Cascade of Classifiers for Vehicle Detection

Daniel Ponsa and Antonio López

Centre de Visió per Computador, Universitat Autònoma de Barcelona
Edifici O, 08193 Bellaterra, Barcelona, Spain
{daniel,antonio}@cvc.uab.es url:www.cvc.uab.es/ADAS

Abstract. Being aware of other vehicles on the road ahead is a key information to help driver assistance systems to increase driver's safety. This paper addresses this problem, proposing a system to detect vehicles from the images provided by a single camera mounted in a mobile platform. A classifier–based approach is presented, based on the evaluation of a cascade of classifiers (COC) at different scanned image regions. The Adaboost algorithm is used to determine the COC from training sets. Two proposals are done to reduce the computation needed for the detection scheme used: a lazy evaluation of the COC, and the customization of the COC by a wrapping process. The benefits of these two proposals are quantified in terms of the average number of image features required to classify an image region, achieving a reduction of the 58% on this concept, while scarcely penalizing the detection accuracy of the system.

1 Introduction

The research in Computer Vision applied to intelligent transportation systems is mainly devoted to provide them with situational awareness [1]. An essential task for demanded applications like Adaptive Cruise Control, or Autonomous Stop-&-Go Driving is determining the position of other vehicles on the road. This provides key information to Advanced Driver Assistance Systems (ADAS) in order to increase driver's safety. Traditionally this task has been addressed using active sensors like radar or lidar. However, since vision sensors (CCD/CMOS) are passive and cheaper, and provide a reacher description of the environment, many research efforts have also been devoted on applying computer vision techniques onto this topic [2]. The variety of vehicle appearances, due to the heterogeneity of this class of objects (many different types of cars, vans and trucks), and due to the uncontrolled acquisition conditions (different daytime and weather conditions, presence of strong shadows, artificial illumination, etc.), poses a big challenge for this detection task.

Our recent work [3] has focused on developing a system to detect vehicles, from a car equipped with a single monochrome camera, mounted close to the rear-view mirror facing the road ahead. The system follows the detection methodology proposed in [4] for face detection, based on scanning video-frames with a cascade of classifiers (COC). Evaluated image regions are sorted out between positive and negative categories (i.e., vehicle vs non-vehicle). The general procedure to construct the COC is the following. Given an initial training set, a classifier is

J. Blanc-Talon et al. (Eds.): ACIVS 2007, LNCS 4678, pp. 980–989, 2007.

learnt, selecting from training data a subset of features that distinguish efficiently both classes. This classifier conforms the first level of the COC. This cascade is then applied on a training sequence, where the generation of false positives (i.e., non–vehicle regions classified as vehicle) will be usually observed. These miss-classifications are collected to construct a new training set, which is then used to learn the next classification level of the COC. This process is iterated, improving the COC until an acceptable performance in the training sequence is reached. With this strategy, we developed a vehicle detector showing qualitatively good results.

This paper extends our previous work on vehicle detection, proposing two techniques to improve the efficiency of evaluating a COC at different regions in an image. First, a lazy evaluation of the classifiers in the COC is proposed, in order to minimize the amount of features computed to classify each inspected region. Given a testing set of video frames, the benefits of this lazy evaluation have been quantified by registering for each inspected region the amount of features needed to evaluate each COC classifier. This information has provided detailed insight into how detection takes place, and allows to identify the bottlenecks of this process. Using this information, a tuning of the used COC is proposed, replacing the *critical* classifiers in the COC with several new ones, which reduce significantly the average amount of features required to classify each inspected region. Experimental work done shows that the resultant COC has a detection accuracy practically identical to the one of the original COC.

The structure of the paper is as follows. Next section justifies why vehicle detection has been posed as a classification problem, and gives details on how the vehicle classifier is constructed. Then, the methodology to scan frames is presented, based on knowledge on the geometry of image formation. Section 3 proposes a lazy evaluation of the vehicle classifier, and presents the study done to determine how vehicle detection based on a COC takes place. Section 4 proposes *tuning* the COC in order to improve detection efficiency, and section 5 quantifies the accuracy of the final vehicle detector, and discusses the obtained results.

