Real-time visual detection of vehicles and pedestrians with new efficient adaBoost features

Fabien Moutarde, Bogdan Stanciulescu and Amaury Breheret

***Abstract*— This paper deals with real-time visual detection, by mono-camera, of objects categories such as cars and pedestrians. We report on improvements that can be obtained for this task, in complex applications such as advanced driving assistance systems, by using new visual features as adaBoost weak classifiers. These new features, the "connected control-points" have recently been shown to give very good results on real-time visual rear car detection. We here report on results obtained by applying these new features to a public lateral car images dataset, and a public pedestrian images database. We show that our new features consistently outperform previously published results on these databases, while still operating fast enough for real-time pedestrians and vehicles detection.**

I. Introduction and Related Work

Autonomous vehicles, as well as most Advanced Driving Assistance System (ADAS) functions, require real-time perception analysis. This environment perception can be done using various sensors such as lidars, radars, ultrasonic devices, etc… However, compared to other sensors, visual perception can provide very rich information for very low equipment costs, if an abstract enough scene analysis can be conducted in real-time.

One of the key bricks required for such an automated scene analysis is efficient visual detection of most common moving objects in car environment: vehicles and pedestrians. Many techniques have been proposed for visual object detection and classification (see eg [10] for a review of some of the state-of-the-art methods for pedestrian detection, which is the most challenging). Of the various machine-learning approaches applied to this problem, only few are able to process videos in real-time. Among those last ones, the boosting algorithm with feature selection was successfully extended to machine-vision by Viola & Jones [4][5]. The adaBoost algorithm was introduced in 1995 by Y. Freund and R. Shapire [1][2], and its principle is to build a *strong classifier*, assembling weighted weak classifiers, those being obtained iteratively by using successive weighting of the examples in the training set.

Most published works using adaBoost for visual object class detection are using the Haar-like features initially proposed by Viola & Jones for face and pedestrian detection.

Manuscript received June 10, 2008.

F. Moutarde, B. Stanciulescu and A. Breheret are all with the Robotics Laboratory of Mines ParisTech, 60 Bd St Michel, 75006 Paris, FRANCE (33-1-40.51.92.92, {Fabien.Moutarde,Bogdan.Stanciulescu}@ensmp.fr).

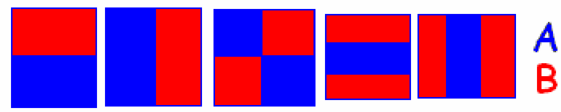

Fig.1: Viola &Jones Haar-like features

These weak classifiers compute the absolute difference between the sum of pixel values in red and blue areas (see figure 1), with the respect of the following rule:

if $\left|Area(A) - Area(B)\right| > Threshold$ **then True else False**

Fig. 2: Some examples of adaBoost-selected Viola-Jones features for car detection (top) and pedestrian detection (bottom).

However, the adaBoost outcome may strongly depend on the family of features from which the weak classifiers are drawn. But rather few investigations have been done on using other kinds of features with adaBoost: Zhu et al. in [13] defined and successfully applied adaBoost features directly inspired from the Histogram of Oriented Gradient (HOG) approach initially proposed (combined with SVM) by Dalal [12]; Baluja et al. in [14] and Leyrit et al. in [15] both use pixel-comparison-based feature very similar, although simplified, to our lab's control-points approach ([6][7][8][9]); very recently Pettersson et al. in [16] proposed efficient gradient-histogram-based features inspired from HOG.

II. CONTROL-POINTS ADABOOST FEATURES

Several years ago, Abramson & Steux [6][7] proposed an original set of features, the control-points, for faster and more illumination-independant adaBoost classifiers.

