Patch-Based Experiments with Object Classification in Video Surveillance

Patch-Based Experiments with Object Classification in Video Surveillance

Rob Wijnhoven[1,2] and Peter H.N. de With[2,3]

[1] Bosch Security Systems B.V., Glaslaan 2, Eindhoven, The Netherlands
[2] Technische Universiteit Eindhoven, Eindhoven, The Netherlands
[3] LogicaCMG, Tech. Softw. Eng., Eindhoven, The Netherlands

Abstract. We present a patch-based algorithm for the purpose of object classification in video surveillance. Within detected regions-of-interest (ROIs) of moving objects in the scene, a feature vector is calculated based on template matching of a large set of image patches. Instead of matching direct image pixels, we use Gabor-filtered versions of the input image at several scales. This approach has been adopted from recent experiments in generic object-recognition tasks. We present results for a new typical video surveillance dataset containing over 9,000 object images. Furthermore, we compare our system performance with another existing smaller surveillance dataset. We have found that with 50 training samples or higher, our detection rate is on the average above 95%. Because of the inherent scalability of the algorithm, an embedded system implementation is well within reach.

1 Introduction

Traditional video surveillance systems comprise of video cameras generating content-agnostic video streams, being recorded by digital video recorders. Recently, there is a shift towards smart cameras that generate a notion of the activity in the monitored scene by means of Video Content Analysis (VCA). State-of-the-art VCA systems comprise object detection and tracking, thereby generating location data of key objects in the video imagery of each camera. For video surveillance, this technology can be used to effectively assist security personnel.

While the detection and tracking algorithms are becoming mature, the classification of the detected objects is still in an early stage. Classification of the detected objects is commonly done using the size of the object, where simple camera calibration is applied to compensate for the perspective. However, effects such as shadows and occlusion negatively influence the segmentation process and thus the object classification (e.g. shadows increase the object size, and occlusion decreases the size). Furthermore, when objects cross each other, they may be combined into one object. For improved scene understanding, more advanced object models are required, taking specific object features from the video into account. The aim of our object modeling is to classify various objects in a reliable way, thereby supporting the decision-making process for a security operator of a CCTV surveillance system.

J. Blanc-Talon et al. (Eds.): ACIVS 2007, LNCS 4678, pp. 285–296, 2007.

In the presented work, we assume that the camera image has been segmented into a static background and moving foreground objects using the algorithm proposed in [1]. Initially, a texture and intensity analysis is applied between the input image and the background reference frame at low-resolution. The resulting initial foreground image blocks are further analyzed at high-resolution to obtain a pixel-true segmentation mask. The extracted objects are represented by a shape and bounding box description and will be referred to as Regions-Of-Interest (ROIs) in the remainder of the paper.

In previous work [2] [3], wire-frame models were matched onto the detected ROIs that represent the detected objects. The disadvantage of this approach is that for each object, such a wire-frame model has to be designed and when the number of objects grows, the classification distance between the models decreases. Furthermore, the computational requirement grows linearly with the number of object models. As an alternative, in this paper we study a patch-based algorithm as proposed by Serre *et al.* [4]. In this technique, the computational expensive stage of template and pattern matching, is independent of the number of object classes and the classification is performed afterwards, on a subset of the data, using feature vectors. Classification results for this algorithm show that a classification rate above 95% is possible. The two approaches are compared under the conditions of a possible implementation in an embedded environment, where the computation power available is strictly limited and scalability of the algorithm is important.

The remainder of the paper is as follows. In Section 2 related work is presented. Section 3 discusses the model that we use for object classification. The dataset used is introduced in Section 4. The results of the algorithm are presented in Section 5, including a discussion on the comparison of the presented algorithm and the previously considered wire-frame approach. The paper ends with conclusions and future work.

2 Related Work

Model-based object classification/detection approaches are based on two different classes of models: rigid (non-deformable) and non-rigid (deformable) models. Rigid models are commonly used for the detection of objects like vehicles, where non-rigid models are typically used for person detection.

In the following, we consider three types of algorithms. In various surveillance systems, classification methods are commonly based on the pixel-size of the object's ROI. More advanced algorithms for traffic surveillance match 3D wire-frame models onto the input image for the purpose of object tracking or classification. Within the domain of generic object recognition in large multimedia databases, various proposed algorithms are based on low-level local descriptors that model the object's appearance. Each of the three methods will now be addressed in more detail.

