

Document downloaded from:

<http://hdl.handle.net/10251/56628>

This paper must be cited as:

Albiol Colomer, AJ.; Sanchis Pastor, L.; Albiol Colomer, A.; Mossi García, JM. (2011).  
Detection of Parked Vehicles using Spatio-temporal Maps. IEEE Transactions on Intelligent  
Transportation Systems. 12(4):1277-1291. doi:10.1109/TITS.2011.2156791.



The final publication is available at

<http://dx.doi.org/10.1109/TITS.2011.2156791>

Copyright Institute of Electrical and Electronics Engineers (IEEE)

Additional Information

# Detection of Parked Vehicles using Spatio-Temporal Maps

Antonio Albiol, Laura Sanchis, Alberto Albiol, José M. Mossi

## Index Terms

Video Analysis, Surveillance, Traffic Planning, Traffic Image Analysis, Parked Vehicle Detection

## Abstract

This paper presents a video-based approach to detect the presence of parked vehicles in street lanes. Potential applications include detection of illegally and double-parked vehicles in urban scenarios and incident detection on roads. The technique extracts information from low-level feature points (Harris corners) in order to create spatio-temporal maps that describe what is happening in the scene.

The method does not rely on any background subtraction or perform any form of object tracking.

The system has been evaluated using private and public data sets and has proven to be robust against common difficulties found in CCTV video such as varying illumination, camera vibration, presence of momentary occlusion by other vehicles, and high noise levels.

## I. INTRODUCTION

Urban traffic is currently a real problem for most medium-sized and large cities. To reduce the problems caused by traffic congestion, intelligent transportation systems (ITS) are being deployed world-wide to achieve a more efficient use of existing infrastructures [1].

One problem that appears with high traffic densities (near congestion) is that any small incident may have a multiplicative effect. For this reason, the development of technologies that help authorities to react quickly to any type of incident is important. Among the technologies applied

This work was funded by the Spanish Government project MARTA under the CENIT program and CICYT contract TEC2009-09146.

ITEAM Research Institute, Universitat Politecnica de Valencia, SPAIN

to ITS, video offers some advantages such as the ability to cover wider areas. Also, most places of interest have monitoring cameras already installed. Moreover, the range of possible automatic analysis tasks that can be done using video is, by far, larger than with any other kind of sensor.

Traditionally, video cameras have mainly been used as a monitoring tool. However, when the number of cameras grows, it becomes unfeasible to monitor the cameras by humans. To overcome this issue, some tools for automatic traffic video analysis have begun to be offered commercially. Examples of a few companies that perform traffic video analysis include [2], [3], [4]. Unfortunately, it is difficult to evaluate the performance and limitations of these solutions since these companies do not often provide any details of their algorithms and implementations.

This paper presents a technique that uses video to automatically detect the presence of parked vehicles in street lanes. Parked vehicles can severely affect the traffic flow especially under near congestion conditions. Bear in mind that although an illegally parked vehicle only blocks one street lane, other vehicles driving down this lane are forced to move to adjacent lanes, thus perturbing the traffic flow in more lanes. Even though our system has an obvious law-enforcement application, we are more interested in reporting the state of the lane in order to be used as an input for *dynamic traffic planning*.

Dynamic traffic planning systems [5] achieve a better regulation of the whole infrastructure by using complex models with many input traffic variables. Some examples of input variables for such systems are: instantaneous and historic values of traffic data in different streets/roads (number of vehicles, average speed, mean stopping times, queue lengths,...); and the number of *available* lanes (not occupied by parked vehicles) on each avenue. This last information is what can be obtained from the algorithm presented in this paper.

Detection of parked vehicles involves detecting objects that remain stopped for more than a certain time. In urban scenarios, the minimum stopping time must be longer than traffic light cycles. The algorithm proposed in this paper has been evaluated with two data sets, one private and one public, described in detail in Sect. VI. Figure 1 shows sample images from two scenarios of the private data set. In the images, the areas of interest where vehicles park illegally have been highlighted .



Figure 1. Two example scenarios indicating the lanes under analysis.

## II. PREVIOUS WORK

The problem of detecting illegally and double-parked vehicles has been gaining more attention in the literature, especially since the i-LIDS vehicle detection challenge was released in 2007 [6].

To our knowledge, almost all previous approaches use some type of background model to detect objects of interest. Once foreground objects are extracted, most approaches use some sort of object tracking. An interesting survey on object tracking can be found in [7].

Examples of algorithms that use background subtraction and object tracking include [8], [9], [10], [11], [12], [13]. For instance, in [8], Bevilacqua and Vaccari present a method to detect stopped vehicles based on the detection of the tracked object's centroid position during short time intervals. They use background subtraction to detect foreground objects. Objects are tracked using an optical-flow motion estimation technique. Then, the centroid position of each car is analyzed within short time intervals. If the centroid remains within a small area for a short time, then the object is considered to be static for a short stability interval. If the object trajectory contains many contiguous short stability intervals, then a long stability interval is detected. The duration of this long stability interval is used to measure the full vehicle stopping time. In our opinion, this approach will have problems if the background changes during the time that a vehicle is parked, either because it is parked for a long time, or because the illumination is changing fast. The stopping times reported in [8] seem to be far shorter than the ones found in the real world.

In [9], Maddalena and Petrosino adopt a self-organized model for both the scene background and foreground in order to distinguish between moving and stopped foreground regions. In [10], the authors employ a DSP-based system for automatic visual surveillance. Block-matching motion detection and MOG-based foreground extraction are proposed in that work.

Other methods propose general frameworks for static object detection in which parked vehicles are a particular case. For example, in [11], Guler et al. (from the company IntuVision [14]) propose using an object tracker that is specially adapted to detect stationary objects. A scene description layer is used. Essentially, it contains the background in the absence of vehicles. Each time that a pixel becomes stationary after having had motion, it is compared with the background; if it is similar to it, then the probability that it belongs to a static object is decreased; if it is quite different, then the probability of being a static foreground object is increased. The algorithm has been tested using the i-LIDS sample data set (section VI-B) for abandoned bags and parked vehicle scenarios. In our opinion, this scheme cannot deal with long parking times or fast illumination changes. A fast change in the appearance of the background would be considered a parked car. On the other hand, if the background changes while an area is occupied by a parked car, the background shown when the car leaves will be different from the one learnt before the car arrived. Moreover, the requirement of obtaining a background image with no cars is not very practical for most busy city scenarios.

In [12], Venetianer et al. propose using the ObjectVideo tracker [15] to detect static objects. ObjectVideo implements a series of generic algorithms for object tracking that can be used for different purposes. For the case of parked vehicles, they used the sample subset of i-LIDS (see section VI-B). Although good results are reported on the sample subset, they recognize that these results do not correspond to the whole i-LIDS data set. In fact, they say that they have purchased the whole Parked Vehicle data set but conclude that further investigation is needed to see how to use ObjectVideo modules on this data. Finally, in [13], tracking is performed in 1-D after a transformation that projects the pixels of one lane of interest.

An example of an approach that uses background subtraction without any object tracking is presented by Porikli in [16]. This method uses two different time constants to estimate two backgrounds using different time-constants. Short-term background captures objects that have recently stopped and ignores moving objects, while long-term background models what is normally understood as background. The difference between the two backgrounds should contain

static objects. They have tested their approach with only one sequence of the sample i-LIDS data set of parked vehicles. Nothing is said about how to reestimate background when vehicles depart, or how to distinguish between vehicles and background in the case of long-term parking.

In our opinion, systems that use pixel-based background subtraction are not appropriate for detecting parked vehicles for long periods of time. If long constants of time are used to estimate background, then fast illumination changes such as those found on windy days with scattered clouds will cause the background to be invalid most of the time. On the other hand, short constants of time will incorporate stopped cars into the background too early. To illustrate the huge background variations in real scenarios, Figure 2 shows images from the two scenarios of Fig.1 at different times of the day and with different weather conditions (rainy, sunny, cloudy).

Most background subtraction methods require knowing if there are parked or moving cars during startup. This requirement constitutes a limitation for unsupervised deployments. Ideally, intra-frame object detectors can be really useful to solve the problem of detecting parked cars without background estimation. An intra-frame object detector is able to detect a specific kind of object using only one image. It is known that intra-frame object detectors exist for faces [17], [18] and pedestrians [19]. However, the huge variability in the aspect of vehicles, depending on point of observation, color, car model, etc. has made that there are no widely used intra-frame vehicle detectors. Nevertheless, some attempts in the literature have taken this direction [20]. An additional limitation of intra-frame detectors is that they usually require a minimum object size and, normally, this required size is larger than that found in traffic monitoring cameras.

