Nighttime Vehicle Detection for Intelligent Headlight Control

Antonio López[1], Jörg Hilgenstock[2], Andreas Busse[3], Ramón Baldrich[1], Felipe Lumbreras[1], and Joan Serrat[1]

[1] Computer Vision Center and Computer Science Dept., Auton. Univ. of Barcelona*
[2] Volkswagen AG, Group Research
[3] Carmeq GmbH, Business Team Surround Sensing
antonio@cvc.uab.es

Abstract. A good visibility of the road ahead is a major issue for safe nighttime driving. However, high beams are sparsely used because drivers are afraid of dazzling others. Thus, the intelligent automatic control of vehicles' headlight is of great relevance. It requires the detection of oncoming and preceding vehicles up to such a distance that only camera based approaches are reliable. At nighttime, detecting vehicles using a camera requires to identify their head or tail lights. The main challenge of this approach is to distinguish these lights from reflections due to infrastructure elements. In this paper we confront such a challenge by using a novel image sensor also suitable for other driver assistance applications. Different appearance features obtained from that sensor are used as input to a novel classifier–based module which, for each detected target, yields a degree of resemblance to a vehicle light. This resemblance is integrated in time using a novel temporal coherence analysis which allows to react in one single frame for targets that are clear vehicle lights, or in only a few frames for those whose type is more difficult to discern.

1 Introduction

The fatal crash rate for nighttime driving is three to four times that of daytime, even though the traffic volume is substantially less [1]. In order to be warned early about hazards, drivers need to look far ahead to see traffic signs, road geometry, other vehicles, pedestrians, etc. However, this task is difficult at night because vision is severely limited: drivers lose the advantage of color and contrast that is available during the day, and depth perception and peripheral vision are also diminished. Accordingly, the headlight system of a vehicle has the aim of providing a safe illumination for driving. The most common system in the market is based on the manual switching between *low* and *high beams* (Fig. 1). In absence of fog, drivers should use high beams under poor ambient lighting but without disturbing others. However, high beams are used less than 25% of the time in

* These authors are partially supported by Spanish MEC research projects Consolider Ingenio 2010: MIPRCV (CSD200700018) and TRA2007-62526/AUT.

J. Blanc-Talon et al. (Eds.): ACIVS 2008, LNCS 5259, pp. 113–124, 2008.

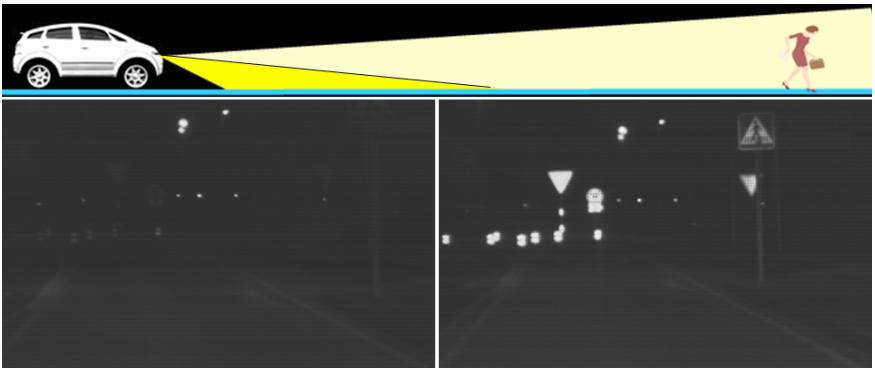

Fig. 1. *Top*: low beams point downward onto the road, while high beams point upward to help drivers identifying far away objects. *Bottom*: nighttime scene imaged with high beams off (*left*) and on (*right*). Notice how clear is the presence of close poles and traffic signs because of the reflection of the light emitted by the car high beams. The energy level of such reflections reaching the image sensor is equal or higher than the one of mid to large distant head and tail lights of other vehicles.

which driving conditions justify their use [2], probably because drivers are afraid of dazzling others by mistake. Thus, our motivation is to develop an intelligent headlight controller for freeing the driver of such a task.

In order to address this problem we must start by selecting the proper sensor technology. Notice that, to avoid dazzling other drivers, oncoming vehicles must be detected up to at least 600 m ahead and preceding ones up to at least 400 m. Such challenging distances rule out active sensors as radar or lidar. Fortunately, vehicle detection at nighttime can be based on vision sensors, which have the additional advantage of being passive and cheaper. In fact, we can already find a monocular camera based system in the market that, according to the presence or not of other vehicles, is able to switch between low and high beams automatically [3]. However, in order to detect vehicles this system relies on a highly specialized vision sensor which makes it difficult to use it for other driving assistance tasks requiring lane markings detection, traffic sign recognition, etc.