Region-of-interest methods are the most simple object models and computationally inexpensive. Systems that segment the camera input images into a static background image and moving foreground images (e.g. [1]), generate the object's ROI, which already provides some information about the detected objects, e.g. pixel-size and -speed. Bose and Grimson [5] use the area of the bounding box and the percentage of foreground pixels within the box as features. Furthermore, the y-coordinate is used to compensate for the perspective in the scene. A different method for obtaining perspective invariance is applied by Haritaoglu *et al.* [6], who use projection histograms in x- and y-direction for tracked objects to make a distinction between various object types.

Wire-frame models have been proposed for the purpose of model-based object detection and tracking [2] [3]. For a more complete overview, we refer to previous work of the authors [7], where rigid object models have been considered for the purpose of vehicle classification. The algorithm is briefly summarized here as it will be discussed later in the paper. Within the already available ROI, the algorithm tries to find the best matching image position for all models in the database. After applying a 3×3 Sobel filter to the image in x- and y-direction, a histogram of gradient orientations is generated, from which the object orientation is extracted. Next, the 3D wire-frame model is projected onto the 2D camera image, using the calculated orientation and the center of the ROI as the object location. The projected 2D line-set is shifted over the image region and calculates a matching error for each pixel position. The position giving the smallest error defines the best matching pixel position. This is performed for all models in the database, and the model with the lowest matching error is chosen as the classified object model.

Low-level image features describing the object appearance are used by several object recognition systems. Haar-wavelets are commonly used, because of the low computational complexity [8], [9], [10].

Mikolajczyk and Schmid [11] compare the performance of various local interest descriptors. They show that Scale Invariant Feature Transform (SIFT) descriptors and the proposed extension of SIFT, Gradient Location and Orientation Histogram (GLOH), outperform other methods. Dalai and Triggs [12] compare the performance of Haar wavelets, PCA-SIFT [13] and Histogram Of Gradient methods (HoG). They show that the HoG method outperforms the others. Mikolajczyk *et al.* [14] generate HoG features for the purpose of person detection, extended with Laplacian-filtered versions of the input images as blob detectors.

Ma and Grimson [15] propose a method based on SIFT for the purpose of vehicle classification in traffic video using a constant camera viewpoint.

Serre *et al.* [4] model findings from biology and neuro-science using a hierarchical feed-forward architecture. The model is shown to have performance in line with human subjects, considering the first 150 ms of the human visual system in a simple binary classification task [16]. Serre *et al.* have shown that the algorithm outperforms SIFT in the generic object-recognition task. As mentioned, the advantage of this approach is that the image analysis part is independent of the amount of object classes. For this reason, the algorithm is suited for embedded implementation and was therefore adopted for further exploration.

3 Algorithm Model

Since humans are good at object classification, it is reasonable to look into biological and neurological findings. Based on findings from Hubel and Wiesel [17], Riesenhuber and Poggio have developed the ”HMAX” model [18] that has been extended recently by Serre [19], [4] and optimized by Mutch and Lowe [20]. We have implemented the model proposed by Serre up to the second processing layer. In his thesis, Serre [16] proposes to extend the model with additional third and fourth layers. For completeness, we will address the working of the algorithm in the following. A simplified graphical representation of the model for classification of objects detected in a video camera is shown in Figure 1, where the first step of object detection is described in [1].

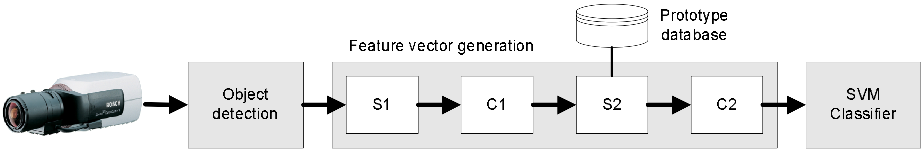

Fig. 1. Architecture for classification of objects in camera image

The algorithm is based on the concept of a feed-forward architecture, alternating between simple and complex layers, in line with the findings of Hubel and Wiesel [17]. The first layer implements line-detectors by filtering the graylevel input image with Gabor filters of several sizes to obtain scale-invariance. The filters are normalized to have zero mean and a unity sum of squares. The filter size of the smallest filter (at scale zero) has a size of 7 × 7 elements, increasing for every scale up to 37 × 37 elements (at scale 15).