2 Vehicle Detector

Traditional approaches to detect vehicles are based on guessing in advance which are the best image features for detecting vehicles and only vehicles. Examples are works proposing the use of line structures and shadows [5], or symmetry measurements [6]. However, these features do not really account for all the different appearances that a vehicle may present, due to the effect of the uncontrolled illumination conditions, and the high variability of appearance of the different sorts of vehicles (figure 1). For this reason, it seems proper to determine the features used to detect vehicles in a learning process. In our work, the Real Adaboost algorithm [7] has been used to construct a vehicle classifier. Given a training set $\mathbf{T} = \{(\mathbf{H}_1, l_1), \ldots, (\mathbf{H}_{n_r}, l_{n_r})\}$, where $\mathbf{H}_i = \{f_i\}_1^N$ is an over-complete set of N Haar-like[1] features describing the i-th example, and $l_i \in \{v, nv\}$ a flag indicating

[1] The responses of filters proposed in [4], and their absolute value are considered.

if this example is a vehicle or not, the Adaboost algorithm selects a subset of $n_f \ll N$ features $\mathbf{F} = \{f_i\}_1^{n_f} \subset \mathbf{H}_i$, each one with an associated weak classifier r_i, that when combined correctly classify the training examples. The resultant classifier follows the expression

$$R(\mathbf{F}) = \sum_{i=1}^{n_f} r_i(f_i) \ , \tag{1}$$

where r_i is a decision stump on f_i that returns a positive (v_i^+) or negative (v_i^-) value according to its classification decision. That is,

$$r_i(f_i) = \begin{cases} v_i^+ \text{ if } f_i \leq (\text{or} \geq) \textit{ threshold} \ , \\ v_i^- \text{ otherwise} \ . \end{cases}$$

Given features $\mathbf{F}$ computed in an image region, $R(\mathbf{F})$ returns a value whose sign provides the final classification decision (positive for vehicles, negative for non–vehicles). Haar–like filters are used because of their reduced computational cost, which is independent of their evaluation scale. This is very relevant for vehicle detection, as vehicles are observed in frames in a wide range of scales (the proposed system considers regions from 24×18 up to 334×278 pixels), and Haar-like features does not demand and explicit size normalization of image regions. In order to achieve a desired detection performance, several classifiers are iteratively learnt and arranged in a cascade. Figure 2 sketches details on the Adaboost learning algorithm.

Once we have a COC, the next step is scanning with it images for detecting vehicles. An scanning process is proposed, derived from the assumption that the road where vehicles move (the one holding the camera, and the observed

Fig. 1. Top) Heterogeneity of the objects to be detected, just from their rear–view. Bottom) Pairs of the same vehicle, acquired under different illumination conditions.

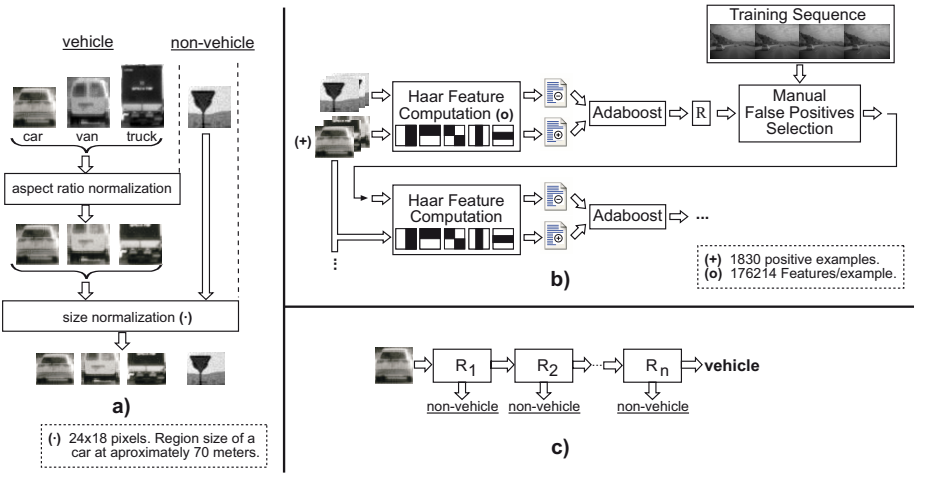

Fig. 2. a) Training set normalization. b) Process to construct the COC. c) Evaluation of the COC. True positives should be processed by all the COC layers.

ones) conforms to a flat surface. Knowing this plane and the geometry of image formation, the regions of the image where putative vehicles at different road locations project is determined, and then these regions are evaluated with the COC to verify the presence of a vehicle (figure 3). It is out of the scope of the paper describing how the road plane is estimated from images. Details can be found in [8]. In our acquisition system, estimating the road plane is equivalent to estimate the pose of the camera with respect to a world coordinate system placed on the road that sustains vehicles. In the experiments done in this paper, this information is obtained from ground truth data.