Our aim is also to develop a real–time vehicle detector reaching the previously mentioned range of detection distances, but based on a monocular image acquisition system such that it would be possible to automatically control the headlight as well as to address other driving assistance applications. To the best of our knowledge no such a system can be found in the market.

Vehicle detection at nighttime consists, in fact, in detecting the corresponding head or tail lights, thus, it may seem a question of simple image thresholding. However, such an intuition underestimates the actual difficulty of the problem: it turns out that the own emitted light is reflected in different infrastructure elements such as traffic signs, fences, poles, etc., in a way that are difficult to distinguish from mid to far distant vehicles (Fig. 1). In order to tackle this

problem we propose the analysis of both the appearance and motion of the detected targets. In this paper, however, we focus on appearance analysis.

The proposed system presents several novelties. First, we use an image sensor based on a novel pattern of color filters on the pixels. To detect taillights, red color is a relevant cue but sensors based on Bayer patterns have poor sensitivity compared to monochrome sensors, specially for our application. Hence, we employ a novel sensor where 75% of the pixels are monochrome and 25% are red. No blue or green pixels are present. This sensor allows to address other driving assistance applications. Second, we have designed a new classifier–based analysis which takes into account that far away lights look different than close ones as well as that head and tail lights look also different. The classifiers use a set of target appearance features obtained from the new sensor. Whenever the previous set of features is not sufficient to make a safe decision, additional appearance features of the target neighborhood are computed. Third, for each detected target the classifier–based analysis provides a degree of resemblance to a vehicle light that is further integrated in time using a new proposed temporal coherence analysis that allows to react in one single frame for targets that are clear vehicle lights, or in only a few frames for targets whose type is more difficult to discern.

The paper is organized as follows. Section 2 summarizes different strategies for automatic headlight control. Section 3 presents our image acquisition system. In Sect. 4 we explain our vehicle detection method. Results are outlined in Sect. 5. Finally, Sect. 6 draws the main conclusions.

2 Intelligent Headlight

A standard headlight system only allows to manually switch between a short range illumination ($\simeq$ 70 m) and a large one ($\geq$ 150 m), *i.e.*, the mentioned low and high beams. Hence, a flexible adaptation of the headlight distribution to the actual traffic situation would improve the scene visibility. In fact, we can find in the market systems that use data from the ego–vehicle to provide illumination distributions ranging from short–wide beams (at low speed) to large–narrow ones (at high speed), swivelling in curves according to the steering angle.

However, such systems do not use environmental sensing to control the headlight distribution. With a sensor detecting oncoming and preceding traffic, different headlight systems of increasing capabilities can be designed. The simplest idea is the automatic switch between low and high beams. A more advanced concept consists in dynamically adapting the position of the illumination cut–off just below the next vehicle in front of the ego–vehicle [4]. Future headlight systems will employ a fully variable light distribution to realize a dazzle–free high beam. In such a system, small regions where other vehicles are detected are dynamically masked out in the high beam light distribution (Fig. 2). These systems can be build using arrays of LEDs [5]. The high beam is composed out of several small LED spots. If another vehicle is detected only the LED which generates the spot for the region of the detected vehicle is turned off, while the rest of regions are still illuminated by the other LEDs.

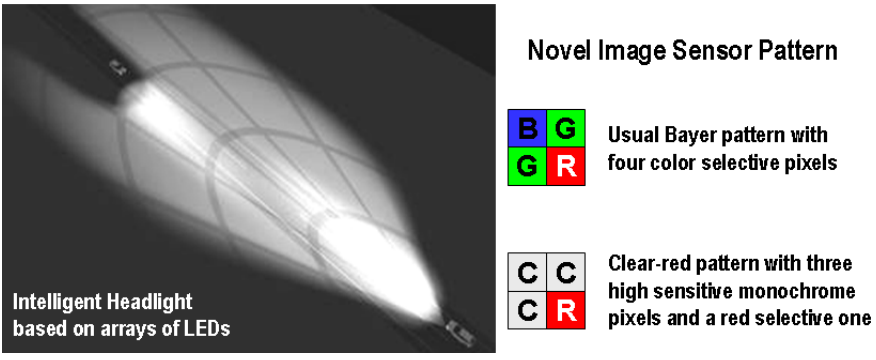

Fig. 2. *Left:* dazzle–free high beam based on arrays of LEDs. This headlight system can mask out potential dazzling light by detecting vehicles with a vision system. *Right:* the novel image acquisition sensor we employ combines the high sensitivity of three monochrome detectors with the color information from a single red selective detector.