The Gabor response is defined by:

$$G(x, y) = exp\ (-\frac{X^2 + \gamma^2 Y^2}{2\sigma^2})\ cos\ (\frac{2\pi}{\lambda} X), \tag{1}$$

where

$$X = x\ cos\ \sigma - y\ sin\ \sigma \tag{2}$$

$$Y = x\ sin\ \sigma + y\ \cos\ \sigma. \tag{3}$$

We use the parameters as proposed by Serre *et al.* [4]. After applying the Gabor filters onto the input image, the results are normalized. This compensates for the image energy in each area of the input image that is used to generate the filter-response. Hence, the final filter response for each filter is defined as:

$$R(I, F) = \left| \frac{\sum I_i F_i}{\sqrt{\sum {I_i}^2}} \right|, \tag{4}$$

where I_i denote pixels of the input image, and F_i denote the actual pixels within the filter aperture. This filter response is called the S1 feature map. An example of such a response for a car image, is shown in Figure 2.

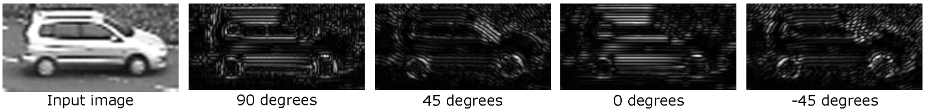

Fig. 2. Gabor filter response (filter size 7×7 elements) on input image of a car (scaled to 140 pixels in height)

3.1 Complex Layer 1 (C1)

The C1 layer from Figure 1 is added to obtain invariance in local neighborhoods. This invariance will be created in both the spatial dimensions and in the dimension of scale. Considering the dimension of scale, two S1 feature maps in consecutive scales (132 elements in height for scale zero) are element-wise maximized. This generates one feature map for every two scales. The combination of several scales results in a band. Next, in order to obtain spatial invariance, the maximum is taken over a local spatial neighborhood around each pixel and the resulting image is sub-sampled. Because of the down-sampling, the number of C1 features is much lower than the number of S1 features. The resulting C1 feature maps for the input image (33 elements in height at band zero and 12 at band 7) of the car image in Figure 2 are shown in Figure 3.

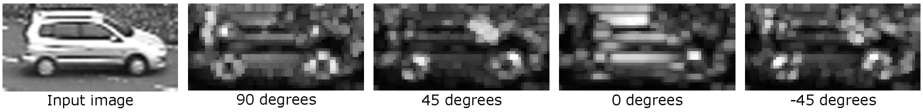

Fig. 3. C1 feature maps for S1 responses from Figure 2 (at band 0). Note that the C1 maps are re-scaled for visualization.

3.2 Simple Layer 2 (S2)

The next layer in the processing chain of the model applies template matching of image patches onto the C1 feature maps. This can be compared to the simple layer S1, where the filter response is generated for several Gabor filters. This template matching is done for several image patches (prototypes). These patch prototypes are extracted from natural images at a random band and spatial location, at the C1 level. Each prototype contains all four orientations and prototypes are extracted at four different sizes: 4×4, 8×8, 12×12 and 16×16 elements. Hence, a 4×4 patch contains 64 C1 elements.

Serre [16] has shown that for a large number of prototypes, the patches can be extracted from random natural images, and do not specifically have to be extracted from the training set.

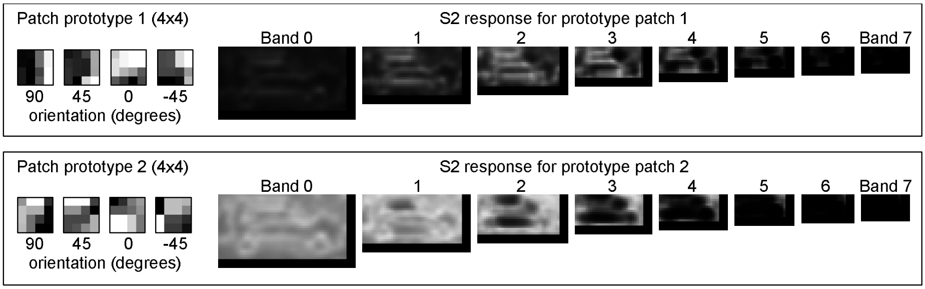

Fig. 4. Patch response for two example patches. The eight images of decreasing size represent the S2 feature maps at each band. Note that the top prototype clearly results in higher responses in the medium bands, where the lower prototype gives a higher reaction in the lower bands. For simplicity, only patches of size 4 × 4 C1 elements are considered.