These features operate directly at pixel level (at one among 3 different possible resolutions) and are illumination-independent. Each of these features can be computed by only a few pixel comparisons, which makes them extremely fast, thus providing very good real-time performances for the resulting detector. Arbitrary points are divided in two groups, one called the positive set and the second called the negative set. Examples are classified as positive, if the following condition applies:

$$\min(\{P_i^+, i=1,\ldots,N_+\}) - \max(\{P_j^-, j=1,\ldots,N_-\}) > V$$

OR

$$\min(\{P_j^-, j=1,\ldots,N_-\}) - \max(\{P_i^+, i=1,\ldots,N_+\}) > V$$

V is the minimum separation threshold between the two point groups, P_i^+ a point from the positive group, P_j^- a point from the negative group, and N_+ and N_- the number of points in the respective groups.

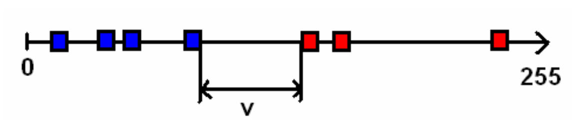

Fig. 3a: Positive-classified example with respect to the threshold V.

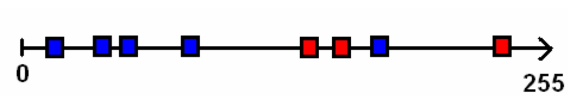

Fig. 3b: Negative-classified example.

In a linear representation of the pixel values, an example is classified as positive if the two point groups are separated by at least the value of threshold V (see figure 3a). Negative examples are those that do not respect this characteristic: values of the control-points of the two groups are interleaved (see figure 3b).

One can see on the figure 4 some examples of control-points features acting on vehicle or pedestrian detection. Each feature operates at either full-, half- or quarter-resolution of the minimal detection window size (80x32 for the lateral car case, and 18x36 for the pedestrian case). An examined image or sub-window is thus resized to those 3 resolutions before the features are applied.

On the upper-left example of figure 4, the feature will respond positively if the 2 pixels values (on the correctly resized image) corresponding to the 2 white squares *all* have higher luminance (with margin ≥ V) than *all* 3 pixels values corresponding to the 3 red squares (or opposite). This particular feature can therefore be interpreted as detecting some usual contrast between the car itself and region just below, with shadow and dark tyres. Similarly, the lower-left feature seems to detect some contrast between pedestrian center and the background. Such interpretation of selected control-points features is not always very clear, however.

AdaBoost requires a ”weak learner”, i.e. an algorithm which will select and provide, for each adaBoost step, a ”good” feature (i.e. with a ”low-enough” weighted error measured on the training set). The weak learner used by Viola and Jones is just an exhaustive search of all ~180,000 possible features in their set of features. But as our control-point family features is absolutely huge (there are more than 10^{35} of them for a 36 × 36 detection window size), a systematic full search is definitely not possible. We therefore use as weak learner a genetic-like heuristic search in feature space: an evolutionary hill-climbing described in more details in [8].

The core of our heuristic search weak-learner is to define specific mutations adapted to the feature-type, and apply them to a population of initially random features. A single mutation of one control-points feature typically consists in adding, moving, or removing one of the points, changing working resolution, or modifying the value of threshold V. When evolution provides no more improvement, the best feature of the population is selected and the weak-learner returns it to be added as the next adaBoost feature.