3 Acquisition System

It is clear that a robust nighttime vehicle detection method is essential for the described advanced lighting systems. Such a system has been introduced into the market by Gentex [3], however, it is based on a highly specialized image sensor that limits its usefulness for addressing other driver assistance tasks. Contrarily, we want to base our solution on a camera which could also be used for other driver assistance applications addressable by a monocular monochrome acquisition system, like lane departure warning or traffic sign recognition.

The camera requirements of these applications differ in terms of field of view, sensitivity and exposure control. For example, traffic sign recognition requires a wide opening angle, while to control the high beam we require a high resolution in the far field rather than a wide opening angle. Settings like exposure control are not a problem since nowadays cameras can work at a high acquisition ratio accepting changes on such parameters in real–time, thus, each application can set its own parameters in a way that inside the driver assistance cycle different acquired images will be used by different applications. Other parameters, however, can not be changed in real–time since these are physical properties of the equipment. Most relevant ones are the lens properties as well as the sensor sensitivity. Such parameters must be set following a trade–off for all the driver assistance applications.

To meet the demands of all applications a horizontal opening angle of more than 40° is required. Such an angle makes very challenging to detect far away taillights. For instance, a taillight with a size of 10×10 cm^2 at a distance of more than 100 m is imaged by less than one pixel on the image sensor. Fortunately, the emitted light forms a bigger cone so that a taillight at 400 m still hits areas of about 4 to 10 pixels. However, if with the aim of exploiting color information

we use an image sensor with a Bayer pattern, such a far away taillight can only be detected at red pixels, which are only the 25% of the sensor. Accordingly, we are using a new CMOS image sensor from Aptina Imaging$^{(TM)}$ with a resolution of 752×480 pixels and a novel pattern that we call *clear–red pattern*. This sensor is composed of 2×2 blocks having one single red selective detector and three standard monochrome detectors, which have more sensitivity than color selective ones and therefore also capture reddish light (Fig. 2). With this new pattern it is possible to distinguish red lights from white ones, while maintaining high sensitivity on the pixels without a color filter. It is also worth to mention that this acquisition system is NIR–free and allows to work with 10 bits per pixel using a logarithmic curve, which is essential to avoid fully saturated light spots, *i.e.*, with most of the appearance information lost.

4 Vehicle Detection

Vehicle detection based on vision sensors gives rise to two different problems depending on whether we are in daytime or nighttime scenarios. At daytime images there are a number of features available to learn the appearance of a vehicle pattern [6]. However, it is difficult to reach detection distances beyond 150 m. Fortunately, this is a sufficient detection range for many driver assistance systems like stop–and–go or automatic cruise control. At nighttime, however, there are not as many features as at daytime, in fact, vehicles are not seen as a whole object since only their head and tail lights are perceived (Fig. 3). This lack of features is a challenge to distinguish mid to far distant vehicle lights from the own emitted light that is reflected in different infrastructure elements such as close traffic signs, fences, poles, etc. On the other hand, the fact of detecting the light emitted by the vehicles rather than the vehicles themselves makes possible to detect the presence of such targets at distances over 600 m for headlights and over 400 m for taillights. That detection range is mandatory for engineering intelligent headlight controllers. The reason is that, for instance, with our high beams on we could dazzle drivers that are closer than such distances.

Having stated the problem of nighttime vehicle detection, it is also worth to mention some realistic simplifications we assume:

- Under 70 km/h a lighting distribution similar to low beams is sufficient, which happens at many urban areas, approaching sharp curves, stops, etc.
- If we find streetlamps at the upper part of the image[1] we assume there is a good illumination and set low beams.
- Near the horizon it is difficult to discern between streetlamps and vehicle lights. This can be a problem for automatically switching on and off the high beams. However, our aim is to use an intelligent headlight controller as the one based on LEDs (Sect. 2). Thus, treating streetlamps near the horizon as vehicles would only prevent to illuminate a world region that is too far away but not the area needed with a good visibility for safe driving.

[1] The method to detect streetlamps is out of the scope of this paper.