The response of a prototype patch P over the C1 feature map C of the input image I is defined by a radial basis function that normalizes the response to the patch-size considered, as proposed by Mutch and Lowe [20].

Examples of image patches (prototypes) are shown in Figure 4 for the car image from Figures 2 and 3. Note that we only show two patch prototypes, each of size 4 × 4 C1 elements.

3.3 Complex Features Layer 2 (C2) and Feature Vector Classification

In this layer, for each prototype patch, the most relevant response is extracted and stored in the final feature vector. This is done by taking the maximum patch-response over all bands and all spatial locations. Therefore, the final feature vector has a dimensionality equal to the number of prototype patches used. In our implementation, we used 1,000 prototype patches. Note that by considering a higher or lower number of C1 patch prototypes, the required computation power can be linearly scaled.

In order to classify the resulting C2 feature vector, we use a one-vs-all SVM classifier with a linear kernel. The SVM with highest output score defines the output class of the feature vector. The Torch3 library [21] was used for the implementation of the SVM. Note that instead of the SVM, also a neural network could have been used for the feature vector classification.

4 Dataset and Experimental Setup

The algorithm model of the previous section was implemented as follows. The S1 layer filters the input image with Gabor filters at several scales, followed by the C1 layer to obtain invariance in both scale and space. In the S2 layer, the C1 feature maps are template matched with a high number of prototype patches. The final C2 layer obtains invariance by taking the global maximum over both scale and space for each prototype patch. For each prototype patch, this maximum value is stored in the final feature vector, which is classified using the support vector machine.

The use of a relevant dataset is very important for objective comparison of the proposed algorithms. Ponce *et al.* [22] discuss the datasets commonly used for generic object detection/recognition. However, these generic datasets are not specific for the typical surveillance case. Most available surveillance datasets have been created for the purpose of object tracking, and therefore contain a strictly limited number of different objects. For the purpose of object classification, a high number of different objects is required. Ma and Grimson [15] presented a limited dataset for separating various car types. Since future smart cameras should be able to make a distinction between more object classes, we have created a new dataset.

A one hour video-recording was made from a single, static camera, monitoring a traffic crossing. The camera image was captured at CIF resolution (352x288 pixels), resulting in object ROIs of 10-100 pixels in height for a person in the distance and a nearby bus, respectively. After applying the tracking algorithm proposed by the authors of [1], the resulting object images were manually adjusted if required, to have a clean performance of the ROI extraction and avoid any possible negative interference with the new algorithm. For this reason, redundant images, images of occluded objects and images containing false detections have been removed. Because of the limited time-span of the recording, the scene conditions do not change significantly. The final dataset contains 9,233 images of objects. The total object set has been split into the following 13 classes: trailers, cars, city buses, Phileas buses (name of a specific type of bus), small buses, trucks, small trucks, persons, cleaning cars, bicycles, jeeps, combos and scooters. Some examples of each object class are shown in Figure 5.

The experiments were conducted on a PC P-IV running at 2 GHz. The average processing time of an object image is about 4 to 5 seconds.

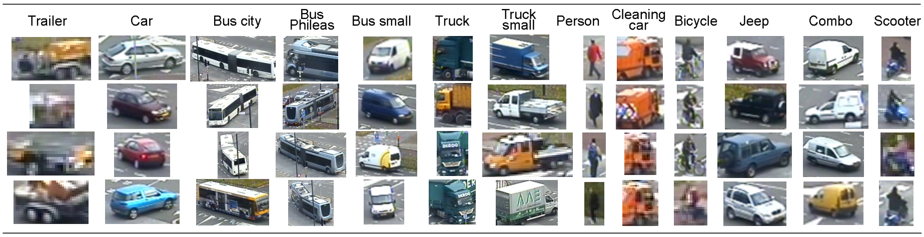

Fig. 5. Surveillance dataset Wijnhoven 2006

5 Results

This section shows the results for the object classification on the surveillance dataset presented in Section 4. Each image is first converted to grayscale and scaled to 140 pixels in height while maintaining the aspect ratio. The total set of images for each class is divided into a training and a test set at random. For the training set, the number is specified (e.g. 30 samples) and the remainder of the images is used for the test set.

Next, the feature vectors for all images are calculated using the methods discussed in Section 3. The SVM classifier is trained with the feature vectors of the images in the training set and tested with the test set. We present the detection rate, being the percentage of images correctly classified. The final detection rate is calculated by averaging the results over ten iterations.

