Benchmarking Haar and Histograms of Oriented Gradients features applied to vehicle detection

No Author Given

No Institute Given

Abstract. This paper provides a comparison between two of the most used visual descriptors (image features) nowadays in the field of object detection. The investigated image features involved the Haar filters and the Histogram of Oriented Gradients (HoG) applied for the on road vehicle detection. Tests are very encouraging with a average detection of 97% on realistic on-road vehicle images.

Keywords: intelligent vehicle, vehicle detection, AdaBoost, Haar filter, Histogram of oriented gradient

1 Introduction

On road vehicle detection is an essential part of the Intelligent Vehicles Systems and has many applications including platooning (i.e. vehicles travelling in high speed and close distances in highways), Stop&Go (similar that precedent situation, but at low speeds), and autonomous driving.

Most of the detecting methods distinguish two basic steps: Hypothesis Generation (HG) and Hypothesis Verification (HV) [20]. HG approaches are simple low level algorithm used to locate potential vehicle locations and can be classified in three categories: knowledge-based (symmetry [2], colour [23, 14], shadows [21], edges [8], corners [4], texture [5], etc), stereo-based (disparity map [10], inverse perspective mapping [3], etc) , and motion-based [9]. HV approaches performs the validation of the Regions of Interest generated by the HG step. They can be classified in two categories: template-based and appearance-based. Template-based methods perform a correlation between a predefined pattern of the vehicle class and the input image: horizontal and vertical edges [18], regions, deformable patterns [7] and rigid patterns [24]. Appearance-based methods learn the characteristics of the classes vehicle and non-vehicle from a set of training images. Each training image is represented by a set of local or global descriptors (features) [1]. Then, classification algorithms can estimate the decision boundary between the two classes.

One of the drawbacks of optical sensors are the considerable time processing and the average robustness. In that way, Viola & Jones [22] developed simple an appearance-based system obtaining amazing results in real time. Their appearance based method uses Haar-based representation, combined with an AdaBoost [11] algorithm. They also introduce the concept of a cascade of classifiers which reach high detection results while reducing computation time.

The present article compares the Haar-based features with the Histograms of Oriented Gradient (HoG) based features using the same cascade architecture.

The next section describes briefly the Haar and the HoG features. Section two introduces the learning classification algorithms based on AdaBoost. We finish the article with the results and conclusions.

2 Features

The common reasons why features are choosen instead of pixels values are that features can code high level object information (segments, texture, ...) while intensity pixel values based system operates slower than a feature based system. This section describes the features used to train the Adaboost cascade.

2.1 Haar filters

Each wavelet coefficient describes the relationship between the average intensities of two neigh-boring regions. Papageorgiou *et al.* [16] employed an over-complete set of 2D wavelets to detect vehicles in static images.

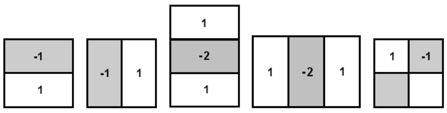

Fig. 1. 2D Wavelet set.

Figure 1 shows basic Haar filters: *two*, *three* and *four* rectangle features, where the sum of the pixels which lie within the white rectangles are subtracted from the sum of pixels in the grey rectangles. We conserve the *two* and *three* rectangle features since vehicles have rectangular shape: diagonal features (*four* rectangle template) doesn't give extra information for this type of pattern.

Viola & Jones [22] have introduced the Integral Image, an intermediate representation for the input image. The sum of the rectangular region in the image can be calculated in four Integral Image references. Then, the difference between two adjacent rectangles, can be computed with only six references, eight in the case of the three rectangle feature.

The Haar feature set is composed of the resulting value of the rectangular filters at various scales in a image.

In figure 2 we can see the results of two rectangular filters (vertical and horizontal) at two scales: 2x2 and 4x4 pixels. Lightness pixels mean important subtraction values (the result is always calculated in modulus). The complete set of Haar's features utilizing the three rectangular filters (see fig. 1) in a 32x32 pixel image at {2,4,8,16} scales is 11378. Every single feature j in the set could be defined as: $f_j = (x_j, y_j, s_j, r_j)$, where r_j is the rectangular filter type, s_j the scale and (x_j, y_j) are the position over the 32x32 image.

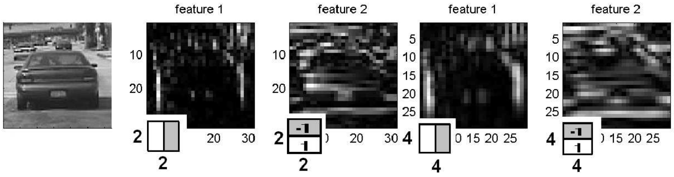

Fig. 2. 2D Haar Wavelet example on a vehicle image.

