Adaptive Multicue Background Subtraction for Robust Vehicle Counting and Classification

Luis Unzueta, *Member, IEEE*, Marcos Nieto, Andoni Cortés, Javier Barandiaran, Oihana Otaegui, and Pedro Sánchez

***Abstract*—In this paper, we present a robust vision-based system for vehicle tracking and classification devised for traffic flow surveillance. The system performs in real time, achieving good results, even in challenging situations, such as with moving casted shadows on sunny days, headlight reflections on the road, rainy days, and traffic jams, using only a single standard camera. We propose a robust adaptive multicue segmentation strategy that detects foreground pixels corresponding to moving and stopped vehicles, even with noisy images due to compression. First, the approach adaptively thresholds a combination of luminance and chromaticity disparity maps between the learned background and the current frame. It then adds extra features derived from gradient differences to improve the segmentation of dark vehicles with casted shadows and removes headlight reflections on the road. The segmentation is further used by a two-step tracking approach, which combines the simplicity of a linear 2-D Kalman filter and the complexity of a 3-D volume estimation using Markov chain Monte Carlo (MCMC) methods. Experimental results show that our method can count and classify vehicles in real time with a high level of performance under different environmental situations comparable with those of inductive loop detectors.**

***Index Terms*—Computer vision, tracking, traffic image analysis, traffic information systems, 3-D reconstruction.**

I. Introduction

CURRENTLY, the most sophisticated vision-based approaches for traffic flow surveillance combine information from cameras with other technologies, such as tags installed in vehicles, laser scanners that reconstruct the 3-D shape of the vehicles, or the Global Positioning System (GPS), to estimate the direction of the casted shadows [1]. Nevertheless, most vehicles do not have tags installed and can be "tricked," and laser scanners increase the cost of systems and are sensitive to weather conditions, analogous to GPS, whose calibration complexity makes satisfactory results more costly to obtain.

Manuscript received February 16, 2011; revised July 4, 2011; accepted October 1, 2011. Date of publication November 25, 2011; date of current version May 30, 2012. This paper was supported by the research projects iToll (http://itoll.clustertil.org/), funded by the Basque Government, and Intelvia (http://www.intel-via.org/), funded by the Spanish Government. The work of L. Unzueta was supported by the Ministry of Education of Spain within the framework of the Torres Quevedo Program. The Associate Editor for this paper was K. Wang.

L. Unzueta, M. Nieto, A. Cortés, J. Barandiaran, and O. Otaegui are with Vicomtech-IK4 Research Alliance, 20009 Donostia-San Sebastián, Spain (e-mail: lunzueta@vicomtech.org; mnieto@vicomtech.org; acortes@vicomtech.org; jbarandiaran@vicomtech.org; ootaegui@vicomtech.org).

P. Sánchez is with IKUSI–Ángel Iglesias S.A., 20009 Donostia-San Sebastián, Spain (e-mail: pedro.sanchez@ikusi.com).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org. provided by the authors.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2011.2174358

Fig. 1. Real installation in a roadway near Donostia-San Sebastián, Spain.

Compared with intrusive technologies such as radar, inductive loop detectors (ILDs), or lasers, computer vision can be used to obtain richer information, such as analyzing the visual features of the vehicles (color, lights, plate number), apart from the geometry (vehicle volume). These advantages and the increasing computational power of processors have made vision-based systems an area of great interest for road operators, particularly in tolling applications.

Therefore, there is a large number of approaches in the literature related to vehicle classification using computer vision, which can be classified in two main groups: 1) sophisticated academic approaches and 2) commercial solutions. According to [2], many commercially available machine vision-based systems rely on simple processing algorithms, such as "virtual detectors" in a similar way to ILD systems, with limited vehicle classification capabilities, in contrast to more sophisticated academic developments [3]–[5]. However, many of the latter usually cannot perform in real time.

As a mixture between these two groups, we have designed a robust and accurate vision-based system to obtain traffic data flow. On the one hand, it operates in real time and works in challenging scenarios (low-cost cameras, poor illumination, and, in the presence of shadows, unknown perspective; Fig. 1 shows some images of the real installation where the system is currently working) and, on the other hand, achieves accurate 3-D vehicle classification into different categories using image sequences from a single camera.

The novelty of our approach relies on a multicue background subtraction procedure in which the segmentation thresholds can adapt robustly to illumination changes, maintaining a high sensitivity level to new incoming foreground objects and effectively removing moving casted shadows and headlight reflections on the road and in traffic jam situations, in contrast to existing approaches that do not cover all of these functionalities at the same time [6]. A tracking module provides the required spatial and temporal coherence for the classification of vehicles, which first generates 2-D estimations of the silhouette of the vehicles and then augments the observations to 3-D vehicle

volumes by means of a Markov chain Monte Carlo (MCMC) method. Its advantage over previous approaches, such as [3]–[5] and [7], is that it can directly apply on existing 2-D tracks to infer the dimension lost due to camera projection. The handling of severe occlusions is out of the scope of this paper, but nevertheless, our approach can be used in practice by applying it to road viewpoints, such as those achievable from cameras installed in highway gantries, with a performance significantly beyond state-of-the-art vision approaches.

We include experimental results with varying weather conditions, on sunny days with moving directional shadows, headlight reflections on the road, rainy days, and traffic jams. We obtain vehicle counting and classification results comparable with those of ILD systems, which are currently the most widely used systems for these types of traffic measurements, while keeping the main advantages of vision-based systems, i.e., not requiring the cumbersome operation or installation of equipment at the roadside or the need for additional technology such as laser scanners, tags, or GPS.

II. Related Work

The main scientific contributions of our work in the field of computer vision are the background subtraction and the tracking stage. The following sections review related work in these two topics.

A. Background Subtraction

Regarding background subtraction, there are methods that model the variation of the intensity values of background pixel with unimodal distributions [8], [9], mixture of Gaussians [10], [11], nonparametric kernel density estimation [12], or codebooks [13], [14]. Unimodal models are fast and simple but are not able to adapt to multiple backgrounds, e.g., when there are trees moving in the wind. The mixture of Gaussian approaches can cope with these moving backgrounds but cannot handle fast variations with accuracy using a few Gaussians, and therefore, this method has problems for the sensitive detection of foreground regions. The nonparametric kernel density estimation described in [12] allows quick adaptation to background changes. However, codebook-based methods are the most sensitive color-based background subtraction methods, applied both indoors and outdoors, even with some motion in the background [13]. The codebook approach, unlike the other methods mentioned, explicitly models the illumination change of pixels, which is the variation that occurs more commonly between foreground and background.

In outdoor scenarios, it is necessary to update the background model in time and adapt it to the new conditions of the scene (illumination, weather conditions, shadows). The most sophisticated approaches are those that learn background selectively, depending on the pixels cataloged as potential vehicles and also of optical flow [15]. In [16], the vehicles are classified according to the height and width of the extracted blobs but without taking into account the suppression of projected shadows. Alternatively, other approaches use gradient cues instead of intensity values, improving the robustness against illumination changes [17]. However, plain regions that may be present in some vehicles are not extracted and still need further processing for discriminating shadows.

Most of these works use the shadow model described in [18]. This model distinguishes penumbra and umbra. Umbra is the shadow that receives illumination coming only from diffuse ambient light, whereas penumbra receives illumination from both the ambient light and a portion of direct light. Therefore, penumbra has more chromatic similarity with respect to its original color than in the case of umbra. According to the taxonomy proposed in [19], shadow suppression methods can be classified as deterministic and statistical. The former uses on/off decision processes, whereas the latter uses probabilistic functions that define different classes. Deterministic methods can be subdivided into model and nonmodel based. Model-based methods use explicit models of the vehicles to be tracked and also of light sources [20], whereas nonmodel-based methods do not [21]. On the other hand, statistical models can be subdivided into parametric and nonparametric. Parametric approaches use a series of parameters that determine the characteristics of the statistical functions of the model [22], whereas nonparametric approaches automate the selection of the model parameters as a function of the observed data during training [9].

According to [19], model-based deterministic techniques can obtain better results for shadow suppression, but it should be remarked that these are excessively cumbersome for a practical implementation in outdoor traffic surveillance. Therefore, nonmodel-based deterministic approaches are most suitable for outdoors applications, whereas nonparametric statistical methods are best for indoors, since the scene is more constant, and thus their statistical description is more effective. In [19], it is also stated that the hue, saturation, and value (HSV) is the color space that can distinguish shadows more precisely, as these do not significantly change the color tone and tend to lower the saturation in the case of penumbra but not umbra. The Sakbot system [21] uses all of these conclusions for counting vehicles from videos. More recently, in [23], it is shown that the improved hue, luminance, and saturation (IHLS) color space is better suited for change detection and shadow suppression than HSV and normalized red, green, and blue (RGB) as it allows the problem of the unstable hue channel at weakly saturated colors to be dealt with.

However, most recent methods for both background subtraction and shadow suppression mix multiple cues, such as edges and color, to obtain more accurate segmentations. For instance, Huerta *et al.* [24] apply heuristic rules by combining a conical model of brightness and chromaticity in the RGB color space along with edge-based background subtraction, obtaining better segmentation results than other previous state-of-the-art approaches. They also point out that adding a higher level model of vehicles could allow for better results as these could help with bad segmentation situations. This is what is done in [7], where the size, position, and orientation of a 3-D bounding box of a vehicle, which includes shadow simulation from GPS data, are optimized with respect to the segmented images. Furthermore, it is shown in some examples that this approach can improve the performance compared with using only shadow detection or shadow simulation. Their improvement is most

evident in cases where shadow detection or shadow simulation is inaccurate. However, a major drawback for this approach is the initialization of the box, which can lead to severe failures.

