [17] and Dalal & Triggs [3] are clearly outperformed and the results are comparable to the approach of Grabner et al. [7]. But the recall is still insufficient. In contrast, by applying the proposed approach the recall-rate is increased to 63% at a precision of 96%! Considering the complexity of the task (i.e., the scene mainly contains persons from the side view) the

[2] http://sourceforge.net/projects/opencvlibrary

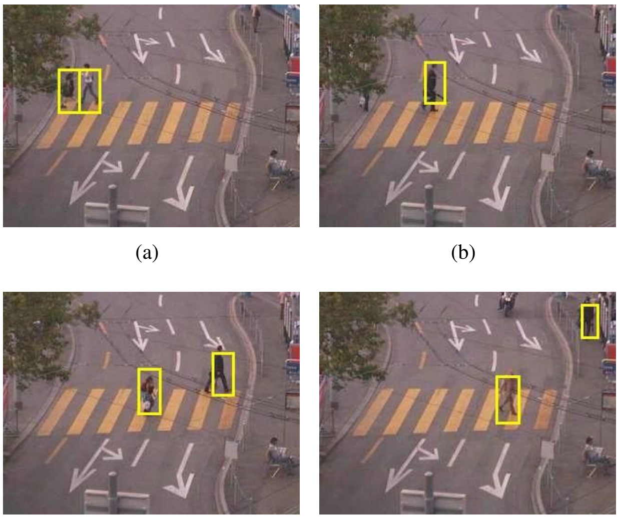
[3] http://pascal.inrialpes.fr/soft/olt

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

method	recall	precision	F-measure
TM	**0.94**	0.04	0.08
Dalal & Triggs [3]	0.19	0.15	0.17
Viola & Jones [17]	0.31	0.13	0.18
Felzenszwalb [4]	0.26	0.32	0.29
BGM [13]	0.93	0.24	0.38
BGM + TM	0.55	0.83	0.66
Grabner et al. [7]	0.55	0.88	0.68
proposed (3.1)	0.67	0.75	0.70
proposed (3.1+3.2)	0.63	**0.96**	**0.76**

Table 2: Comparison of different approaches for the pedestrian detection experiment.

obtained results are quite competitive. Finally, in Figure 7 we show some illustrative detection results for the proposed grid-based detector.



(a)

(b)

(c)

(d)

Figure 7: Exemplary detection results for the pedestrian detection experiments obtained by the proposed approach.

4.2 Car Detection

The previous experiment was focused on pedestrian detection. In order to demonstrate that this approach is not limited to person detection, we apply it for car detection. In particular, the experiment was carried out on a highway surveillance sequence with a resolution of 720 × 576, which contains 1000 frames.

Compared to person detection car detection is a more challenging task, because there is more variability in the appearance (i.e., we have to detect limousines, minivans, compact cars, etc.). Thus, more complex classifiers are required to obtain detection results of sufficient accuracy. In particular, for our experiments we applied classifiers containing 70 selectors, each consisting of 30 weak classifiers. The parameters for this experiment are summarized in Table 3.

In addition, we used pre-selected features and included an off-line prior, i.e., an off-line trained classifier with 500 weak classifiers, such as described in Section 3.2 and Section 3.3. In order to be less liable to label noise and to increase the performance of our object detection system, we modified the update strategy. Instead of updating the patches in all arriving images, the classifiers were updated only every fifth frame. The other frames were used for evaluation, but no updates were performed.

Parameter	Value
Patch size	50 x 50
Overlap of grid elements	92%
Number of classifiers	1.163
Number of selectors per classifier	70
Number of weak classifiers per classifier	30

Table 3: Parameters for the car detection experiment.

For evaluation purposes, we evaluated three classifiers in parallel: (a) fixed off-line pre-trained classifier, (b) adaptive on-line grid-based classifier, and (c) a classifier combining off-line and on-line information using Eq. (5). The corresponding precision-recall curves are shown in Figure 8.

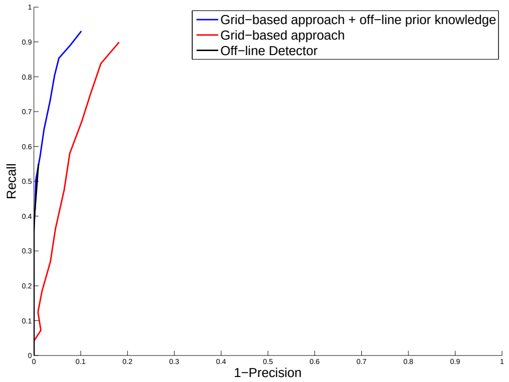


Figure 8: Recall-Precision curves for the car detection experiment: off-line detector, grid-based detector, and combined detector.