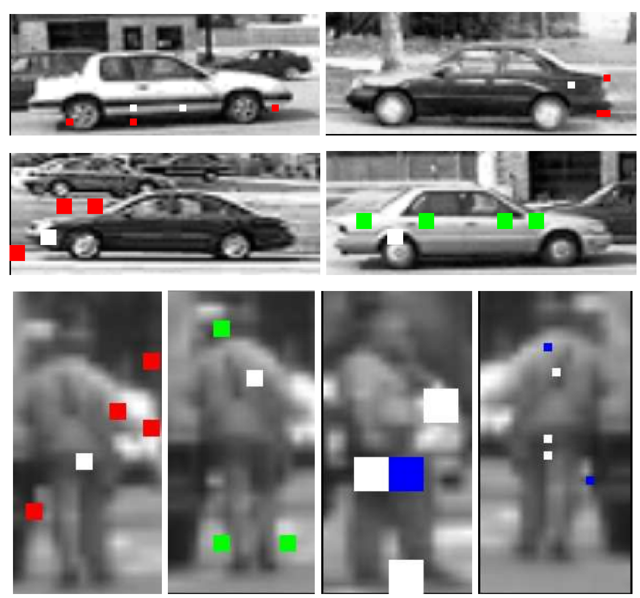

Fig. 4: Some examples of adaBoost-selected Control-Points features for car detection (top) and pedestrian detection (bottom line). Some features operate at full resolution of detection window (eg rightmost bottom), while others work on half-resolution (eg leftmost bottom), or even at quarter-resolution (third on bottom line).

III. New "connected-Control-points" features

As presented in [9], we have recently explored new types of adaBoost features in the context of rear car detection. It turned out that among those, the new "connected control-points" significantly outperformed all others. This feature is a particular form of the control-points feature. It contains 2 up to 12 points, and the principle is exactly the same as described in II. The difference is that the "control-points" of a given feature are constrained to remain connected with 8-connectivity, which implies each point must touch another one *at least by a corner*.

As mentioned in section II, the classical control-points features family is extremely large, and therefore difficult to search efficiently by the weak-learner. By imposing the 8-connectivity constraint, the search-space size decreases to $\sim 3x10^{19}$ possible combinations instead of $\sim 10^{35}$, which makes it easier to explore efficiently for our heuristic. Besides, the connectedness constraint will force each feature to focus on a more localized part of the detection window.

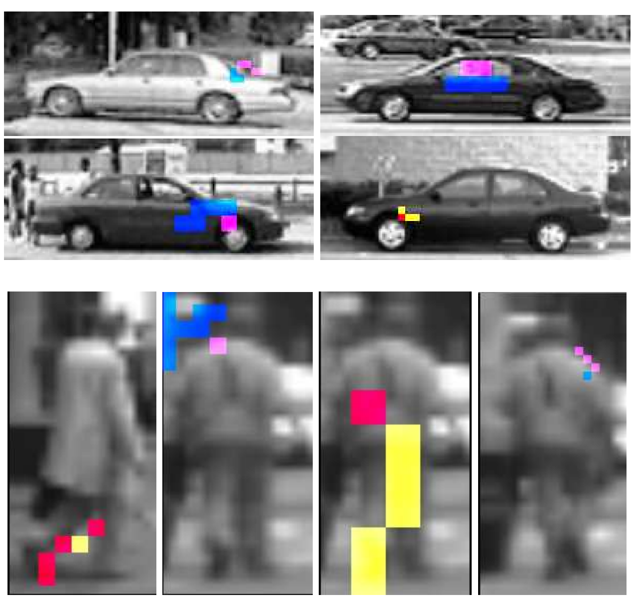

Fig. 5: Some examples of adaBoost-selected new "connected control-points" features for lateral car detection (top) and pedestrian detection (bottom line).

In figure 5 are shown some examples of the "connected control-points" features resulting from the adaBoost training process for cars and pedestrians. The evolutionary heuristic weak-learner we use is exactly the same as for standard control-points, except that the mutation operator has been modified to maintain the connectedness constraint. As can be seen by comparison to figure 4, because of the connectedness constraint, each of the new features tend to operate on a particular region (as can readily be seen on figure 5), contrary to basic control-points features whose points positions are sometimes disseminated throughout the detection window (see eg bottom right on figure 4). As a result, our connected-control-points features are in some way a kind of generalization of Haar-like features, but much more flexible in shape so that they can adapt themselves to detect any particular contour or contrast geometry. Note on examples of figure 5 that the features we obtain are even more general and flexible than the generalization of Viola&Jones type features proposed by Treptow and Zell in [17], with which they had obtained better detection performances than with standard Viola&Jones Haar-like features.