### III. SYSTEM OVERVIEW

As mentioned in the introduction, the final goal is to detect the presence of parked vehicles in a particular street lane, rather than attempting to detect each individual vehicle. The street pavement normally appears as a smooth flat surface (except for the road markings). However, within the space that a car occupies in the image, it is possible to find more textured information. To take advantage of this fact, corner points are used as an important clue of the possible presence of vehicles. The use of corner points to distinguish foreground from asphalt was also proposed by [21] in the context of counting vehicles in motorways. We would like to stress that detecting features that are normally contained in cars, is different from precisely detecting cars. Upon careful reflection on the need to precisely segment individual cars in order to detect blocked

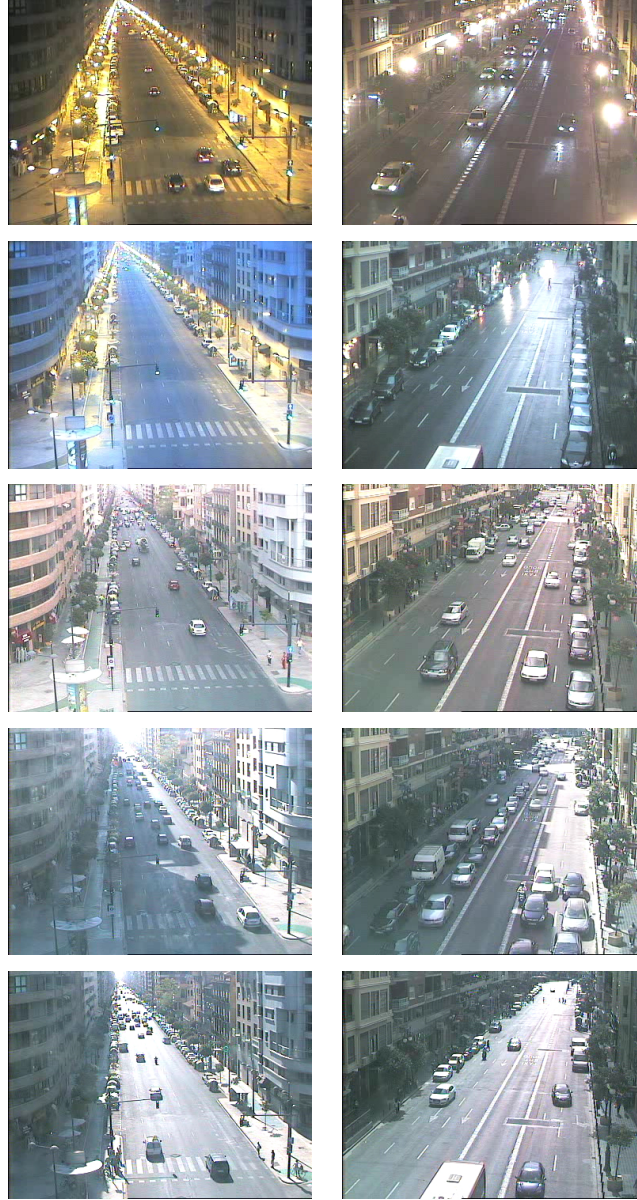


Figure 2. Sample views of two different scenarios at different times to illustrate the variations around the clock.

lanes, we have concluded that this requirement is excessive. Even though our proposal is an indirect approach, it is much more robust to changes in illumination and performs better in extremely difficult scenarios with high traffic densities where it is almost impossible to segment individual vehicles.

An additional reason for choosing not trying to detect vehicles, but features related with them, is the wide range of car sizes that it is necessary to cover due to perspective. In typical scenes



where cars approach or depart, vehicles at the bottom of the image have a relatively large size while those at the top have a very small size (widths of less than 5 pixels).

From the above reasoning, the algorithm starts by detecting all corner points using the Harris algorithm [22]. Hereafter, the shorter term *corners* will be used to refer to these feature points. These corners are then classified into two categories, static and dynamic. Dynamic corners most probably correspond to moving objects such as vehicles or pedestrians. However, static corners may appear on stopped cars and also on the background. Although the asphalt pavement is normally a smooth surface (without corners), the presence of road markings causes the detection of static corners at arrows, line ends, etc. Fortunately, since static cameras mounted on street poles are being used, these points always appear at the same locations and can be removed, as will be explained below.

From the point of view of traffic, a lane can be considered as a one-dimensional entity (in the longitudinal dimension). The area monitored by cameras located at junctions normally extends vertically (see the sample scenarios of Fig.1). Ignoring the transverse (horizontal) dimension of corners within the lane of interest, a one dimensional vector (for each frame) can be obtained. This vector has the same length as the longitudinal (vertical) dimension of the lane, and each element contains the number of corners at a given depth. Remember that these corners will likely correspond to vehicles. It is interesting to emphasize the information compression achieved by the process of discarding one coordinate. This greatly reduces the computational complexity, similarly to [13]. Appending these one-dimensional vectors for different frames allows the creation of spatio-temporal maps (2D) that contain the time evolution of the location of corners. The fine details about spatio-temporal maps will be presented in section IV. These maps are then conveniently filtered to take into account the fact that only vehicles that stop for a time longer than a threshold are being detected. Details about the filtering will be explained in section IV-C. Finally, after filtering, a classifier is applied using portions of maps corresponding to small time intervals to determine the lane occupancy.

The proposed system can be used for both real-time operation and off-line statistical analysis of lane occupancy. In real-time operation, the maps are filtered as they are built and the most recent portion of the map is used to make a decision about the current lane status.



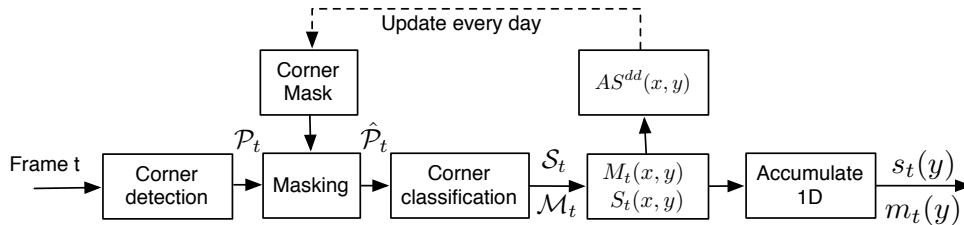


Figure 3. Flowchart of the processing performed for each frame.



Figure 4. a) Corners detected using Harris detector. b) Corners detected using Tomasi-Kanade features detector.

#### IV. DETAILED ALGORITHM DESCRIPTION

In this section, a detailed explanation of each of the modules will be presented.

##### A. Processing of each video frame

Figure 3 shows a flowchart with the processing performed for each frame. The rest of this section will describe these blocks in detail.

Given the frame at time  $t$ , the first step is to detect image corners. The well-known Harris corner detector [22] has been used. We have also tried the Tomasi-Kanade feature detector [23]. The performance in terms of detection of parked cars is the same as Harris'. In figure 4 it is shown the results of corner detection using both methods. The important point is that both of them produce large clusters of static corners at the locations of parked cars (although not exactly in the same locations).

In any case, there is no need for sub-pixel precision in this corner detection step. Good open source implementations of many corner detectors types can be found in the OpenCV Library [24]. The result of corner detection is a list of corner points  $\mathcal{P}_t$  for each frame. Next, a masking process selects a subset of relevant corners  $\hat{\mathcal{P}}_t$  from  $\mathcal{P}_t$ . Initially, the mask is manually configured to

cover the lane of interest as in Fig.1. The mask may also contain *holes* which are used to discard static corners at selected coordinates within the lane of interest. These coordinates correspond to background corners and are automatically detected. The process of automatic detection of background corners is described below.

In the next step, selected corners  $\hat{\mathcal{P}}_t$  are classified into static or dynamic, obtaining two disjoint lists of corners  $\mathcal{S}_t$  and  $\mathcal{M}_t$ , respectively. Any motion estimation method, such as block-matching or optical flow, could have been used to this end. However, a much simpler technique has been used because the magnitude or direction of the motion are not needed; it is only necessary to know whether or not the corner is in motion.

Hence, for each corner  $c_i \in \hat{\mathcal{P}}_t$ , with coordinates  $(x_i, y_i)$

$$d_i = \max_{(x', y') \in \mathcal{N}(c_i)} \{|I_t(x', y') - I_{t-1}(x', y')|\} \quad (1)$$

is computed, where  $I_t(x, y)$  is the image gray-level, and  $\mathcal{N}(c_i)$  is a small neighborhood around corner  $c_i$ . A neighborhood of size  $11 \times 11$  has been used in all our experiments. If  $d_i$  is over a threshold  $d_{th}$ , then  $c_i$  is classified as a dynamic corner; otherwise, as a static one. Typical values of  $d_{th}$  fall in the range of 10 to 25 for 8-bit quantized grayscale images. Note the great computational saving that this simple approach achieves compared with any possible motion estimation algorithm. It might be argued that this simple approach would fail in the case of high noise, sudden illumination change, or camera vibration. However, this happens rarely and, as will be explained later, our system does not rely on precise classification of corners but on the accumulation of static corners at a certain spatial location within a time interval (of the order of a few seconds). In fact, all these potential difficulties happen in the tested sequences and have not affected the algorithm performance.

