# Unsupervised vehicle detection in traffic scene using distributed one class classifiers

Adrien Foucart
Laboratory of Image Synthesis and Analysis
Ecole Polytechnique
Universite Libre de Bruxelles
Brussels, Belgium
Email: afoucart@ulb.ac.be

Olivier Debeir
Laboratory of Image Synthesis and Analysis
Ecole Polytechnique
Universite Libre de Bruxelles
Brussels, Belgium
Email: odebeir@ulb.ac.be

***Abstract*—Background detection is common in object detection and tracking and has been widely addressed, in particular for automatic video traffic analysis where vehicle detection can be achieved by comparing the current frame to a background image. For traffic applications, background detection models have to cope with varying conditions such as traffic density, camera movement or scene illumination changes and computation power limitation when image computing is embedded into the camera itself. We introduce an unsupervised technique making the hypothesis that there is a strong bias for the background in usual traffic surveillance images. Indeed, for narrow angle traffic acquisition, most of the frames are occupied by background. We explore the possible use of One Class classifier paradigm to efficiently detect background in images, and therefore detect when the background is not present (i.e. vehicle). The proposed method is also locally adaptive, to deal with possible complex background such as road markings, shadows or road irregularities. We show that this unsupervised, continuously adaptive, approach gives robusts results both in detection efficiency and computing load. Results are compared to a classical mixture of gaussian approach and human reference.**

## I. Introduction

Vehicle detection in traffic videos has mostly been studied for wide views englobing multiple lanes, in both colour and grayscale images. However, to extract as much information on the vehicle as possible, and in particular to be able to read the license plate, it is necessary to deal with narrow fields of view. The task of identifying moving objects in a video sequence has received a lot of attention in computer vision applications. A common approach consists in creating a background model and using it as a reference to compare with the current frame [1]. Foreground is then deduced by subtracting the background model from the current frame. These methods are heavily dependant on the reliability of the background model. Common background models use Mixture of Gaussians, optical flow or Hidden Markov Models for each pixel on the image. Cheung and Kamath [2] identified the main challenges of any such algorithm : robustness against changes in illumination, non-stationary background objects, stationary foreground objects, changing climatic conditions, small camera movements... As an alternative to the model based approach, authors proposed methods based on supervised examples, with a classifier trained to recognize foreground and background regions of the image [3]. However, such a supervised method has limitations in particular when it is used with a different acquisition setup, since the system needs to be re-trained. We propose to use unsupervised SVM classifiers which will be able to adapt to new settings with few user-set parameters. After this introduction, we will describe in the following section II the sequences and the requirement of our traffic surveillance application, in section III we briefly recall the basics of the one class classification and we present our approach, in section IV we compare the results with a classical mixture of gaussian model with several manually supervised sequences of various lighting conditions and traffic intensity.

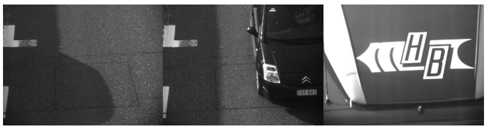

Fig. 1. Three frames from one of our test sequences, illustrating some of the difficulties of this setting.

## II. Material and requirements

While other vehicle detection applications in the litterature deal either with wide views or with a moving camera, this work deals with the case where the camera is fixed, with a narrow view covering one lane only, in free flowing traffic. The main benefit of such a setup is that it allows for the automatic reading of the license plate. For the examples shown in this paper, the native images resolution is 1024x780 pixels (30 fps), subsampled at 780x640 (30 fps) (single 8-bit channel). The desired output is the localisation of the area occupied by a moving vehicle. The system should be able to run with minimal setup, calibration or supervision.

We identified several aspects to be tackled : (i) any vehicle will only be visible for about five frames for cars at regular speed, (ii) for most of these frames, the vehicle will only be partly visible, (iii) vehicles hide huge portions of the background, or even all of it for trucks, (iv) ground markings may occupy a significant part of the background, which makes background subtraction very sensitive to small vibrations of the camera (see figure 1).

