Auto-iris Compensation for Traffic Surveillance Systems

Rita Cucchiara, Rudy Melli, Andrea Prati

***Abstract*— This paper addresses auto-iris compensation. Auto-iris can be really troublesome for motion detection and tracking techniques based on background or frame differencing, since it can change quickly the average intensity of the current frame. To cope with this, we introduced a two-step auto-iris compensation approach in our traffic monitoring system. First, the auto-iris detection is based on the computation of the average of the luminance difference obtained by background suppression. Then, if an auto-iris is detected, the compensation phase is started. In this phase, the auto-iris' behaviour is empirically modelled and, thus, compensated. Experimental results demonstrate the accuracy of the proposed approach, with both quantitative measures and visual analysis.**

I. INTRODUCTION

Vision-based traffic surveillance systems are nowadays very diffuse. This has been made possible by the recent development in hardware and software for image processing that leaded, on the one hand, to cheaper cameras and acquisition devices, and, on the other hand, to the advances in image processing and computer vision techniques, capable to become more reliable and efficient. There are, however, many cases in which computer vision techniques are likely to fail in automatic traffic surveillance. These cases are frequent in real conditions, when a system is supposed to work on a 24/7 basis. Some examples are: working during night or in extreme weather conditions, such as fog or snow; working in case of heavy traffic and with traffic in both directions; handling occlusions due to other vehicles, trees, poles, and so on; detecting high-speed vehicles; being robust to both natural (e.g., due to clouds occluding the sun) and artificial (e.g., due to camera auto-iris) illumination changes; etc.

This work is the result of a collaboration between one of the leading companies in vision-based traffic surveillance, Traficon N.V. (Belgium), and our group at University of Modena and Reggio Emilia (Italy). This collaboration brings together the expertise of Traficon in making these systems working in real life and our expertise in tracking and segmentation of moving objects, with the aim of providing reliable solutions to a real system. This project has the final aim of implementing on a DSP board (based on the TMS320DM642 Video/Imaging Fixed-Point DSP) advanced algorithms for traffic surveillance. Unfortunately, the computational resources of the DSP used are limited: in particular, the lack of a floating-point unit makes many operations inefficient. Moreover, the DSP board is used also for many other tasks (video communication, trip-wire emulation, and so on), further reducing the availability of resources for our implementation. This limit, together with the request to work in almost every real condition makes our job a great challenge.

Existing commercial solutions rarely exploit temporal information of the scene: objects are detected by motion detection techniques based, for example, on *inductive loops emulation* [1] or *trip-wire emulation*. These approaches, however, are not able to discriminate between real and apparent moving objects. With these premises, tracking of moving objects is fundamental to keep history and identity of objects. In general, a tracking algorithm should be robust to shape and appearance changes, robust to occlusions, and computationally efficient.

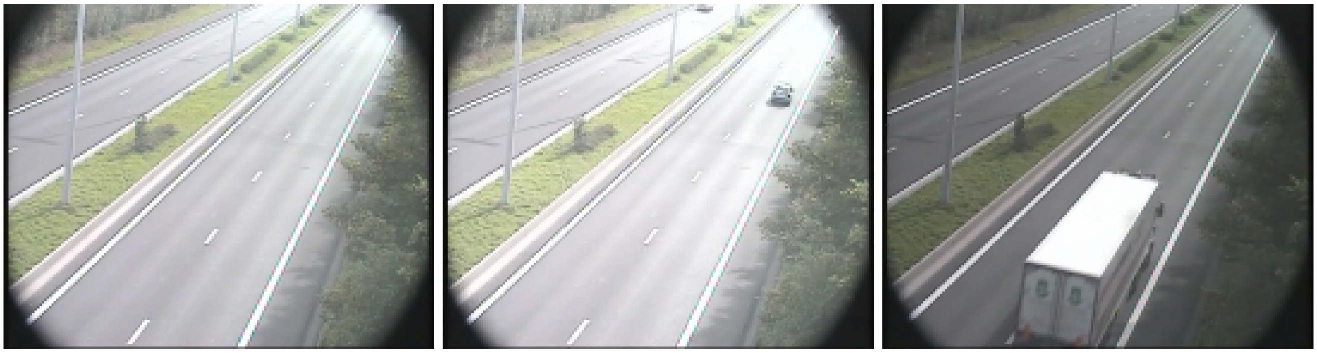
Among the many requirements, one of the more challenging is being robust to illumination changes. In particular, camera auto-iris can be very problematic in the case of background suppression techniques. In fact, when the camera reacts to a change in the scene by varying the camera's iris opening, that is increasing or decreasing the average luminance of the image, the difference between the current frame and the reference background image can result in many false positives and in the merge of the moving objects in a single, image-wide object. This compromises tracking performance and, thus, computed traffic parameters. Fig. 1 reports two examples of positive (i.e., the average luminance is increased) and negative (i.e., image is darker) auto-iris compared with an unmodified frame.