IV. Experiments and results

Encouraged by our good results on rear car detection [9], we decided to test our new "connected control-points" features for other kind of objects encountered in vehicle environment: lateral cars, and pedestrians. In order to allow comparisons with other published methods, we have chosen to work on publicly available databases: the "lateral cars" by UIUC [11], and the pedestrian database collected by Munder and Gavrila [10].

A. Lateral cars database

The lateral cars database contains 500 positive examples and 500 negative examples, all of size 100x40 pixels. For evaluation, we use, as in [11], the set of 108 wider field independent images containing 139 lateral cars at various scales, ranging from roughly 0.8 to 2 times the size of cars in the training images. This test set comes with an associated ground truth allowing automated computation of correct detection and false alarm rates.

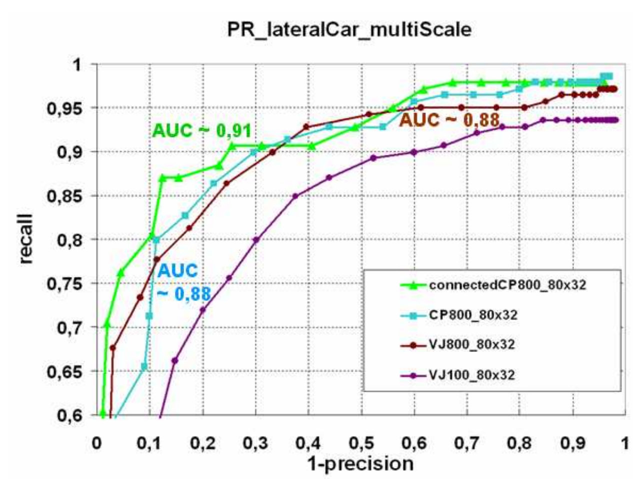

Fig. 6: Precision-recall curves for adaBoost lateral car detectors obtained with connected-control-points features (upper green curve), standard control-points (cyan curve), and ViolaJones Haar-like features (maroon)

All trainings were done for 800 boosting steps. Figure 6 shows precision-recall curve of resulting detectors obtained with various feature families. We use precision-recall metrics in order to allow easy comparison with (rather poor) results of the method presented in [11] on the same database. Our new "connected control-points" features (upper curve, and best Area Under Curve with 0.91, instead of 0.88) outperforms both our usual simple control-points, and Haar-like features.

Figure 7 shows some detection results on test wider-field images by our connected-control-points adaBoost classifier. These illustrate the robustness to at least moderate occlusion, of classifiers built with our new features.

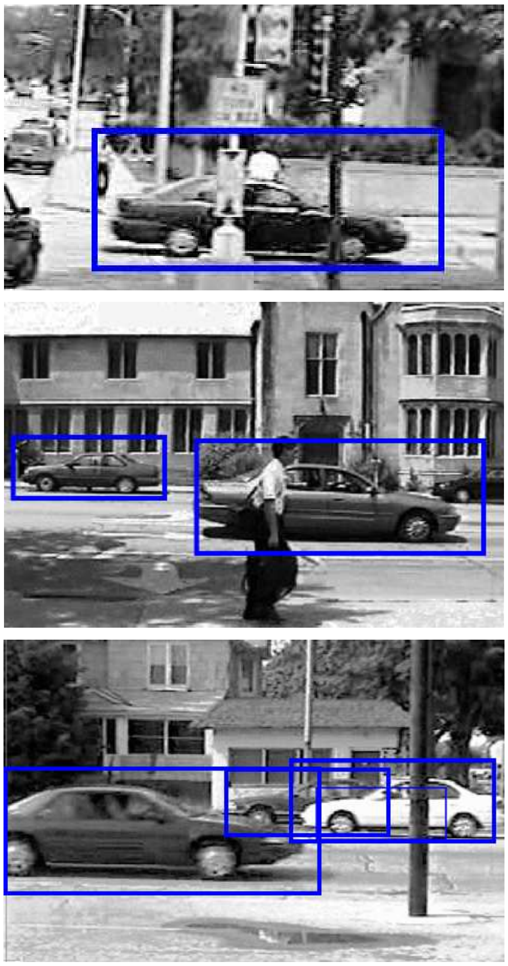

Fig. 7: Some detection results with our connected-control-points adaBoost classifier, which illustrates its robustness to at least moderate occlusion.

