A Monocular Vision Based Advanced Lighting Automation System for Driving Assistance

Julien Rebut, Benazouz Bradai, Julien Moizard, Adrien Charpentier
VALEO, Driving Assistance Domain, FRANCE
{julien.rebut, benazouz.bradai, julien.moizard, adrien.charpentier}@valeo.com

***Abstract*—This paper presents a night time vehicle detection system performing automatic headlamp control in the frame work of driving assistance. In such application dealing with sensor, processing and actuator, we focus on image processing techniques developed in this project. From our embedded camera, image processing enables to detect vehicles ahead and estimates their positions in order to increase driver visibility by adjusting headlamps. We review algorithms (segmentation, classification, tracking and position estimation) in detail and present results comparing driver dazzling between static headlamps and intelligent headlamps. Our system detection range is above 600m for headlamps and about 400m for tail lamps which is sufficient to avoid glaring of other road users. Classification performances are above 97% of true positive rate evaluated on a validation database (frame by frame detection). The final vehicle detection is guaranteed at 100% of recognition by attributing a minimum confidence accumulated over successive fames. By way of conclusion, we introduce perspective of advanced lighting automation.**

I. Introduction

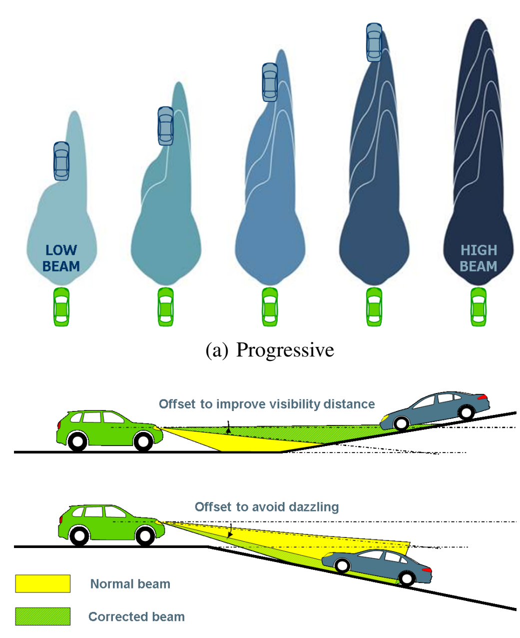
Lighting automation (high/low beam) is a widely studied subject for about 5 years now. It's a driving assistance problem having an interest in terms of comfort but also safety as 55% of the driving fatalities occur at night while it represents only 25% of the driving period ([1]). Typically, a car at 90kph needs about 110m to be fully stopped. A low beam light does not provide enough visibility range (70m) to stop the car if an obstacle appears at 70m whereas a high beam lights up to 200m. In [2], the University of Michigan Transportation Research Institute has shown that high beam is used at only 25% when it is possible, driver seems to forget to switch from low to high beam or is bored by incessant switch manipulations. This figure could be greatly reduced by innovative technologies such as camera-assisted lighting automation.

Gentex Coorp. ([3]) has been the first automotive supplier in serial production on lighting automation system called Smart Beam on Lincoln Mercury in 2004 in United States and on BMW in Europe since 2007. Now, several other suppliers such as MobileEye ([4]) and Valeo ([5], [6]) offer such system. In [7], authors present a night time vehicle detection algorithm that performs lighting automation. From a B&W camera, bright blobs (headlamp and tail lamp) are extracted using an adaptive thresholding, tracked with a Kalman filter and finally classified (distinguishing between traffic sign and vehicle lighting) using a Support Vector Machine (SVM). Headlamps are detected between 300m and 500m while tail lamps detection range is poor and below 80m.

This kind of system uses imaging sensor and image processing algorithms to detect the vehicles ahead and monitor road environment. It has to detect oncoming traffic, leading traffic and city environment without be confused with traffic signs. Based on Valeo commercialized headlamp, we defined the minimum detection range at 350m which is the distance where an oncoming or leading vehicle starts to be dazzled by our high beam. In terms of acceptance of the function by the driver, we consider a minimum detection range of 600m for oncoming traffic even if at such distance high beam does not glare.