Figure 5 shows an example with the result of our corner classification approach. In this example, the mask (light blue area) is set to cover most of the avenue and no corners are discarded within this area. It can be seen that dynamic (green) corners are located on vehicles in motion. On the other hand, static (red) corners tend to concentrate on stopped/parked vehicles (see, for instance, the white parked van on the right) and on road markings. It can also be seen that a smarter design of the mask, which had avoided the road lines between lanes, would have discarded most static background corners.

To automatically discard most of these static background corners, an accumulator image

$AS_d(x, y)$  is built during the day  $d$ . Each pixel in  $AS_d(x, y)$  counts the number of static corners detected at that specific coordinate. Let  $S_t(x, y)$  be a binary image which is built at instant  $t$  using the list of static pixels  $\mathcal{S}_t$  as follows:

$$S_t(x, y) = \begin{cases} 1 & \text{if } \mathcal{S}_t \text{ contains a static corner at } (x, y) \\ 0 & \text{otherwise} \end{cases} . \quad (2)$$

Then, the accumulator of static points  $AS(x, y)_d$  is built as:

$$AS_d(x, y) = \sum_{t=0}^{N_t} S_t(x, y) \quad (3)$$

where  $N_t$  denotes the number of frames in one day. Notice that  $AS_d(x, y)$  can be incrementally computed during the day and the images  $S_t(x, y)$  do not need to be stored.

Large values in  $AS_d(x, y)$  correspond to corners on the road, and possibly to corners on vehicles that have been parked for a really long time. This criterion is used to select which points will be ignored during the next day ( $d + 1$ ). A threshold value corresponding to 2 hours has been used.

It is possible to think that discarding background corners caused by parked cars on day  $d - 1$  might be a problem for detecting corners on day  $d$ . However, there is a small probability that parked vehicles on day  $d$  contain a corner at *exactly the same coordinates* as one that parked on day  $d - 1$ . Even in this case, since many corners are found within a car, the probability that *all corners* match these masked coordinates is almost null. Figure 6 shows an example of locations where the accumulator of static corners  $AS_d(x, y)$  is above the threshold.

Points have been dilated for a better visualization. As illustrated in Fig.3, the mask is updated once every day using  $AS_d(x, y)$ .

In a similar way, using the list of dynamic corners  $\mathcal{M}_t$ , it is possible to obtain a binary image of dynamic corners  $M_t(x, y)$ :

$$M_t(x, y) = \begin{cases} 1 & \text{if } \mathcal{M}_t \text{ contains a dynamic corner at } (x, y) \\ 0 & \text{otherwise} \end{cases} . \quad (4)$$

Dynamic corners will also be of interest in the determination of whether or not the lane is blocked as is shown in the next section.

The processing of each frame finishes by marginalizing the traverse coordinate  $x$  in both  $S_t(x, y)$  and  $M_t(x, y)$ :

$$s_t(y) = \int S_t(x, y) dx \quad (5)$$

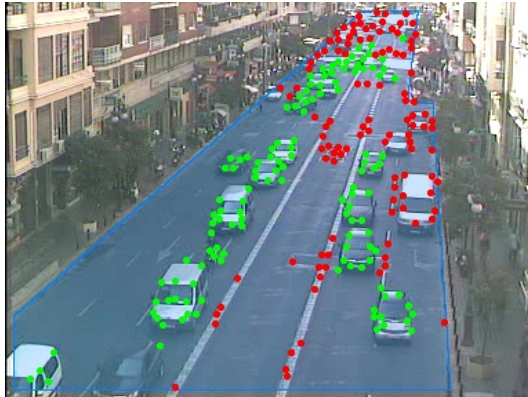


Figure 5. Corner detection and classification sample results. Static corners are shown in red, while moving corners are depicted as green.



Figure 6. Locations (in red) where static corners will be discarded based on the accumulator of the day before. Notice that some points are located on road markings or dirt. Points on smooth areas are due to cars that were parked at those locations the day before for a long enough time.

$$m_t(y) = \int M_t(x, y) dx \quad (6)$$

### B. Raw Map formation

After processing each frame, the next step is to fuse information from different time instants into a compact representation. For that purpose, spatio-temporal maps are built for both static and dynamic corners by appending the column vectors  $s_t(y)$  and  $m_t(y)$  of different time instants horizontally. Therefore, the vertical dimension of the maps corresponds to depth (longitudinal dimension of the lane) and the horizontal to time. In [25], we also proposed the idea of spatio-

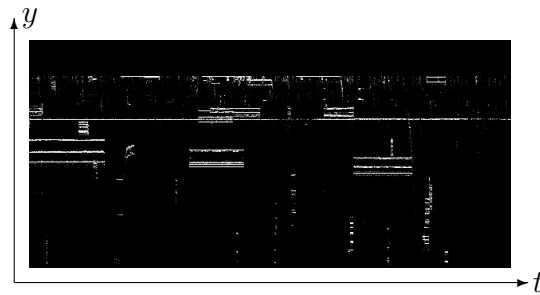


Figure 7. Example of static raw spatio-temporal map corresponding to 60 minutes. The horizontal axis corresponds to time while the vertical corresponds to the longitudinal lane dimension.

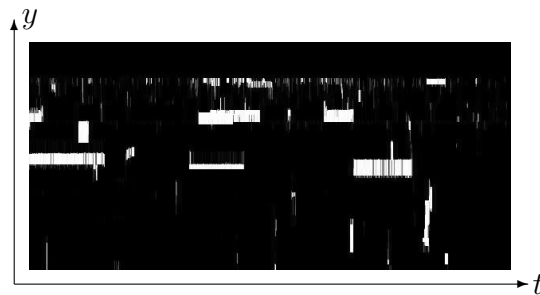


Figure 8. Result of the spatial filtering applied to the static map of Fig.7.

temporal maps for counting people. However, the information embedded on those maps was completely different.

The initial spatio-temporal maps are low-pass filtered in the time direction and subsampled in time so that each column corresponds to the summary of a 5-second interval. This time resolution is more than enough for our application. Spatio-temporal maps will be denoted as  $M_{map}(t, y)$  and  $S_{map}(t, y)$  for dynamic and static maps, respectively.

The spatio-temporal maps obtained as described above will be called *raw maps*. Figure 7 shows an example of a raw static spatio-temporal map  $S_{map}(t, y)$  for the lane of the scenario of Fig.1a. Note that spatio-temporal maps are *not* binary images; the value of a pixel  $(t, y)$  in the map represents the number of corner points (static or dynamic depending on the type of map) that have contributed in a 5-second interval around instant  $t$  and at a fixed depth  $y$ .

### C. Spatial filtering of static raw maps

As Fig.5 shows, corners on a vehicle do not form a dense grid. This implies that corners on a car might not appear as a connected region in a raw spatio-temporal map. This issue is clearly



Figure 9. Example of static spatio-temporal map after spatial (vertical) filtering and temporal (horizontal) filtering.

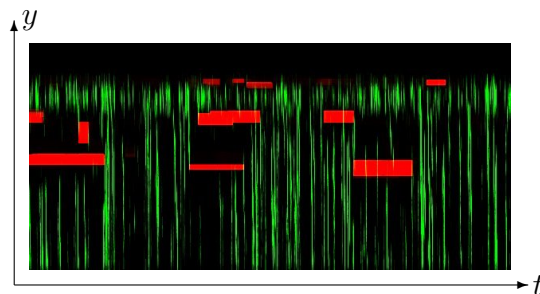


Figure 10. Combined spatio-temporal map obtained from Fig.9. The static map is in red whereas the dynamic map is in green.

visible in Fig.7. These regions in the map can be connected if it is assumed that there will be a maximum separation in the vertical dimension between the corners of a particular vehicle. A morphological (grayscale) close operator [26], with a vertical structuring element that is slightly longer than the maximum gap between corners of the same vehicle in the  $y$  dimension, is used for this purpose.

A careful look at Fig.7 shows the presence of some long horizontal lines that are not close to any other and have a height of one pixel. These lines correspond to background static corners and are normally caused by road dirt (that was not present the day before). These spurious horizontal lines can be easily removed with an additional filtering. A morphological grayscale open filter, with a vertical structuring element that is shorter than the minimum expected size of a vehicle is used to this end. Figure 8 illustrates the result of these two morphological filters applied to the spatio-temporal map of Fig.7. It should be mentioned that dynamic maps are not spatially filtered since the intrinsic motion of their corners integrates the information in the vertical dimension.