It can be seen that the off-line detector has an excellent precision but the recall is too low in practice. In contrast, the pure on-line classifier has a high recall and a precision, which might be sufficient for the task. But if both approaches are combined we obtain a high recall-rate at a high precision. Thus, it is clear that the combination introduced in Section 3.3 can be beneficial for practical applications. In this way, starting from a fixed prior the on-line classifier assures the required adaptivity.

Exemplary detection results are shown in Figure 9. The yellow bounding boxes indicate the detections whereas the big white box is the region where the detector is applied. In this case, the actual task was to detect a car once within the detection area and to further track those cars that were detected (e.g., for speed measurement).

4.3 Long-term Experiment

Finally, we demonstrate the proposed approach in a long-term experiment for two purposes. First, we want to show that the proposed grid-based detector does not drift even if a large number of updates is performed. Second, we want to emphasize the need for an adaptive object detection system.

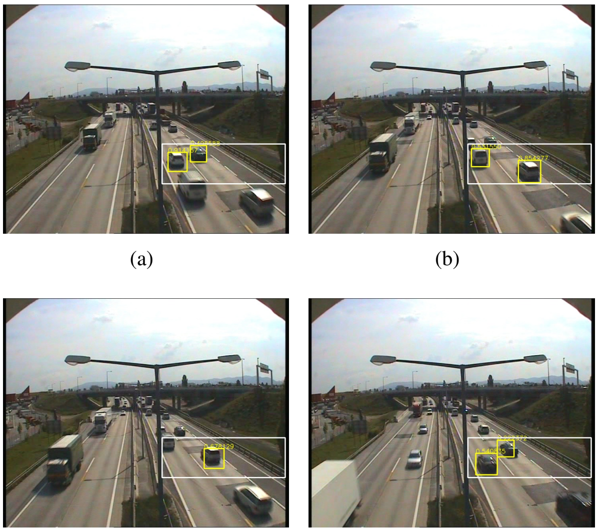

(a) (b) (c) (d)

Figure 9: Exemplary detection results for the car detection experiment. The white rectangle indicates the detection region.

Thus, in this experiment we observed a corridor in a public building for 24 hours and processed in total 150,000 frames with a size of 240 × 320. In fact, each classifier corresponding to a grid-element was updated as a new frame arose.

Similar to the experiments carried out in Section 4.1 each classifier holds 10 selectors consisting of 20 weak classifiers. A complete list of all required parameters for this setup is given in Table 4. In addition, to increase the performance, we used a sub-set of features and included an off-line prior such as described in Sections 3.2 and 3.3.

Parameter	**Value**
Patch size	32×64
Overlap of grid elements	90%
Number of classifiers	1.255
Number of selectors per classifier	10
Number of weak classifiers per classifier	20

Table 4: Parameter settings for the long-term experiment.

During the 24 hours we evaluated the performance at four different points in time (afternoon, evening, morning, and afternoon). In particular, after 100, 31,500, 110,000, and 149,000 processed frames we evaluated a fixed sequence of 500 frames (to show the stability the last sequence contains 1000 frames) and computed the precision-recall curves, which are shown in Figure 10.

One can clearly see that the performance of the system is not decreased, even after 150.000 unsupervised updates. The slightly different performance curves can be explained by slightly the different complexity of the selected sequences (i.e., lightening condition, number of persons within the sequence, etc.). In fact, the worst results are obtained for frames 100 - 599 right after the experiment was started!

Finally, we show some illustrative detection results in Figure 11, that were obtained during 24 hours. It clearly can be seen that the environmental conditions (i.e., mainly illumination) are drastically changing over time, which arises the need for an adaptive system. However, as can be seen the proposed approach can handle this challenging task.

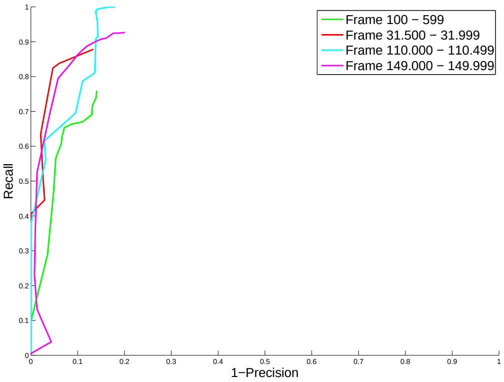


Figure 10: Recall-Precision curve for four different sequences during the 24 hours experiment.