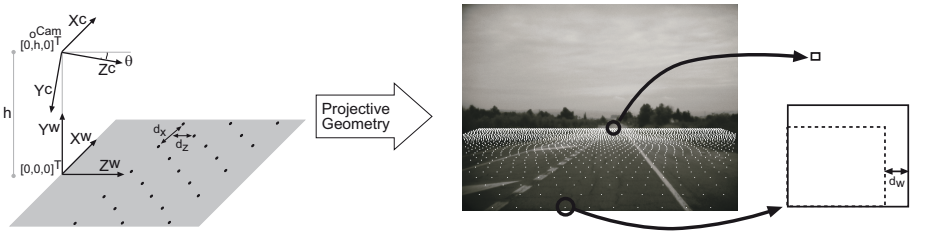

Fig. 3. Frame Scanning Process. For each inspected road point, rectangular regions of different widths and aspect ratios are evaluated.

The image regions to be scanned are determined from the projection onto image coordinates of a regular grid on the ground plane, inspecting the road ahead up to 70 meters away. For each projected grid point, several regions of different sizes are considered, to account for vehicles of different widths. Obviously, different grid points projecting onto the same image pixels are considered just once.

For a dense scanning of the road ahead (that is, $dx = 10$ cm and $dz = 10$ cm in figure 3), this means classifying between 350.000 and 500.000 image regions per frame, depending on the acquisition system parameters. This is a remarkably huge number of regions, if it is compared with the amount inspected in other application domains (for instance, in [9] a dense scanning to detect pedestrians consists on evaluating just 12.800 regions per frame). Thus, improving the efficiency in the COC evaluation is very important for the described application. Once a frame has been scanned, a list of image regions that may contain a vehicle is obtained. As a same vehicle is usually detected in several neighboring overlapping regions, a clustering algorithm fuses them in order to provide a single detection per vehicle.

3 Lazy COC Evaluation

Each level of the COC is constituted by a classifier R as defined in (1). Since classification is done in terms of the sign of the value returned by R, it is not always necessary to evaluate all their weak classifiers r_i to give the classification decision. The classification decision can be taken, as soon as the sum of the accumulated r_i responses has a magnitude bigger than the summation of responses of opposite sign in the remaining rules. More formally, given a classifier R and the set of features $\mathbf{F} = \{f_i\}_1^{n_f}$ of the region evaluated, the number of features n_{eff} required to establish a classification decision corresponds to the minimum n accomplishing

$$\sum_{i=1}^{n} r_i(f_i) > \sum_{j=n+1}^{n_f} v_j^{-sign\left(\sum_{i=1}^{n} r_i(f_i)\right)} \quad .$$

Thus, the number of features of $\mathbf{F}$ evaluated at each COC level depends on the image region content. It is proposed to take advantage on that to minimize the computation required by the described vehicle detection process. To quantify the significance of this lazy evaluation scheme, the presented vehicle detector has been applied on a set of testing frames, registering for each region processed the details of the COC evaluation, namely:

- the number of COC layers evaluated to give a classification decision [2],
- the number of features evaluated at each COC layer.

Figure 4 displays the statistics of the obtained results, showing the percentage of processed regions that receive a final classification decision at each COC layer, and for each layer, the percentage of regions that require the evaluation of a given number of features. For each layer, the expectation of the number of features evaluated vs the number of features n_f of the classifier is presented. Results show that, on average, the standard evaluation of the COC requires computing 102.82 features per region, while the lazy evaluation requires just 76.06. This means a reduction of the 26% in the number of features computed.

[2] Note that only positive regions are expected to be evaluated in all COC layers.