B. Vehicle Tracking

To provide temporal coherence to the measurements, tracking methodologies are typically applied. Two main trends in techniques can be distinguished: 1) those considering 2-D image objects and 2) works that infer 3-D volumes and positions using camera calibration.

Regarding 2-D tracking schemes, the Kalman filter has been shown to offer great estimation results in rectified images, where the dynamics of the vehicles are undistorted and thus can be assumed to be linear [25]. Nevertheless, 2-D estimations lack the required accuracy in classification strategies: The viewpoint of the camera is a critical aspect for these strategies. The problem of tracking vehicles is tackled typically with pole-mounted cameras, with a large angle, looking down on to the road [26]. This way, the perspective effect is reduced, and the length and width of vehicles can be measured with lower error. However, more flexible approaches should consider several potential road viewing angles. This issue directly affects the maximum accuracy that a 2-D approach can provide, and only 3-D methods can reliably determine vehicle measurements in such situations [27].

For this purpose, many 3-D estimation alternatives have been proposed, which in their essence fit a 3-D model of a vehicle to the observed image. We have found that the most sophisticated methods for vehicle tracking typically use expensive high definition cameras [3]. In addition, they are usually designed for simplified urban scenarios, with reduced vehicle speeds, and do not consider overtaking maneuvers, which makes the classification problem easier [4]. Some 3-D classification methods have used vehicle models as prior information, such as wireframe fixed models [5]. However, as a general remark, in most situations, the fitting accuracy of these methods is much lower than the detail of the wireframe, making such complex vehicle models ineffective. Cuboid models have been proposed as an acceptable tradeoff [7], [28].

The type of tracking method can also be used to classify works. Some use simple data association between detections in different time instants [29]. However, Bayesian filtering like Kalman filters [30], the extended Kalman filter [31], and, as a generalization, particle filter methods [28], [32] have been shown to provide more efficiency and robustness in results.

Particle filters, although they represent the most powerful alternative, require the use of many particles to converge to a correct target posterior distribution [33]. In traffic flow applications, the average number of vehicles in the scene at the same time instant can be up to 10 or 12 with a mean of 4 or 5, and the required number of particles must be approximately 1000.

This number can be excessively high and has been the motivation behind the proposal of a novel sampling approach more suitably adapted to the problem, which is based on a predefined set of possible models that guide the sampling strategy and help to dramatically reduce the required number of evaluations of the posterior distribution. This method is explained in Section IV.

III. Multicue Background Subtraction

Fusing different cues has been proven to be the current best approach to obtain accurate background subtraction results. Two main issues arise at this point: 1) which cues are to be used and 2) in which way they are fused. Typical useful cues are pixel color, pixel intensity (gray level), and edges (obtained from image gradients), which have been used in works such as [9], [24], and [34]. Although these approaches present interesting conclusions for their application in the measurement of traffic data, such as vehicle counting and classification, they do not take into account several necessary issues for real and affordable video surveillance systems. The most important issue is the background update over time, since in real applications, training periods cannot be separated from processing periods. The background must be trained continuously while the system observes the scene to adapt to global changes, which affect the background mean and standard deviation values and, thus, the brightness and chromaticity distortions. Another important factor is the complexity of the algorithms. The computation power needed by image processing methods is related to the resolution and frame rate, even if these algorithms can be parallelized in multiple cores and many-core Graphics Processing Units (GPUs) programmable with parallel languages such as OpenMP, TBB, CUDA, or OpenCL. Typically, further processing steps, which are not necessarily parallelizable, are needed, such as vehicle tracking and classification procedures. Hence, some simplifications and optimizations are desirable for the background subtraction itself.

Thus, in this paper, we present a new multicue segmentation architecture to fuse different image cues, which combine bottom-up and top-down strategies to solve global/local illumination changes, and to obtain a conditional background model learning. As shown in the experiments carried out in Section V, it improves existing segmentation approaches, resulting in a step forward on the current state of the art in vehicle counting and classification using surveillance cameras. It concretely enhances the following aspects.

1) The bottom-up strategy includes a color model with a higher sensitivity to scene changes to the typically used cylindrical RGB and conical RGB models [14], [34] and gradient cues that efficiently reinforce the segmentation masks. This allows to easily define different segmented regions according to its luminance and chromaticity characteristics. Thus, we can distinguish projected moving shadows, vehicle headlight reflections, global illumination changes due to camera autoexposure when a light colored truck passes through the image, or environmental changes.
2) The top-down strategies include scene information coming from shadow directions detected directly from the images without the need for extra sensors, camera lens distortion and perspective, and historical tracking data to obtain an improved background model that can also be used in sunny days and with dense traffic situations, ignoring other moving objects such as flies, raindrops, etc.
3) The proposed background subtraction strategy is demonstrated to perform well in this scenario since the following

$D_l \leftarrow$ Per xy pixel: $|I^y_{xy} - B^y_{xy}| - k_l \cdot (\Sigma_l)_{xy}$
$D_c \leftarrow$ Per xy pixel: $\sqrt{(I^{c_x}_{xy} - B^{c_x}_{xy})^2 + (I^{c_y}_{xy} - B^{c_y}_{xy})^2} - k_c \cdot (\Sigma_c)_{xy}$
Apply CLAHE [36] to D_l and D_c
t_l & $t_c \leftarrow t_{(l,c)} = min(D_{(l,c)}) + o_{(l,c)}$
if $(D_l)_{xy} > t_l \cup (D_c)_{xy} > t_c$ (per xy pixel) **then**
  if $B^y_{xy} > I^y_{xy} \cap |B^y_{xy} - I^y_{xy}| < o_s$ **then**
    $Mask_{xy}$ = $shadow$
  else if $B^y_{xy} > I^y_{xy} \cap |B^y_{xy} - I^y_{xy}| \geq o_s$ **then**
    $Mask_{xy}$ = $black$
  else if $I^y_{xy} - B^y_{xy} > o_h \cap I^y_{xy} \leq o_w$ **then**
    $Mask_{xy}$ = $highlight$
  else if $I^y_{xy} > o_w$ **then**
    $Mask_{xy}$ = $white$
  else
    $Mask_{xy}$ = $foreground$
  end if
  if $Mask_{xy}$ = $(shadow \cup black) \cap (D_c)_{xy} > t_c \cap (D_l)_{xy} - t_l < (D_c)_{xy} - t_c$ **then**
    $Mask_{xy}$ = $foreground$
  end if
else
  $Mask_{xy}$ = $background$
end if
$E_I, E_B \leftarrow$ Apply Sobel$_{xy}$ to I and B with t_g
Add $(E_I - E_B)$ to $Mask$ as $foreground$
Crop $highlight$ regions in $Mask$ (Fig. 6)
Watershed(t_f): Fill $foreground$ & $highlight$ & $white$ blob inbetweens as $foreground$
Reinforce $Mask$ from shadow direction (Fig. 8)
Update B, Σ_l, Σ_c (Fig. 10)
return $Mask$

Fig. 2. Multicue background subtraction algorithm.

2-D/3-D strategy heavily relies on it to obtain the position and volume estimates of the vehicles. The ability of the method to distinguish between projected moving shadows and global illumination changes reduces the number of tracking errors and, thus, allows the system to achieve the low false positive and false negative rates.

4) Furthermore, we have parallelized our algorithms in our implementation, where possible, achieving real-time performance.

A. Image Pixel Classification

Fig. 2 summarizes the sequential steps to be given for the proposed multicue background subtraction. In this algorithm, I is the current image (I^y_{xy}, $I^{c_x}_{xy}$, and $I^{c_y}_{xy}$ are the luminance and chromaticity coordinates for each xy pixel in the IHLS color space); B is the mean background (B^y_{xy}, $B^{c_x}_{xy}$, and $B^{c_y}_{xy}$ are the luminance and chromaticity coordinates for each xy pixel in IHLS color space); Σ_l is the background luminance variance; Σ_c is the background chromaticity variance; k_l and k_c are the proportionality constants for determining the background/foreground disparity maps (D_l and D_c); o_l, o_c, o_s, and o_w are the offsets applied for pixel classification; and finally, t_g and t_f are the thresholds for gradient (Sobel$_{xy}$) and blob-filling (watershed [35]) procedures.

The algorithm classifies pixels according to multiple cues: 1) luminance and chromaticity *disparities*; 2) image gradients; and 3) higher level observations such as blob detections and the estimated shadow direction. The parameters that control the algorithm can be divided in two groups: 1) those related to the foreground/background (FG/BG) segmentation ($k_{(l,c)}$ and $o_{(l,c)}$) and 2) those related to the segmented pixel classification ($o_{(s,h,w)}$, t_g, and t_f).

The first group, i.e., the segmentation thresholds, is determined by keeping the pixel segmentation false alarm rates below a certain threshold [13], whereas the references for the second, i.e., the classification thresholds, are the observed dark shadow segmentations in sunny days and the headlight segmentations when there is lower ambient light.

In practice, these thresholds can be set as follows: Once the system has learned a background model (B, Σ_l, Σ_c), we take a frame containing passing vehicles as a reference to check the segmentation quality. Initially, $o_{(l,c)}$ are set to zero, whereas $k_{(l,c)}$ are set with a high number. Then, $k_{(l,c)}$ parameters are lowered separately until the vehicle shapes are extracted without surrounding segmentation noise. Next, even if this frame is correctly segmented, due to the noise of video streams, it is recommendable to raise the segmentation offset values $o_{(l,c)}$ so that the pixel segmentation false alarm rate is kept very low ($< 0.1\%$ in our case). It is desirable to set them to the lowest possible so that the system is the most sensitive possible to scene changes.