2.2 Histogram of Oriented Gradient

The Histograms of Oriented Gradient (HoG) is another way to encode an input image to obtain a vector of visual descriptors. This local descriptor, based on Scale Invariant Feature Transform (SIFT) [15], uses the gradient magnitude and orientation around a keypoint location to construct an histogram. Orientations are quantized by the number of bins in the histogram (four orientations are sufficient). For each histogram bin, we compute the sum in the region of all the magnitudes having that particular orientation. The histogram values are then normalised by the total energy of all orientations to obtain values between 0 and 1.

Gepperth [13] train a neural network classifier using these features for a two class problem: vehicle, non-vehicle. First, a ROI is subdivided into a fixed number of regions called *receptive fields*. From each *receptive field*, they obtain an oriented histogram feature.

The HoG features set is composed of histograms calculated inside a rectangular region on the original image. We evaluate the the gradient of the image using the Sobel filters to obtain the gradient magnitude and orientation.

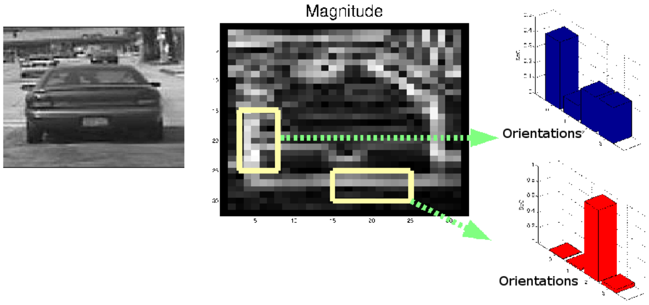

Fig. 3. HoG example on a vehicle image.

There are three types of rectangle regions: r_1 square l*l, r_2 vertical rectangle l*$2l$, r_3 horizontal rectangle $2l$*l. Considering $l : \{2, 4, 8, 16\}$ scales, we have a total of 4678 features. A single histogram j in the set could be defined as: $h_j = (x_j, y_j, s_j, r_j)$, where r_j is the rectangular filter type, s_j the scale and (x_j, y_j) are the position over the 32x32 image.

3 AdaBoost

As we saw in previous sections, Haar and HoG representations are used to obtain a vector of visual descriptors describing an image. The size of this vectors is clearly bigger than the number of pixel in the image. Using the total number of features to carry out a classification is inadequate from the computing time point of view of the and the robustness, since many of these features do not contain important information (noise). Different methods: statistics [17], PCA, genetic algorithms [19], etc. can be used to select a limited quantity of representative features.

Among these methods, the Boosting [11] classification method improves the performance of any algorithm. It finds precise hypothesis by combining several *weak classifiers* which, on average, have a moderate precision. The *weak classifiers* are then combined to create a *strong classifier*:

$$G = \begin{cases} 1 \sum_{n=1}^{N} \alpha_n g_n \geq \frac{1}{2} \sum_{n=1}^{N} \alpha_n = T \\ 0 \text{ otherwise} \end{cases} \tag{1}$$

Where G and g are the strong and weak classifiers respectively, and α is a coefficient wheighting each feature result. T is the strong classifier threshold.

Different variants of boosting are known such as Discrete Adaboost [22], Real AdaBoost [12], Gentle AdaBoost, etc. The procedures (pseudo-code) of any of this variants are widely developed in the literature.

We need, however, to study the construction of the weak classifier for both cases: Haar and HoG features.

3.1 Haar Weak classifier

We define the weak classifier as a binary function g:

$$g \begin{cases} 1 \text{ if } p_j f_j < p_j \theta_j \\ 0 \text{ otherwise} \end{cases} \tag{2}$$

where f_j is the feature value, θ_j the feature threshold and p_j the threshold parity.

3.2 HoG Weak classifier

This time, instead of evaluate a feature value, we estimate the distance between an histogram h_j of the input image and a model histogram m_j. The model is calculated like the mean histogram between all the training positive examples. For each histogram h_j of the feature set, we have the corresponding m_j. A vehicle model is then constructed and AdaBoost will found the most representative m_j which best separate the vehicle class from the non-vehicle class.

We define the weak classifier like a function g:

$$g \begin{cases} 1 \text{ if } d(h_j, m_j) > \theta_j \\ 0 \text{ otherwise} \end{cases} \tag{3}$$

where $d(h_j(x), m_j)$ is the Bhattacharyya distance [6] between the feature h_j and m_j and θ_j is the distance the feature threshold.

4 Test and Results

4.1 Dataset

The images used in our experiments were collected in France using a prototype vehicle. To ensure data variety, 557 images where captured during different time, and on different highways.