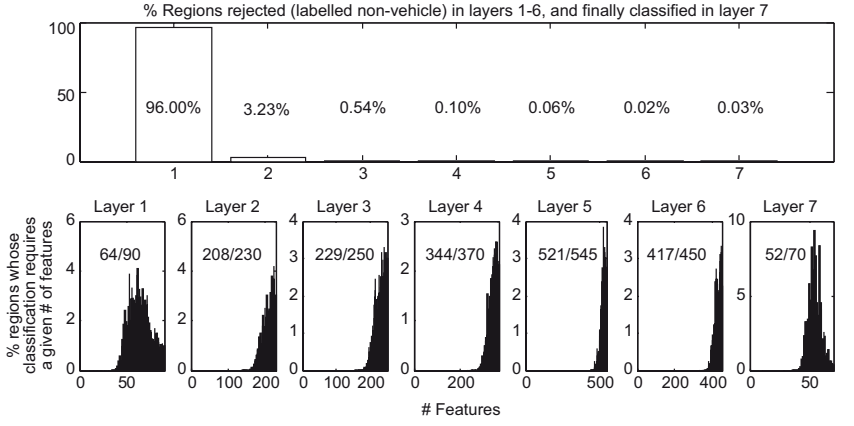

Fig. 4. Statistics of the lazy evaluation of a COC. For each layer, the average number of features evaluated vs. the total number of features of its classifier is shown.

Results also show that on average, the 96% of regions are discarded (i.e. classified as non–vehicles) at the first COC layer. This comes from the fact that processed images present a large homogeneous area (the road), and the image regions evaluated there are easy to distinguish from vehicles. However, although most image regions just require the evaluation of a single COC layer, they require on average evaluating 64.34 features, which results in a noteworthy amount of computation, due to the big amount of image regions that are inspected. In order to obtain a more efficient vehicle detector, less features should be used to discard this greater part of analyzed regions. Next section proposes a methodology to tune the learned COC in order to achieve that.

4 Tuning a COC

In order to implement with lower computational cost the task of a given level of a COC, it is proposed to substitute its corresponding classifier R by another COC. Ideally, this COC should achieve an equivalent classification performance, requiring the analysis of a fewer amount of features when a frame is processed. The method proposed is based on a partition of the training set $\mathbf{T}$ used to generate R, in order to obtain new classifiers of lower complexity. Let's denote $\mathbf{T}\oplus$ and $\mathbf{T}\ominus$ the positive and negative examples in $\mathbf{T}$ (i.e. $\mathbf{T} = \mathbf{T}\oplus \cup \mathbf{T}\ominus$). Using the classifier R learned from $\mathbf{T}$, elements in $\mathbf{T}\ominus$ are classified, selecting then the ones whose classification remain negative during the evaluation of the last 90% of the weak rules r_i in (1). This selection groups negative examples according to the similarity of how they are classified (that is, that from the evaluation of the first 10% of weak classifiers in R, they are always considered as negative). This partitions $\mathbf{T}\ominus$ in two groups:

- one with elements easily distinguishable from positive examples ($\mathbf{T}\ominus_1$);
- the other with elements more difficult to classify ($\mathbf{T}\ominus_2$).

Heuristicly it is guessed that from these two sets new classifiers will be learned that jointly require a lower complexity than R. From the set $\{\mathbf{T} \oplus \cup\mathbf{T}\ominus_1\}$, as contains *clearly* negative examples, it seems logical to expect classifying them with less features. For $\{\mathbf{T} \oplus \cup\mathbf{T}\ominus_2\}$ it is also possible to obtain a classifier of lower complexity, as the Adaboost will select a different subset of features $\mathbf{F}_2$ specially tuned to distinguish just the elements in $\mathbf{T}\ominus_2$ [3]. Thus, in this paper we propose to recursively apply such a divide and conquer strategy, attempting to obtain classifiers of a desired complexity. This procedure can be seen as a wrapper method devoted to iteratively select negative examples that simplify (in terms of the number n_f in R) the learned classifier. Figure 5 sketches the specific proposed strategy. The subset $\mathbf{T}\ominus_\mathbf{1}$ is recursively purged using the described method, until either a classifier with a constrained maximum complexity is obtained, or the complexity of the classifier obtained does not decrease. Then, the examples discarded during this process are grouped in a new training set $\mathbf{T}\ominus'$, and the process is started again. The process is stopped when no significant improvement is achieved.

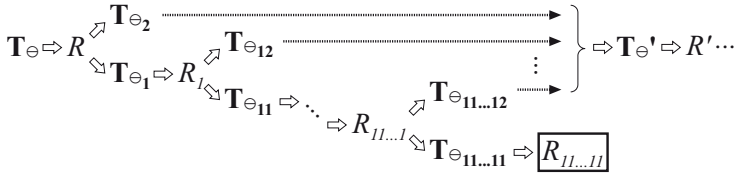

Fig. 5. Strategy used to substitute a classifier R by a COC

Using this strategy, the first level of the cascade analyzed in figure 4 has been replaced by 4 new sub-levels which, when applied on testing frames, display the statistics of figure 6.