978-1-4244-5997-1/10/$26.00 ©2010 IEEE

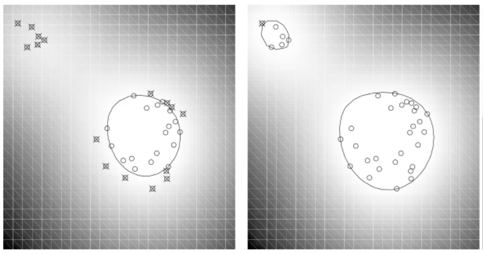

Fig. 2. Data points and decision frontier for two different values of $\nu$, and the same kernel function. Left : $\nu = 0.5$, right : $\nu = 0.1$. With a lower $\nu$, the penalty for ignoring the points on the top left is too strong, and they are considered as inliers. From [6, p.16]

## III. Method

### A. One Class classifier

Support Vector Machines have been used extensively since the late nineties in applications ranging from handwriting recognition to regression problems. The main motivation behind the development of SVMs is to create a learning machine with a bound on its generalization performance [4]. The basic idea of an SVM classifier is, given a training set $(x_i, y_i), i = 1, ..., l$ with $x_i$ the data point and $y_i$ the target class, to map the data points to a higher dimensional space where the classes can be linearly separated by a hyperplane, with the largest possible margin [5]. The parameters of such a classifier will be the penalty for incorrectly classified training data, and the parameters of the kernel function used to map the data to a higher dimensional space.

The one-class SVM introduced by Schölkopf et al. [6] extends the method to unsupervised problems. Here, the input of the classifier is a learning set $(x_i), i = 1, ..., l$, and the classifier will try to find the region where most of the data is found, with a parameter $\nu \in \{0...1\}$ regulating how many point in the training set may be considered as outliers of the distribution. In other words, if $\nu$ is close to zero, a huge penalty will be set for any point outside the decision frontier. This influence is illustrated in the figure 2.

This kind of classifier is therefore extremely useful when studying a system whith a large, majority class, with occasional, potentially very different, outliers. This is very relevant to our problem, since a typical portion of a road in a traffic image will most of the time be occupied by the background, and the passage of a vehicle will be an occasional event.

### B. Outline of the algorithm

The main idea of the algorithm is to have a set of classifiers, each trained to recognize the background of a small region of the image. Once trained, predictions from all the classifiers are collated to determine if a vehicle is present, and the region it occupies. By having each classifier trained on a specific region, we reduce the influence of features-rich background region, such as the ground markings.

The different steps of the algorithm are as follow :

1) Define region of activity of the classifiers, and parameters of the SVMs.
2) For each region, aggregate a training set of descriptors over N frames.
3) Train SVMs.
4) For each new frame : collate all predictions and determine the region where a vehicle is present.
5) Retrain every N frames.

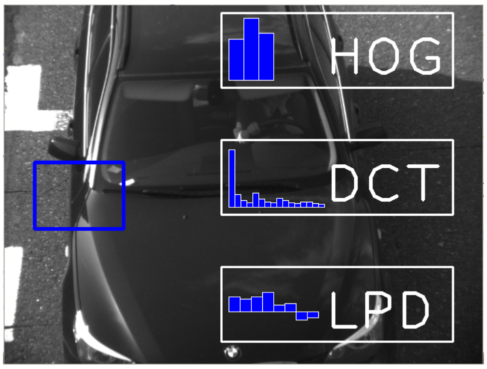

Fig. 3. Result of the computation of the HOG, DCT and LPD descriptors for a region of an example frame. HOG descriptor shows the dominance of a vertical border, the different modes of the DCT corresponds to the most represented spatial frequencies, and the LPD shows the presence of some pattern.

The first step will be a trade-off between how precise our classifiers will be in defining the region where the vehicle is, and how much information a classifier will have to make their prediction. If the regions are too small, the classifiers will be much more sensible to noise ; if the regions are too big, they will not be able to provide very precise localisation information. In our implementation, we used overlapping regions of 120x90 pixel. The overlapping allows for a better spatial precision while keeping each region big enough to have a significant portion of the image visible.

The different descriptors tested in this work will be presented hereafter.

For the retraining process, the main thing to consider is that we must be reasonably sure that the frames used for training are representative enough of the distribution. If the traffic is too dense, the SVMs might try to fit the decision frontier around the vehicles instead of the background. If the traffic is too light, they will not have any example of outliers.

### C. Low-level image descriptors

The descriptors will have to be chosen so as to best be able to make the distinction between background and vehicles for every region. They must be robust against rapid changes in illumination, and small vibrations of the camera. Therefore, descriptors based on the intensity value of the pixels are to be avoided, in favor of descriptors based on edges and patterns informations. Three such descriptors are presented here : (i) the Histogram of Oriented Gradients (HOG) similar to Dalal and Triggs [7], (ii) the Discrete Cosine Transform and (iii) the Local Pattern Descriptor. The three descriptors are illustrated in the figure 3.