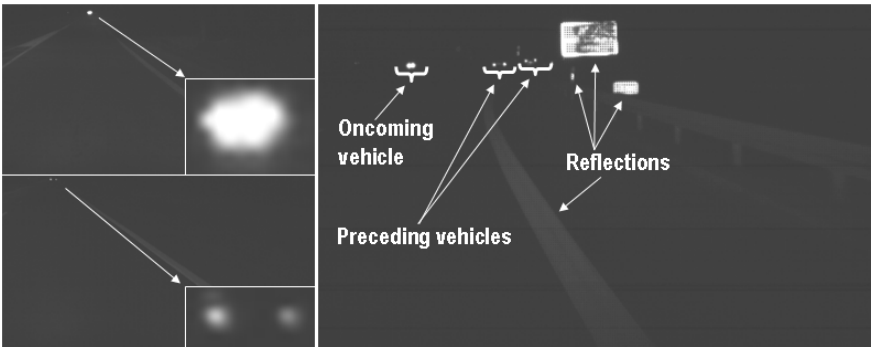

Fig. 3. *Left:* oncoming (*top*) and preceding (*bottom*) vehicles at a distance over 600 m and 400 m resp. At such distances the vehicles are near the horizon. *Right:* typical scene with oncoming and preceding vehicles as well as reflections from traffic signs. Oncoming vehicles are closer than 400 m and preceding ones are closer than 200 m.

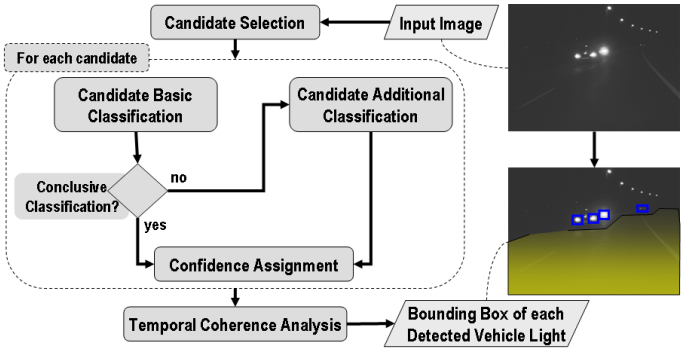

Fig. 4. Proposed vehicle detection method for nighttime images

Now we focus in the main difficulty, *i.e.*, to distinguish light emitted by other vehicles from reflections coming from infrastructure elements. Our proposal is based on both appearance and motion analysis. In this paper, however, we only focus on appearance, taking into account the following observations:

- The light spots captured by the camera are quite different for close vehicles than for far away ones.
- The same happens comparing head and tail lights, the former are supposed to be more intense and white while the latter are dim and reddish.
- For close vehicles the system must react faster than for far away ones.

With all these considerations in mind we have designed the processing method of Fig. 4. The details of each module are provided in the following.

4.1 Candidate Selection and Basic Classification

Thresholding and connected component analysis is used to obtain a set of blobs, *the candidates*, which must be later classified as *vehicles* or *non–vehicles*. It is worth to mention that the simplicity of this step is crucial for reaching real–time.

We term as *blob feature* to a single–valued measure that can be obtained by using image information from only the pixels that form the blob. Early work on this problem reveals that no single feature is sufficient for distinguishing vehicle blobs from others. There are some features expected to be especially useful like the maximum grey level, or some measure of reddishness, but they are not sufficient. Thus, our approach has been to define a set of potentially useful blob features and using a learning machine to obtain different classifiers.

Currently we use a set of features of different types: (1) binary (area, centroid, elongation, bounding box covering, main axes, etc.); (2) intensity from only monochrome pixels (maximum, mean, standard deviation, position of the maximum inside the blob, histogram skewness, mean gradient magnitude, etc.); (3) the same from only red pixels; and (4) features that try to provide color information by different comparisons between monochrome and red pixels (*e.g.*, the ratio between the mean of monochrome level and the mean of red level).

In [6] we successfully used a Real–AdaBoost learning machine for daytime vehicle detection, thus, it has been also our choice for nighttime. Briefly, given a set of labelled examples and counter–examples, each one described by a set of features, the Real–Adaboost algorithm provides a discriminative linear rule for classification. Given the features of a new sample, the rule yields a number whose sign indicates whether the sample is of the modelled class (+) or not (-), and whose absolute value acts as a degree of confidence on the classification.