If we compare detectors with similar computation loads (in this particular setup, control-points features operate ~ 8 times faster than our implementation of ViolaJones Haar-like features), then the superiority of our new connected control-points features over Haar-like features is even clearer (see figure 7). It should be noted however that our ViolaJones classifiers were obtained using the same heuristic weak-learner as for control-points (with adapted mutation operator), rather than usual full-search which would anyway have been prohibitively long for a 80x32 detection window size.

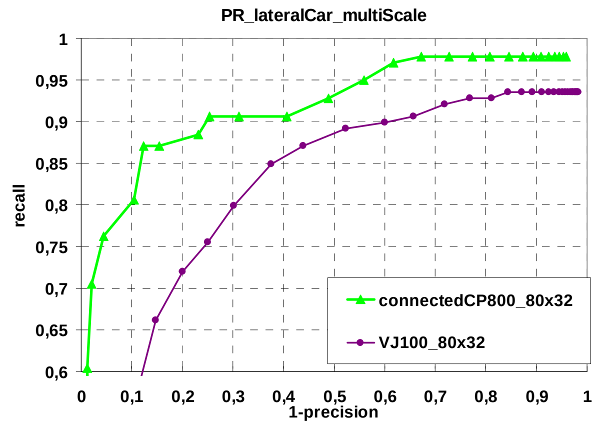

Fig. 7: Precision-recall for adaBoost lateral car detection, when comparing detectors with similar computation loads. At equivalent computation time, our new connected-control-points features clearly outperform ViolaJone Haar-like features.

B. Pedestrians database

The pedestrian database comprises 3 training sets and 2 test sets (each one of the 5 sets with 4800 positive examples and 5000 negative ones). As suggested in [10], 3 independent trainings were conducted on unions of 2 of the 3 training sets, and the evaluation was done on the 2 test sets, producing a total of 6 evaluations, to be averaged, for each feature type. In each training, 2000 boosting steps were allowed, therefore producing adaBoost detectors assembling 2000 weak-classifiers.

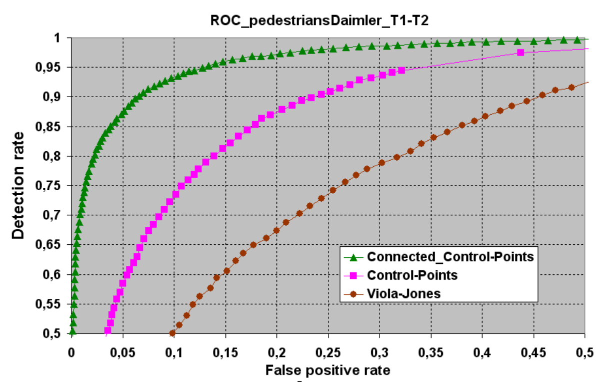

Fig. 9: Averaged ROC curves for adaBoost pedestrian classifiers obtained with various feature families

As one can see in figure 9, the classifiers obtained with the new “connected control-points” features have by far the best classification results. The Viola-Jones performs rather poorly, even when compared to “ordinary control-points”.

We also compared the performance of our new classifier to the Viola-Jones classifier performance reported in [10], which was obtained with openCV implementation. As can be seen on figure 10, our “connected control-points” pedestrian classifier has a significantly better performance, which confirms the results obtained with our own implementation (with which we did not use cascade for our comparisons).