This paper addresses the problem of detecting and managing auto-iris in the case in which this functionality can not be disabled on the camera, either because there are other tasks that need it or because it can not be disabled at all. The auto-iris detection relies on the assumption that, in case of auto-iris, the points detected as foreground (i.e., different from the background) are numerous. Once the "auto-iris condition" is detected, two choices are possible: the first is to consider the extracted information as unreliable and to suspend the processing, avoiding to output any result; the second is to try to compensate the effects of auto-iris, continuing the moving object segmentation process. Since, in our cases, the auto-iris is very frequent, suspending the processing will result in a completely unreliable tracking. Therefore, we propose a new method that tries to compensate negative and positive auto-iris modifications to obtain a good segmentation (and, consequently, a robust tracking) also in these cases.

This work was supported by Traficon N.V., Belgium

R. Cucchiara and R. Melli are with Dipartimento di Ingegneria dell'Informazione, University of Modena and Reggio Emilia, 41100 Modena, Italy {cucchiara.rita,melli.rudy}@unimore.it

A. Prati is with Dipartimento di Scienze e Metodi dell'Ingegneria, University of Modena and Reggio Emilia, 42100 Reggio Emilia, Italy prati.andrea@unimore.it



(a) Unmodified frame (b) Example of positive auto-iris (c) Example of negative auto-iris

Fig. 1. Example of simultaneous crossing of two objects.

II. RELATED WORKS

The existing techniques for the detection and the filtering of the illumination changes are divided on pixel-level and region-level approaches. Most of the motion detection systems use background suppression. In these systems, illumination changes are really troublesome since they affect frame differencing, making highly probable to detect as moving also parts of the image that are still. An approach to manage these changes is to (implicitly or explicitly) select the detection threshold based on the estimated noise of the difference frame [2] or based on the gray level distribution of the background points (as in W^4 [3], Pfinder [4], the mixture of Gaussians [5][6]). Aach and Kaup in [7] propose a region-based change detection that employ a Bayesian formulation using a Gibbs/Markov Random Fields framework. Moreover, they use a noniterative multiple threshold algorithm to adapt the region differences to the context changes. This approach starts with the hypothesis that moving objects tend to have a compact shape with smooth boundaries and false alarms manifest themselves an irregular speckles spread randomly over the image plane, not always true in real installations. Bichsel [8] employed the distribution of both image derivatives and image difference for moving object detection. He assumed that the moving objects are enclosed by significants edges. However, this might not be always true.

These approaches can detect and suppress well the noise, but are sensitive to illumination changes due to the comparison of just intensity distribution at region level. Region-based illumination independent statistics have been also proposed in the literature. Skifstad and Jain [9] presents a Shading Model (SM) which uses the variance of the intensity ratio calculated in the corresponding regions of two images to detect changes. However, they present few tests with limited illumination changes. Liu et al. [10] defined Circular Shift Moment (CSM) to define a difference metric for detecting structural changes between consecutive frames under time-varying illumination.

All these methods employ ratio of intensities or sum of intensities to describe structural characteristics of a region. This means they may perform poorly over the dark regions in images due to the denominator of the ratio becoming insignificant for these regions.

Li and Leung [11] use a texture difference measure integrated with an intensity difference to detected the changes. Several works studied the use of homomorphic image filtering [12] [13]. For Lambertian surfaces with constant reflectance the relation between observed intensity y, illumination i and reflectance r is multiplicative and, under the assumptions reported in [14], it is possible to approximately compensate the illumination changes by taking the logarithm of the image before applying a linear high-pass filter. Unfortunately, in these cases the camera noise distribution must be known, which is not always possible. In addition, all these works are proposed as a general filter for illumination changes and do not work properly with specific strong changes caused by auto-iris.