The average correct detection rate in the case of 30 training samples per class is 87.7%. The main misdetections are bicycles and scooters (13%), and combos and small buses (13%).

For some simple applications, the classification between four object classes is already significant. A camera that can make a distinction between cars, buses, persons and bikes with high accuracy adds functionality to the camera that only comprises object detection and tracking. Therefore, the total dataset of 9,233 object images has been redivided into a new dataset, containing only the mentioned four object classes. Applying the same tests as mentioned before, result in an increase in detection rate. Furthermore, because there are less classes with a low number of object images, the number of learning samples can be increased. Table 1 shows that the detection rate of such a four-class system increases to 94.6% for 30 samples and up to 97.6% when 100 samples are learned.

Furthermore, we have compared our system with the system of Ma and Grimson [5]. As can be seen in Table 2, our system outperforms the proposed SIFT-based system for the car-van problem, in contrast to the sedan-taxi problem. Where our proposed algorithm has been designed to limit the influence of small changes within an object class, the SIFT-based algorithm focuses on describing more specific details of the test objects. This explains the differences in performance.

Table 1. Detection rates for the four-class classification problem

Training samples	Car	Bus	Person	Bike	Average
1	62.7%	38.8%	64.9%	66.6%	58.3%
5	86.8%	73.1%	91.8%	84.0%	83.9%
10	87.3%	91.9%	93.5%	89.4%	90.5%
20	93.1%	94.6%	95.2%	92.3%	93.8%
50	96.7%	96.4%	97.1%	93.4%	95.9%
100	97.3%	99.4%	98.2%	95.6%	97.6%

Table 2. Detection rates for the traffic dataset from Ma and Grimson [5]

	Ma Grimson	Our method	Difference
Car-van	98.5%	99.25%	+0.75%
Sedan-taxi	95.76%	95.25%	-0.49%

5.1 Wire-Frame Models vs. Feature-Based Object Modeling

In discussing the differences between the wire-frame approach and the patch-based techniques, we focus specifically on the trade-off between computational requirements and performance, which is very important for implementation in an embedded system.

Scale invariance is reached in the wire-frame approach by calibration of the camera. This results in correct projections of the 3D models onto the 2D camera image. With this *a-priori* knowledge, we scale the models to the correct size, so they are relevant for the image pixel-position they are considered at. The requirement of the calibration makes the wire-frame approach inherently sensitive to the object size.

In contrast with this, the patch-based algorithm implements scale-invariance by filtering with a set of Gabor filters of different size. By taking a global maximum in both scale and space in the C2 feature generation step, the algorithm is not influenced by the actual object size.

It should be noted, that the variation factor of object sizes in typical camera settings is quite limited. If they are large, scale-invariance can be reached by up- or down-sampling of the original image pixels.

Scalability in required computation power in the patch-based approach is reached by changing the number of C1 patch prototypes used in the template matching process, which is the most expensive part of the system. Furthermore, the parameters for the Gabor filters in S1 can be changed (e.g. number of orientations and scales considered). This filtering can be implemented in a fully parallel way. The generation of the feature vector is independent of the number of object classes considered, where in the case of wire-frame models, each model of the total set of 3D models needs to be matched.

A second aspect is that the template matching cost grows quadratically with the image resolution. Changing the input resolution of the object images directly results in a change of the required computation power. In the case of wire-frame models, the complexity of the calculation of the orientation using the gradient orientation histogram has a quadratic dependence on the image resolution, just as the calculation of the matching error.

The level of camera calibration required for VCA systems is important for the installer of a security system. Requesting a large number of parameters is impractical and therefore, a semi-automatic approach is preferred. In the case of wire-frame models, the installer only needs to calibrate the extrinsic camera parameters, since the intrinsic parameters are defined by the camera. The database of 3D models does not depend on the camera calibration.

In the patch-based approach however, for optimal performance, the classification system needs to be trained with training examples, coming from the actual setting of the camera. There is some robustness for small changes in the camera setting.

6 Conclusions and Future Work

We have presented a scalable patch-based algorithm, suited for parallel implementation in an embedded environment. The algorithm has been tested on a new dataset extracted from a typical traffic crossing. When the total set of object images is divided into 13 classes and 30 samples per class are used for training, a correct classification rate of 87.7% has been obtained. This performance increases to 94.6% when the set is split into only four classes and reaches 97.6% with 100 training samples. Furthermore, we have shown comparable performance with the SIFT-based algorithm by Ma and Grimson [5] using their dataset.