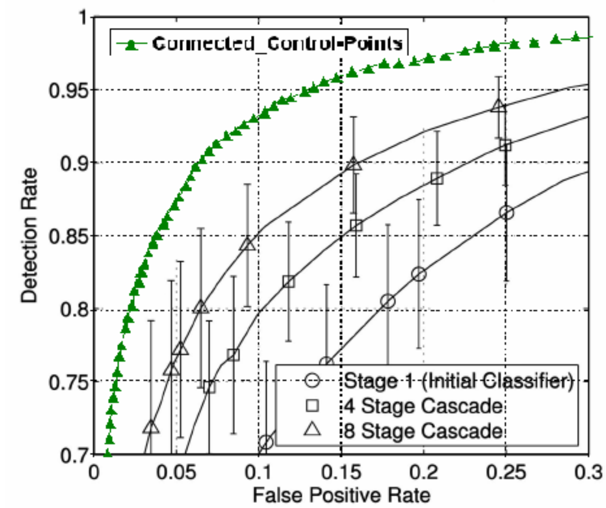

Fig. 10: ROC curves comparing our boosted “connected control-points” (upper curve, green) to boosted ViolaJones cascade result reported in [10].

Moreover, we finally compare to the best methods reported in [10] on figure 11, where one can see that boosting with our new features seems to be even better than the best algorithms (namely quadratic and RBF SVM, and NN-LRF) reported in [10].

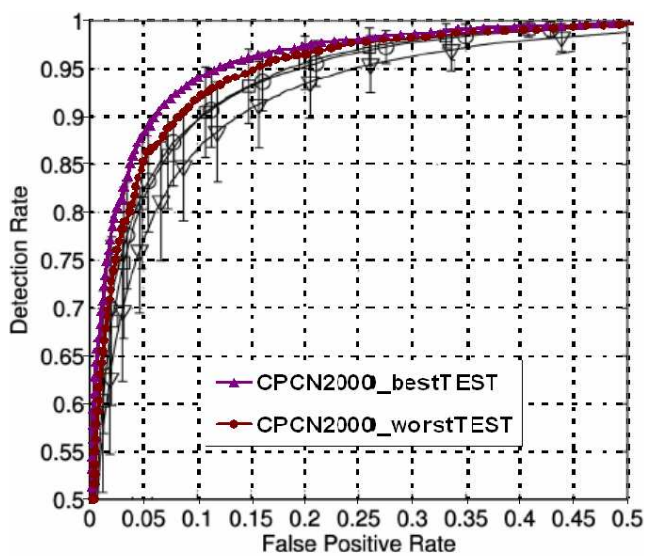

Fig. 11: ROC curves comparing our boosted connected control-points (two upper curves) to best algorithms results reported in [10].

It should be noted that the best algorithms from [10], to which we compare on figure 11, are reported in [10] to operate at ~ 250 ms per test sample on a 3.2 Ghz Pentium IV PC, while our boosted classifier containing 2000 “connected control-points” features requires only ~0.4ms per test image from the database, on a 2 GHz Intel Core2 laptop.

V. Conclusions and perspectives

We have presented a new feature type, which we call “connected control-points”, for adaBoost training of visual object classifiers.

We report here on test of these new features on two publicly available databases: one for lateral cars, and one for pedestrians on which many classification algorithms have already been tested and results published. It turns out that the adaBoost strong classifiers obtained with our new features, while being extremely fast (~0.4ms per pedestrian image classification on a 2Ghz laptop), clearly outperform both standard Viola-Jones boosted cascade and even the most powerful (but very slow) classification algorithms reported so far on the pedestrian database.

Given previous tests conducted by us on real-time visual rear car detection application [9] that have also shown these new “connected control-points” features to provide better results than other features used in boosting, we think these new features have a very promising potential for improving real-time detection performance of visual object classes in general, and particularly the kind of objects that should be efficiently detected and tracked in intelligent vehicle applications.