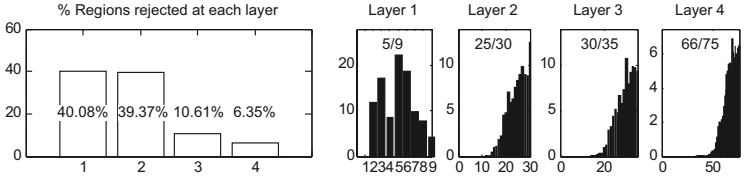

Fig. 6. Statistics of the COC levels that replace the first layer of the COC in figure 4

The joint performance of these new 4 layers is compared in figure 7 with the performance of the replaced layer. Now the 96% of analyzed regions require on average the evaluation of 33 features, when the original COC required 64 features. Considering the overall COC performance, the average number of features required per inspected region is now 43.35, which with respect to the 76.06 of the original COC, it means a reduction of the 43%.

[3] If this does not happen, one can just use the original R for classifying $\mathbf{T}\ominus_2$.

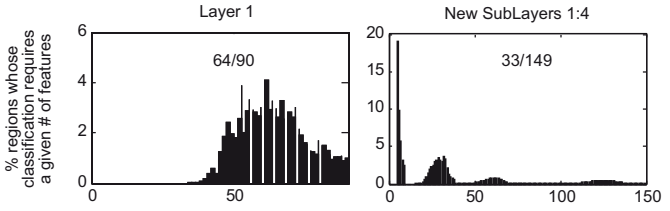

Fig. 7. Performance of the initial COC layer vs. the new learned sub-layers

5 Classification Rate Evaluation

To objectively evaluate the performance of the proposed method, the following experiment has been carried out. First, sequences different to the ones used for training has been used, which where acquired using three different vehicles with different video cameras. Each camera has premounted optics, and has been roughly calibrated assuming a pin–hole camera model with zero–skew. The images provided by each camera are significantly different, due to their different behavior with respect to the automatic control of the camera gain, and their spectral sensitivity. Sequences has been acquired in different times of the day (midday and sunset) and environmental conditions (cloudy day and sunny day, etc.). From them, 500 frames has been selected in order to construct a testing set to validate the system. The selection criterion has been collecting frames significant with respect to the different kinds of vehicles acquired and to the lighting conditions (presence of shadows, specularities, under-illuminated environments, etc.). All selected frames accomplish the restriction that a user can easily annotate a planar surface approximating the observed road. This annotation is easy if parallel road structures (lane markings, road limits, etc.) are clearly observed in the image. The annotated plane provides the ground truth information used to determine the frame regions that are inspected. With this information, an ideal scanning of video frames is carried out, and the best performance achievable for the proposed method can be quantified. The vehicles in testing frames have also been manually annotated, being labeled depending of if their detection should be mandatory, or if they can be miss-detected due to some of the following causes:

- present partial occlusions;
- are farther than the maximum operative detection distance (70 meters);
- lay in a plane different than the one used for scanning the image.

The labeling of observed vehicles in these two disjoint classes is done to better quantify the detection performance (i.e., count properly the number of false positives and false negatives). The miss-detection of a *miss-detectable* vehicle do not have to be interpreted as a false negative, as the objective in this papers is not evaluating the detection performance in this challenging cases. On the other hand, *miss-detectable* vehicles, being detected or not, are counted neither as true nor false positives, in order to do not distort results. Thus, classification ratios are computed taking into consideration just vehicles that should be

detected obligatorily. Table 1 shows the results obtained for a dense scanning of testing frames, using the original and the tuned COC respectively. Using the tuned COC, a slightly lower detection rate is achieved (93.91% versus the 94.13% of the original COC), but also a lower false positive rate per region evaluated. The detection accuracy achieved is remarkable, due to the complexity of the faced problem (detection of vehicles up to 70 meters away), and the challenging conditions considered in the testing (different acquisition cameras, daytime conditions, frontal and rear vehicle views, etc.).