Once the segmentation thresholds have been established, we take a frame in which vehicles projecting shadows in a sunny day can be observed. The rule to set the classification parameters is to segment vehicles, i.e., only the best possible, with respect to shadows in this case. This means that we set o_s and t_g so that pixels classified as *black* and *foreground*, corresponding to dark and gradient regions, lie mostly on the vehicle projections and not in shadows. Next, we establish o_h and o_w using a frame with vehicles with the headlights on. Again, the rule is to segment vehicles only, the best possible, with respect to headlight projections in this case. Thus, we first set o_h to obtain headlight projection pixel classifications of *highlight* only, which will be consequently cropped through the cropping algorithm explained in the next section. As vehicle internal regions can also be classified as *highlight*, we set o_w in order to obtain pixel classifications of *white* instead of *highlight* and thus to avoid cropping them. Finally, t_f is set following again the rule of improving the segmentation of vehicles only, the best possible, by observing how filling the inbetweens of *foreground*, *highlight*, and *white* regions with *foreground* affecting this criteria.

It may seem that this parameterization does not allow the approach to be generic, but as it will be shown in the tests section, these parameters are quite independent from environmental changes, and therefore, they need to be established only once for all cases.

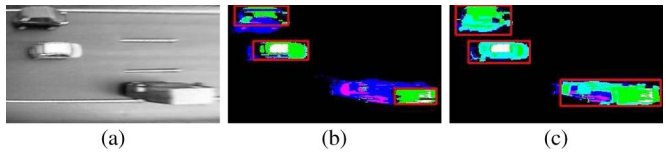
The resulting segmentation thresholds $t_{(l,c)}$ can adapt automatically to environmental changes because they correspond to the minimal values of the disparity maps $D_{(l,c)}$, calculated at each time instant, biased by the user-defined offsets $o_{(l,c)}$, and therefore allow the system to run satisfactorily in real installations. For the practical implementation of the approach, it is also recommended to normalize the image resolution to a

predefined number of lines, keeping the aspect ratio. This way, it is also possible to make parameters that depend on it more independently from installation to installation.

The disparity images $D_{(l,c)}$ are established by comparing the values of pixel luminance and chromaticity differences with respect to their corresponding temporal variances. If the difference is higher than the variance, then it can be considered not to be due to the image noise or moving background features and, therefore, be part of a change in the scene due to a moving object (vehicle, flies, raindrops, etc.) or an illumination variation. The thresholds to extract the nonbackground regions are established automatically with minimal disparity values and thus can adapt to outdoor scene variations. As there can be some vehicles with higher disparity values than other vehicles in the same image and as we also consider user-defined offsets for thresholds, to diminish false negative segmentations, we apply a contrast-limited adaptive histogram equalization procedure [36] to the disparity images to enhance the local contrast of disparities before thresholding. To remove small segmented areas due to noise, we also apply morphologic procedures such as erosion and dilation at this stage.

Initially, a mask is extracted by measuring the luminance and chromaticity disparities of the current image with respect to a mean background updated through time. This mask contains pixels cataloged according to the following labels: *background*, *shadow*, *black*, *foreground*, *highlight*, and *white*, where each of these is represented by a gray-level value in the mask. *Shadow* and *black* labels are set to darker pixels, depending on the luminance disparity amount with respect to B. The reason for including a *black* pixel category, besides from *shadow* for darker regions, is that this category is more plausible to be inside the segmented blobs instead of their boundaries, as can be observed experimentally. After the first cataloging stage, these darker regions are revisited again to check if their chromaticity disparity has more importance than that of luminance and relabel them as *foreground* in the case that it has. In a similar manner, lighter pixels are cataloged as *highlight* and *white*. The proposed categorization allows improvement in the segmentation quality with respect to previous ones, such as the four well-known classes (*background*, *shadow*, *highlighted background*, and *foreground*) originally presented in [9], as will be shown in the tests section.

In the second stage of the process, the mask is reinforced with subtracted gradient cues, which is particularly useful for detecting features that correspond to dark vehicles, which can be mistaken with shadows according to its model definition. It is recommendable to dilate the E_B image before subtracting as the edge subtraction can contain noisy features that should be removed. On the other hand, edges can appear in the shadow boundaries, and therefore, a morphological kernel-based erosion should be applied. It is preferable to do this erosion only when it has been detected that there might be a noticeable shadow projection, which is done in the algorithm shown in Fig. 7, as small dark vehicles without projected shadows could be cropped in the process. This decision can easily be automated through the relation of the number of pixels in the shadow direction reinforcement mask with respect to those of full blobs.



(a) (b) (c)

Fig. 3. Image segmentation results. Image (b) shows the obtained segmentation using the luminance and chromaticity disparity maps only, whereas (c) also includes the gradient cues. The pixel classification colors are *background* = black, *shadow* = blue, *black* = magenta, *foreground* = cyan, *highlight* = green, and *white* = white. It can be observed in (c) how the number of pixels classified as foreground is higher.

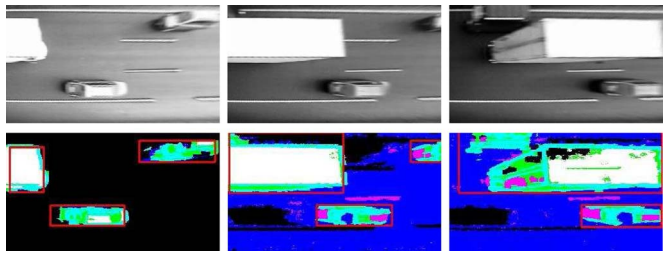


Fig. 4. Sudden global illumination change of the surrounding background image due to camera autoexposure when a light colored big truck passes through the image.

(a) (b) (c)

Fig. 5. Sudden local illumination change due to vehicle headlights. Image (b) shows the initially obtained segmentation, and (c) the resulting cropped blob using the proposed approach.

Then, as will be explained in next section, the mask will be processed to remove highlighted regions corresponding to sudden illumination changes due to weather variability or vehicle headlights. Additionally, the blob mask inbetweens are filled by applying the watershed procedure [35] to *foreground*, *highlight* and *white* inbetweens, filling them with *foreground*. As watershed increases the size of the resulting compact blobs, a proportional erosion is applied to avoid it. Thus, we reinforce the dark vehicle segmentation and prepare them for a later stage in which the shadow direction will be estimated to reinforce the mask even more. Fig. 3 shows an example of this segmentation procedure.

B. Sudden Illumination Changes Processing

We will ignore darker regions to build vehicle blob candidates for the tracking stage, but highlighted regions require further processing to remove those generated by sudden illumination changes coming from weather variations or headlights (see Figs. 4 and 5). Fig. 6 shows the algorithm that does this, taking into account lane geometry, where x is the direction along the lane, and y is the transversal in the images that, as will be shown in next section, are rectified so that these directions match with x and y of the image. Inside a lane, the pixels corresponding to a line perpendicular to the lane are summed, and if all nonbackground pixels correspond to the *highlight* category, then those pixels are set to *background*. Vehicle projections usually have more than one pixel category in

```
for all lane do
    for x = 0, lane_length do
        count = 0
        hCount = 0
        for y = lane_y, lane_width do
            if Mask_xy ≠ background then
                count = count + 1
                if Mask_xy = highlight then
                    hCount = hCount + 1
                end if
            end if
        end for
        if hCount = count ∩ count ≠ 0 then
            for y = lane_y, lane_width do
                Mask_xy = background
            end for
        end if
    end for
end for
return Mask
```

Fig. 6. Highlight cropping algorithm.

lines perpendicular to lanes, and therefore, using this approach, blobs can be cropped per lane, removing fully highlighted areas but not the vehicles. Lane geometry is considered in this process as there can be vehicles in different lanes, parallel to highlighted regions, which could interfere in the cropping. The *white* category is also included in the mask, apart from *highlight*, because vehicles painted in white with low texture characteristics can appear, and applying this procedure, they could be cropped when they should not be. This procedure can also be applied in road verges in the same way.

C. Shadow Direction Estimation and Mask Reinforcement

Fig. 7 shows the procedure to reinforce the segmented mask by estimating the shadow mean direction and setting the blob region as *highlight*, ignoring the *shadow* category, opposite to it. Using its complementary mask, the opposite region is explicitly set as *shadow* to remove any remaining pixels wrongly cataloged as foreground, mainly due to edge cues in shadow boundaries. This is suitable for distinguishing dark vehicles from casted shadows better, because it allows the automatic relabeling of pixels that have more chance to lie inside the vehicles and not in shadow regions, as we know that the vehicle is located at the opposite side to its projected shadow. Morphologic procedures can also be applied to G_1 and G_2 images to obtain a more compact and better suited reinforcing (G_1–G_2) image, as depicted in Fig. 8. The number of pixels N for image displacement is set experimentally according to the resolution. It is recommended to set a value with a size similar to those of vehicle widths.

D. Conditional Background Update

Fig. 9 shows the algorithm to update the B, Σ_l, and Σ_c background images, where I_{t-1} is the previous frame, T is a matrix that stores the time passed for each *static* pixel, $blobs_{\text{tracked}}$ are the tracked blobs, as will be shown in the next section, Δt is the time passed from the previous frame, l_{rate} is the background learning rate, and t_{update} is the time threshold for updating *static* blobs. *Static* pixels are those that lay inside tracked blobs with no displacement from frame to frame.

This procedure allows the robustness of the system in terms of background update to be increased. Within this type of scenario, it is critical to keep an updated background model even in situations of high traffic density, where vehicles spend more time in the scene. Our algorithm is capable of identifying these situations and adapting the background update, improving the performance of update schemes that work at the pixel level. Fig. 10 illustrates the effect of applying the proposed conditional background update approach compared to pixel-level strategies, such as the running-average [8] or the layered conditional update [38], in a challenging example scenario.