The approaches proposed in the literature are, generally, quite complex and often rely on floating-point computation. For these reasons, they resulted to be unsuitable to our case, because of the DSP limitations reported in section I.

III. SYSTEM ARCHITECTURE

The system is composed by three main modules: segmentation, tracking, and scene understanding (Fig. 2).

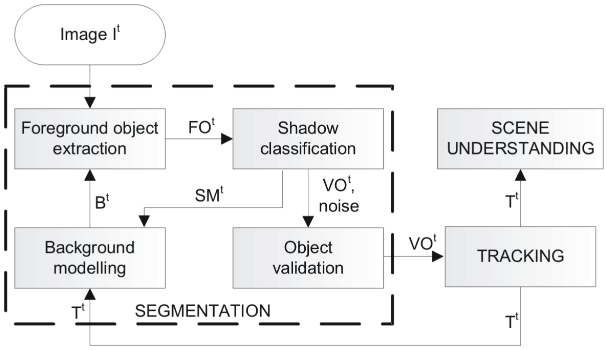


Fig. 2. The architecture of the system.

The segmentation module aims at extracting the *visual objects*. The first step uses the background suppression by subtracting the current background model B^t from the current frame I^t. The points are extracted and grouped with a labelling process into a set FO^t of foreground objects at instant time t. This set contains both relevant objects and other outliers, such as shadows and noise.

To identify shadow points we used a deterministic approach, similar to that proposed in [15], but duly modified for computational reasons to work directly in the YCbCr space (instead of the HSV space adopted by [15]). A point p (resulted from the segmentation) is classified as shadow point if its value in the mask $SM^t(p)$ is 1:

$$SM^t(p) = \begin{cases} 1 & if \;\; \alpha \leq \frac{I_Y^t(p)}{B_Y^t(p)} \leq \beta \wedge \; \Psi \leq \tau_S \wedge \; \Phi \leq \tau_H \\ 0 & otherwise \end{cases} \tag{1}$$

where the subscript Y denotes the Y component of a vector in the YCbCr space, and $\alpha, \beta \in [0, 1]$. The values Ψ and Φ can be obtained from equations reported in [15] approximating the hue H with the angle between vectors formed by Cb and Cr and the saturation S with the respective modules difference:

$$\Psi = \left| \sqrt{(I_{Cr}^t)^2 + (I_{Cb}^t)^2} - \sqrt{(B_{Cr}^t)^2 + (B_{Cb}^t)^2} \right|$$

$$\Phi = \frac{\left(I_{Cb}^t B_{Cb}^t + I_{Cr}^t B_{Cr}^t\right)^2}{\left((I_{Cb}^t)^2 + (I_{Cr}^t)^2\right)\left((B_{Cb}^t)^2 + (B_{Cr}^t)^2\right)}. \tag{2}$$

Experimental results proved that this transformation does not affect significantly the performance [16]. Objects in the set FO^t considered too small (taking the perspective distortion into account) are discarded as noise. The set VO^t of visual objects obtained after the size-based validation is processed by the tracking module that computes for each frame t a set of tracks $T^t = \{T_1^t, ..., T_m^t\}$ and assigns to each track T_i^t a status label: *moving*, *stopped*, *new*, or *undetected*. An object is classified as *stopped* when it is detected as still in the current frame and in at least a certain number of previous frames.

We use a basic *directional* tracking module. The tracking process starts with the association between the VO^t and T^{t-1} sets. To this aim a Boolean *correspondence matrix* is created. The basic rule for the association between VOs and tracks is the direction. If a VO has a direction divergent with the track and its state is not *stopped*, the correspondence is forbidden. Otherwise, supposing each VO and each track T are provided together with its bounding box BB and its centroid c, a visual object and a track are marked as correspondent if the *Bounding Box Distance* (BBd) defined in Eq. 3 is lower than a threshold:

$$\begin{aligned} & BBd(VO_j, T_k) = \\ & = \min_{\substack{x_k \in BB_k, \\ y_j \in BB_j}} \left(\min\left\{\left\|c_j, x_k + \vec{e}_k\right\|, \left\|c_k + \vec{e}_k, y_j\right\|\right\}\right) \end{aligned} \tag{3}$$

This threshold is decreased in the case of stopped vehicles, since, for them, looking for higher distances increases the probability to match with a wrong blob. Once the check on the minimum BBd is successful, the value in the correspondence matrix is set to the sum between the BBd and the weighted difference ratio of the areas. Both values are calculated with real world coordinates using the calibration parameters provided with the videos.