We take into account evident differences in the vehicles' light class to split its variability beforehand by building different classifiers: small and non–small blobs look different, as well as head and tail lights. We denote by $\mathcal{P}_{h,s}$ the small blobs labelled as headlights, $\mathcal{P}_{h,ns}$ the non–small blobs labelled as headlights. $\mathcal{P}_{t,s}$ and $\mathcal{P}_{t,ns}$ are the analogous for taillights. Thus, assuming those sets of examples and being $\mathcal{N}_{h,s}, \mathcal{N}_{h,ns}, \mathcal{N}_{t,s}$ and $\mathcal{N}_{t,ns}$ the corresponding sets of labelled counter–examples, four classifiers are learned: $\mathcal{C}_{h,s}, \mathcal{C}_{h,ns}, \mathcal{C}_{t,s}$ and $\mathcal{C}_{t,ns}$.

Now, let's denote by $\mathcal{N}$ the set of all the labelled counter–examples. Since discriminative machines like Real–Adaboost generalize better by using examples and counter–examples around the discriminative frontier, for learning $\mathcal{C}_{h,s}$ we only want to use those counter–examples in $\mathcal{N}$ that are actually similar to small headlights, the set of such counter–examples is what we have previously denoted as $\mathcal{N}_{h,s}$. The same applies for $\mathcal{C}_{h,ns}$ with $\mathcal{N}_{h,ns}$, and so on.

We have devised the following procedure to obtain $\mathcal{N}_{h,s}, \mathcal{N}_{h,ns}, \mathcal{N}_{t,s}$ and $\mathcal{N}_{t,ns}$. First, given the labelled examples, we learn the following classifiers:

- $\mathcal{C}_{h,t,s}$: by using $\mathcal{P}_{h,s}$ as examples and $\mathcal{P}_{t,s}$ as counter–examples.
- $\mathcal{C}_{h,t,ns}$: from $\mathcal{P}_{h,ns}$ and $\mathcal{P}_{t,ns}$.

With $\mathcal{C}_{h,t,s}$ we classify the small blobs included in $\mathcal{N}$ as belonging to $\mathcal{N}_{h,s}$ if the classifier yields a positive value or to $\mathcal{N}_{t,s}$ otherwise. For non–small blobs the process is analogous, *i.e.*, the classifier $\mathcal{C}_{h,t,ns}$ is used to build $\mathcal{N}_{h,ns}$ and $\mathcal{N}_{t,ns}$.

During execution only the four classifiers $\mathcal{C}_{h,s}, \mathcal{C}_{h,ns}, \mathcal{C}_{t,s}$ and $\mathcal{C}_{t,ns}$ are applied according to the set of features of each blob, then, following a conservative approach (*better a false vehicle than missing a true one*) we take the maximum value of the four outputs. Therefore, at the end of this process each blob has a classification value assigned: positive values indicate that the blob resembles a vehicle light and negative values just the opposite. However, since no classification process is perfect for doubtful cases, we discretize the classification values so that they are translated to pre–defined weights. Let c be the classification value, then the corresponding weight w is assigned following:

$$\omega = \begin{cases} \omega_+ & \text{if} \quad c \geq t_+ \\ \omega_0 & \text{if} \quad t_0 \leq c < t_+ \\ \omega_- & \text{if} \quad t_- \leq c < t_0 \\ 0 & \text{if} \quad c < t_- \end{cases} , \tag{1}$$

where the t_+, t_0 and t_- are thresholds that must be set for each classifier, while the ω_+, ω_0 and ω_- are the corresponding weights that must be also defined for each classifier. Such weights will be posteriorly used to assign a confidence value to each blob (Sect. 4.3). Over t_+ we are confident about classifying a blob as vehicle light, and below t_- about classifying a blob as non–vehicle light. We consider the range from t_- to t_+ as a non conclusive output of the classifier. From t_- to t_0 the blob is assumed to be more similar to a non–vehicle light while from t_0 to t_- it is assumed to be more similar to a vehicle light.

4.2 Candidate Additional Classification

For those candidates such that the basic classification process has not been conclusive, *i.e.*, $t_- \leq c < t_+$, we propose the use of additional features such that are not sufficiently generic as to be part of the basic feature set but that can help when computable. One example we are currently using are what we call *pairing* features. Given a blob, the pairing features provide cues about the presence of a *twin blob*. The idea is that most vehicles have left and right head and tail lights, thus, there are many chances of observing the corresponding two bright spots in the image. Of course, this is true for vehicles like cars, but not for motorbikes. Therefore, we can not use such features as part of the basic set.