Table 1. Detection results of the original (top) and tuned (bottom) COC

Original COC - True Positives Detection rates									
	Car		Van		Truck			Acum.	
Rear	547/570	95.96%	163/169	96.45%	67/78	85.90%	⇒	777/817	95.10%
Front	67/80	83.75%	11/12	91.67%	11/11	100.00%	⇒	89/103	86.41%
	⇓		⇓		⇓			⇓	
Acum.	614/650	94.46%	174/181	96.13%	78/89	87.64%	⇒	866/920	94.13%

Original COC - False Positives Detection rates	
FP per Window evaluated: 1.509e-004	FP per Frame: 1.07

Tuned COC - True Positives Detection rates									
	Car		Van		Truck			Acum.	
Rear	545/570	95.61%	162/169	95.86%	68/78	87.18%	⇒	775/817	94.86%
Front	67/80	83.75%	11/12	91.67%	11/11	100.00%	⇒	89/103	86.41%
	⇓		⇓		⇓			⇓	
Acum.	612/650	94.15%	173/181	95.58%	79/89	88.76%	⇒	864/920	93.91%

Tuned COC - False Positives Detection rates	
FP per Window evaluated: 1.426e-004	FP per Frame: 1.02

The detector has a better performance in detecting the back of vehicles, probably due to the fact that frontal views are underrepresented in the training set (they constitute less than the 10% of positive training examples). Concerning the type of vehicles, the ones more difficult to detect are trucks. We guess that this is due to two factors. On one hand, trucks conform a class more heterogeneous than other types of vehicles. On the other hand, the appearance of their back side usually vary very significantly depending on the camera viewpoint. This does not happen with the other type of vehicles, where their backside commonly conforms approximately a vertical plane, and for this reason their appearance scarcely varies with the camera viewpoint. Another point worth to mention is the number of false positives. On average 1.02 false positives per frame are generated, but this does not mean that when a real sequence is processed, a false alarm is generated at every frame. In real sequences it can be seen that false positives do not present spatio–temporal coherence, while true vehicles do. Using this fact, it is easy to differ false from true detections with the help of tracking.

6 Conclusions

A system has been presented to detect vehicles from images acquired from a mobile platform. Based on the Adaboost algorithm, a COC has been learnt from training data. Two proposals have been presented to reduce computational cost in the detection process, namely the lazy evaluation of classifiers, and a wrapping process to tune the initial learned COC. Thanks to these two proposals, the average number of features computed per inspected region has reduced from the 102.82 of the original COC with standard evaluation, to the 43.35 of the tuned COC with lazy evaluation (a reduction of around the 58%). The detection accuracy of the tuned COC is scarcely inferior to the one of original COC, showing also an inferior false detection rate.

Acknowledgments. This research has been partially funded by Spanish MEC project TRA2004-06702/AUT.

References

1. Dickmanns, E.: The development of machine vision for road vehicles in the last decade. In: Int. Symp. on Intelligent Vehicles, Versailles, vol. 1, pp. 268–281 (2002)
2. Sun, Z., Bebis, G., Miller, R.: On-road vehicle detection: A review. IEEE Trans. on Pattern Analysis and Machine Intelligence 28, 694–711 (2006)
3. Ponsa, D., López, A., Serrat, J., Lumbreras, F., Graf, T.: 3d vehicle sensor based on monocular vision. In: Int.Conf. Intel. Transportation Systems, pp. 1096–1101 (2005)
4. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: IEEE Conf.Computer Vision and Pattern Recognition, pp. 511–518. IEEE Computer Society Press, Los Alamitos (2001)
5. Maurer, M., Behringer, R., Fürst, S., Thomanek, F., Dickmanns, E.D.: A compact vision system for road vehicle guidance. In: 13th Int. Conference on Pattern Recognition, Vienna, Austria, vol. 3, pp. 313–317 (1996)
6. Broggi, A., Cerri, P., Antonello, P.: Multi-resolution vehicle detection using artificial vision. In: IEEE Intelligent Vehicles Synposium, pp. 310–314. IEEE Computer Society Press, Los Alamitos (2004)
7. Schapire, R.E., Singer, Y.: Improved boosting using confidence-rated predictions. Machine Learning 37, 297–336 (1999)
8. Sappa, A., Gerónimo, D., Dornaika, F., López, A.: On-board camera extrinsic parameter estimation. IEE Electronics Letters 42, 645–747 (2006)
9. Zhu, Q., Avidan, S., Yeh, M.C., Cheng, K.T.: Fast human detection using a cascade of histograms of oriented gradients. In: IEEE Computer Society Conference on Computer vision and Pattern Recognition, vol. 2, pp. 1491–1498. IEEE, Los Alamitos (2006)