Correspondence matrix is solved parsing the matrix for VOs. In the case of one-to-one $VO - T$ correspondence, we investigate the track to look for another VO that matches better with this track. If there is not, the track is labelled as *moving* or *stopped* depending on its motion. Otherwise, the correspondence is not performed and the VO is labelled as *notNew*, that means a new track will not be created. In the case of a one-to-zero correspondence, a track is created and labelled as *new*. Instead, in the case of one-to-m $VO - T$ correspondence, all the tracks in the set are investigated. If one of these has a better matching value with another VO, it is removed from the set. If at the end of investigation the set is empty, this VO is labelled as *notNew* and a new correspondent track is not created, because there is an high probability that a vehicle has been divided by the segmentation process into more parts. If the set of tracks is not empty and contains only one element, this is managed as one-to-one correspondence. Eventually, if in the set there are more than one tracks, we choose the track that with longer trajectory and closer to the VO. After the corresponding process, all the tracks not matched with any VO are classified as *undetected*.

The knowledge about VO and their status is exploited by a selective background model. To this aim we adopt the model proposed in [17] but with some changes. In particular the above-mentioned track status is used to selectively update the background model and to exclude stopped vehicles from the updating [16].

Scene understanding is a high-level module. It receives the information from the tracking and switches on an alert when a stopped vehicles is detected.

IV. AUTO-IRIS DETECTION AND SUPPRESSION

The system described in section III relies heavily on a correct background suppression, since subsequent tasks are deeply influences by this. Thus, when an auto-iris occurs, like in the cases reported in Fig. 1, these techniques are likely to fail. From the analysis of the previous approach reported in section II, two classes of approaches are possible to cope with the auto-iris problem: the first is to pre-process the input frame to make background differencing invariant to illumination changes, while the second includes two-step approaches in which firstly the auto-iris is detected and then, if present, compensated. Since the first class is typically computationally intensive, our approach belongs to the second class.

A. Detection

Our auto-iris detection is based on the computation of the average of the luminance difference obtained by background suppression. The rationale is that auto-iris leads to the quick increase of the average of the luminance difference. To distinguish between real foreground points (belonging to moving objects) and points detected as foreground due to the auto-iris, we make the assumption that real moving points differ from the background more than those due to auto-iris. Therefore, the points whose difference w.r.t. to the background is limited are counted by computing the cardinality Np^t of the following set D^t:

$$D^t = \{p(x,y) \in I^t | \\ |\Delta Y^{t-1} - T_L \leq I_Y^t(x,y) - B_Y^t(x,y) \leq \Delta Y^{t-1} + T_H\} \quad (4)$$

where I^t and B^t indicate the current image and background, respectively, and the subscript Y stays for the luminance in the YCbCr color space. ΔY^{t-1} is the average variation computed on the previous frame as follows:

$$\Delta Y^t = \frac{\sum\limits_{p(x,y) \in D^t} I_Y^t(x,y) - B_Y^t(x,y)}{Np^t} \quad (5)$$

The two thresholds T_H and T_L are set to two different values (e.g., $T_L = 70$ and $T_H = 50$) to account for the different behaviours in the case of positive and negative auto-iris. If ΔY^t is greater than a threshold the auto-iris is detected and the compensation phase is started. This method has demonstrated a good accuracy in detecting auto-iris, resulting in a 8.5% of false positives and a 0.6% of false negatives (at frame level) over a test set of four sequences.

B. Compensation

In section II we reported examples of adaptive background (or frame) differencing, in which, for instance, the detection thresholds are adaptively selected. Some more advanced methods for adaptive thresholding have been proposed, as that in [12] where the detection threshold is modified in function of the number of points detected as moving in the neighborhood of the current point. The rationale is that moving pixels corresponding to moving objects are spatially close. However, this approach does not consider that the value of the change in luminance is proportional to the current value of the pixel.