More specifically, for each blob b a window of a size proportional to its bounding box is placed to its left and to its right. Inside each window (left and right) we search for other blobs whose centroid is included in the window. When such a possible twin blob is found, namely b^t, some of its basic features are compared to the ones of b. Each comparison consists in computing a ratio, for instance, the ratio of the maximum grey levels of b and b^t. Such ratios form a set of features that is appended with others like distance between centroids, absolute maximum grey value, etc. Currently, all them form a set of features that is used for learning a classifier with the Real–Adaboost machine. Again, we could define more than one classifier of this type.

During execution the candidates for which the basic classification is not conclusive are submitted to the pairing classification. The new classification result

is used to modify the weight ω that the blob obtained during the basic classification. Currently, our approach consists in the following rule that we have found quite useful: if the pairing classifier yields a negative value (no twin blob) we do not modify ω, otherwise this weight is promoted to its upper possibility, *i.e.*,

$$\omega_- \longrightarrow \omega_0 \text{ or } \omega_0 \longrightarrow \omega_+ \ . \tag{2}$$

4.3 Confidence Assignment

At this stage we assign to each candidate a degree of resemblance to a vehicle light that we call *confidence* or just υ. Let M_g be the maximum possible grey level of the images, and g the maximum grey level reached by a candidate after normalizing by M_g. Then as confidence for that candidate we propose:

$$\upsilon = \omega \times g \ , \tag{3}$$

where ω is the weight assigned to the candidate by the classification process, *i.e.*, following Eq. (1), perhaps rectified by Eq. (2). Thus, we follow the scheme *confidence = classification evidence* $\times$ *physical evidence*. Close vehicles are expected to produce brighter blobs in the image than far away ones. Hence, blobs classified as vehicle light and corresponding to close vehicles will have a very high confidence. If the vehicle is further away, the confidence will be high but lower. A bright blob coming from a reflection can have a not too high confidence if it is properly classified, and such a confidence will be even lower if the reflection comes from an infrastructure element which is far away.

4.4 Temporal Coherence Analysis

The current system version assumes no tracking of candidates. Thus, to determine if the confidence assignments are coherent in time, we propose to use an *accumulation array* as well as a *state array*, namely **A** and **S** respectively. In [7] there is a more detailed description of the algorithm, but here we outline the main steps. **A** and **S** are updated so that we perform a kind of two–dimensional hysteresis as temporal coherence criterion: **A** keeps the hysteresis value (ranging from 0 to a given real number $M_{\mathbf{A}}$) and **S** the hysteresis states (*true* or *false*). At each frame, the temporal coherence analysis only considers the set of blobs not fully distrusted by classifiers, *i.e.*, blobs with associated positive confidence. At first frame we start with $<\mathbf{A},\mathbf{S}> \leftarrow <\mathbf{0},\mathbf{False}>$. For a new frame the temporal coherence analysis is done following these main ideas:

1. **Clean A and S.** Let $\mathcal{C}$ be the set of coordinates included in the bounding boxes of the blobs classified as *vehicle* at previous frame. All the cells of **A** out of $\mathcal{C}$ are set to 0 and, analogously, in **S** are set to *false*.
2. **Decrease A.** A decay of accumulation is done applying $\mathbf{A} \leftarrow \max(0, \mathbf{A} - d)$, where d is a fixed real number that controls the decay ratio: starting at $M_{\mathbf{A}}$, $M_{\mathbf{A}}/d$ steps are required to reach 0. In fact, at different cells of **A**, d can take different values out of two possibilities, namely $d = d_t$ if the state in **S** associated to the cell equals *true* and $d = d_f$ otherwise.

3. **Spread A and S.** With the aim of actually combining confidences coming from the same target from frame to frame, the values of $\mathbf{A}$ and $\mathbf{S}$ are *spread* following the expected motion of the targets.
4. **Increase A**. Let C_b be the set of coordinates that form the blob b, and $v_b > 0$ the blob confidence. Then, we apply the update $\mathbf{A}_{C_b} \leftarrow \min(\mathbf{A}_{C_b} + v_b, M_\mathbf{A})$, where $\mathbf{A}_{C_b}$ stands for the cells of $\mathbf{A}$ whose coordinates are in C_b.
5. **Hysteresis on S.** For each valid coordinate (i, j):
   - if $\mathbf{A}_{i,j} = 0$ then we set $\mathbf{S}_{i,j} \leftarrow$ *false* ;
   - if $\mathbf{A}_{i,j} \geq M_\mathbf{A}/2$ then $\mathbf{S}_{i,j} \leftarrow$ *true* ;
   - if $0 < \mathbf{A}_{i,j} < M_\mathbf{A}/2$ then $\mathbf{S}_{i,j}$ does not change its value.