*1) Histogram of Oriented Gradients:* Histograms of Oriented Gradients have first been introduced for human detection [7], but have also been used before for vehicles [8]. The principle of the HOG is to compute, for each pixel of the image, the gradient magnitude and orientation of the illumination ; then, for each region, magnitudes are summed along a set of quantified orientations. The resulting histogram for each region will be representative of the edges visible in the region. In our implementation, we create three gradient images by convolution with these kernels :

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}, \begin{bmatrix} 2 & \frac{1}{2} & -2 \\ \frac{1}{2} & -2 & \frac{1}{2} \\ -2 & \frac{1}{2} & 2 \end{bmatrix}$$

The gradient images correspond to horizontal, vertical and adirectional gradients. The absolute value of the gradient images are summed over each region, resulting in a three-bins histogram for each. Since these are edge-based descriptors, they should be invariant to illumination. However, many small edges on the road only become apparent when the light is strong enough, making this descriptor sensitive to shadows and quick variations in sunlight.

*2) Discrete Cosine Transform:* The Discrete Cosine Transform is widely used both for image compression (such as the JPEG standard) and for spectral analysis of a signal. It provides a lot of information on the presence of edges and patterns with a low computational cost. For each region, we can compute the DCT. The DCT is then split into 4x4 blocks, and the average power for each block is put into a 16-bins histogram.

*3) Local Pattern Descriptor:* This descriptor is based on the Local Binary Patterns introduced by Ojala et al. [9] and described for background subtraction by Heikkila and Pietikainen [10]. The idea is to characterize the pattern of its neighbours. If $p_c$ is the value of the central pixel, and $p_k, k = 0, ..., N-1$ a set of N pixels on a circle around it, then the LBP descriptor of C will be :

$$LBP(C) = \sum_{k=0}^{N-1} s(p_k - p_c)2^k, s(x) = (x \geq 0)$$

If all pixels surrounding C are darker than C, then $LBP(C) = 00...00$, if they are all brighter, then $LBP(C) = 11...11$. Our Local Pattern Descriptor takes the same concept, but with a per-block approach. Each region is divided in 3x3 blocks. Let $B_C$ be the center block, and $B_k, k = 0...8$ the neighbouring blocks. We compute the average intensity value for each block $m_i$, and define the descriptor :

$$LPD(C) = (m_c - m_i), i = 0...8$$

If the neighouring blocks are darker than the central block, the descriptor will be filled with positive values. This descriptor is very robust to changes in illumination, since even when new borders appear due to a brighter light, they aren't large enough to contribute significantly to the difference in block averages. Likewise, the use of block averages prevent small vibrations of the camera to have a strong influence on the descriptor.

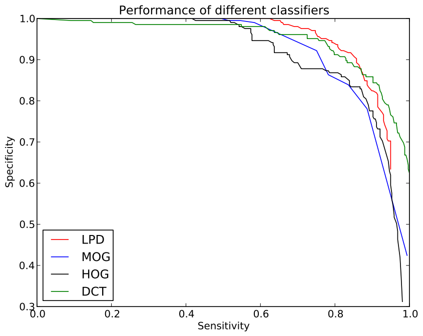

Fig. 4. Results of the classification for the LPD (red), MoG (blue), HOG (black) and DCT (green) classifiers on a sensitivity (x-axis) / specificity (y-axis) graph on the first test sequence.

## IV. Results

### A. Methodology

As a first test of the interest of our method, we compare its ability to simply detect the presence or absence of a vehicle with that of a standard Mixture of Gaussian (MoG) algorithm (using the implementation of KaewTraKulPong and Bowden [11] in OpenCV). The Mixture of Gaussian is a background subtraction approach consisting in updating a model of the intensity values for each pixel, fitted to separate gaussian distributions, with the most common distribution identified as the background. For each new frame, the pixels are fitted on the gaussians, and the most probable distribution decides if the pixel is background or foreground.

To decide whether or not there is a vehicle present, we compute the area that the classifier predicts is in the foreground. For our method, we compare the results with the three descriptors. For the Mixture-of-Gaussian, we kept the parameters which gave the best results for our sequences.

Our first test sequence is composed of 1522 successive frames. The classifiers are trained on the first 1000 frames, and the predictions are made on the 522 last. The conditions are that of a normal afternoon traffic (57 % of the frames have at least a portion of a vehicle in it), with strong shadows. Our second test sequence has 2000 frames (1000 for training, 1000 for predictions), with a lighter traffic (21 % of the frames with a vehicle) and early morning light (with some cars with their headlights on, some without). Our third sequence is also 2000 frames long, with a very sparse traffic (7 % of the frames with a vehicle) but good lights conditions.

### B. Results

Figure 4, 5 and 6 shows the sensitivity-specificity graph for the three test sequences and all the tested classifiers. We see that the MoG classifier performs better in very sparse traffic, but that its performance degrades when confronted to heavier traffic. The one-class SVM, particularly with our Local Pattern Descriptors, is less dependant on the conditions.