IV. Three-Dimensional Vehicle Tracking and Classification

The multicue mask is used as the observation of the tracking process that estimates the position and volume of vehicles. Tracking is carried out in a two-step process that first obtains 2-D bounding boxes of the projection of the vehicles in the road plane and then estimates their 3-D volume according to the calibration of the camera. Both steps are modeled as dynamic systems in which the values of a set of random variables must be estimated using instantaneous observations, namely, the multicue mask, and a prior knowledge of the scene, mainly given by the spatiotemporal coherence of estimations. A previous version of the proposed strategy is described in more detail in [39]. The next sections enter into details about the design of the estimation filters associated with each step as well as a brief discussion about the necessary preprocessing and calibration.

A. Image Rectification and Camera Calibration

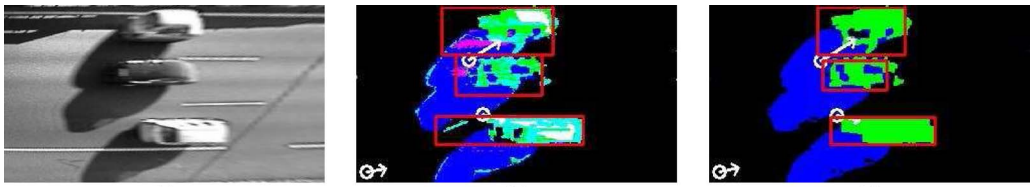
First, the radial distortion of the lens must be corrected to make the imaged lines actually correspond to the lines in the road plane. It can be done by defining five points on the original image $\mathbf{p}_i$, $i = 1, \ldots, 5$ that are collinear in the road plane. In our case, we ignore the tangential distortion, since its effect is negligible in most commercial surveillance camera lenses, and use the following second-order distortion model:

$$\mathbf{p}'_i = \mathbf{p}_i \left(1 + K\|\mathbf{p}_i\|\right) \tag{1}$$

where K is a parameter that can be determined through an optimization process that minimizes the sum of distances between points. We solve this using the Levenberg–Marquardt algorithm [40].

On the other hand, the perspective correction (see Fig. 11) is solved through a rectification of the road using a planar homography obtained from four points manually selected in the image. Next, the relation between pixel size and real world distances (in the longitudinal dimension of the road) is established manually based on elements with known length. Finally, the lane lines are also marked so that the system can detect in which of the lanes the vehicle projections lie.

This procedure is simple and practical for real installations but satisfactory enough to obtain valid height values for classification purposes. If necessary, to obtain a more accurate



(a) (b) (c)

Fig. 7. Shadow direction reinforcement algorithm.

1: $blobs \leftarrow$ get non-$background$ blobs [37]
2: $S \leftarrow$ get image with $shadow$ & $black$ regions
3: $F \leftarrow$ get image with $foreground$, $highlight$ & $white$ regions
4: **for all** $blobs$ **do**
5: $G \leftarrow$ paint single blob in an image
6: $S_1 \leftarrow G \cap S$
7: $F_1 \leftarrow G \cap F$
8: $s \leftarrow$ calculate centroid of S_1
9: $f \leftarrow$ calculate centroid of F_1
10: $\alpha \leftarrow$ calculate angle of vector joining s and f
11: **end for**
12: $\bar{\alpha} \leftarrow$ calculate mean from α angles & previous $\bar{\alpha}$ when $blobs \neq 0$. Otherwise, maintain previous $\bar{\alpha}$
13: $G_1 \leftarrow$ get image with blobs containing $black$, $foreground$, $highlight$ & $white$ regions
14: $G_2 \leftarrow$ displace G_1 in $\bar{\alpha} + \pi$ direction N pixels
15: Add $(G_1 - G_2)$ to $Mask$ as $highlight$ and set the rest of the mask as $shadow$
16: **return** $Mask$

Fig. 8. Shadow direction estimation and mask reinforcement. Image (b) shows in the bottom left corner the mean shadow direction, and (c) the *highlight* reinforcement and *shadow* "cleaning" set to the blobs from it.

1: $C \leftarrow$ paint $blobs_{tracked}$ distinguishing $static$ and $moving$ blobs with different gray-level values
2: **for all** xy pixels in $< I, I_{t-1}, B, \Sigma_l, \Sigma_c, T, C >$ **do**
3: **if** $C_{xy} = static$ **then**
4: $T_{xy} = T_{xy} + \Delta t$
5: **if** $T_{xy} > t_{update}$ **then**
6: $B_{xy} = I_{xy}$
7: $T_{xy} = 0$
8: **end if**
9: **else if** $C_{xy} = moving$ **then**
10: $T_{xy} = 0$
11: **else**
12: $(\Delta r, \Delta g, \Delta b) = (I - B)_{xy}^{(r,g,b)}$
13: $\Delta d_B = \sqrt{\Delta r^2 + \Delta g^2 + \Delta b^2}$
14: $factor = l_{rate}/\Delta d_B$
15: $B_{xy}^{(r,g,b)} = B_{xy}^{(r,g,b)} + factor \cdot (\Delta r, \Delta g, \Delta b)$
16: **end if**
17: $\Delta y = |I_{xy}^{y} - (I_{t-1})_{xy}^{y}| - (\Sigma_l)_{xy}$
18: $\Delta c = \sqrt{(I_{xy}^{c_x} - (I_{t-1})_{xy}^{c_x})^2 + (I_{xy}^{c_y} - (I_{t-1})_{xy}^{c_y})^2} - (\Sigma_c)_{xy}$
19: $\Delta d_\Sigma = \sqrt{\Delta y^2 + \Delta c^2}$
20: $factor = l_{rate}/\Delta d_\Sigma$
21: $(\Sigma_l)_{xy} = (\Sigma_l)_{xy} + factor \cdot \Delta y$
22: $(\Sigma_c)_{xy} = (\Sigma_c)_{xy} + factor \cdot \Delta c$
23: **end for**
24: **return** B, Σ_l, Σ_c

Fig. 9. Conditional background update algorithm.

calibration, we could as well include metric information of the normal direction of the road plane by selecting a vertical reference on the image.

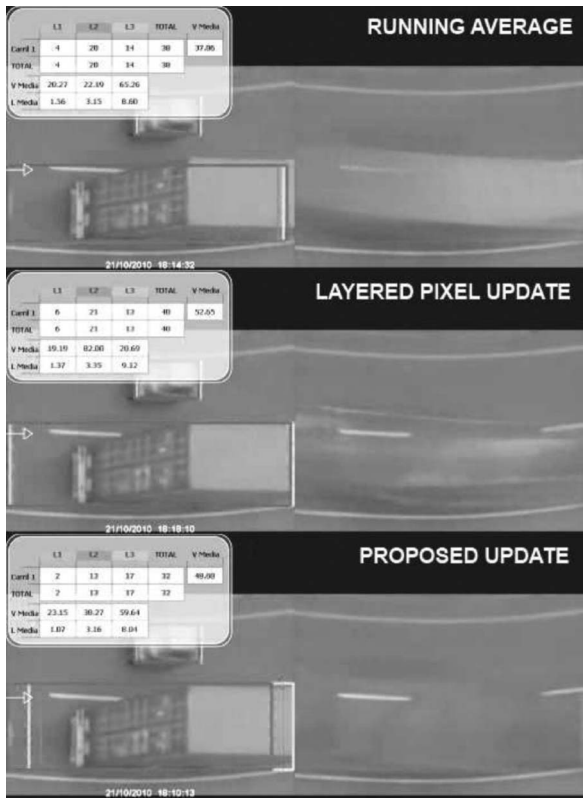


Fig. 10. Comparison between different background update methods. The upper image represents the basic approach at pixel level. The middle image depicts the result after applying the proposed algorithm but without the use of regions (pixel level), and the bottom image shows the excellent result provided by the procedure described in Fig. 9.

(a) (b)

Fig. 11. Results of (a) camera lens distortion correction and (b) perspective correction.

B. Linear Tracking—2-D Silhouette Estimation

In this step, the state vector is defined as a set of rectangles on the rectified plane that characterize the position and size of the projection of the vehicles in these images. Provided that the rectification process has removed perspective and affine distortion from the images, the motion of the vehicles can be

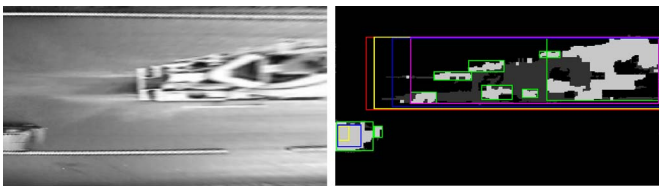


Fig. 12. Example of the box candidate merging process. The box colors are *previous* = red, *prediction* = yellow, *candidates* = green, *merged* = magenta, and *final* = blue.

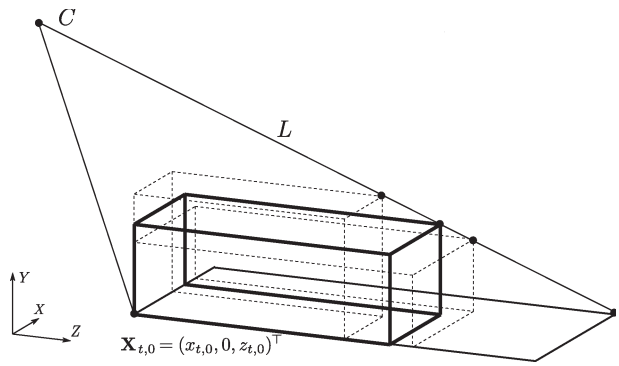


Fig. 13. Projective ambiguity. A given 2-D observation in the $Y = 0$ plane of a true 3-D cuboid (with thick solid lines) may also be the result of the projection of a family of cuboids (with dotted lines) with respect to camera C.

safely modeled as a linear process, and a Kalman filter can be used for each vehicle. The state vector for each vehicle is defined as $\mathbf{x}_t = (x_t, y_t, w_{2d}, l_{2d}, \dot{x}_t, \dot{y}_t)^\top$, which is a traditional position–velocity model.