Valeo has designed 2 levels of automation:

- *BeamAtic*®: it is the core module which detects traffic and monitor road surrounding. Based on these information, function controls headlamp in low or high beam position.
- *BeamAtic Plus*®*:* it is an added layer to *BeamAtic*®. Indeed, vehicle lights are previously detected. Then, one has to build vehicles based on lights, estimates their distances and vertical angles. It controls continuously headlamp range (from low beam to high beam, see figure 1a) and leveling (see figure 1b) according to the position (distance and vertical angle) of the closer vehicle ahead.



(a) Progressive

(b) Leveling

Figure 1: BeamAtic Plus

In this paper, we present work done during the last 4 years on lighting automation at Valeo. The paper is organized as follows: in section II, we detail the algorithms developed to detect, classify and track bright objects corresponding to the vehicle lights. Then, we focus on vehicle building and position estimation (distance and angle) needed to perform *BeamAtic Plus®* function. Finally we present in section III results of a case study. We compare the glaring of a car followed by another one in low beam mode or in automatic mode (intelligent headlights).

II. Algorithm

Such systems operate exclusively at night time condition. It has to identify all vehicles ahead of the ego car in order to adapt headlamp beam. By night, the main criterion to detect a vehicle is the bright spot generated by a headlight and a tail light. Once the object is detected, a classification procedure enables to keep the relevant lights such as headlamps and tail lamps and removes the irrelevant objects such as reflections on road signs or reflexes bounding the road. This is one of the main tasks to distinguish a reflection from a relevant light because it has a similar intensity than a headlamp as it is generated by ego car lighting. Second part is focused on *BeamAtic Plus®* function which consists in building some vehicles from detected lights and in estimating their positions: distance and vertical angle. The distance estimation is difficult to achieve with enough accuracy in mono-vision. Moreover, angle estimation depends on estimated distance; that's why estimation has to be precise enough to prevent any angle error.

Figure 2 shows the overall application diagram. Only the core modules (in green) are necessary to perform *BeamAtic®* while other stages (in orange) are necessary to build *BeamAtic Plus®* system.

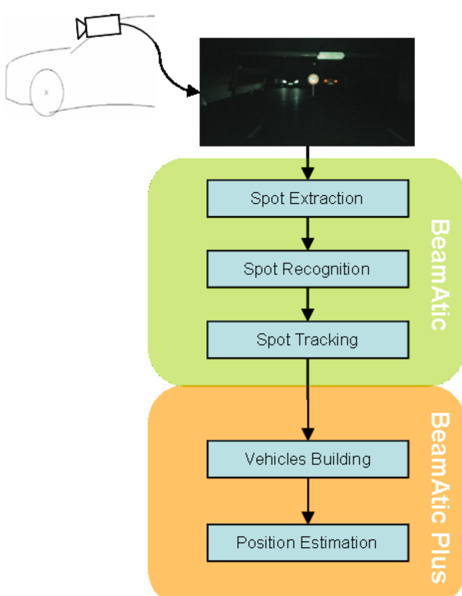


Figure 2: System Diagram

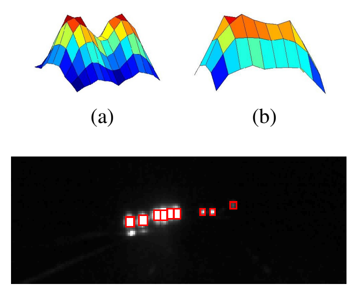
A. Spot Extraction

Image segmentation to extract vehicle lights could be simple as background is dark and lights are similar to bright spots. In such case, a threshold should be sufficient to extract such objects. Unfortunately, a road scene is not so simple. First of all, headlamps and tail lamps have different light intensity levels which fluctuate according to distance from camera and light power (dirty...). Additionally, background is not always uniformly dark and its intensity can be as bright as very distant tail lamps. In some complex scenes such as highway environment, there are many cars which can be in staggered rows. Image does not contain well separated objects but rather a Gaussian mixture[1].

This step is crucial for further processing especially pattern recognition which analyzes deeper content of lights. The extracted objects have to be well delimited. For all of these reasons, we apply first a local maximum algorithm to find at least one pixel belonging to each spot and then a growing process to find objects boundaries.

Local maximum: Our local maximum algorithm extracts saturated pixels or pixels for which region of interest around has a pyramid shape. Connected components are clustered to obtain one label per object.