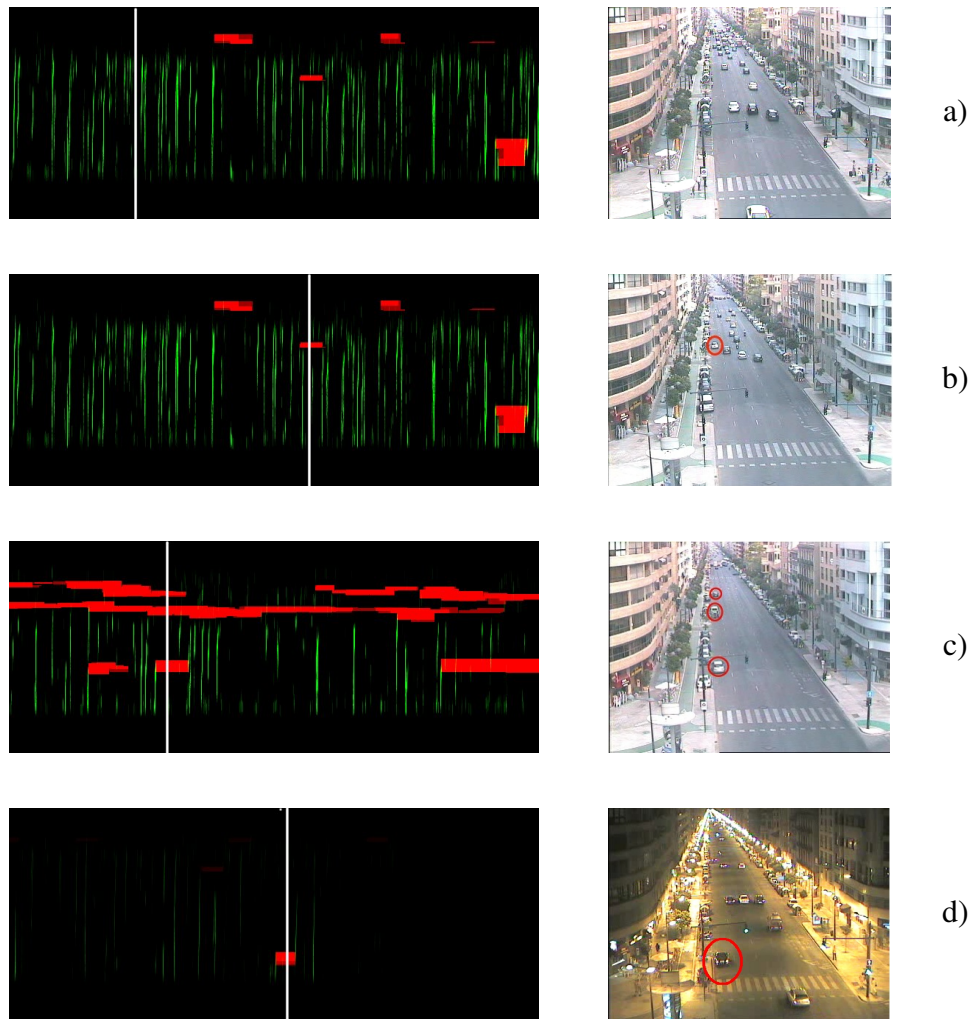


Figure 11. Combined map and camera view at the time marked by the white cursor line. a) In this case, no cars are parked on the left-most lane. b) In this case, one car is parked. Notice that the vertical coordinates of the parked car and the red region are the same. c) Three cars are parked on the lane under analysis at different depths. d) Despite the tremendous change in the illumination conditions, parked vehicles can still be detected. The absence of green lines indicates very low traffic during late night. All maps in this figure correspond to one hour.

#### *D. Temporal filtering of static spatio-temporal maps*

As illustrated in Fig. 8, after spatial filtering of a static map  $S_{map}(t, y)$ , many connected regions appear which correspond to stopped vehicles. The horizontal dimension of these regions is directly related to the time that they remained static. A morphological open with a horizontal structuring element is used to discard events that are shorter than a minimum parking time.

The output of the spatio-temporal filtering of  $S_{map}(t, y)$  will be denoted as  $\hat{S}_{map}(t, y)$  in the



sequel. The result of temporal filtering of the map in Fig. 8 is displayed in Fig. 9. In this example, a minimum stopping time of 1 minute (that corresponds to a structuring element of size 12) has been used. It is interesting to observe that temporal filtering introduces a delay in real-time operation. This delay is related to the fact that one must wait at least one minute to conclude that a vehicle has been stopped for one minute. Another point that it is interesting to note is how a temporal filtering of the order of magnitude of minutes can be efficiently implemented using spatio-temporal maps.

Dynamic spatio-temporal maps are not temporally filtered.

### *E. Combined spatio-Temporal Maps*

Maps of moving and static corners can be aligned both spatially and temporally. Therefore, it is possible to combine them into a single image using different color channels for better visualization. The red channel is used for the static map and the green one for the dynamic map creating a color spatio-temporal map like the one depicted in Fig.10. This representation provides a quick view of what has happened in the lane of interest during a time interval. Each green line corresponds to a car driving up/down a lane (depending on the view), hence, indicating traffic flow. Red stripes indicate the presence of parked cars.

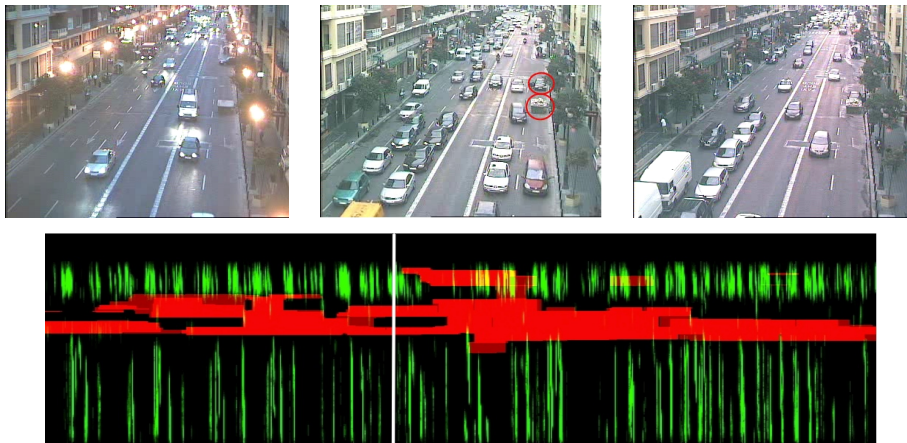


Figure 12. Long Term Parked Container. Top Row: sample camera views of different instants of the presence of the container. Bottom Row: Corresponding one hour map. The white line indicates the time of the top central view, where the container is detected together with a car.

To better understand the relation between maps and camera views, Fig. 11 shows a few

examples. The particular instant at which each image was captured is highlighted with a vertical white line in the corresponding combined spatio-temporal map.

Information such as the number of parked vehicles, how long they are parked, how parked vehicles influenced the traffic flow, etc. can be obtained from these maps.

Just to illustrate how our system is able to continuously detect the presence of objects that remain on the road for long periods of time, the example of Fig.12 shows a garbage container left on the street for about two hours at sunrise. In the map, the presence of the container can be seen as a red stripe that extends horizontally across the total width of the map.

## V. SPATIO-TEMPORAL MAPS ANALYSIS

The original objective of this research was to decide when a lane was blocked by parked vehicles. However, the information provided by spatio-temporal maps is much richer and allows additional information about the following to be obtained:

- **Parking times:** Measuring the horizontal dimension of red regions in the map gives the parking duration of a parked vehicle. These measures can be statistically modeled with mean and variance as a function of the hour of the day and the day of the week.
- **Frequent parking locations:** Vertical positions of red regions in the map are related to vertical positions on the original image.
- **Typical conflictive hours** that may require more active police surveillance to prevent illegal parking.

In any case, obtaining this sort of extra information is out of the scope of this paper. In this section, we will focus only on how to determine the state of the lane.

In order to determine the state of the lane and its availability for traffic planning, a simple four state classifier has been implemented. It uses the following two features:

$$r_{vol}(t) = \int_0^L \int_{t-\Delta t}^t \hat{S}_{map}(t, y) dt dy \quad (7)$$

$$g_{vol}(t) = \int_0^L \int_{t-\Delta t}^t M_{map}(t, y) dt dy \quad (8)$$

These features represent the red (static) and green (dynamic) volume of the combined spatio-temporal map in a certain interval  $\Delta t$  of analysis, respectively. In the equations,  $L$  stands for the total length of the lane.

The features  $r_{vol}(t)$  and  $g_{vol}(t)$  define four different lane states as follows:

- No parked cars and no traffic. This situation normally happens at night (*A* region in Fig.13).
- Parked cars and no traffic on the lane. This is what happens if a double-parked vehicle blocks the lane, thus reducing the total capacity of the street (*B* region in Fig.13).
- Parked cars and traffic present. In this case, a parked car is blocking the lane but traffic still flows before or after it. This situation is more frequent in avenues when the traffic density of the avenue is relatively high and corresponds to the *C* region in Fig.13.
- No parked vehicles and traffic flowing. The lane is contributing to the total traffic capacity of the street (*D* region in Fig.13).