First, the bounding boxes of segmented blobs, ignoring *shadow* regions, with at least a certain area $A_{\min}$, are detected using the approach described in [37]. Then, a process to merge these observed box candidates is undertaken, depending on the region of the image in which they lie, as well as depending on whether they can be considered as new observations corresponding to already tracked boxes.

Thus, the suitability for merging the observed box candidates is checked by analyzing their overlapping with respect to the boxes predicted by the Kalman filters applied to the tracked boxes. The decision to merge is done by measuring the area and height of the overlapping regions. If they are at least a certain amount in both cases, then box candidates are merged into a bigger one. The resulting boxes are considered to be the current observations of the tracked ones and are used to update the vehicle models through Kalman filters. Tracked vehicles that disappear from the image at one of the sides of the image are considered as vehicles that have exited the scene and hence are deleted from the tracking list. Fig. 12 shows an example of this candidate merging process.

On the other hand, the observed box candidates that lie in regions where new vehicles could appear and have not been matched with any previously tracked boxes are considered to be new vehicles if they meet certain geometric requirements. Thus, they are joined recursively into larger bounding boxes that satisfy $d_x < t_X$, $d_y < t_Y$ and $h_{\text{candidate}} \leq \nu h_{\text{lane}}$, where d_x and d_y are the minimal distances in X and Y from box to box, and t_X and t_Y are the corresponding distance thresholds. The projection's height is h, ν is a constant proportion, and "candidate" refers to the resulting rectangle. The recursive process stops when no larger rectangles can be obtained that meet both conditions. The resulting rectangles will feed the Kalman filters for 2-D tracking.

Note that this tracking process does not limit the size of candidates, i.e., the length of vehicles according to the transformed view's layout, because large vehicles, such as trucks or buses, may have very different lengths without a clear length limit. We have found that checking the spatiotemporal behavior of the tracked objects allows us control of vehicles or other object types, such as raindrops or insects on the camera's lens. Additionally, it is also useful for detecting traffic anomalies such as vehicles driving in the wrong direction.

C. 3-D Volume Estimation

The aim of this step is to refine the obtained estimations of the state vector associated with each vehicle according to its expected 3-D volume. This step is necessary to consider the potential distortion that the perspective effect causes on the estimated dimensions of the vehicles.

The proposed solution is based on the following principle: there are infinite points on the ray that are projected in the same image point and therefore correspond to a solution to the parameters of the cuboid, as shown in Fig. 13. Nevertheless, there are a number of constraints that bind the solution to a segment of the ray: positive and minimum height, width, and length. Furthermore, only some particular width, height, and length configurations are realistic representations of existing vehicles, such as cars, trucks, etc. Although some works have defined vehicle models more detailed than our box model [41], we propose to use a set of configurations that describe predefined vehicle models. We propose to define a posterior density function that identifies the probability of each point of the ray to represent the correct box hypothesis.

We propose to define a posterior density function that gathers these sources of information and identifies the probability of each point of the ray to yield the correct volume hypothesis.

The posterior distribution can be defined following Bayes' rule [33] as

$$p(\mathbf{x}_t|\mathbf{z}_{1:t}) \propto p(\mathbf{z}_t|\mathbf{x}_t) \sum_{i=1}^{N_s} p\left(\mathbf{x}_t|\mathbf{x}_{t-1}^i\right) \tag{2}$$

where $\mathbf{x}_t = (w_t, h_t, l_t)^\top$ is the state vector defining each vehicle as a measure of its width, height, and length; and $\mathbf{z}_t$ is the 2-D box obtained from the previous stage. The likelihood function can be expressed as a function of the distance to the projected ray (being maximum if the defined box exactly projects into the 2-D observed box). The prior function $p(\mathbf{x}_t|\mathbf{x}_{t-1}^i)$ can be defined, for simplicity, as a normal distribution around the previous state $\mathbf{x}_{t-1}$.

The likelihood function must be any function that fosters volume hypotheses near the reprojection ray. For the sake of simplicity, we choose a normal distribution on the point-line distance. The covariance of the distribution expresses our confidence about the measurement of the 2-D silhouette and

the calibration information. The likelihood function can be written as

$$p(\mathbf{z}_t|\mathbf{x}_t) \propto \exp\left((\mathbf{y}_t - \mathbf{x}_t)^\top S^{-1}(\mathbf{y}_t - \mathbf{x}_t)\right) \tag{3}$$

where $\mathbf{x}_t$ is a volume hypothesis, and $\mathbf{y}_t$ is its projection onto the reprojection ray. The position of $\mathbf{y}_t$ can be computed from $\mathbf{x}_t$ as the intersection of the ray and a plane passing through $\mathbf{x}_t$ and whose normal vector is parallel to the ray. For this purpose, we can represent the ray as a Plücker matrix $L_t = \mathbf{a}\mathbf{b}^\top - \mathbf{b}\mathbf{a}^\top$, where $\mathbf{a}$ and $\mathbf{b}$ are two points of the line, e.g., the far-most point of the 2-D silhouette, and the optical center, respectively. These two points are expressed in the WHL coordinate system (i.e., width, height, and length). Therefore, provided that we have the calibration of the camera, we need a reference point in the 2-D silhouette. We have observed that the point with less distortion is typically the closest point of the quadrilateral to the optical center, whose coordinates are $\mathbf{X}_{t,0} = (x_{t,0}, 0, z_{t,0})^\top$ in the XYZ world coordinate system. This way, any XYZ point can be transformed into a WHL point as $\mathbf{x}_t = R_0\mathbf{X}_t - \mathbf{X}_{t,0}$. Nevertheless, the relative rotation between these systems can be approximated to the identity, since the vehicles typically drive parallel to the OZ axis. The plane is defined as $\pi_t = (\mathbf{n}_t^\top, D_t)^\top$, where $\mathbf{n}_t = (n_x, n_y, n_z)^\top$ is normal to the ray L_t, and $D_t = -\mathbf{n}_t^\top\mathbf{x}_t$. Therefore, the projection of the point on the ray can be computed as $\mathbf{y}_t = L_t\boldsymbol{\pi}_t$.

MCMC methods can be used to generate a sample-based approximation, as in (2), and from which we can obtain point estimates of the best volume hypothesis as the mean of the distribution. Nevertheless, we have found that using a predefined set of typical sizes of vehicles can reduce the number of samples to use and thus can be applied to dramatically increase the speed of the algorithm.

Therefore, we can reduce the state space to a discrete number of states, namely, $\{\mathbf{y}_m\}_{m=1}^M$, which are the projections on the ray of M models $\mathbf{x}_{t,m}$, such as car (1.42, 1.6, 4.2), bus (2.1, 3.4, 12), etc. In our case, we are using a set of $M = 24$ models to have enough granularity in the detections.

The posterior distribution of all the discrete states is evaluated, and the one with the highest value is selected as the best hypothesis and the value of the state vector at time t.

V. Tests and Discussion

For testing the performance of our approach, we performed two kinds of tests: 1) a background subtraction performance to check the sensitivity of our approach to scene variations with respect to recent alternatives and 2) a counting and classification test through a set of videos with different and challenging situations for vision-based systems (see Fig. 16).

A. Background Subtraction Performance

The recent alternatives with which our approach is compared to are the following: 1) the cylindrical RGB model described in [13] and [14]; 2) the conical RGB model described in [24]; 3) the HSV-based model used in [21]; and 4) a variant based on hue, lightness, and saturation (HLS), which also distinguishes luminance from chromaticity in a similar way to HSV but replacing the brightness (or value) by the lightness. To make a fair comparison of their sensitivity to scene changes, we integrate these background luminance and chromaticity modelings into the same framework of disparity images presented in this paper. This way, we let the system choose automatically the threshold to separate background from foreground.

Fig. 14. Some samples of 3-D tracking results.

The parameters of each background subtraction approach are determined experimentally so that false alarm rates are kept below 0.1% in the same way as explained in Section III, whereas vehicle segmentations have a similar quality in all methods. This comparison is carried out using the perturbation method [13], which measures the percentage of image pixels considered as foreground (without distinguishing subclasses) with respect to the total number of pixels of the image when a random color perturbation is applied to them. This random perturbation is made by adding a Δ size vector with random direction to each pixel in RGB color space. If Δ is high, almost all pixels of the image should be cataloged as foreground for all methods. This way, we can measure the sensitivity of background subtraction methods in detecting low contrast targets against the learned background. The lower the Δ values that are needed for detecting foregrounds, the more sensitive the background subtraction approach will be, and hence the better the results. Therefore, for this test, it is not necessary to observe scenes with moving vehicles. Observing background scenes is enough as foreground pixel candidates are artificially generated. During the observation of the background, we increase the value of Δ by 1 from frame to frame, where its units correspond to pixel values with 8 bits per color channel. During the perturbation period, the background is not further updated.