Growing process: The boundaries of an object have to be finely extracted to enable further pattern recognition analysis. To do so, the algorithm starts from the gravity center of a previously extracted label (containing a local maximum) and extends its boundaries by scanning in four 1D directions (left, right, top and bottom) the gray level profile. Figure 3 shows spots extraction in simple and difficult cases.



(a) (b)

(c)

Figure 3: Spots Extraction (a & b respectively simple and difficult cases, c is the final result)

B. Spot Recognition

The goal of this step is to suppress at the end irrelevant objects such as traffic signs, reflections on road.... Moreover, it has to classify the kind of detection between *headlamp* and *tail lamp* and to provide a confidence level. A typical road scene is composed of an oncoming vehicle, a leading vehicle and a back reflection of the ego beam on a traffic sign. Indeed, this kind of system is usually confused by the reflections of the ego beam on reflective areas which have the same intensity

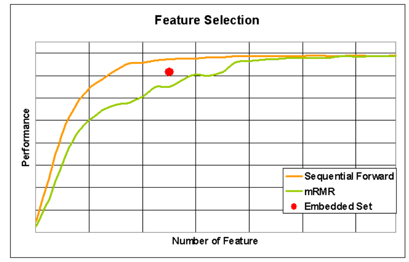
[1]A light is assumed to follow a Gaussian distribution

than the ego beam. This effect is accentuated even in high beam.

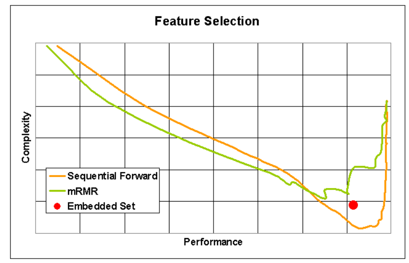
Recognition task is carried out by a SVM classifier (Support Vector Machine [10]). The extracted 2D objects in the picture have to be represented by some descriptors (also called feature) to feed the classifier. A lot of features have been studied to characterize a light. We have started with an exhaustive list of features commonly used to describe an object in terms of intensity, shape, texture, information... Data have been collected on several hundred of kilometers and in various conditions to build a headlamp/tail lamp database. Keeping in mind that the final application has to run on an embedded hardware, the only relevant features, without redundancy, have to be kept. This is one of the interests of feature selection method. Feature selection consists in finding a subset of features that minimizes classification error. This procedure reduces the cost of recognition by reducing the number of features and leads to a better classification accuracy (see [8] for an evaluation of feature selection methods). Our feature selection method has been mostly inspired from work in [9]. It is a forward procedure: variables are selected one by one. Evaluation of the feature relevance is based on cross-validation classification error of our SVM classifier. Lights database is divided into a training database of N samples and a validation database of L samples. Let's consider M features $X = \{x_i, i = 1 \ldots M\}$ and $S_{k=1\ldots M}$ a set of k features. Initially, the feature set is empty $S_k = \{\emptyset\}$ and classification error e_k is set to 100%. First search consists in finding the first feature x_1^*, denoted Z_1, (where * denotes the optimal feature) from X that leads to the largest error reduction. Then, from the set $\{X - Z_1\}$, second search selects the second feature x_2^* where $Z_2 = \{Z_1, x_2^*\}$ which minimizes again the validation error. This sequential search is repeated until X is empty. At the end, the final features set S_m of m features is the one for which cross-validation error improvement is no significant regarding increasing of number of vector support (complexity). Figure 4 shows a comparison of our algorithm with the mRMR selection method (max-Relevance and Min-Redundancy [9])[2]. Basically, the algorithms have the same trend. They decrease the number of features by a factor of 3. Our algorithm provides better performances on this specific problem: better recognition rate of 2.5% while complexity reduced by 30% (less support vector). But these conclusions should be checked on a larger database. Moreover, we do not consider here execution time as algorithm is done in an off-line manner. Finally, for real-time constraints on our embedded hardware, we had to replace some of selected features by others less time consuming providing performances represented by the red dot.

Table I shows headlamps and tail lamps recognition rates on training and validation data set. Recognition rate is higher than 96%. For security reasons, we attribute a minimum confidence level to avoid any missed detection.

[2]HTTP://research.janelia.org/ peng/proj/mRMR/index.htm



(a) Perf. vs. feat. number



(b) Complexity vs. perf.