Examples of each of the four states can be seen in Fig.11. A simple classifier that thresholds  $r_{vol}(t)$  and  $g_{vol}(t)$  independently allows us to efficiently obtain the different lane states over time.

## VI. DATA SETS

Two data sets have been used for testing. One is an internal data set, and the other one is a publicly available data set.

### A. Valencia data set

This data set was provided by the traffic authorities of the city of Valencia (Spain) in the context of a national research project [27]. The video data comes from 10 different traffic cameras. For each camera, one week continuous sequences were recorded at 25 images per second, with CIF resolution. The total volume of this data set is about 1.5TB of compressed video (11 MB/min.). A sample of these videos can be downloaded from [28]. The videos show:

- Changes in illumination: sequences were obtained at all times of the day.
- Changes in weather conditions: sunny, cloudy and rainy.
- Presence of fast illumination changes on windy days with sun and clouds.
- Large variability in traffic density, from nights to rush hours in some very busy streets.
- *Real* stopped cars.
- Some very long-term stopped cars (more than one hour).

Unfortunately for legal reasons, this data set cannot be publicly released.

## B. I-LIDS data set

The i-LIDS library has been sponsored by the UK Government to evaluate the performance of vision-based equipment. The main purpose is to evaluate the performance, under realistic conditions, of equipment for possible adoption by the UK authorities. I-LIDS proposes several application scenarios, among which it includes detection of parked vehicles.

This scenario contains three different stages (see Fig.14). The i-LIDS data is organized in four subsets:

- The **sample set** contains 4 sequences, named as *easy*, *medium*, *hard*, and *night*, which can be freely downloaded from internet [6]. These sequences are very short and each one contains just one parking event. The sample set was proposed as a challenge in the AVSS 2007 Conference and has been widely used by other authors addressing this problem.
- The **training set** is a large collection of short duration videos from the 3 stages. Unlike the sample set, this data set must be purchased, at the cost of the media, and is delivered on a 500GB hard disk. Sequences have a resolution of  $720 \times 576$  and 25 frames/s. They have been compressed using MJPEG at a relatively high bit-rate. This does not mean that the quality of the images is good, since in many cases, especially at night, the images contain high noise levels. Different illumination and weather conditions can be found. They are intended to develop algorithms and parameter tuning for each stage.
- The **test set** is meant to self-evaluate performance. It is a collection of longer videos that may contain many parked vehicles or, in some cases, none. The idea is to evaluate performance with this data set with the parameters obtained using the training data set. This data set must also be purchased and is again delivered on a 500GB hard disk. Ground truth is available for Training and Testing data sets. The testing data set contains a total of 216 events in about 22 hours of video (including the 3 stages). The parking times range from 63 seconds to about 29 minutes; in total, only 14 vehicles stop for more than 10 minutes, and the median stopping time is about 90 seconds. In general, these stopping times are much shorter than those in the Valencia data set.
- The **evaluation set** is *not* public and is retained by the UK Home Office. The idea is that periodically the UK authorities launch a call for evaluation. Manufacturers or research groups that have achieved good enough performance on the test set can submit a machine

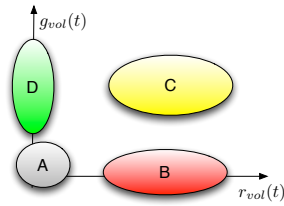


Figure 13. Lane state regions in the  $r_{vol}(t)$ - $g_{vol}(t)$  space.



Figure 14. I-LIDS stages for Parked Vehicle Detection.

to be evaluated by the Home Office. The machines that pass the Home Office's tests are included in a list of candidate suppliers.

In the case of the Parked Vehicle scenario, an event means that a new vehicle has been stopped for more than 60 seconds in a non-parking area. Events must be reported within the next 10 seconds. In other words, an event must be triggered between the seconds 60 and 70 after the vehicle stops. No event shall be generated for parkings shorter than 60 seconds. Also, no event must be signaled when the vehicle leaves. However, an additional event must be generated if a second vehicle arrives and stops for more than 60 seconds while the first one remains parked.

### C. Data Set Comparison

In both data sets, Valencia and i-LIDS, a number of *distractions* occur that make the problem more difficult. Examples of such distractions are the presence of other vehicles and pedestrians moving near the stopped car, people that get in and out the vehicles, vehicles that temporarily occlude totally or partially the stopped vehicle, short duration stops, camera vibration, etc. Distractions act as *background noise* that contributes to causing false alarms or missed detections.

In both scenarios, distractions can be considered *real* since they reflect what normally may happen in scenarios of this kind. However, in the Valencia data set, parking events are *real* in the sense that they correspond to what really happened in some streets during a certain time period. On the other hand, i-LIDS events are simulated, at least most of them. Also, the duration of many parking events are artificially made to last slightly more than one minute.

## VII. PERFORMANCE METRICS FOR PARKED VEHICLE DETECTION

In order to evaluate the performance of any system, it is essential to define the metrics to be used. These metrics are different depending on the final application. The following metrics can be considered for the problem of Parked Car Detection:

### A. Hit-Miss metric

This metric measures the ability of the system to detect individual vehicles that park. Standard precision and recall measurements can be used to measure performance:

$$P = \frac{a}{a + b} \quad (9)$$

$$R = \frac{a}{a + c} \quad (10)$$

where

- $a$ : number of vehicles that park that are correctly detected (true positives).
- $b$ : number of detections that do not correspond to real parked vehicles (false alarms).
- $c$ : number of vehicles that park and are not detected (missed detections).

### B. Precise Event Detection Metric

This metric measures the accuracy of event detection. If an event occurs at  $\tau_i$  and it is detected at  $t_i$ , then the quadratic error for that detection is defined as:

$$e_i^2 = (\tau_i - t_i)^2 \quad (11)$$

A global value of the accuracy is obtained by averaging the values of  $e_i^2$  for a relatively large number of events:

$$\sigma_t^2 = \frac{1}{N_{events}} \sum_{i=1}^{N_{events}} e_i^2 \quad (12)$$

The down-side of this measure is that it is only available for correctly detected events. Therefore, it should be accompanied with precision and recall measurements for a good performance evaluation.

In the case of the i-LIDS challenge, a good value on this feature has no extra merit (it is equally good to detect an event within 1 second or within 8 seconds).

### *C. Occupation Time Metric*

This metric measures the percentage of time that a lane contains parked vehicles for longer than a minimum time. In this case, the parking of new vehicles while the lane contains previous parked vehicles is not relevant.

Precision (P) and recall (R) measurements can also be used where:

- $a$  is the number of seconds that the lane contained parked vehicles that were correctly detected.
- $b$  is the number of seconds that the system indicated parked vehicle presence that was not true (false alarms).
- $c$  is the number of seconds that the lane contained parked vehicles that were not detected (missed detections).

For the application that motivated this research, this is the most appropriate metric. We are not interested in how many cars are parked but rather if the lane is available. We consider that the precise event detection metric has a more academic or research value, and it could provide a way to compare systems that have similar precision and recall. On the other hand, the Hit-Miss metric is probably the best for surveillance scenarios where parking events are rare.

## VIII. RESULTS

This section presents the results using our algorithm and some comparison with previously published results. Section VIII-A provides quantitative performance results using the occupation time metric on the Valencia data set. A comparison with other previous works using the i-LIDS sample data set is presented in Sect.VIII-B. Finally, the results obtained with our approach using the full i-LIDS test data set are provided in Sect.VIII-C.



### A. Quantitative results on the Valencia data set

The Valencia data set has been evaluated using the occupation time metric. The first day of each sequence has been used for training and tuning parameters and the other 6 days for testing. The parameters learned from the first day are:

- The value of  $d_{th}$  used to classify pixels as static or moving (page 9). A value of  $d_{th} = 25$  has been used for most sequences.
- The size of the spatial filters (section IV-C). These parameters are related to the expected vehicle size and depend on the particular scenario geometry.
- The accumulator of static corners  $AS_d(x, y)$  of the previous day (Eq.3). Notice that  $AS_d(x, y)$  is updated every day.
- Threshold on  $r_{vol}(t)$  to consider that the lane contains a stopped car (Eq. 7).

Only 6 of the 10 available scenarios in the data set contain parked vehicles. The number of parked vehicles depends on the day, the hour, and the scenario. However, a number of about 100 parked vehicles per day can be considered as a typical value (the lanes are blocked about 40% of time). The scenarios without parked vehicles are also useful to evaluate false alarms.

In the Valencia data set, a perfect recall has been achieved:

$$R = 1.0$$

which means that it was possible to always detect that the lane was blocked for more than the minimum required time (2 minutes in this data set). We are aware that this does *not mean* that our system cannot miss parked vehicles; it only means that it did not happen in our data set (which is quite large). Some cases of missed parked vehicles on the i-LIDS dataset are shown in Sect. VIII-C.