Fig. 15 shows the sensitivity results obtained in the five models in four videos with different types of backgrounds without moving vehicles present: 1) a scene with noticeable global illumination changes due to the sun disappearing behind clouds and appearing again; 2) a scene with diffuse illumination and noticeable image noise particularly around the white color due to analog–digital conversion; 3) a scene with diffuse illumination but with less contrast; and 4) a nighttime scene recorded in gray level instead of color. These are short sequences of a few seconds, in which if perturbations are not applied, all the

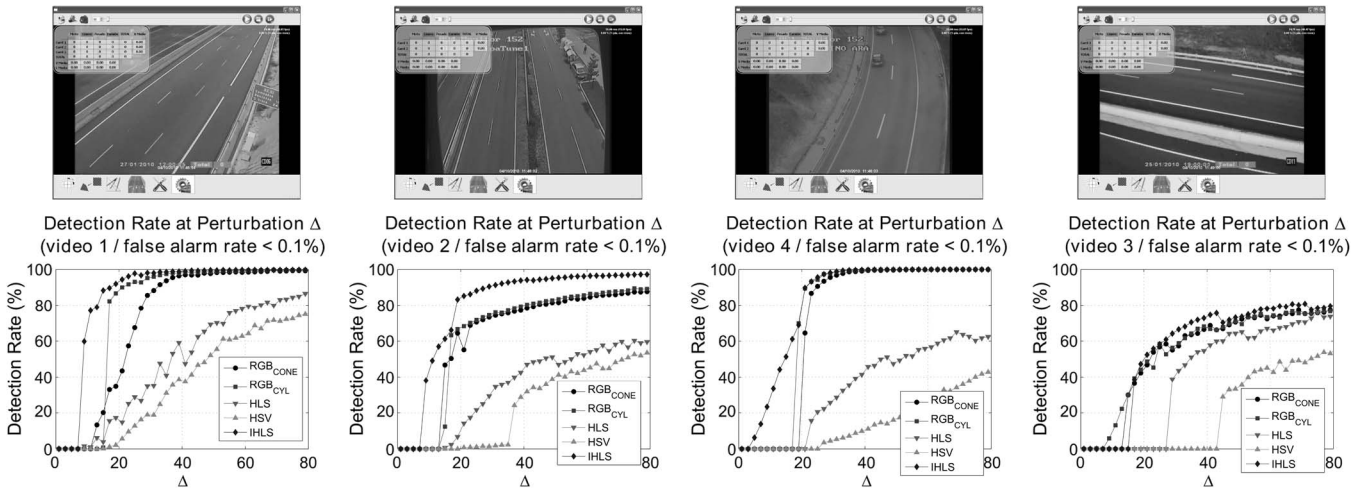


Fig. 15. Color model sensitivity comparison of the FG/BG disparity approach using IHLS, conical RGB, cylindrical RGB, HSV and HLS luminance, and chromaticity models through the color perturbation method [13] applied in four different road videos.

pixels are labeled as background. The higher Δ is, the more the chance that background subtraction methods will have to label pixels as foreground. Thus, it can be observed how, in general, our approach (marked as IHLS) needs lower perturbations for detecting higher rates of foreground pixels than the rest of the evaluated methods, particularly if color images are used.

Additionally, to check the sensitivity and influence of segmentation parameters on the performance of background subtraction, we have observed how the system reacts to this test with different parameter values. Most sensitive approaches continue being those of cylindrical RGB, conical RGB, and IHLS models, with slight differences among them. However, taking into account the results obtained in [23] for IHLS with respect to RGB and normalized RGB color models for image segmentation in low saturation regions, we consider that IHLS is the better option.

B. Counting and Classification Results

Fig. 16 shows an image of each of the four videos used for this test: 1) a nightfall with diffuse ambient light in which vehicles have headlights turned on (called *Nightfall*); 2) a sunny day where vehicles project shadows (called *Sunny*); 3) a rainy day in which the sun appears and causes vehicles to project shadows (called *Transition*); and 4) a rainy day that includes a traffic jam in which, at times, vehicles are completely stationary in some lanes (called *Jam*). Each video has a duration of 5 min, and the traffic flow in all of them is significantly dense, with several passing maneuvers and vehicles in parallel. The vehicle types include cars, motorbikes, heavy trucks, articulated trucks, vans, and buses, but for this test, we consider three classes, depending on tracked region geometric characteristics: *Two Wheels*, *Light Vehicle*, and *Heavy Vehicle*. Thus, *Two Wheels* and *Light Vehicle* categories will have the same length limit but different width and height limits. Vehicles longer and taller than these will be considered as *Heavy Vehicles*.

In this test, we have compared our segmentation approach with other vision-based recent alternatives, maintaining the same tracking procedure: 1) modified codebook (MCB), which is a modified version of [14]; and 2) modified fusion (MFS), which is a modified version of [24]. Both methods have been adapted to our context to update the background and use adaptive luminance and chromaticity thresholds in the same way as ours, which is referred in the test as adaptive multicue (AMC). The main differences between MCB with respect to the color-only part of the AMC come from the use of cylindrical RGB color space instead of IHLS and the segmented categories. In the case of MFS, differences with respect to AMC come from the use of conical RGB color space instead of IHLS, the way in which different cues are fused and the segmented categories. All approaches maintain the same parameters for all the scenarios, as in the case of a real installation.

Fig. 16. Samples of (a) *Nightfall*, (b) *Sunny*, (c) *Transition*, and (d) *Jam* videos used for testing the performance of our approach for vehicle counting and classification.

Table I shows the vehicle counting and classification results obtained in the test, where I_R and I_D refer to the real and detected number of vehicles, F_N and F_P refer to the false negative and positive detections, and P, R, and F refer to the precision, recall, and F-measure, respectively, expressed as

$$P = \frac{T_P}{I_D} \qquad R = \frac{T_P}{I_R} \qquad F = 2 \cdot \frac{P \cdot R}{P + R} \tag{4}$$

where T_P refers to the obtained true positives.

It can be seen that in these videos, we obtain overall vehicle precision, recall, and F-measure results of 96.25%, 92.69%, and 94.44%, respectively, which is significantly beyond state-of-the-art vision approaches such as MCB and MFS, which is also in a similar range to those achievable by current ILD systems and is thus satisfactory enough to consider a replacement of an ILD with this vision system.

TABLE I
RESULTS OBTAINED FOR THE COUNTING AND CLASSIFICATION OF TWO WHEELS (TW), LIGHT VEHICLE (LV), AND HEAVY VEHICLE (HV) CATEGORIES

GROUND TRUTH		*NIGHTFALL*			*SUNNY*			*TRANSITION*			*JAM*			**TOTAL**		
I_R	TW	17			11			0			7			35		
	LV	442			320			212			284			1258		
	HV	17			7			16			22			62		
MCB		I_D	F_N	F_P	I_D	F_N	F_P	I_D	F_N	F_P	I_D	F_N	F_P	I_D	F_N	F_P
Detections	TW	31	12	10	37	6	9	0	17	9	25	4	14	93	39	42
	LV	374	22	8	238	49	0	131	65	0	41	217	0	784	353	8
	HV	55	0	0	25	0	8	31	1	8	33	9	2	144	10	18
Confusion Matrices	$R \setminus D$	*TW*	*LV*	*HV*	*TW*	*LV*	*HV*	*TW*	*LV*	*HV*	*TW*	*LV*	*HV*	*TW*	*LV*	*HV*
	TW	4	1	0	5	0	0	0	0	0	3	0	0	12	1	0
	LV	17	363	40	23	236	12	8	129	10	8	39	20	56	767	82
	HV	0	2	15	0	2	5	0	2	13	0	2	11	0	8	44
Overall Results		*P(%)*	*R(%)*	*F(%)*	*P(%)*	*R(%)*	*F(%)*	*P(%)*	*R(%)*	*F(%)*	*P(%)*	*R(%)*	*F(%)*	**P(%)**	**R(%)**	**F(%)**
		83.04	80.25	81.62	82.00	72.78	77.12	87.65	62.28	72.82	53.54	16.93	25.73	**80.61**	**60.74**	**69.28**
MFS		I_D	F_N	F_P	I_D	F_N	F_P	I_D	F_N	F_P	I_D	F_N	F_P	I_D	F_N	F_P
Detections	TW	26	6	7	83	0	70	2	0	1	3	3	0	114	9	78
	LV	422	2	9	292	22	2	194	13	0	171	101	0	1079	138	11
	HV	37	0	1	13	0	0	28	0	8	36	1	2	114	1	11
Confusion Matrices	$R \setminus D$	*TW*	*LV*	*HV*	*TW*	*LV*	*HV*	*TW*	*LV*	*HV*	*TW*	*LV*	*HV*	*TW*	*LV*	*HV*
	TW	11	0	0	7	4	0	0	0	0	3	1	0	21	5	0
	LV	8	412	20	6	284	8	1	194	4	0	168	15	15	1058	47
	HV	0	1	16	0	2	5	0	0	16	0	2	19	0	5	56
Overall Results		*P(%)*	*R(%)*	*F(%)*	*P(%)*	*R(%)*	*F(%)*	*P(%)*	*R(%)*	*F(%)*	*P(%)*	*R(%)*	*F(%)*	**P(%)**	**R(%)**	**F(%)**
		90.52	92.23	91.36	76.29	87.57	81.54	93.75	92.11	92.92	90.48	60.70	72.66	**86.84**	**83.76**	**85.27**
AMC		I_D	F_N	F_P	I_D	F_N	F_P	I_D	F_N	F_P	I_D	F_N	F_P	I_D	F_N	F_P
Detections	TW	14	4	1	13	0	1	1	0	0	6	2	0	34	6	2
	LV	441	5	3	311	9	1	210	3	1	244	32	0	1206	49	5
	HV	16	0	0	6	1	0	15	0	0	28	1	0	65	2	0
Confusion Matrices	$R \setminus D$	*TW*	*LV*	*HV*	*TW*	*LV*	*HV*	*TW*	*LV*	*HV*	*TW*	*LV*	*HV*	*TW*	*LV*	*HV*
	TW	12	1	0	9	2	0	0	0	0	3	2	0	24	5	0
	LV	1	434	2	3	306	2	1	207	1	3	237	12	8	1184	17
	HV	0	3	14	0	2	4	0	2	14	0	5	16	0	12	48
Overall Results		*P(%)*	*R(%)*	*F(%)*	*P(%)*	*R(%)*	*F(%)*	*P(%)*	*R(%)*	*F(%)*	*P(%)*	*R(%)*	*F(%)*	**P(%)**	**R(%)**	**F(%)**
		97.66	96.64	97.15	96.67	94.38	95.51	97.79	96.93	97.36	92.09	81.79	86.63	**96.25**	**92.69**	**94.44**

The main difficulty in the *Nightfall* sequence is correctly segmenting the headlight reflections on the road with respect to the vehicle. The classification errors in light vehicles are mainly derived from this issue as the measured length in some cases includes parts of reflections. In these cases, the reflection intensity is very high, and therefore, some pixels are not classified as *highlight*. Therefore, the reflection is not completely cropped. In the case of MCB and MFS approaches, as they do not apply any postprocessing for removing this kind of region, they have more false positives and classification errors.