In terms of speed, the time for the whole process (training and predictions together) on 2000 frames have been computed

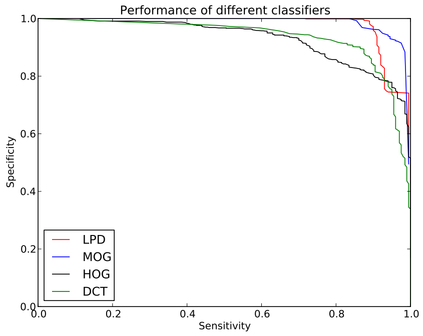

Fig. 5. Results of the classification for the LPD (red), MoG (blue), HOG (black) and DCT (green) classifiers on a sensitivity (x-axis) / specificity (y-axis) graph on the second test sequence.

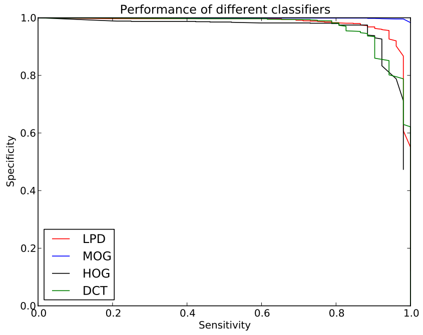

Fig. 6. Results of the classification for the LPD (red), MoG (blue), HOG (black) and DCT (green) classifiers on a sensitivity (x-axis) / specificity (y-axis) graph on the third test sequence.

on a regular DELL laptop with two Intel(R) Core(TM) 2 Duo CPU @2.20GHz and 3.4 Go of memory, with an implementation of the algorithms on OpenCV 2.1. The results are as follows for the different methods :

- Histogram of Gradients : 66.87 s, or 29.9 fps
- Discrete Cosine Transform : 145.67 s, or 13.73 fps
- Local Pattern Descriptors : 41.78s, or 47.87 fps
- Mixture of Gaussian : 100.8s, or 19.8fps

## V. Conclusion

Our preliminary results seem to show performances at least as good as that of standard background subtraction algorithms. The main improvement of our method is its completely unsupervised nature, which makes it easily portable to new settings, new conditions, and new similar problems of detecting transient events. When using quick, robust descriptors such as the Local Patterns, it can easily be used in real-time applications. Future work will have to focus on robust ways to detect situations which would make re-training difficult, and to further test the robustness towards more difficult meteorological conditions, as well as night time sequences.

## Acknowledgment

The authors would like to thank the Brussels Institute for Research and Innovation (www.innoviris.be) for funding this project, as well as Macq (www.macq.eu) for starting it and providing the image sequences.

## References

[1] S. Elhabian, K. El-Sayed, and S. Ahmed, “Moving object detection in spatial domain using background removal techniques - state-of-art,” *Recent Patents on Computer Science*, vol. 1, pp. 32–54, 2008.

[2] S. Cheung and C. Kamath, “Robust techniques for background substraction in urban traffic video,” *Proc. Visual Communications and Image Processing*, vol. 5308, pp. 881–892, 2004.

[3] J. Zhou, D. Gao, and D. Zhang, “Moving vehicle detection for automatic traffic monitoring,” *IEEE Trans. on Vehicular Technology*, vol. 56, no. 1, pp. 51–59, January 2007.

[4] C. Burges, “A tutorial on support vector machines for pattern recognition,” *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, 1998.

[5] C.-W. Hsu, C. Chang, and C. Lin, “A practical guide to support vector classification,” Published online : http://www.csie.ntu.edu.tw/ cjlin, April 2010.

[6] B. Schölkopf, J. Platt, J. Shawe-Taylor, A. Smola, and R. Williamson, “Estimating the support of a high-dimensional distribution,” *Neural Computation*, vol. 13, no. 7, pp. 1443–1471, July 2001.

[7] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, vol. 1, pp. 886–893, 2005.

[8] P. Negri, X. Clady, and L. Prevost, “Benchmarking haar and histograms of oriented gradients features applied to vehicle detection,” *Proceeding of the 4th International Conference on Informatics in Control Automation and Robotics ICINCO07*, p. 359364, 2007.

[9] T. Ojala, M. Pietikinen, and D. Harwood, “Performance evaluation of texture measures with classification based on kullback discrimination of distributions,” *Proceedings of the 12th IAPR International Conference on Pattern Recognition (ICPR 1994)*, vol. 1, pp. 582–585, 1994.

[10] M. Heikkila and M. Pietikainen, “A texture-based method for modeling the background and detecting moving objects,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 657–662, 2006.

[11] P. KaewTraKulPong and R. Bowden, “An improved adaptive background mixture model for real-time tracking with shadow detection,” *Proc. 2nd European Workshop on Advanced Video Based Surveillance Systems, AVBS01*, September 2001.