On the other hand, false alarms may occur in certain situations. False alarms usually happen when the lane under analysis is half in sun and half in shade. Figure 15 illustrates one of such challenging situations; in the spatio-temporal map, it looks as if there was a parked car. This adverse situation happens only on sunny days (and not in every scenario) and depends on the shape of the shadow (which in turn depends on the shape of the roofs of the buildings). The duration of these false alarms is different for each scenario, but, in general, if they occur, it is less than 30 minutes per day. The impact of these situations on precision varies depending on

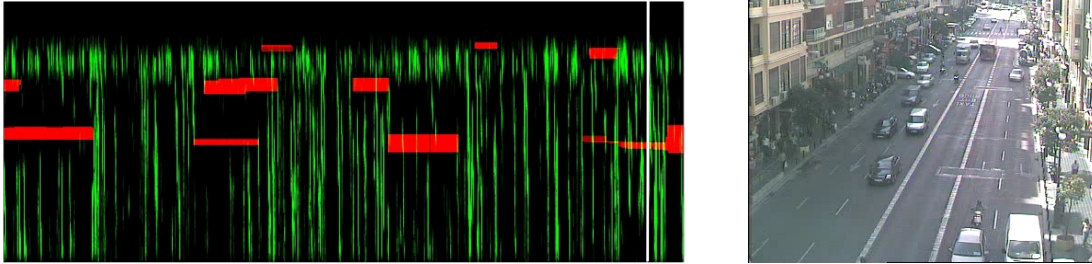


Figure 15. Example of camera view when the lane is half in shade and half in sun. Depending on the shape of the roofs of the buildings static corners can be detected and false alarms can be generated.

whether or not there are parked cars when it happens. On average, a precision value of:

$$P \approx 0.97$$

is obtained for lanes that remain blocked by parked vehicles about 40% of the time. Notice that this is important because, according to Eq.9, precision also depends on the percentage of time that the lane is blocked (a false alarm of just one minute in a lane that is never blocked yields a precision of  $P = 0$ ). Availability can be defined as the percentage of the day when the system is reliable. Excluding the 30 minutes per day where shadows may cause trouble, the proposed system provides an availability of about 98%.

### B. Results on *i-LIDS* sample data set

In order to compare our results with those obtained by other researchers such as [8], [9], [10], [11], [12] and [16], our method has also been tested using the sample *i-LIDS* subset.

Figure 16 shows examples of frames from the four sequences where it is possible to see the cars that have to be detected in each sequence. For the *i-LIDS* challenge, these sequences were precisely annotated with the arrival and departure of each car.

The spatio-temporal maps of the four sequences are shown in Fig.17. The white line indicates the time instant that corresponds to the images of Fig.16. The presence of the vehicle that is parking is clearly visible in the four maps. Green vertical lines correspond to cars driving down the lane under analysis. Notice that the presence of other traffic does not affect the detection of the stopped car. In all cases, stopped cars can be detected regardless of camera vibration, differences in car size, nearby traffic, low illumination, or high noise levels.

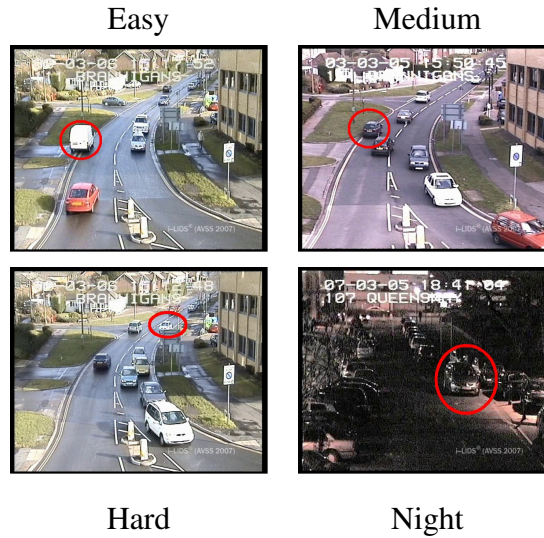


Figure 16. i-LIDS image samples of the easy, medium, hard, and night sequences.

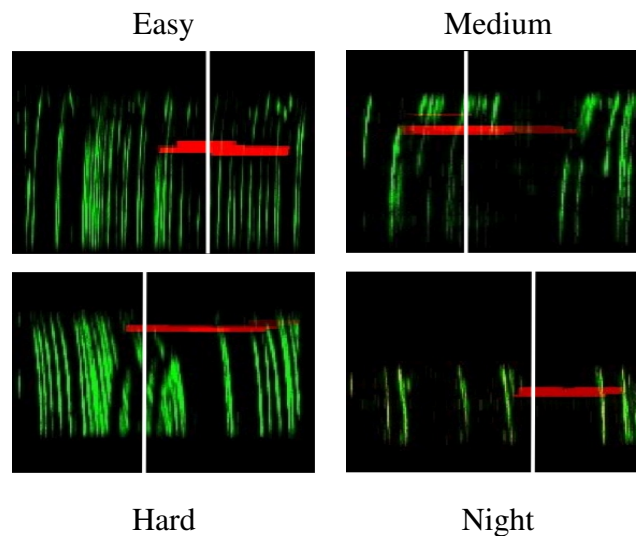


Figure 17. i-LIDS data set spatio-temporal maps. Notice that the vertical position of the red blob corresponds to the vertical coordinate of stopped cars in Fig.16.

Table I shows a summary of the results obtained by other authors with this data set. Notice that all authors use the Precise event detection metric (see Sect.VII-B) with this data set. The results on this data set are all very similar except that there is missing information for some sequences and authors.

In our opinion, this small data set can only provide limited insight about algorithm perfor-

mance. For this reason, larger data sets have been used to evaluate our approach. Also, as stated above, we think that the Hit-Miss or Occupation Time Metrics are more relevant for our application.

Author	Easy	Medium	Hard	Night
Our System	Yes	Yes	Yes	Yes
Bevilacqua and Vaccari [8]	Yes	Yes	Yes	N/A
Lee et al. [13]	Yes	Yes	Yes	N/A
Maddalena and Petrosino [9]	Yes	Yes	Yes	N/A
Boragno et al. [10]	Yes	Yes	Yes	Yes
Guler et al. [11]	Yes	Yes	Yes	Yes
Venetianer et al. [12]	Yes	Yes	Yes	Yes
Porikli [16]	N/A	Yes	N/A	N/A

Table I

COMPARISON OF DETECTION RESULTS IN I-LIDS SAMPLE DATA SET. YES MEANS THAT THE PARKED VEHICLE WAS DETECTED. N/A MEANS THAT THE AUTHORS DO NOT REPORT RESULTS ON THAT SEQUENCE.

### C. Results on the testing i-LIDS data set

First of all, we would like to emphasize that, to our knowledge, there are no previous published results using this public data set. Due to the large size of results using this data set, this section presents just a summary of them. A more detailed description is provided in Appendix A.

Using the training i-LIDS data set, the same parameters as in the Valencia data set were determined (see Sect.VIII-A).

In our implementation, input images were downsized from  $720 \times 576$  to  $240 \times 180$ . This has no impact on the results, but it makes processing much faster.

Figure 18 shows the static corner maps corresponding to the first 60 minutes of the rightmost lane of the stage 1 sequence PVTEA101a (see Fig.14). It is interesting to observe how the temporal filter removes events that are shorter than one minute.

In the i-LIDS sequences, it is quite common for new cars to park next to previously parked vehicles. An event like this is highlighted in Fig.18 with a red circle, and the corresponding images are shown in Fig. 19. Detecting the arrival instant of the second vehicle is really difficult