In the *Sunny* sequence, the main difficulty comes from the detection of dark vehicles that project shadows. In some cases, there may be some dark vehicles that do not have sufficient gradient features to identify that they are vehicles. MCB has particular problems in distinguishing these cases as it does not take into account cues other than color. On the other hand, MFS has many false positives cataloged as *Two Wheels* because many vehicle shadows have some regions, corresponding to the frontal window projected shadow, in which gradient cues are segmented as foreground regions with a similar size to those of *Two Wheels* category vehicles. In this case, it might have seemed desirable to apply extra morphologic operations, such as erosion, but it was not appropriate because other good regions would have been lost for the test. Thus, the proposed approach for improving the segmentation by estimating the shadow direction shows its relevance in this scenario.

In the *Transition* sequence, the main challenges come from the visible slipstreams behind vehicles due to the wet road and the sudden illumination changes, but the proposed system behaves well in most cases. On the contrary, both MCB and MFS have problems with sudden illumination changes as they do not handle them explicitly, provoking more false positives.

Finally, in the *Jam* sequence, the most challenging of all, the proposed conditional background update allows the effect of vehicle stops on the detections to be diminished in our approach. However, in the case of MCB and MFS, there are situations in which big vehicle candidates, which result from the jam and sudden illumination changes due to passing large trucks, occupy the entire image for a long time and do not let track other vehicles passing through those regions, which results in many false negatives. This occurs as passing vehicles do not let the segmented candidate to remain static a sufficient time to be absorbed as background. Another difficulty in this scenario is the occlusion that occurs when a large truck is in the middle lane and smaller vehicles pass it in the left lane, which happens frequently because of the dense traffic. This is another reason to have more false negative detections than in other cases. Additionally, this issue also generates negative and positive detection errors as the vehicles are very close among them and the tracking procedure joins segmented regions corresponding to different vehicles. Nevertheless, even if handling explicitly severe occlusions is beyond the scope of this paper, as vehicle segmentation quality is improved with respect to MCB and MFS, our approach obtains better counting and classification results also in this challenging scenario.

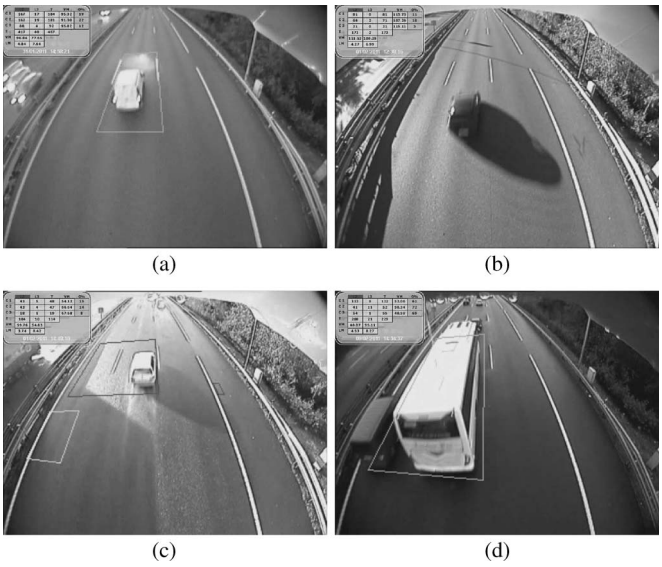


Fig. 17. Detection and classification failure examples in each video. (a) Wrong classification due to a bad segmentation of headlight reflection. (b) Nondetected dark vehicle. (c) A false positive due to sudden global illumination change. (d) Missed vehicle due to partial occlusion.

Fig. 17 shows some examples of these detection and classification failures. We also include the processed videos as supplementary material (available at http://ieeexplore.ieee.org). We encourage readers to view these videos as they provide better visualization of the experiments.

As mentioned in the previous section, the cylindrical RGB, conical RGB, and IHLS models have similar sensitivities when varying the segmentation parameters, and therefore, the obtained color masks have similar segmentation characteristics, with differences in low saturation regions [23]. The big differences arise when gradient cues are added and when masks are reinforced with procedures to handle local illumination changes explicitly. The parameters that control them are those of o_s, o_h, o_w, t_g, and t_f. We normalize the images to a predefined resolution to have the influence of t_g and t_f more under control for different installations. In practice, it is not necessary to have special care in setting their values so that they are optimal, because there is a wide range of t_g values that can work satisfactorily for our purpose, and small t_f values are sufficient to get compact enough vehicle segmentations. All the vehicle moving shadow projections we have observed have darker shadow regions closer to the vehicle itself; therefore, the value o_s can be maintained from installation to installation after setting it, as explained in Section III. On the other hand, it is recommended to set the o_h parameter with a small value so that the system can classify lighter regions with a high sensitivity. Those regions that correspond to illumination changes and not to vehicles with color lighter than the background, according to our experiments, also have other neighboring pixels classified with other categories than *highlight* and are thus removed by the cropping algorithm. The parameter o_w should be set so that almost absolute white regions are included in the $White$ category, which can be found in big trucks, but not usually in the case of headlight reflections.

Finally, regarding the computation time of our implementation, it runs at around 30 FPS using images captured at 768 $\times$ 576 but converted to 320 $\times$ 240 for processing on an Intel Core2 Quad CPU Q8400 at 2.66 GHz with 3-GB RAM and a NVIDIA 9600GT, which is sufficient for our purpose. The program has been implemented in C/C++ using OpenMP and CUDA for multicore and GPU programming, respectively. In our implementation, the average time consumption percentages of the different stages with respect to the total are, from higher to lower, the following: 1) color mask calculation, 32.3%; 2) edge mask calculation, 29.2%; 3) highlight cropping, 17.5%; 4) conditional update, 16.3%; 5) shadow reinforcement, 2.6%; 6) 2-D tracking, 2.0%; and 7) 3-D tracking, 0.1%. We have applied CUDA tools for the color and edge mask calculations and also for the conditional update. On the other hand, we did not observe significant changes on computational timing due to parameter variations, except in the case of t_f for the watershed blob filling, which in our implementation can decrease significantly the performance if the parameter value is big. However, we keep the resolution of the images under control, and we need only a small value to improve the segmented mask, so in practice, it is not a problem for the performance of the system.

VI. Conclusion

In this paper, we have presented a novel computer vision system devised to track and classify vehicles with the aim of replacing ILDs, particularly on highways. The system has been tested with different kinds of weather conditions (including rainy and sunny days that make passing vehicles cast shadows), obtaining similar results to ILDs. Additionally, this system distinguishes itself from other computer-vision-based approaches in the way in which it can handle casted shadows without the need for any hardware other than cameras, such as GPS to estimate the direction of the shadows. Hence, we believe that this is a viable alternative to replace ILDs, other technologies such as tags installed in vehicles, laser scanners that reconstruct the 3-D shape of the vehicles, or other computer-vision-based approaches, whose installation and maintenance are more cumbersome than using cameras only. GPU and multicore programming allow us to achieve real-time performances and with off-the-shelf hardware components.

To extend the approach to viewpoints in which more severe occlusions may occur, it would be necessary to include an interaction model between objects that can be achieved by adding Markov Random Field factors to the posterior distribution expression. Therefore, even when the image projections of two or more vehicles intersect, the 3-D model understands that they cannot occupy the same space at the same time.

References

[1] Nat. Renewable Energy Lab. (NREL), (2000) SOLPOS 2.0, Distributed by NREL. Center for Renewable Energy Resources. Renewable Resource Data Center. [Online]. Available: http://rredc.nrel.gov/solar/codesandalgorithms/solpos/

[2] S. Birchfield, W. Sarasua, and N. Kanhere, "Computer vision traffic sensor for fixed and Pan-Tilt-Zoom cameras," Transp. Res. Board, Washington, DC, Tech. Rep. Highway IDEA Project 140, 2010.