After these steps, for each blob b with $v_b > 0$ we compute the *logic or* at $\mathbf{S}_{C_b}$, where $\mathbf{S}_{C_b}$ stands for the cells of $\mathbf{S}$ whose coordinates are in C_b. If the logic or yields *true* then b is classified as *vehicle* and as *non–vehicle* otherwise.

The most important step of this method is the *spread* operation. The basic idea consists in performing a kind of *dilation* similar to the one of grey level mathematical morphology, however, being the structuring element different at each accumulation cell. The difference accounts for the expected motion of the different targets in the image space: targets near the horizon stay quite static from frame to frame while the position of close ones varies more; oncoming vehicles move fast towards the image bottom while preceding vehicles do not. In fact, we use a pair $< \mathbf{A}, \mathbf{S} >$ for blobs that are more similar to headlights than to taillights and a different pair for the rest, being the pair selected according to the classifier that had the higher output during the basic classification stage.

4.5 System Settings

Candidate selection. In order not to miss far away targets a low threshold must be used, but it must not be too low if we do not want to have many irrelevant blobs that could slow down the processing. Currently we use $30\%M_g$.

Basic and additional classification. We drove through traffic ways close to Barcelona recording images. Since we used a standard vehicle, we manually switched on and off the high beams according to the traffic situation. We covered highways, secondary roads and city streets, with scenarios of dense traffic, only a few vehicles and no vehicles at all, and including many types of vehicles, traffic signs, fences and poles. From such a huge amount of data we labelled about 12000 headlights, 31000 taillights, 18500 traffic signs and 14000 poles/fence reflectors.

Candidate basic classification. After testing several criteria we found sufficiently useful to define *small blobs* as those with an area lower than 25 pixels. Equation (1) is set according to the thresholds and weights of Fig. 5*Right*. These thresholds have been set so that no samples are wrongly classified in the testing set (see Sect. 5 for details about this set). For setting the weights we have taken into account the performance of the classifiers using such thresholds. This performance, shown also in Fig. 5*Right*, suggests we set the weights trusting a lot the classifier for non–small headlights, *i.e.*, $\mathcal{C}_{h,ns}$, followed by $\mathcal{C}_{t,ns}, \mathcal{C}_{h,s}$ and $\mathcal{C}_{t,s}$, which confirms that the most challenging targets are far away (small) taillights.

Classification according to the sign of each classifier output		Properly classified (vehicles / non-vehicles)	Wrongly classified (vehicles / non-vehicles)
Taillights Classifiers	small	88 % / 89 %	12 % / 11 %
	non-small	97 % / 98 %	3 % / 2 %
	paring (small)	97 % / 95 %	3 % / 5 %
Headlights Classifiers	small	93 % / 92 %	7 % / 8 %
	non-small	99 % / 99 %	1 % / 1 %

For having zero wrongly classified samples	Headlights Classifiers		Taillights Classifiers	
	small	non-small	small	non-small
$(t, \omega)_+$	(1, 1)	(1, 1.5)	(1, 0.5)	(1, 1)
$(t, \omega)_0$	(0, 0.5)	(0, 1)	(0, 0.25)	(0, 0.25)
$(t, \omega)_-$	(-2, 0)	(-2, 0)	(-2, 0)	(-2, 0)
Properly classified (vehicles / non-vehicles)	78 % / 75 %	97 % / 97 %	60 % / 65 %	94 % / 93 %
Non conclusive (vehicles / non-vehicles)	22 % / 25 %	3 % / 3 %	40 % / 35 %	6 % / 7 %

Fig. 5. *Left:* classifiers performance according to the sign of their output. *Right:* settings for Eq. (1) so that no samples of the testing set are wrongly classified. The performance using these thresholds (t) is still over 90% for non–small blobs, thus, the associated weights (w) employed for the temporal hysteresis are *extreme*: 0 for blobs given as *non–vehicle* and ≥ 1 for the ones given as *vehicle*. Small blobs require more conservative weights, *i.e.*, confirming a blob as *vehicle* requires a few frames, which is acceptable because they correspond most often to targets that are far away.