The training set contains 745 vehicle sub-images of typical cars, sport-utility vehicles (SUV) and minivan types. We duplicate this quantity flipping the sub-images around y-axis, obtaining 1490 examples. We split this new set keeping 1000 of the examples for training and the others for validation: the training set (TS) contains 1000 sub-images aligned to a resolution of 32 by 32 pixels, the validation set (VS) contains 490 vehicle sub-images with the same resolution. The negative examples come from 3196 images without vehicles.

The test set contains 200 vehicles in 81 images.

4.2 Single stage detector

First experiments were carried out with a strong classifier constructed with 100, 150 and 200 Haar or HoG features using the Discrete Adaboost algorithm [22].

We used the TS for the positive examples. The non-vehicle (negatives) examples were collected by selecting randomly 5000 sub-windows from a set of 250 non-vehicle images at different scales.

To evaluate the performance of the classifiers, the average detection rate (DR) and the number of false positives (FP) were recorded using a three-fold cross validation procedure. Specifically, we obtain three sets of non-vehicle sub-windows to train three strong classifiers. Then, we test these classifiers on the test set.

4.3 Multi stage detector

This section shows the test realised using a cascade of strong classifiers [22]. The multi stage detector increases detection accuracy and reduces the computation time. Simpler classifiers (having a reduced number of features) reject the majority of the false positives before more complex classifiers (having more features) are used to reject difficult sub-windows.

Stages in the cascade are constructed with the Adaboost algorithm, training a strong classifier which achieves a minimum detection rate ($d_{min} = 0.995$) and a maximum false positive rate ($f_{max} = 0.40$). The training set is composed of the TS positive examples and the non-vehicle images separated in 12 different folders (the maximum number of stages). Subsequent classifiers are trained using those non-vehicle images of the corresponding folder which pass through all the previous stages.

An overall false positive rate is defined to stop the cascade training process (F = $43 * 10^{-7}$) within the maximum number of stages.

This time, the average accuracy (AA) and false positives (FP) where calculated using a five-fold cross validation procedure. We obtain five detectors from five differents TS and VS randomly obtained.

4.4 Results

Classifier	DR (%)	FP	Time
HoG - 100 fts	69.0	1289	3,52
HoG - 150 fts	72.5	1218	4,20
HoG - 200 fts	83.1	1228	5,02
Haar - 100 fts	96.5	1443	2,61
Haar - 150 fts	95.7	1278	3,93
Haar - 200 fts	95.8	1062	5,25

Table 1. Single stage detection rates (Haar and HoG classifiers)

Table 1 shows the detection rate of the single stage detector trained either on Haar features or on HoG features with respectively 100, 150 and 200 features. These results are very interesting though quite predictible. As seen before, HoG classifiers computes a distance from the test sample to a "vehicle model" (the mean histograms). These are generating classifiers. When the number of considered features increases, the model is refined and the detection rate increases while the number of false positives keeps stable. On the other hand, Haar classifiers are discriminative classifier evaluating a fronteer between positive and negative samples. Now, the fronteer is refined - and the number of false positives decreases - when the number of features increases. Figure 4 presents the ROC curves for each detector. As told before, for a given detection rate, the number of false positives is lower for Haar classifiers than for HoG classifiers.

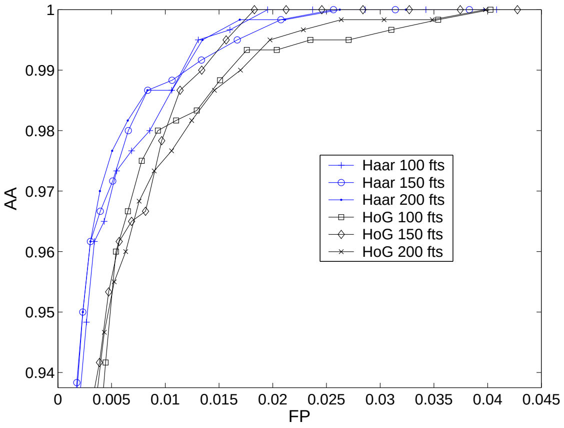

Fig. 4. ROC curves for Haar and HoG detectors.

Table 2 shows results of cascade detectors using Haar and HoG based features. We also tested the effect of increasing the size of the negative set in each training stage.

Classifier	Stages	# Features	Negatives	DR (%)	FP	t (seg)
HoG	9	87	2000	96.8	1107	0.57
HoG	11	152	3000	97,5	1032	0.71
Haar	12	332	2000	94	447	0.83
Haar	12	386	3000	94,5	387	0.80

Table 2. Multi stage detection rate (Haar and HoG classifiers)

The behavior of each detector is the same as described before. HoG detector try to construct a finer vehicle model to take into account the new negatives. The number of features used increases twice and as the model refines, the detection increases slowly. But the number of false positives does not changes significantly. Haar detector refines the fronteer using somemore features and the number of false positives decreases while the detection keeps quite stable.