[3] C. C. C. Pang, W. W. L. Lam, and N. H. C. Yung, "A method for vehicle count in the presence of multiple-vehicle occlusions in traffic images," *IEEE Trans. Intell. Transport. Syst.*, vol. 8, no. 3, pp. 441–459, Sep. 2007.
[4] N. Buch, J. Orwell, and S. A. Velastin, "Urban road user detection and classification using 3D wire frame models," *IET Comput. Vis. J.*, vol. 4, no. 2, pp. 105–116, Jun. 2010.
[5] M. Haag and H. H. Nagel, "Incremental recognition of traffic situations from video image sequences," *Image Vis. Comput.*, vol. 18, no. 2, pp. 137–153, Jan. 2000.
[6] Z. Mayo and J. R. Tapamo, "Background subtraction survey for highway surveillance," in *Proc. Annu. Symp. PRASA*, Stellenbosch, South Africa, 2009, pp. 77–82.
[7] B. Johansson, J. Wiklund, P. Forssén, and G. Granlund, "Combining shadow detection and simulation for estimation of vehicle size and position," *Pattern Recognit. Lett.*, vol. 30, pp. 751–759, 2009.
[8] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 780–785, Jul. 1997.
[9] T. Horprasert, D. Harwood, and L. Davis, "A statistical approach for real-time robust background subtraction and shadow detection," in *Proc. IEEE ICCV*, 1999, pp. 256–261.
[10] C. Stauffer and W. E. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE CVPR*, 1999, vol. 2, pp. 246–252.
[11] J. Hu, T. Su, and S. Jeng, "Robust background subtraction with shadow and highlight removal for indoor surveillance," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2006, pp. 4545–4550.
[12] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *Proc. Eur. Conf. Comput. Vis. Part II*, 2000, vol. LNCS 1843, pp. 751–767.
[13] T. Chalidabhongse, K. Kim, D. Harwood, and L. Davis, "A perturbation method for evaluating background subtraction algorithms," in *Proc. IEEE Joint Int. Workshop VS-PETS*, 2003, pp. 1–7.
[14] K. Kim, T. Chalidabhongse, D. Harwood, and L. Davis, "Background modeling and subtraction by codebook construction," in *Proc. ICIP*, 2004, pp. 3061–3064.
[15] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, "Statistic and knowledge-based moving object detection in traffic scenes," in *Proc. IEEE Intell. Transp. Syst.*, 2000, pp. 27–32.
[16] S. Gupte, O. Masoud, R. Martin, and N. Papanikolopoulos, "Detection and classification of vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 3, no. 1, pp. 37–47, Mar. 2002.
[17] D. Aubert, F. Guichard, and S. Bouchafa, "Time-scale change detection applied to real-time abnormal stationarity monitoring," *Real-Time Imag.*, vol. 10, no. 1, pp. 9–22, Feb. 2004.
[18] R. Mech and J. Ostermann, "Detection of moving cast shadows for object segmentation," *IEEE Trans. Multimedia*, vol. 1, no. 1, pp. 65–76, Mar. 1999.
[19] A. Prati, I. Mikic, M. M. Tridevi, and R. Cucchiara, "Detecting moving shadows: Algorithms and evaluation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 7, pp. 918–923, Jul. 2003.
[20] D. Roller, K. Daniilidis, and H. H. Nagel, "Model-based object tracking in monocular image sequences of road traffic scenes," *Int. J. Comput. Vis.*, vol. 10, no. 3, pp. 257–281, Jun. 1993.
[21] R. Cucchiara, C. Grana, G. Neri, M. Piccardi, and A. Prati, "The Sakbot system for moving object detection and tracking," in *Video-Based Surveillance Systems-Computer Vision and Distributed Processing*, P. Remagnino, G. A. Jones, N. Paragios, and C. S. Regazzoni, Eds. Berlin, Germany: Springer-Verlag, 2001, ch. 12, pp. 145–157.
[22] I. Mikic, P. Cosman, G. Kogut, and M. Trivedi, "Moving shadow and object detection in traffic scenes," in *Proc. IEEE Int. Conf. Pattern Recognit.*, 2000, pp. 321–324.
[23] P. Blauensteiner, H. Wildenauer, A. Hanbury, and M. Kampel, "Motion and shadow detection with an improved colour model," in *Proc. IEEE Int. Conf. Signal Image Process.*, 2006, pp. 627–632.
[24] I. Huerta, M. Holte, T. Moeslund, and J. González, "Detection and removal of chromatic moving shadows in surveillance scenarios," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2009, pp. 1499–1506.
[25] J. C. McCall and M. M. Trivedi, "Video-based lane estimation and tracking for driver assistance: Survey, system, and evaluation," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 1, pp. 20–37, Mar. 2006.
[26] C. Maduro, K. Batista, P. Peixoto, and J. Batista, "Estimation of vehicle velocity and traffic intensity using rectified images," in *Proc. IEEE Int. Conf. Image Process.*, 2008, pp. 777–780.
[27] N. K. Kanhere, S. J. Pundlik, and S. T. Birchfield, "Vehicle segmentation and tracking from a low-angle off-axis camera," in *Proc. IEEE CVPR*, 2005, pp. 1152–1157.
[28] F. Bardet and T. Chateau, "MCMC particle filter for real-time visual tracking of vehicles," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, 2008, pp. 539–544.
[29] B. Coifman, D. Beymer, and P. McLauchlan, "A real-time computer vision system for vehicle tracking and tracking surveillance," *Transp. Res. Part C, Emerging Technol.*, vol. 6, no. 4, pp. 271–288, Aug. 1998.
[30] X. Zou, D. Li, and J. Liu, "Real-time vehicles tracking based on Kalman filter in an ITS," in *Proc. Int. Symp. Photoelectron. Detection Imag.*, 2007, vol. SPIE 6623, p. 662 306.
[31] P. L. M. Bouttefroy, A. Bouzerdoum, S. L. Phung, and A. Beghdadi, "Vehicle tracking by non-drifting mean-shift using projective Kalman filter," in *Proc. IEEE Intell. Transp. Syst.*, 2008, pp. 61–66.
[32] X. Song and R. Nevatia, "Detection and tracking of moving vehicles in crowded scenes," in *Proc. IEEE Workshop Motion Video Comput.*, 2007, pp. 4–8.
[33] Z. Khan, T. Balch, and F. Dellaert, "MCMC-based particle filtering for tracking a variable number of interacting targets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 11, pp. 1805–1819, Nov. 2005.
[34] I. Huerta, D. Rowe, and M. Mozerov, "Background subtraction fusing colour, intensity and edge cues," in *Proc. Conf. AMDO*, 2008, vol. LNCS 5098, pp. 279–288.
[35] E. Preteux, "Watershed and skeleton by influence zones: A distance-based approach," *J. Math. Imag. Vis.*, vol. 1, no. 3, pp. 239–255, Sep. 1992.
[36] A. M. Reza, "Realization of the contrast limited adaptive histogram equalization (CLAHE) for real-time image enhancement," *J. Math. Imag. Vis.*, vol. 38, no. 1, pp. 35–44, Aug. 2004.
[37] S. Suzuki and K. Be, "Topological structural analysis of digitized binary images by border following," *Comput. Vis., Graph., Image Process.*, vol. 30, no. 1, pp. 32–46, 1985.
[38] K. Kim, D. Harwood, and L. Davis, "Background updating for visual surveillance," in *Proc. ISVC*, 2005, vol. LNCS 3804, pp. 337–346.
[39] M. Nieto, L. Unzueta, A. Cortés, J. Barandiaran, O. Otaegui, and P. Sánchez, "Real-time 3D modeling of vehicles in low-cost monocamera systems," in *Proc. Int. Conf. Comput. VISAPP*, Algarve, Portugal, Mar. 2011, pp. 459–464.
[40] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[41] Y. Goyat, T. Chateau, and L. Trassoudaine, "Tracking of vehicle trajectory by combining a camera and a laser rangefinder," *Mach. Vis. Appl.*, vol. 21, no. 3, pp. 275–286, Apr. 2010.

Luis Unzueta (M'10) received the M.S. and Ph.D. degrees in mechanical engineering from the University of Navarra, Donostia-San Sebastián, Spain, in 2002 and 2009, respectively.

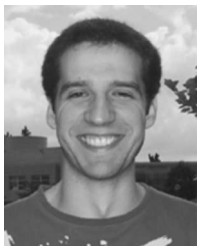
He is currently a Researcher with the Intelligent Transportation Systems and Engineering Area, Vicomtech, Donostia-San Sebastián. His current research interests include video surveillance systems and human–computer interaction.

Marcos Nieto received the M.S. and Ph.D. degrees in electrical engineering from the Technical University of Madrid, Madrid, Spain, in 2005 and 2010, respectively.

He is currently a Researcher with the Intelligent Transportation Systems and Engineering Area, Vicomtech, Donostia-San Sebastián, Spain. His actual research interests include optimization methods for probabilistic models in computer vision.

Andoni Cortés received the M.S. degree in computer science from the University of the Basque Country, Donostia-San Sebastián, Spain, in 2001.

He is currently a Researcher with the Intelligent Transportation Systems and Engineering Area, Vicomtech, Donostia-San Sebastián. His research interests include pattern recognition and augmented reality.



Javier Barandiaran received the M.S. degree in computer science from the University of the Basque Country, Donostia-San Sebastián, Spain, in 2004.

He is currently a Researcher with the Intelligent Transportation Systems and Engineering Area, Vicomtech, Donostia-San Sebastián. His research interests include augmented reality and 3-D reconstruction.

Oihana Otaegui received the M.S. and Ph.D. degrees in mechanical engineering from the University of Navarra, Donostia-San Sebastián, Spain, in 1999 and 2005, respectively.

She is currently the Head of the Intelligent Transportation Systems and Engineering Area, Vicomtech, Donostia-San Sebastián. Her research interests include satellite navigation and transport fields.

Pedro Sánchez received the B.S. degree in physics (electronics and automation) from the University of the Basque Country, Bilbao, Spain, in 2001.

He is currently the Coordinator of R&D projects with the Intelligent Transportation Systems and Security Business Unit, IKUSI–Ángel Iglesias S.A., Donostia-San Sebastián, Spain. His research interests include transport fields and automated systems.