Figure 4: Feature Selection

	Recognition rate	
	Learning	Validation
Headlamp	99.9%	96.8%
Tail lamp	100%	97.7%

Table I: Classifier Recognition Rate

C. *Spot Tracking*

Tracking has several functions. First, it has to temporal match spots over successive frames. This is necessary to adjust the confidence of a light. Once it has reached a sufficient level, the light is considered relevant. Secondly, tracking enables to analyze trajectory of moving spots. The trajectory enables to discriminate false detections and adjust the reaction time of the system (in case of overtaking for instance). We use the well known Kalman Filter [11] to estimate the position $(u_{prediction}, v_{prediction})$ of a spot at frame t based on information collected on preceding frames. Then, a shortest path algorithm is used to match detections at t with predicted ones at $t-1$ in finding the best global solution. This is necessary to avoid bad matching appearing especially in curve as illustrated on the figure 5.

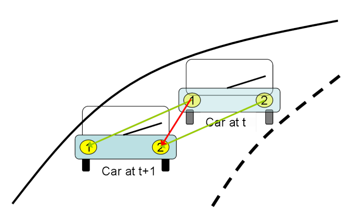


Figure 5: Tracking (green and red mean respectively good and bad matching)

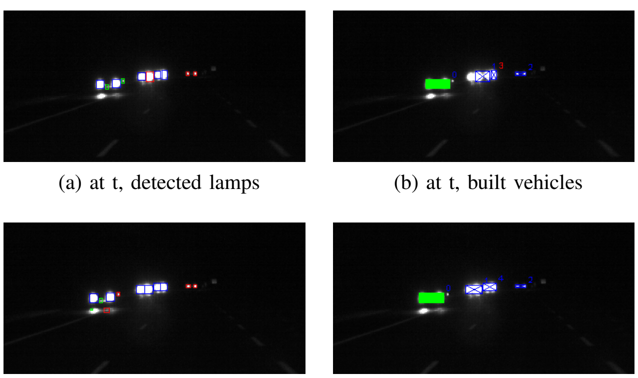
At this step of the algorithm, the software has sufficient knowledge to perform high/low beam switching. However, a dedicated strategy module is required to adjust system behavior such as deactivation and reactivation time according to the situation ahead. In part III, we provide performances of automatic high/low beam function called *BeamAtic*™.

D. Vehicle Building

A vehicle can be a car or a motorbike which is respectively built based on 2 lamps or 1 lamp. Thus, a car is built by the association of two lights which should be similar. This similarity notion is assessed by a cost function defined based on features selected before (see II-B):

$$cost = \frac{1}{N}\sum_{i=1}^{n}\left(x_i^{left} - x_i^{right}\right)^2,\ x_i \in Z_n$$

However, we can't guarantee to not have a bad association of lamps due to the scene complexity. Let's consider now a road scene (see fig. 6) which contains 3 oncoming vehicles and one leading vehicle. At a certain time t, headlamps of the 3 oncoming vehicles are not all visible due to partial occlusions. With lack of information at t, lamps association module build some wrong cars in such situation. As a consequence, distance and angle are over or under estimated. In other words, it could glare other road users or degrade driver visibility. Moreover, due to the tracking step, a wrong association could remain tracked during a long time. That's why the system is very restrictive and re-evaluates every frame all existing vehicles.



(a) at t, detected lamps

(b) at t, built vehicles

(c) at t+1, detected lamps

(d) at t+1, built vehicles

Figure 6: Vehicle Building

The spots not associated before are considered as motorbike (vehicle equipped of only one lamp). Note that these vehicles do not always correspond in the reality to a motorbike due to partial occlusions.

Once vehicles are built and identified as car or motorbike, next step consists in estimating their position.

E. Position Estimation

The position of a vehicle is defined by its distance from the ego car and its vertical angle in the headlamp basis. The distance is necessary to adjust beam range and the vertical angle is necessary to adjust beam leveling.

1) Car distance estimation: A car is composed of two lamps. Its distance estimation is based on the assumption of its real width. Measuring the number of pixels separating the lamps and knowing the real width, the distance can be easily estimated.