References

[1] Yoav Freund, Robert E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting”, ‘95 European Conference on Computational Learning Theory, pages 23–37, 1995.

[2] Yoav Freund Robert E. Schapire, “A short introduction to boosting”, Journal of Japanese Society for Artificial Intelligence 14(5), pages 771–780, 1999.

[3] R. Meir and G. Ratsch, “An introduction to boosting and leveraging”, S. Mendelson and A. Smola, Editors, Advanced Lectures on Machine Learning, LNCS. Springer Verlag, pages 119–184, 2003.

[4] Paul Viola, Michael Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features", IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’2001), Volume 1, page 511, Kauai , Hawai, USA, 2001.

[5] Paul Viola, Michael J. Jones, Daniel Snow. « Detecting pedestrians using patterns of motion and appearance”, IEEE International Conference on Computer Vision (ICCV’2003), pages 734–741, Nice, France, 2003.

[6] Yotam Abramson, Bruno Steux, "Illumination-independent pedestrian detection in real-time", internal report, CAOR, Mines ParisTech, 2003.

[7] Yotam Abramson, Bruno Steux, Hicham Ghorayeb, “YEF (Yet Even Faster) Real-Time Object Detection “, 2005 Proceedings of International Workshop on Automatic Learning and Real-Time (ALART’05), Siegen, Germany, page 5, 2005.

[8] Yotam Abramson, Fabien Moutarde, Bruno Steux, Bogdan Stanciulescu, "Combining adaBoost with a Hill-Climbing evolutionary feature search for efficient training of performant visual object detectors", Proceedings of FLINS2006 Conference on Applied Computational Intelligence, Genova, Italy, pages 737-744, 2006.

[9] Bogdan Stanciulescu, Amaury Breheret, Fabien Moutarde, "Introducing New AdaBoost Features for Real-Time Vehicle Detection", Proceedings of COGIS2007 Cognitive Systems with Interactive Sensors, Stanford University, USA, 2007.

[10] Stefan Munder and Dariu M. Gavrila. “An Experimental Study on Pedestrian Classification”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.28, nr 11, pp. 1863-1868, 2006.

[11] Shivani Agarwal, Aatif Awan, and Dan Roth, “Learning to Detect

Objects in Images via a Sparse, Part-Based Representation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.26, nr 11, pp. 1475-1490, 2004.

[12] Navneet Dalal and Bill Triggs, "Histograms of Oriented Gradients for Human Detection", proceedings of 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), held in San Diego, CA, USA, 20-26 June 2005.

[13] Qiang Zhu, Mei-Chen Yeh, Kwang-Ting Cheng, Shai Avidan, "Fast Human Detection Using a Cascade of Histograms of Oriented Gradients", proceedings of 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006), held in New York, NY, USA, 17-22 June 2006.

[14] Shumeet Baluja, Mehran Sahami, and Henry A. Rowley, "Efficient face orientation discrimination", proc. *International Conference on Image Processing (ICIP '04)*, pp. 589-592 Vol. 1, 24-27 Oct. 2004.

[15] Laetitia Leyrit, Thierry Chateau, Christophe Tournayre and Jean-Thierry Lapresté, "Association of AdaBoost and Kernel Based Machine Learning Methods for Visual Pedestrian Recognition", proc. *IEEE* Intelligent Vehicles Symposium (IV 2008), Eindhoven (Nederland), June 2008.

[16] Niklas Pettersson, Lars Petersson and Lars Andersson, "The Histogram Feature – A Resource-Efficient Weak Classifier", proc. *IEEE* Intelligent Vehicles Symposium (IV 2008), Eindhoven (Nederland), June 2008.

[17] A. Treptow and A. Zell, "Combining adaBoost learning and evolutionary search to select features for real-time object detection", In Proc. of IEEE Congress on Evolutionary Computation (CEC 2004), Portland, Oregon, vol. 2, pp. 2107-2113, IEEE Press (2004).