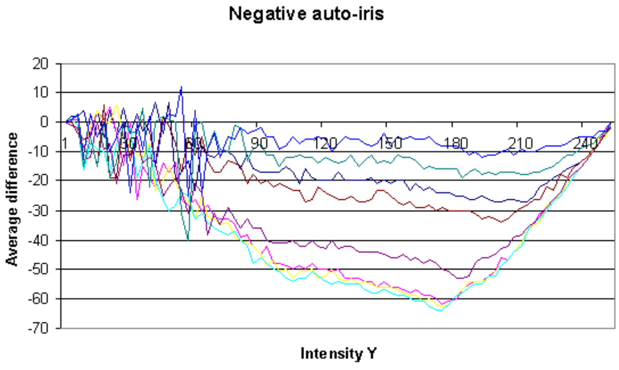


Fig. 3. Channel Y, negative auto-iris. Each series corresponds to a different frame.

Our approach is based on empirical evaluation of the results over a large test set of videos. From it, the graphs in Figs. 3 and 4 report the histogram distributions of the difference w.r.t. to the background in luminance. Different series in the graphs are obtained for successive frames correspondent to an auto-iris and collecting the values only in an area of the image in which actual motion is absent.

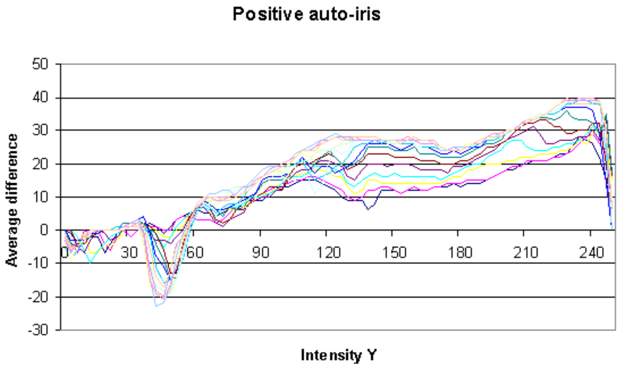


Fig. 4. Channel Y, positive auto-iris. Each series corresponds to a different frame.

These graphs are used to infer a model for the effects of the auto-iris in order to compensate them.

From Figs. 3 and 4 it is evident that the modification of the luminance due to auto-iris is a function of the gray level (x axis). In particular, it is proportional to the ΔY^t value reported in equation 5. From this analysis, a common peak in the average difference can be identified. In the case of negative auto-iris, denoting this peak with P, from our empirical evaluation we can approximate the corresponding intensity P_Y and the corresponding difference $P_{\Delta Y}$ proportionally with respect to the average luminance of the current frame, $\overline{I_Y^t}$:

$$P_Y = \overline{I_Y^t} + 35 \quad (6)$$

$$P_{\Delta Y} = -\overline{I_Y^t} * 1.1 \quad (7)$$

The auto-iris compensation model has been defined as non-linear with a square function for the portion of the histogram at the left of the peak P, and a linear function for the portion at the right. The compensated luminance of the pixel $\widetilde{I_Y^t}(x,y)$ is calculated as:

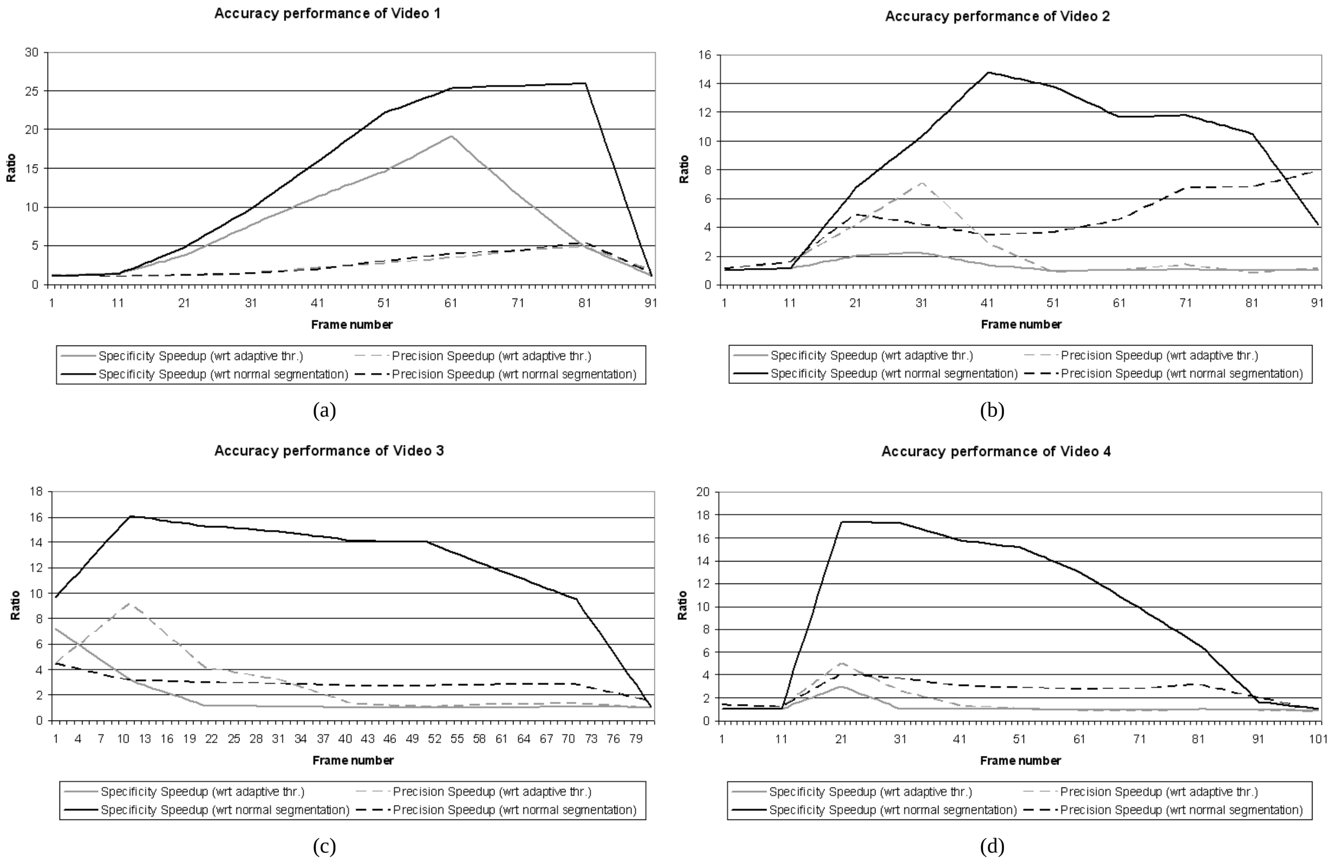
$$\widetilde{I_Y^t}(x,y) = \begin{cases} \left(\frac{\left(P_Y - I_Y^t(x,y)\right)^2}{\frac{P_Y^2}{P_{\Delta Y}}} \right) - P_{\Delta Y} & if \;\; I_Y^t(x,y) \leq P_Y \\ \frac{\left(255 - I_Y^t(x,y)\right) * P_{\Delta Y}}{255 - P_Y} & otherwise \end{cases} \quad (8)$$

Since positive variations of the luminance due to auto-iris had different behavior (see Fig. 3), different values are used:

$$P_Y = 240 \\ P_{\Delta Y} = \overline{I_Y^t} * 1.8 \quad (9)$$

In this case, the experiments highlight the independence of P_Y from the average luminance of the frame $\overline{I_Y^t}$. The auto-iris compensation model for the positive variations has been defined with two linear functions at the left and at the right of the peak P. The compensated luminance of the pixel is computed as:

$$\widetilde{I_Y^t}(x,y) = \begin{cases} \frac{I_Y^t(x,y) * P_{\Delta Y}}{P_Y} & if \;\; I_Y^t(p) \leq P_Y \\ \frac{\left(255 - I_Y^t(x,y)\right) * P_{\Delta Y}}{255 - P_Y} & otherwise \end{cases} \quad (10)$$



(a)

(b)

(c)

(d)

Fig. 5. Segmentation results.

To improve the model, chromatic channels (Cb and Cr) have been also investigated, but their enhancement in auto-iris compensation is negligible.