Candidate additional classification. Given the performance of the basic classifiers (Fig. 5*Left*), we decided to use just one pairing classifier to reinforce the classification decision regarding small taillights, *i.e.*, the most difficult targets.

Temporal coherence analysis. We have set $M_{\mathbf{A}} \leftarrow 2$, thus, the hysteresis associated to a cell of an accumulation array reaches the state of *true* over $M_{\mathbf{A}}/2 = 1$ and it is not *false* until reaching the null value again. The decay control is set as $< d_t, d_f > \leftarrow < 45, 15 >$ frames, which means that when a vehicle disappears our system illuminates the new free area in about 2 s.

5 Results

Currently we are using testing sequences to quantitatively evaluate the performance of both the classifiers alone and the whole system. These sequences were acquired around Wolfsburg, which implies that the testing set comes from a different camera and vehicle than the training set. Besides, this vehicle was progressively using preliminary versions of our vehicle detector to control a headlight system with variable cut–off line. Thus, in the same images we have lights from vehicles and reflections from infrastructure elements, something that is not possible by only turning on the high beams when no other vehicles are present.

In order to evaluate the performance of the classifiers, we labelled a number of samples from the testing sequences of the same order of magnitude than the mentioned for training. Using the sign of each classifier output as threshold we obtain the performance shown in Fig. 5*Left*. We think such a performance is quite high since it is over the 90% for both vehicles and non–vehicles, except for the case of far away (small) taillights where the performance is just slightly below the 90%. This last case is compensated by the good performance obtained using the pairing classifier, which we think will be often applicable.

Regarding the performance of the whole system now we can only provide a qualitative impression since we are still working in the quantitative analysis, which requires the labelling of many full sequences. After visually examining many testing sequences employing the settings explained in Sect. 4.5, the impression we have is that oncoming vehicles are mostly detected in a single frame if they are at distances under 300 m and in two or three frames otherwise. For taillights two or three frames are needed at distances under 300 m and three to five otherwise. Notice that when a vehicle appears the overall reaction time of the headlight control should be less than half a second. This implies that the vehicle detection should be within about 200 ms because of the slow actuators in some headlamps. Then, for a case when five frames are required, we must process each frame in less than 40 ms. This requirement is satisfied by our current C++ implementation since it takes less than 20 ms using a 2 GHz Pentium Mobile.

6 Conclusion

We have presented a novel proposal for real–time vehicle detection at nighttime, fulfilling the requirements for an intelligent headlight controller as well as without restricting the incorporation of additional driver assistance applications. The main challenge is to discern between image spots actually due to vehicle lights and those coming from reflections in different infrastructure elements. To confront that challenge we have used an image sensor with a novel clear–red pattern, and we have proposed a new classifiers–based architecture and temporal coherence analysis that are able to reach quite high detection rates for very low false positives ones, in a way that the intelligent headlight controller could react from one single frame for targets that are clear vehicle lights, or from only a few frames for targets whose type is more difficult to discern most often because they are small dim spots. Future work will focus on increasing the classification performance for such difficult targets investigating the use of new features.

References

1. U.S. National Highway Traffic Safety Admin. Traffic safety facts (2000)
2. Michigan Univ. Transportation Research Inst. Use of high-beam headlamps (2006)
3. Schaudel, C., Falb, D.: Smartbeam — a high–beam assist. In: Proc. of the 7th International Symposium on Automotive Lighting, Darmstadt, Germany (2007)
4. Könning, T., Amsel, C., Hoffmann, I.: Light has to go where it is needed: Future light based driver assistance systems. In: Proc. of the 7th International Symposium on Automotive Lighting, Darmstadt, Germany (2007)
5. Hummel, B.: The future of the LED headlamp (in German). In: Haus der Technik - LED in der Lichttechnik, Essen, Germany (2008)
6. Ponsa, D., López, A.: Cascade of classifiers for vehicle detection. In: Blanc-Talon, J., Philips, W., Popescu, D., Scheunders, P. (eds.) ACIVS 2007. LNCS, vol. 4678, pp. 980–989. Springer, Heidelberg (2007)
7. López, A., Hilgenstock, J., Busse, A., Baldrich, R., Lumbreras, F., Serrat, J.: Temporal coherence analysis for intelligent headlight control. In: 2nd Workshop on Planning, Perception and Navigation for Intelligent Vehicles, Nice, France (2008)