From the number n of pixels between the lamps, we calculate the distance as follows:

$$d = \frac{w \times f}{n \times P_x}$$

Where:

- d is the distance in meter
- f is the camera focal length in meter
- w is the real width hypothesis in meter
- P_x in the camera pixel size in meter

Figure 7 compares distance estimation with the ground truth (from 300m to 0m).

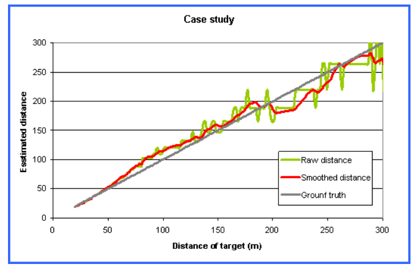


Figure 7: Distance function of width hypothesis

It is important to note that 1 pixel error in the width estimation has no consequence at 20m but can lead to a 40m error at 200m. To limit the effects of such an error, we estimate the gap between the lamps with a sub pixel resolution.

Concerning the real width hypothesis, we made an average based on several vehicles. If the detected vehicle is a Smart or a truck, the distance will be either over or under estimated.

2) Motorbike distance estimation: The motorbike distance estimation is a tricky problem because of the assumption used for a car is not applicable for a motorbike. Unfortunately, there are not many criteria to evaluate the distance of an object. Distance can be estimated based on vertical position (estimator 1). The issue is that this distance is not accurate, depends on body car pitch and is no more suitable further 70m! Then, a relation between object intensity and distance can be determined (estimator 2).

Estimator 1, vertical position: Doing the flat world hypothesis, the vertical position of an object in the picture can be converted, knowing the exact camera parameters, to a distance using the following formula:

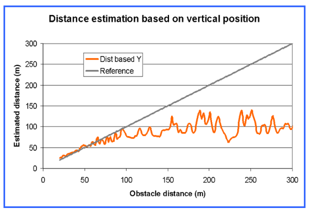
$$d = \frac{H_{cam}}{\tan\left[\theta \arctan\left(\frac{Y_{pix} \times P_x}{f}\right)\right]} \times \left(1 - \frac{H_{obst}}{H_{cam}}\right)$$

where

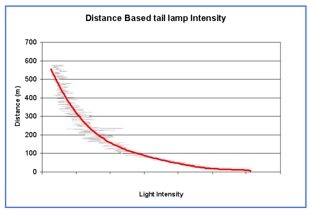
- H_{cam} is the camera height
- H_{obst} is obstacle lamp height
- θ is the camera inclination angle
- Y_{pix} is the object vertical position in the picture (pixel)

Figure 8a shows distance estimation compared with real values. Distance has been corrected using the vehicle pitch. This estimation provides acceptable results below 70m. Note that this analysis has to be nuanced by the height hypothesis! Indeed, the only part of the car which is visible in the picture at night time is its lamps. Lamp height H_{obst} has to be taken into account in the distance formula. A bad tail lamp height hypothesis of about 40cm can lead to a distance error of about 50%!

Estimator 2, intensity: First estimator does not provide accurate results further 70m. To extent the estimation range provided by the first estimator, the light intensity is used as a rough estimation. Of course, the reliability is contestable as the light intensity received by the imaging sensor depends in a large part of the lamp by itself. But at least, it provides a rough estimation. Indeed, a car far away from the camera provides a lower intensity on the sensor than a car closer. Figure 8b shows an example of leading car distance estimation.



(a) Estimator 1



(b) Estimator 2

Figure 8: Motorbike distance estimation

3) Distance filtering: Estimators introduced before provide distances quite unstable from frame to frame. Indeed, the light intensity fluctuates as well as vertical position and gap between two lamps. Moreover, at far distance, a small variation on the assessed value leads to chopped distance estimation. For instance, 1 pixel error in the width estimation leads to 1m error at 30m and 40m error at 200m.

A Kalman filter is used to smooth these values. The system noise has been studied to be taken into account in this filter. Indeed, our estimators are quite accurate at near range and low accurate at far range. This has been introduced in the Kalman filter noise model. Figure 9 compares different filtering. The adaptive noise provides the best results.