V. EXPERIMENTAL RESULTS

Thanks to our collaboration with Traficon N.V., we have the availability of tens of existing CCTV source sites to experiment our system in many different conditions. All videos provided by Traficon N.V. were acquired at 720x576 from analog interlaced camera sources at 25 fps. For this work, the frames have been downscaled to 360x288 to reduce computational complexity; in addition, the nearest neighbor algorithm used for the scaling has also reduced the interlaced effect. These videos suffer the classical problems of real camera installations: auto-iris, low quality image, auto white-balance, sudden illumination changes. In section IV, we explained how we detect with accuracy the auto-iris. Results reported in this section are meant to demonstrate this accuracy comparing our method with the normal background difference (without auto-iris compensation) and with a global *adaptive thresholding* method. The *adaptive thresholding* method calculates a new value of the detection threshold looking at the average difference of luminosity between current frame and background.

In order to evaluate the performance, a manual ground truth segmentation has been performed on four salient sequences with string auto-iris effect, three in different outdoor scenes and the fourth in a tunnel scene. To measure the accuracy, we used *precision* η (called also *positive predictive value*) and *specificity* ξ [18] calculated as:

$$\eta = \frac{TP}{TP+FP} \quad ; \quad \xi = \frac{TN}{TN+FP} \qquad (11)$$

where FP indicates the number of *false positives* (points of background classified as foreground), FN the number of *false negatives* (points of foreground classified as background), TP the number of *true positives* (points of foreground correctly classified) and TN that of *true negatives* (points of background correctly classified). To emphasize the improvement achieved by our approach, the ratio between the precision obtained with our method and that obtained with the normal segmentation is used. Similarly, we compute the ratio for the specificity. These values (reported with dark lines in Fig. 5) measure the speedup in precision and specificity obtained by our approach. We also report (in gray lines in Fig. 5) these values with respect to the *adaptive thresholding*. Graphs in Fig. 5 evidence how our algorithm works very well, compensating the illumination changes due to auto-iris in all the sequences.

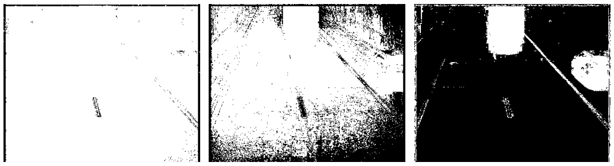
Figs. 6-9 show some snapshots of the segmentation results, where white pixels correspond to foreground points. It is noticeable that the segmentation is significantly improved.

Eventually, since auto-iris does not last many frames,

TABLE I

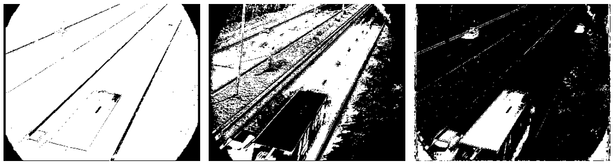
TIME PERFORMANCE PER FRAME.

	Video 1	Video 2	Video 3	Video 4	Average
Normal bkg diff.	3.97 msec	3.80 msec	3.84 msec	3.72 msec	3,83 msec
adaptive thresh.	3.92 msec	3.88 msec	3.64 msec	3.72 msec	3.79 msec
proposed appr.	4.20 msec	4.90 msec	6.76 msec	4.02 msec	4.97 msec



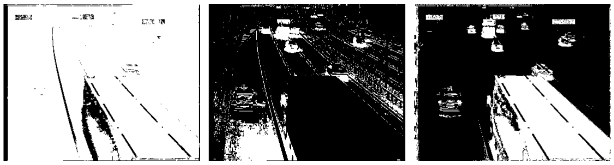
(a) Without auto-iris compensation (b) With adaptive thresholding (c) With proposed approach

Fig. 6. Segmentation results on Video 1



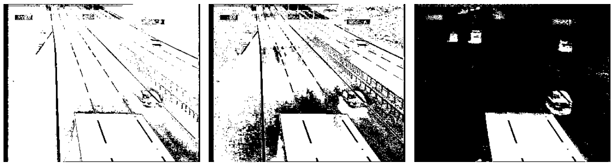
(a) Without auto-iris compensation (b) With adaptive thresholding (c) With proposed approach

Fig. 7. Segmentation results on Video 2



(a) Without auto-iris compensation (b) With adaptive thresholding (c) With proposed approach

Fig. 8. Segmentation results on Video 3



(a) Without auto-iris compensation (b) With adaptive thresholding (c) With proposed approach

Fig. 9. Segmentation results on Video 4

the computational overhead introduced by our algorithm with respect to adaptive thresholding or normal background differencing is negligible, as evidenced in Table I where the figures are related to the background differencing task only per frame (on a P4 processor at 3.4 GHz and 1 GB of RAM).