The previously mentioned performance can be further improved by exploiting application-specific information. Object-tracking algorithms provide useful information that can be taken into account in the classification step. Viola and Jones [23] show a performance gain by using the information from two consecutive frames. Another potential improvement can be made as follows. Extracting a sub-set of relevant features (C1 patch prototypes in our case) which are specific to our application, can give a performance gain as shown by Wu and Nevatia [24].

For future research, it is interesting to know how much sensor information is required to obtain a decent classification system. One of the first experiments would be to measure the influence of the input image resolution on the classification performance.

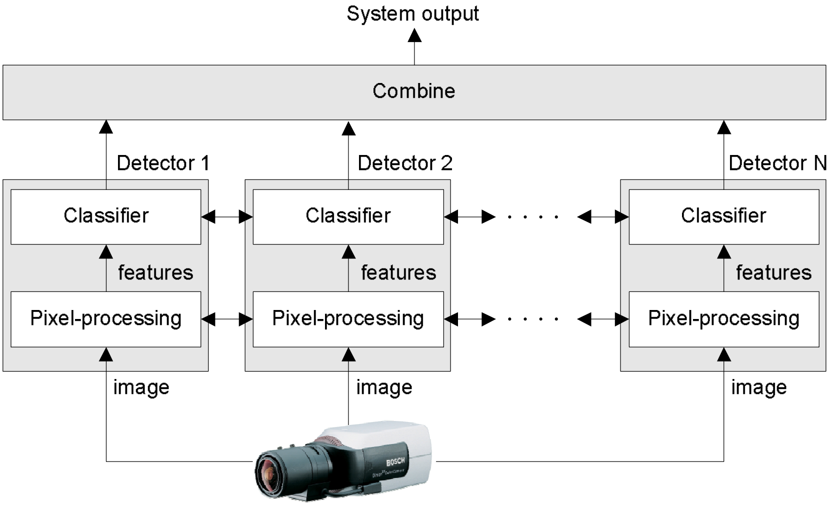

Fig. 6. Generic object modeling architecture, containing multiple detectors

A generic object modeling architecture can consist of several detectors that include pixel-processing elements and classification systems. We propose a generic architecture as visualized in Figure 6, where detectors can exchange both features extracted at the pixel level and classification results. For the purpose of person detection, Mohan *et al.* [9] propose multiple independent component detectors. The classifier output of each component is used in a final classification stage. In contrast to this fully parallel implementation, Zuo [25] proposes a cascaded structure with three different detectors to limit the computational cost in a face-detection system.

Recently, the authors have considered a 3D wire-frame modeling approach [7] that is completely application-specific. This means that for each typical new application, 3D models have to be manually generated. Furthermore, addition of a new object class requires a new model that differs from the other models and implies the design of a new detector. On the opposite, the patch-based approach is a more general approach which generates one feature vector for every object image and the SVM classifier is trained to make a distinction between the application-specific object classes.

In our view, when aiming at a generic object modeling architecture, we envision a convergence between application-specific techniques and application-independent algorithms, thereby leading to a mixture of both types of approaches. The architecture as shown in Figure 6 should be interpreted in this way. For example, in one detector the pixel processing may be generic whereas in the neighboring detector the pixel processing could be application-specific. The more generic detectors may be re-used for different purposes in several applications.