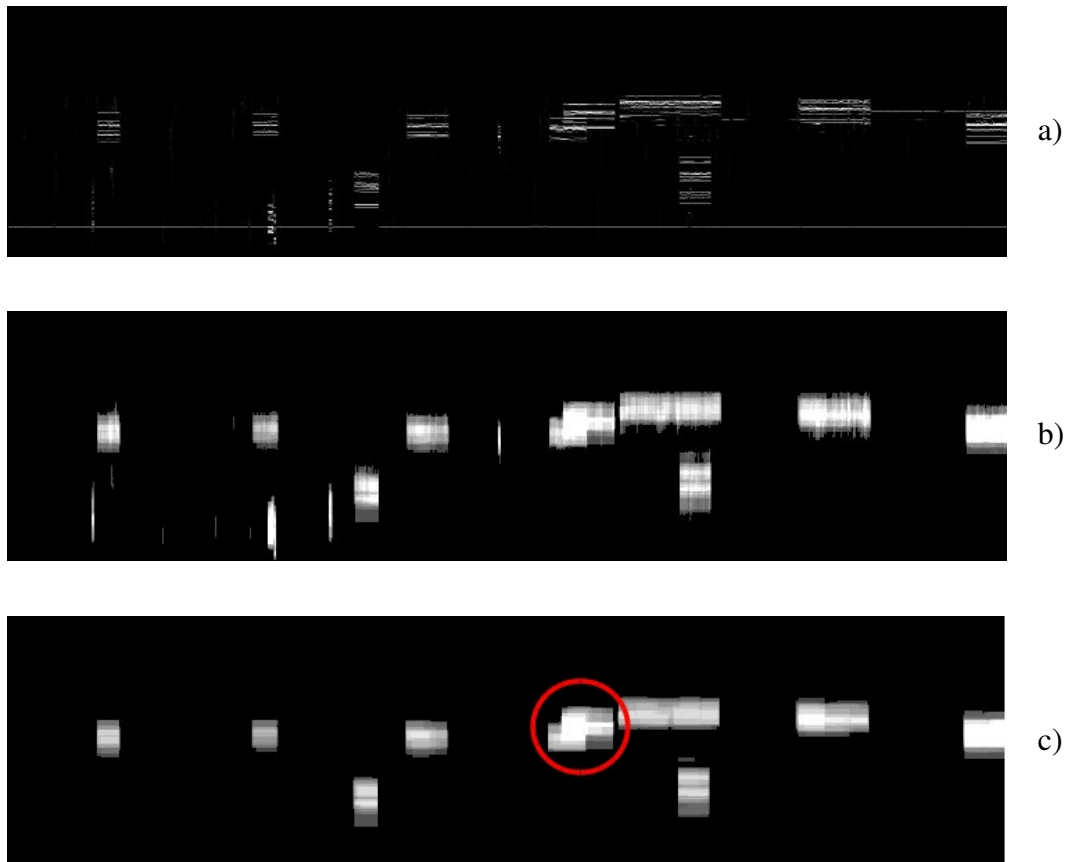


Figure 18. Maps corresponding to the first 60 minutes of the bottom lane of sequence EA101a. a) Raw map of static corners. b) Spatially filtered map. c) Result of temporal filtering of spatially filtered map.

for the algorithm proposed in this paper. However, as can be seen from the maps, the system will continue to detect the presence of the second car until it leaves even though it was occluded for some time by the vehicle that arrived first. Missing the arrival instant of the second car would lead to a missed detection using the i-LIDS hit-miss metric. However, the occupation time metric gets a perfect recall ( $R = 1$ ) and no false alarm ( $P = 1$ ) in this fragment (it was possible to detect when the lane was blocked).

Each stage contains two lanes that were processed independently. Since the i-LIDS test set has a total duration of about 22 hours, it contains a total of  $22 \times 2 = 44$  hours of lane analysis.

False alarms, for the reasons detailed in the appendix, were present for 13 minutes. This gives a probability of false alarm of 0.5% of total time. Total parking time is 10h36'. For several reasons, that are detailed in the appendix, a total of 25 minutes containing parked vehicles are missed. This gives the following figures for the time occupancy metric:



Figure 19. Camera views corresponding to the encircled portion in the map of Fig.18. (a) One vehicle is parked. (b) A second car parks right behind the first one.

$$P = 0.98 \quad R = 0.96$$

The main conclusions from this data set are similar to those obtained with the Valencia data set.

- If vehicles are not occluded and they fall (mostly) in the lane mask, they are always detected.
- All the missed detections occurred with vehicles that were partially occluded by tree branches or in areas that were excluded from the detection mask.
- False alarms occurred in a small portion of time; they were caused by two things:
  - Shadows in the middle of the lanes; however, keep in mind that not all shadows produce false alarms (it depends on the shape and contrast of the shadows).
  - Very slow moving traffic queues. In this case, the system was not able to cope with the fact that the stopped vehicle was not always the same (the queue advanced very slowly). See Fig.22a.

## IX. CONCLUSIONS

In this paper, an approach for dealing with the problem of parked vehicle detection has been presented. It is based on the analysis of spatio-temporal maps of static image corners. Our solution to the problem does not require estimating background or tracking any object.

Using a 2 Ghz Core 2 Duo Processor, our implementation can process more than 150 CIF images per second. This means that several video streams can be analyzed in real time in

the same machine. The task that requires the most computational power is the Harris corner detection (located at the pixel-level processing layer). Corner classification (static/dynamic), map formation, and analysis represent a negligible computational load.

The proposed approach has been thoroughly evaluated on two huge data sets, one private and one public. As far as we know, it is the first time that a system for detecting parked vehicles has been evaluated in real conditions using databases of this size. With this extensive evaluation, we found that the proposed approach is robust to:

- Illumination changes, fast and slow.
- Presence of moving lights due to traffic.
- Thermal noise in night conditions.
- Presence of pedestrians around stopped cars.
- Passengers getting in and out of the vehicles.
- Hazard warning lights flashing.
- Occlusions by cars passing near the stopped vehicle.
- Different illumination conditions: night, day, dawn, dusk,...
- Different weather conditions: cloudy, sunny, rainy. In the case of rain, the system has proved to be resilient to specular reflections on the wet pavement.
- Automatic day/night camera mode switching.
- Camera Automatic Gain Control.
- Camera shakes due to wind.

Robustness against these problems is achieved by the statistical nature of the algorithm, which is based on the fact that a car generates a large number of static corners around the position and time where it is located. For this reason, failing to detect static corners for a short time (occlusion) or misclassifying some corners as dynamic instead of static (noise, camera vibration, etc.) during one frame have no impact in the detection capability if, *on average*, a large number of static corners are around the same spatial location during a short time interval (5 seconds). The spatial and temporal average filters also contribute to increasing the robustness of the system. The presented system has the very practical additional advantage that it does not require any supervision at startup.



## REFERENCES

- [1] European Commission, “Managing urban traffic congestion: Complete edition - isbn 9282101282,” *SourceOECD Urban, Rural and Regional Development*, vol. 2007, no. 8, pp. i–294, May 2007.
- [2] “Autoscope,” <http://http://autoscope.com>.
- [3] “Traffic control,” <http://www.trafficcontrol.es>.
- [4] “Citilog,” <http://www.citilog.com>.
- [5] N. Taylor, “The contrans dynamic traffic assignment model,” *Network and Spatial Economics*, , no. 3, pp. 297–322, 2003.
- [6] “ilids dataset for avss 2007,” <ftp://motinas.elec.qmul.ac.uk/pub/iLids/>.
- [7] Alper Yilmaz, Omar Javed, and Mubarak Shah, “Object tracking: A survey,” *ACM Computing. Surveys*, vol. 38, no. 4, december 2006.
- [8] A. Bevilacqua and S. Vaccari, “Real time detection of stopped vehicles in traffic scenes,” *Advanced Video and Signal Based Surveillance, IEEE Conference on*, vol. 0, pp. 266–270, 2007.
- [9] L. Maddalena and A. Petrosino, “Self organizing and fuzzy modelling for parked vehicles detection,” in *ACIVS09*, 2009, pp. 422–433.
- [10] S. Boragno, B. Boghossian, J. Black, D. Makris, and S. Velastin, “A dsp-based system for the detection of vehicles parked in prohibited areas,” in *AVSS '07: Proceedings of the 2007 IEEE Conference on Advanced Video and Signal Based Surveillance*, Washington, DC, USA, 2007, pp. 260–265, IEEE Computer Society.
- [11] S. Guler, J.A. Silverstein, and I.H. Pushee, “Stationary objects in multiple object tracking,” in *Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on*, Sept. 2007, pp. 248–253.
- [12] P. L. Venetianer, Z. Zhang, W. Yin, and A. J. Lipton, “Stationary target detection using the objectvideo surveillance system,” *Advanced Video and Signal Based Surveillance, IEEE Conference on*, vol. 0, pp. 242–247, 2007.
- [13] J.T. Lee, M.S. Ryoo, M. Riley, and J.K. Aggarwal, “Real-time illegal parking detection in outdoor environments using 1-d transformation,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 19, no. 7, pp. 1014–1024, July 2009.
- [14] “Intuvision,” <http://www.intuvisiontech.com>.
- [15] “Object video,” [www.objectvideo.com](http://www.objectvideo.com).
- [16] F. Porikli, “Detection of temporarily static regions by processing video at different frame rates,” *Advanced Video and Signal Based Surveillance, IEEE Conference on*, vol. 0, pp. 236–241, 2007.
- [17] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, Hawaii, December 2001.
- [18] Ming-Hsuan Yang, D.J. Kriegman, and N. Ahuja, “Detecting faces in images: a survey,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 1, pp. 34–58, Jan 2002.
- [19] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Computer Vision and Pattern Recognition*, 2005, pp. 886–893.
- [20] C. P. Papageorgiou and T. Poggio, “A trainable object detection system: Car detection in static images,” Tech. Rep. AI-Memo-1673, CBCL-180, MIT,, Oct 1999.
- [21] N.K. Kanhere and S.T. Birchfield, “Real-time incremental segmentation and tracking of vehicles at low camera angles using stable features,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 9, no. 1, pp. 148 –160, mar. 2008.
- [22] C. Harris and M.J. Stephens, “A combined corner and edge detector,” in *Alvey Vision Conference*. University of Manchester, september 1988.