5 Conclusion

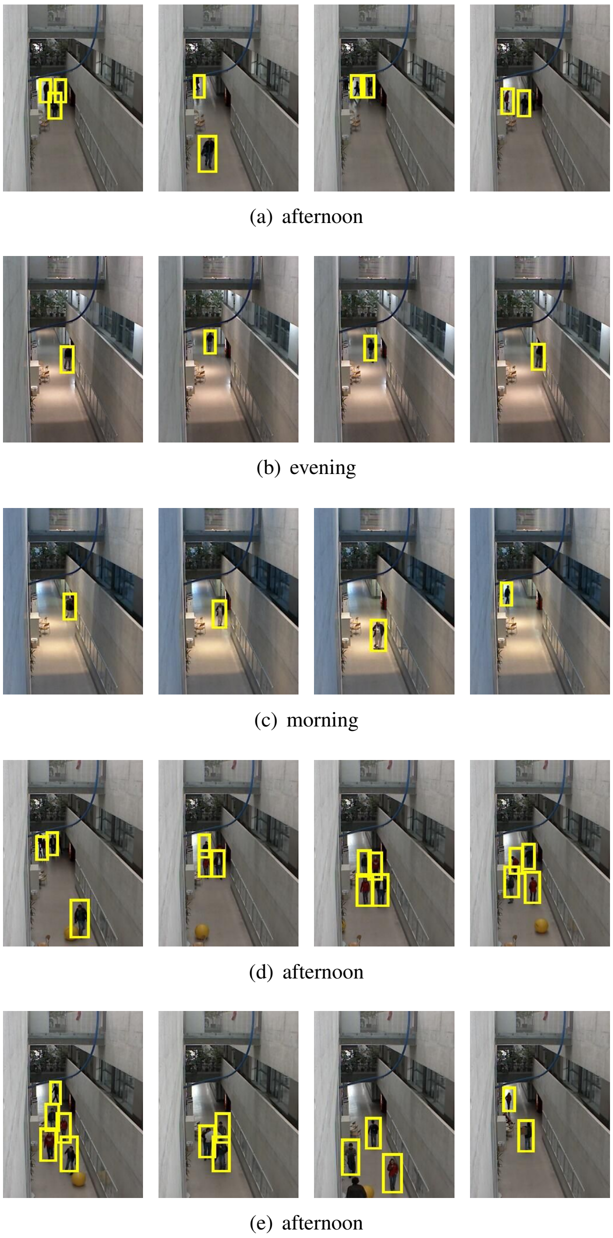
In this paper, we presented a robust adaptive method for object detection from static cameras. In particular, we extended the idea of classifier grids in a way that it can be applied for real-world scenarios. The main idea is to on-line train a specific classifier by on-line boosting for each image location, which allows to adapt to changing environmental conditions. To avoid drifting, i.e., ensure that the classifier does not get corrupted during on-line updating, the positive representation is kept fixed whereas only the negative representation is updated. In addition, to improve the classification results we pre-select a small subset of highly valuable features and optionally include an off-line trained classifier in our decision function. The experimental results show that the proposed approach does not only outperforms simple detection methods such as background subtraction or template matching but also state-of-the-art object detectors. In addition, in a long-term experiment we showed that the proposed update strategy is stable over time and that the system can be applied in a real-world 24/7 scenario.

Acknowledgment

This work was supported by the FFG project HI-MONI under the COMET programme in co-operation with FTW, by the FFG project AUTOVISTA (813395) under the FIT-IT programme, and by the Austrian Joint Research Project Cognitive Vision under projects S9103-N04 and S9104-N04. In addition, this work was supported by the EU-project SCOVIS under grant agreement no 216465.

References

[1] S. Agarwal, A. Awan, and D. 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.


(a) afternoon

(b) evening

(c) morning

(d) afternoon

(e) afternoon

Figure 11: Exemplary results of the long term experiment.

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

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 886–893, 2005.

[4] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2008.

[5] Y. Freund and R. E. Shapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55:119–139, 1997.

[6] H. Grabner and H. Bischof. On-line boosting and vision. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 260–267, 2006.

[7] H. Grabner, P. M. Roth, and H. Bischof. Is pedestrian detection really a hard task? In *Proc. IEEE Intern. Workshop on Performance Evaluation of Tracking and Surveillance*, pages 1–8, 2007.

[8] M. Heikkilä, M. Pietikäinen, and J. Heikkilä. A texture-based method for detecting moving objects. In *Proc. British Machine Vision Conf.*, pages 187–196, 2004.

[9] H. Kruppa, M. Castrillon-Santana, and B. Schiele. Fast and robust face finding via local context. In *Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2003.

[10] B. Leibe, K. Schindler, and L. v. Gool. Coupled detection and trajectory estimation for multi-object tracking. In *Proc. IEEE Intern. Conf. on Computer Vision*, October 2007.

[11] A. Levin, P. Viola, and Y. Freund. Unsupervised improvement of visual detectors using co-training. In *Proc. IEEE Intern. Conf. on Computer Vision*, volume I, pages 626–633, 2003.

[12] L.-J. Li, G. Wang, and L. 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.

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

[14] N. C. Oza and S. Russell. Online bagging and boosting. In *Proc. Artificial Intelligence and Statistics*, pages 105–112, 2001.

[15] C. Rosenberg, M. Hebert, and H. Schneiderman. Semi-supervised self-training of object detection models. In *IEEE Workshop on Applications of Computer Vision*, pages 29–36, 2005.

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

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