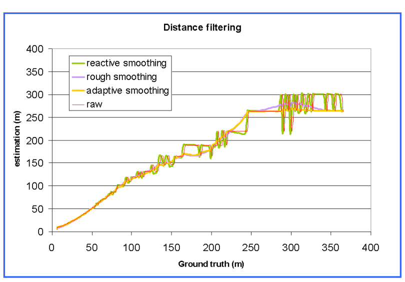


Figure 9: Distance filtering

4) Angle estimation: Vertical angle of each vehicle is necessary to adjust headlamp leveling. In the camera basis, the vertical angle is defined as follows:

$$\tan\left(\theta + \alpha_{cam}\right) = \frac{v}{f}$$

and

$$\frac{\tan(\theta)+\tan(\alpha_{cam})}{1-\tan(\theta)+\tan(\alpha_{cam})} = \frac{v}{f}$$

where

- θ is the camera inclination angle
- α_{cam} is the angle we aim to calculate
- v is the spot vertical position
- f is the focal length

With small angles approximations, we obtain the following relation:

$$\alpha_{cam} = \frac{v+f\times\theta}{v\times\theta-f}$$

As we can see on the figure 10, the vertical angle α_{cam} estimated in the camera basis is different from the one necessary to adjust the beam (α_{lamp}). The angle α_{cam} is projected from the camera basis to the headlamp basis and becomes α_{lamp}. To do this projection, the distance of considered vehicle needs to be known as expressed in the following relation:

$$\alpha_{lamp} = \frac{H_{cam}-H_{proj}+dist_{obst}\times\alpha_{cam}}{dist_{obst}-L}$$

where

- H_{cam} is the camera height
- H_{proj} is the headlamp height
- L is the distance between camera and headlamp

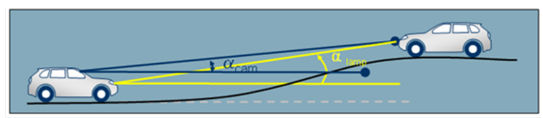


Figure 10: Angle Calculation

III. Results

Our system runs at 30 frames per second on a current computer and a version is also available on an embedded hardware (see fig. 11). The detection range is more than *600m* for oncoming traffic and about *400m* for leading traffic (it exceeds dazzling requirements introduced in I). These values are averaged because they depend of vehicle lighting quality, atmosphere attenuation and angle between camera and lamps. System reaction time depends of the situation ahead. It can switch from high to low beam in less than *300ms* and spends more time (till confidence sufficient) in case of very far vehicle. However, one guarantees to switch to low beam, even if the object is not recognized by the learning algorithm, by attributing a minimum confidence level.

Figure 11: Embedded camera

BeamAtic Plus® has same performances than *BeamAtic®* in terms of detection (range and reaction time) but it has to deal with distance and angle estimation. Indeed, the target is to enhance driver visibility by adjusting headlamp beam just below the bumper of the vehicle ahead without dazzling other road users. We have equipped a vehicle with *BeamAtic Plus*® and another one with a camera (fixed parameters) placed on the back window at the same height of the driver eyes and looking backward. The purpose of this camera is to assess the driver dazzling due to the vehicle behind (equipped or not of intelligent headlamps). Our test track is about 5.2km long and we present results on a hilly portion (blue segment, slope up to +/- 8%, see fig. 12).

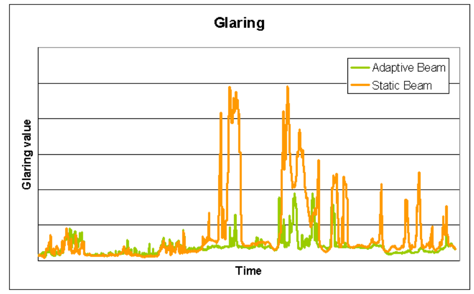
(a) Test track (b) Hilly landscape

Figure 12: Test Track

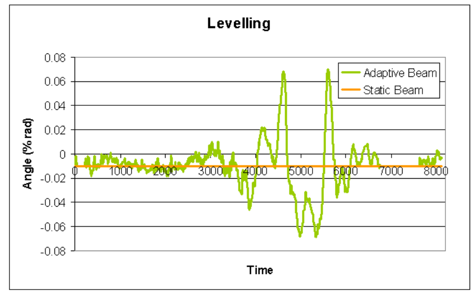
The manipulation consists in recording dazzling effect of a vehicle $V_{followed}$ followed with a car equipped or not of smart headlamp $V_{follower}$. The distance between cars fluctuates from 50m up to 200m. The recorded data are dazzling values from $V_{followed}$ and headlamp parameters (range and angle) of $V_{follower}$. One record has been made with $V_{follower}$ in low beam and another one with $V_{follower}$ in automatic mode. Figure 13 shows results. $V_{followed}$ is much less glared by $V_{follower}$ (Fig. 13a) when it is in automatic mode while $V_{follower}$ has an extended visibility (Fig. 13c) compared with a classical low beam range (60m). Figure 13b shows headlamp angle on this track which is centered on classical -1% radian low but which can reach $\pm 8\%$ in hilly section.