- [23] C. Tomasi and T. Kanade, "Detection and tracking of point features," Tech. Rep., Carnegie Mellon University, April 1991.
- [24] "Opencv library," <http://sourceforge.net/projects/opencv/>.
- [25] A. Albiol, I. Mora, and V. Naranjo, "Real-time high density people counter using morphological tools," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 2, no. 4, pp. 204–217, 2001.
- [26] Jean Serra, *Image Analysis and Mathematical Morphology Vol. I*, Ac. Press, 1982.
- [27] "Marta (mobility & automotion through advanced transport networks)," <http://www.cenitmarta.org>.
- [28] "Sample videos," [http://www.gpiv.upv.es/traffic\\_videos/](http://www.gpiv.upv.es/traffic_videos/).

#### AUTHOR BIOGRAPHIES

**Antonio Albiol** received the M.S. degree in Telecommunication Engineering from Universidad Politécnică of Madrid in 1987. He got his PhD degree also in Telecommunication Engineering in 1993 from Universidad Politécnică de Valencia. Currently he is a professor teaching Digital Signal Processing. His research interest include automatic analysis of video, specially for surveillance applications.

**Laura Sanchis** received the M.S. degree in Telecommunication Engineering from Universidad Politécnică of Valencia (UPV) in 2009. She is currently working at the Image and Video Processing Group (GPIV) of the Institute of Telecommunications and Multimedia Applications (I-TEAM) of UPV. Her main research activity has been focused on computer vision applications such as face classification, video surveillance, people counting and automatic video-based traffic analysis.

**Alberto Albiol** graduated in Telecommunications Engineering at the Universidad Politecnica de Valencia (UPV) in 1995. From 1999 until 2003 he stayed several times in Purdue University, IN (USA) pursuing his PhD on multimodal video indexing, which finally obtained from UPV in 2004. He started his teaching activities in 1995 in the University of Zaragoza and currently, he is Associate Professor of Signal and Image Processing at UPV. His research interests include video analysis, image face recognition, pattern recognition and computer vision applications. He is author of more than 26 refereed publications in journals and in international conferences, and actively participates on the review process of many related journal and conference proceedings.

**Jose M. Mossi** was born in Valencia, Spain. He received the Ingeniero de Telecomunicacion degree in 1991 from the Universidad Politecnica de Madrid and the PhD in 1998 from the Universidad Politecnica de Valencia(UPV). He joined the Departamento de Comunicaciones at the UPV in 1999 as a Lecturer and participated in several national and european research projects.

His current interests include image processing, coding, intelligent traffic vision applications and digital television.

#### APPENDIX

This appendix provides detailed results using the testing i-LIDS data set. Sample images for the cases where the algorithm has failed are also presented. The data set has been evaluated using the occupation time metric. False alarm time is used as a measure of performance for sequences that do not contain parked vehicles.

## STAGE 1 RESULTS

Seq. name	P	R	Remarks
PVTEA101a	1.0	1.0	Rainy day. Cars park with windscreen wipers on and are still detected.
PVTEA101b	1.0	1.0	Night sequence. Some cars drive with lights off. Hazard warning lights cause some erroneous classification of corners; however, the parked vehicle is correctly detected.
PVTEA102a	0.85	1.0	Sunny and windy day. Fast illumination changes. Some tree shadows cause false alarms for about 5 minutes (out of 52 minutes) in one of the two lanes . No missed detection.
PVTEA102b	1.0	1.0	Rainy and windy. Camera vibration. Specular reflections on wet pavement at some moments. Relatively fast background changes.
PVTEA103a	1.0	1.0	Daylight. No rain, no shadows.
PVTEA103b	1.0	1.0	
PVTEA104a	1.0	1.0	Night sequence. Poor street illumination. Some cars park with the lights off and are still detected.
PVTEA104b	1.0	1.0	
PVTEN102d	1.0	1.0	Daylight. Cloudy.

Table II

STAGE 1. DETAILED RESULTS FOR SEQUENCES CONTAINING EVENTS.

Seq. name	Duration	Min. FA	Remarks
PVTEN101a	20 min.	0	Sunny day. Shadows on the lanes did not generate false alarms.
PVTEN101b	25 min.	0	Sun creates reflections on wet ground.
PVTEN102a	19 min.	0	Daylight. Variable weather. Shadows from trees create some static corners but not enough to cause a detection
PVTEN102c	20 min.	0	
PVTEN102b	10 min.	0	Daylight. Raining.
PVTEN102f	20 min.	0	
PVTEN102e	30 min.	0	Daylight. Cloudy.
PVTEN103a	25 min.	0	Night. Good street illumination.
PVTEN103b	20 min.	0	
PVTEN104a	20 min.	0	

Table III

STAGE 1. DETAILED RESULTS FOR SEQUENCES THAT DO NOT CONTAIN EVENTS.

## STAGE 2 RESULTS



Figure 20. a) Missed detection in sequence PVTEA201c. b) Missed detection in sequence PVTEA202b .



Figure 21. Sequence of images showing a missed detection in sequence PVTEN201b. The occlusion by the SUV car and the tree leaves make this sequence extremely difficult.

## STAGE 3 RESULTS



Figure 22. a) Queue that caused a false alarm in PVTEA301c. b) Van that was not detected in sequence PVTEA301c because it was mostly outside of the lane mask. c) Vehicle not detected in sequence PVTEN301c because the overlaid text area was excluded from the detection mask.



Seq. name	P	R	Remarks
PVTEA201a	1.0	1.0	Daylight. No rain, no shadows.
PVTEA201b	1.0	1.0	
PVTEA201d	1.0	1.0	
PVTEA202a	1.0	1.0	
PVTEA201c	1.0	0.83	Night sequence. Poor street illumination. Some cars park with the lights off and are still detected. Two cars that park with lights off in badly illuminated zones are missed. See Fig.20a.
PVTEA202b	1.0	0.82	Rain and wind. Camera vibrations. Specular reflections on wet pavement at some moments. Relatively fast background changes. A tree is occluding the upper left part of the lane. Two cars partially occluded by the tree are not detected. See Fig.20b.
PVTEN202a	1.0	0.86	Night sequence. Poor street illumination. Some cars park with the lights off. One car is missed because it parks with the lights off in a specially badly illuminated zone. See Fig.20a.
PVTEN201a	1.0	0.5	
PVTEN201b	-	0.0	Daylight. Cloudy. This sequence contains only one parked vehicle, and it is missed due to occlusions (see Fig.21). This sequence illustrates the difficulty of this data set very well. Precision can not be computed in this case since nothing is detected in this sequence.
PVTEN201c	1.0	1.0	Daylight. A tree occludes the upper part of the lane. Our lane mask excludes this area. Any car parking in that zone would have been missed (see Fig.20b). Comparing Figures 20b and 14 we can see that the tree leaves occlude a large

Seq. name	P	R	Remarks
PVTEA301a	1.0	1.0	Daylight. Cloudy.
PVTEA301b	1.0	1.0	
PVTEA301e	1.0	1.0	
PVTEA301c	0.82	0.88	Daylight. In this sequence a false alarm was generated for 2 minutes because of a queue (Fig.22a). A van that is located outside of the lane mask was not detected (Fig.22b).
PVTEA301d	1.0	1.0	Night sequence with good illumination. Parked vehicles with warning lights on are correctly detected.
PVTEN301b	1.0	1.0	Daylight. Variable weather with moments of sunshine that generate shadows on the lanes. No false alarm produced.
PVTEN301c	N/A	0.0	Daylight. Rain. The sequence contains two cars that park at the top of the image. None of them is detected because the lane mask does not contain that region (the detection mask is indicated by i-LIDS for each stage and does not include the location where these two cars stop). The region where these two cars park corresponds to overlaid text. Fig.22c shows the car that is missed. Precision is not computed since nothing is detected in that sequence.

Table V

STAGE 3. DETAILED RESULTS FOR SEQUENCES CONTAINING EVENTS.

Seq. name	Duration	Min. FA	Remarks
PVTEN301a	30 min.	2	False alarm of 2 minutes duration caused by queue similar to that in Fig.22a.
PVTEN301d	20min.	2	Daylight. False alarm of 2 minutes duration caused by queue similar to that in Fig.22a.
PVTEN301e	20min.	0	Daylight. Cloudy. Camera vibrations
PVTEN302a	30min.	0	Night sequence with good street illumination
PVTEN302b	30min.	0	Twilight. Wet pavement with specular reflections from vehicles headlights.
PVTEN303c	30min.	0	Night sequence. Raining. Specular reflections of street lights on wet pavement. Camera vibration. Water drops on the camera lens.

Table VI

STAGE 3. DETAILED RESULTS FOR SEQUENCES THAT DO NOT CONTAIN EVENTS.