References

1. Muller-Schneiders, S., Jager, T., Loos, H., Niem, W.: Performance evaluation of a real time video surveillance system. In: Proc. of 2nd Joint IEEE Int. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS), pp. 137–144. IEEE Computer Society Press, Los Alamitos (2005)
2. Kollnig, H., Nagel, H.: 3d pose estimation by directly matching polyhedral models to gray value gradients. Int. Journal of Computer Vision (IJCV) 23(3), 283–302 (1997)
3. Lou, J., Tan, T., Hu, W., Yang, H., Maybank, S.: 3-d model-based vehicle tracking. IEEE Transactions on Image Processing 14(10), 1561–1569 (2005)
4. Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M., Poggio, T.: Robust object recognition with cortex-like mechanisms. Transactions on Pattern Analysis and Machine Intelligence (PAMI) 29(3), 411–426 (2007)
5. Bose, B., Grimson, W.E.L.: Improving object classification in far-field video. In: Proc. of IEEE Computer Vision and Pattern Recognition (CVPR), Washington, DC, USA, vol. 2, pp. 181–188. IEEE Computer Society Press, Los Alamitos (2004)
6. Haritaoglu, I., Harwood, D., Davis, L.: W4: real-time surveillance of people and their activities. In: IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 22, pp. 809–830. IEEE Computer Society Press, Los Alamitos (2000)
7. Wijnhoven, R., de With, P.: 3d wire-frame object-modeling experiments for video surveillance. In: Proc. of 27th Symposium on Information Theory in the Benelux, pp. 101–108 (2006)
8. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proc. of the 2001 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR)., vol. 1, pp. 511–518. IEEE, Los Alamitos (2001)
9. Mohan, A., Papageorgiou, C., Poggio, T.: Example-based object detection in images by components. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 23(4), 349–361 (2001)

10. Oren, M., Papageorgiou, C., Sinha, P., Osuna, E., Poggio, T.: Pedestrian detection using wavelet templates. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), San Juan, Puerto Rico, pp. 193–199. IEEE Computer Society Press, Los Alamitos (1997)
11. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 27(10), 1615–1630 (2005)
12. Dalai, N., Triggs, B.: Histogram of oriented gradients for human detection. In: Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 886–893. IEEE Computer Society Press, Los Alamitos (2005)
13. Ke, Y., Sukthankar, R.: Pca-sift: A more distinctive representation for local image descriptors. In: Proc. of IEEE Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. 506–513. IEEE, Los Alamitos (2004)
14. Mikolajczyk, K., Schmid, C., Zisserman, A.: Human detection based on a probabilistic assembly of robust part detectors. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3021, pp. 69–81. Springer, Heidelberg (2004)
15. Ma, X., Grimson, W.: Edge-based rich representation for vehicle classification. In: Proc. of IEEE Int. Conf. on Computer Vision (ICCV), vol. 2, pp. 1185–1192. IEEE Computer Society Press, Los Alamitos (2005)
16. Serre, T.: Learning a Dictionary of Shape-Components in Visual Cortex: Comparison with Neurons, Humans and Machines. PhD thesis, Massachusetts Institute of Technology Computer Science and Artificial Intelligence Laboratory (April 2006)
17. Ullman, S., Vidal-Naquet, M., Sali, E.: Visual features of intermediate complexity and their use in classification. Nature Neuroscience 5, 682–687 (2002)
18. Riesenhuber, M., Poggio, T.: Models of object recognition. Nature Neuroscience 3, 1199–1204 (2000)
19. Serre, T., Wolf, L., Poggio, T.: Object recognition with features inspired by visual cortex. In: Proc. of Computer Vision and Pattern Recognition (CVPR), pp. 994–1000 (2005)
20. Mutch, J., Lowe, D.: Multiclass object recognition with sparse, localized features. In: IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR), vol. 1, pp. 11–18. IEEE Computer Society Press, Los Alamitos (2006)
21. Collobert, R., Bengio, S., Mariethoz, J.: Torch: a modular machine learning software library. Technical report, Dalle Molle Institute for Perceptual Artificial Intelligence, PO Box 592, Martigny, Valais, Switzerland (October 2002)
22. Ponce, J., Berg, T., Everingham, M., Forsyth, D., Hebert, M., Lazebnik, S., Marszalek, M., Schmid, C., Russell, B., Torralba, A., Williams, C., Zhang, J., Zisserman, A.: Dataset issues in object recognition. In: Ponce, J., Hebert, M., Schmid, C., Zisserman, A. (eds.) Toward Category-Level Object Recognition. LNCS, vol. 4170, Springer, Heidelberg (2006)
23. Viola, P., Jones, M., Snow, D.: Detecting pedestrians using patterns of motion and appearance. In: Proc. of the Ninth IEEE Int. Conf. on Computer Vision (ICCV), vol. 2, pp. 734–741. IEEE Computer Society Press, Los Alamitos (2003)
24. Wu, B., Nevatia, R.: Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In: Proc. of the 10th IEEE Int. Conf. on Computer Vision (ICCV), vol. 1, pp. 90–97. IEEE Computer Society, Washington, DC, USA (2005)
25. Zuo, F.: Embedded face recognition using cascaded structures. PhD thesis, Technische Universiteit Eindhoven, The Netherlands (October 2006)