5 Conclusions

This communication deals with a benchmark comparing Haar-like features and Histograms of Oriented Gradients features applied to vehicle detection. These features are used in a classification algorithm based on Adaboost. Two strategies are implemented: a single stage detector and a multi-stage detector. The tests - applied on realistic on-road images - show two different results: for the HoG (generative) features, when the number of considered features increases, the detection rate increases while the number of false positives keeps stable; for the Haar-like (discriminative) features, the number of false positives decreases. Future works will be oriented to combined these behaviors. An approach could be build using simultaneously both feature types. We should also select relevant features.

References

1. S. Agarwal and D. Roth. Learning a sparse representation for object detection. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part IV*, pages 113–130, London, UK, 2002. Springer-Verlag.
2. A. Bensrhair, M. Bertozzi, A. Broggi, P. Miche, S. Mousset, and G. Toulminet. A cooperative approach to vision-based vehicle detection. In *Proceedings on Intelligent Transportation Systems*, pages 207–212, 25-29 August 2001.
3. M. Bertozzi and A. Broggi. Vision-based vehicle guidance. *Computer*, 30(7):49–55, 1997.
4. M. Bertozzi, A. Broggi, and S. Castelluccio. A real-time oriented system for vehicle detection. *Journal of Systems Architecture*, 43(1-5):317–325, 1997.
5. T. Bucher, C. Curio, J. Edelbrunner, C. Igel, D. Kastrup, I. Leefken, G. Lorenz, A. Steinhage, and W von Seelen. Image processing and behavior planning for intelligent vehicles. *IEEE Transactions on Industrial Electronics*, 50(1):62–75, Febrary 2003.
6. S.H. Cha and S. N. Srihari. On measuring the distance between histograms. *Pattern Recognition*, 35(6):1355–1370, 2002.
7. J.M. Collado, C. Hilario, A. de la Escalera, and J.M. Armingol. Model based vehicle detection for intelligent vehicles. In *International Symposium on Intelligent Vehicles*, pages 572–577, 2004.

8. F. Dellaert. Canss: A candidate selection and search algorithm to initialize car tracking. Technical report, Robotics Institute, Carnegie Mellon University, 1997.
9. C. Demonceaux, A. Potelle, and D. Kachi-Akkouche. Obstacle detection in a road scene based on motion analysis. *IEEE Transactions on Vehicular Technology*, 53(6):1649 – 1656, November 2004.
10. U. Franke. Real-time stereo vision for urban traffic scene understanding. In *Proceedings IEEE Intelligent Vehicles Symposium 2000*, pages 273–278, Detroit, USA, October 2000.
11. Y. Freund and R. Schapire. Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, pages 148–156, 1996.
12. J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *The Annals of Statistics*, 28(2):337–374, April 2000.
13. A. Gepperth, J. Edelbrunner, and T Bocher. Real-time detection and classification of cars in video sequences. In *Intelligent Vehicles Symposium*, pages 625–631, 2005.
14. D. Guo, T. Fraichard, M. Xie, and C. Laugier. Color modeling by spherical influence field in sensing drivingenvironment. In IEEE, editor, *Intelligent Vehicles Symposium*, pages 249–254, Dearborn, MI, USA, 2000.
15. D.G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision*, pages 1150–1157, 1999.
16. C.P. Papageorgiou and T. Poggio. A trainable object detection system: Car detection in static images. Technical report, MIT AI Memo 1673 (CBCL Memo 180), 1999.
17. H. Schneiderman and T. Kanade. A statistical method for 3d object detection applied to faces and cars. In *ICCVPR*, pages 746–751, 2000.
18. N. Srinivasa. Vision-based vehicle detection and tracking method for forward collision warning in automobiles. In *IEEE Intelligent Vehicle Symposium*, volume 2, pages 626–631, June 2002.
19. Z. Sun, G. Bebis, and R. Miller. Object detection using feature subset selection. *Pattern Recognition*, 37(11):2165–2176, 2004.
20. Z. Sun, G. Bebis, and R. Miller. On-road vehicle detection: A review. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(5):694–711, 2006.
21. M.B. van Leeuwen and F.C.A. Groen. Vehicle detection with a mobile camera. Technical report, Computer Science Institute, University of amsterdam, The Netherlands, October 2001.
22. P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Conference on Computer Vision and Pattern Recognition*, pages 511–518, 2001.
23. T. Xiong and C. Debrunner. Stochastic car tracking with line- and color-based features. *Advances and Trends in Research and Development of Intelligent Transportation Systems: An Introduction to the Special Issue*, 5(4):324–328, December 2004.
24. H. Yang, J. Lou, H. Sun, W. Hu, and T. Tan. Efficient and robust vehicle localization. In *International Conference on Image Processing*, volume 2, pages 355–358, Thessaloniki, Greece, 7-10 October 2001.