VI. CONCLUSIONS

In this paper, we present a new two-step approach to auto-iris compensation. The method, although empirical, has proved to be very effective in many different situations, such as outdoor and tunnel scenes. More sophisticated approaches have been investigated, but the limited resources of the DSP board on which our system is (at the end) implemented make these approaches unsuitable. In fact, we also demonstrate that the proposed two-step approach is computationally inexpensive.

REFERENCES

[1] L. Di Stefano and E. Viarani, "Evaluation of inductive-loop emulation algorithms for UTC systems," in *Proc. of Sixth Int'l Conference on Control, Automation, Robotics and Computer Vision (ICARCV)*, 2000.

[2] P. L. Rosin, "Thresholding for change detection," in *Proc. of IEEE Int'l Conference on Computer Vision*, 1998, pp. 274–279.

[3] I. Haritaoglu, D. Harwood, and L. Davis, "W4: real-time surveillance of people and their activities," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 8, pp. 809–830, Aug. 2000.

[4] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: real-time tracking of the human body," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, no. 7, pp. 780–785, July 1997.

[5] C. Stauffer and W. Grimson, "Learning pattern of activity using real-time tracking," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 8, pp. 747–757, Aug. 2000.

[6] N. Friedman and S. Russell, "Image segmentation in video sequences: A probabilistic approach," in *Proc. Thirteenth Conf. on Uncertainty in Artificial Intelligence*, 1997.

[7] T. Aach and A. Kaup, "Bayesian algorithms for adaptive change detection in image sequences using markov random fields," in *Signal Processing: Image Communication*, vol. 7, Aug. 1995, pp. 147–160.

[8] M. Bichsel, "Segmenting simply connected moving objects in a static scene," in *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, no. 11, Nov. 1994, pp. 1138–1142.

[9] K. Skifstad and R. Jain, "Illumination independent change detection for real world image sequences," in *Computer Vision, Graphics, and Image Processing*, vol. 46, no. 3, June 1989, pp. 387–399.

[10] S. Liu, C. Fu, and S. Chang, "Statistical change detection with moments under time-varying illumination," *IEEE Trans. Image Processing*, vol. 7, no. 9, pp. 1258–1268, Sept. 1998.

[11] L. Li and M. K. H. Leung, "Robust change detection by fusing intensity and texture differences," in *Proc. of IEEE Int'l Conference on Computer Vision and Pattern Recognition*, vol. 1, 2001, pp. 777–784.

[12] D. Toth, T. Aach, and V. Metzler, "Illumination-invariant change detection," in *IEEE Southwest Symposium on Image Analysis and Interpretation*, Apr. 2000, pp. 3–7.

[13] j. Lou, H. Yang, W. Hu, and T. Tan, "An illumination invariant change detection for visual surveillance," in *Asian Conference on Computer Vision*, 2002.

[14] C. Moloney, "Methods for illumination-independent processing of digital images," in *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, vol. 2, May 1991, pp. 811–814.

[15] A. Prati, I. Mikic, M. Trivedi, and R. Cucchiara, "Detecting moving shadows: Algorithms and evaluation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 25, no. 7, pp. 918–923, July 2003.

[16] R. Cucchiara, R. Melli, A. Prati, and L. De Cock, "Predictive and probabilistic tracking to detect stopped vehicles," in *Proc. of Workshop on Applications of Computer Vision*, Jan. 2005, pp. 388–393.

[17] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, "Detecting moving objects, ghosts and shadows in video streams," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 25, no. 10, pp. 1337–1342, Oct. 2003.

[18] A. P. Bradley, "The use of the area under the roc curve in the evaluation of machine learning algorithms," in *Pattern Recognition*, vol. 30, 1997, pp. 1145–1159.