IV. Conclusions

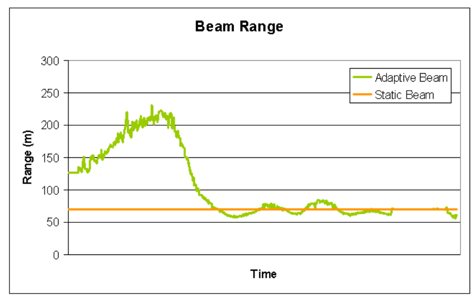
In this paper we have presented a night time vehicle detection software performing automatic headlamp beam adjustment. It is declined in two versions: one performs automatic On/Off switching (*BeamAtic*®) and the other one adapts the headlamp *range* and *leveling* to extend driver visibility up to car ahead (*BeamAtic Plus*®). It has been shown that such system reduce dazzling effect compared with static headlamp in hilly situations while it increases visibility. Our next development is called *BeamAtic Premium*®. It controls the headlamp over three axes: *range*, *leveling* and *bending*. It keeps the high beam lights at all times with only blanking out the zones in which on coming or preceding vehicles are located.



(a) Glaring measurement



(b) Headlamp angle



(c) Headlamp range

Figure 13: *BeamAtic Plus*® Results

References

[1] Dubrovin A., Lelevé J., Prevost A., Canry M., Cherfan S., Lecoq P., Kelada J. M., Kemeny A., “Application of real-time lighting simulation for intelligent front-lighting studies”, Proceedings of the Driving Simulation Conference, pp.333-343, Paris, September 2000.

[2] Sullivan, J.M., Adachi, G., Mefford, M.L., Flannagan, M.J. 2004, “High-Beam Headlamp Usage on Unlighted Rural Roadways”, Lighting Research and Technology, vol. 36, no. 1 (2004) p. 59-67.

[3] Joseph S. Stam, Gentex Corp, “Automatic vehicle high-beam headlamp control system”, SAE 2001 World Congress, March 2001, Detroit, MI, USA.

[4] Mobileye, Adaptive Headlight Control, [Online] Available: http://www.mobileye-vision.com

[5] B. Reiss, J. Moizard, “Advanced high beam/ low beam transitions: progressive beam and predictive leveling”, V.I.S.I.O.N 2008, October 7 & 8, Versailles Satory, France.

[6] P. Reilhac, J. Moizard, M. Grimm, B. Reiss, “Valeo innovative lighting systems Enhancing road safety”, ATZ, 03/2008, p. 210-217.

[7] P.F. Alcantarilla, L.M. Bergasa, P. Jiménez, M.A. Sotelo, I. Parra, D. Fernandez, S.S. Mayoral, “Night Time Vehicle Detection for Driving Assistance LightBeam Controller”, Intelligent Vehicles Symposium, IV2008, Eindhoven, June 4-6, 2008, p.p. 291-296.

[8] Jain A., Zongker D., ”Feature Selection: Evaluation, Application, and Small Sample Performance”, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 19, no. 2, pp. 153-158, Feb. 1997.

[9] H. Peng, F. Long, C. Ding, ”Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy”, Pattern Analysis and Machine Intelligence, IEEE Transactions on Volume 27, Issue 8, Aug. 2005 Page(s):1226 - 1238.

[10] Bernhard E. Boser, Isabelle M. Guyon, Vladimir N. Vapnik, “A Training Algorithm for Optimal Margin Classifiers”, in Fifth Annual Workshop on Computational Learning Theory, pages 144–152, Pittsburgh, ACM, 1992.

[11] Kalman R. E., “A New Approach to Linear Filtering and Prediction Problems”, Transaction of the ASME—Journal of Basic Engineering, 82(Series D), 35-45.