Multimed Tools Appl (2016) 75:17711–17743
DOI 10.1007/s11042-016-3773-8

# QuickSpot: a video analytics solution for on-street vacant parking spot detection

**Elena Màrmol[1] · Xavier Sevillano[1]**

Received: 15 January 2016 / Revised: 31 May 2016 / Accepted: 7 July 2016 /
Published online: 25 July 2016


**Abstract** Vehicles searching for a vacant parking spot on the street can amount to as much as 40 % of the traffic in certain city areas, thus largely affecting mobility in urban environments. For this reason, it would be desirable to create integrated smart traffic management systems capable of providing real-time information to drivers about the location of available vacant parking spots. A scalable solution would consist in exploiting the existing and widely-deployed video surveillance camera networks, which requires the development of computer vision algorithms for detecting vacant parking spots. Following this idea, this work introduces QuickSpot, a car-driven video analytics solution for on-street vacant parking spot detection designed as a motion detection, object tracking and visual recognition pipeline. One of the main features of QuickSpot is its simplified setup, as it can be trained on external databases to learn the appearances of the objects it is capable of recognizing (pedestrians and vehicles). To test its performance under different daytime lighting conditions, we have recorded, edited, annotated and made available to the research community the QuickSpotDB video database for the vacant parking spot detection problem. In the conducted experiments, we have evaluated the trade-off between the accuracy and the computational complexity of QuickSpot with an eye to its practical applicability. The results show that QuickSpot detects parking spot status with an average accuracy close to 99 % at a 1-second rate regardless of the illumination conditions, outperforming in an indirect comparison the other car-driven approaches reported in the literature.

**Keywords** Smart parking · Computer vision · Vacant parking spot detection

✉ Xavier Sevillano
xavis@salleurl.edu

Elena Màrmol
emarmol@salleurl.edu

[1] GTM - Grup de Recerca en Tecnologies Mèdia, La Salle - Universitat Ramon Llull, Quatre Camins 30, 08022 Barcelona, Spain

## 1 Introduction

In response to society's concerns to make cities more sustainable, accessible, livable and greener, researchers are exploring new approaches to exploit the synergies between societal challenges and Information and Communication Technologies (ICTs). As a result, innovative solutions for day-to-day problems are constantly being proposed in an attempt to transform urban nuclei into Smart Cities.

In this context, one of the key challenges to push forward urban environments towards Smart Cities is transportation. Indeed, city inhabitants use public and/or private means of transportation to perform their daily activities, investing personal, material and economic resources in the process. Provided that transportation has a direct impact on the quality of life of nearly every citizen and on the planning of urban environment development, numerous approaches have been proposed to analyze, mine and exploit transportation related data, with applications ranging from the construction of road traffic models [29] to the inference of buildings use from transportation data analysis [59].

In parallel, many cities have undergone a process of *sensorization*, i.e. the deployment of networks of sensors designed to capture transportation related data with diverse purposes [4]. As a consequence, cities have become a source of huge volumes of multimedia data related to transportation, like for instance, urban traffic data. Examples of this include acoustic sensors to create traffic noise level maps [6, 45], Bluetooth sensors to measure traffic conditions [35], magnetic sensors to detect vacant parking spots [53] or video camera networks to monitor traffic status [48].

This latter modality is one of the most relevant as regards the mining and analysis of transportation data, as the proliferation of CCTV cameras in urban environments makes them a source of valuable information for automatic traffic activity monitoring at large scale. For this reason, video analytics has been applied in this context with a fairly diverse range of purposes, such as smart payment systems based on license plate registration [52], traffic incident classification [57], congestion points information systems [56, 58], or automated road surveillance [47].

This work focuses on video analytics applied to one of the most common daily traffic activities: the search for a vacant parking spot on the street. According to some studies, between 28 % and 41 % of traffic in specific times and urban areas corresponds to drivers looking for parking [46]. This constitutes a relevant factor affecting urban mobility, besides a non-negligible source of atmospheric and acoustic pollution, as well as a source of stress and a waste of time for the driver. Therefore, the development of systems capable of providing real-time information about the availability of on-street parking spots at city scale would constitute a step forward towards the implementation of efficient urban mobility policies, and certainly, to promote Smart Transportation.

Most existing approaches to on-street vacant parking spot detection entail the deployment of application-focused sensors, such as pneumatic road tubes, and magnetic or infrared sensors [23]. However, the cost of deploying and maintaining this type of sensor networks constrains its practical applicability, restricting its implementation to geographically limited areas within the city (e.g. San Francisco in the USA, and Santander in Spain are examples of cities with sensorized neighbourhoods). Moreover, the deployment of such sensor networks breaks with the idea of Smart Cities, which envisages the reuse of already existing cities' capabilities to tackle new challenges, reducing the investment in self-centered solutions whilst improving the cities' services.

In fact, we believe that a strategy to reduce the car-loitering in the search of parking space would require the creation of integrated smart traffic management systems capable of providing real-time information to the drivers about the location of available vacant parking spots by combining the data gathered by small-scale sensor networks and widespread CCTV camera networks. By doing so, reusable, sustainable and scalable solutions could be developed.

As a step forward towards this aim, this paper introduces QuickSpot, an on-street parking spot detection system based on video analytics. The proposed system uses a single stationary camera to monitor the status of the parking spots within its field of view, being also capable of detecting, tracking and recognizing different types of objects (e.g. vehicles and pedestrians) while being robust to natural changes in illumination.

In order to assess the reliability of our proposal, the QuickSpot system performance has been evaluated through a set of experiments thoroughly planned to replicate a real-world scenario. In this sense, due to the scarcity of publicly available video material for the vacant parking spot detection application, we generated the QuickSpotDB video database after recording, pruning, post-processing and labeling more than five hours of video of a parking lot. Moreover, the recording was carefully prepared to collect external factors affecting the scene characteristics, so as to test QuickSpot's ability to operate under different daytime illumination conditions.

Five are the main contributions of this work. First, the design of QuickSpot has taken into account several aspects related to its practical deployment. To this end, we have simplified the setup of QuickSpot in a new location by training the system with images from an external database, which reduces its time-to-operation to a large extent. Second, we have also analyzed the trade-off between the robustness and accuracy of the system and its computational complexity, finding the configuration that allows real time execution with the highest parking spot detection accuracy. Third, we have implemented a foreground mask post-processing step that allows QuickSpot perform reliably under varying illumination conditions. Fourth, we have designed a scheme based on temporal hysteresis and visual recognition that ensures the reliability and robustness of parking spot status decision. And finally, we describe and make available to the research community the QuickSpotDB database, a completely annotated brand new video database for vacant parking spot detection that presents strong daytime illumination variations.[1]

The remainder of the paper is organized as follows. Section 2 reviews recent work on video based vacant parking spot detection. The architecture of the proposed system is presented and described in Section 3. Next, Section 4 describes the video data recorded and employed for evaluating the QuickSpot system. Section 5 presents the experiments conducted to test the system under different experimental conditions, and the conclusions of this work are discussed in Section 6.

## 2 Related work

The use of image and video analysis for detecting vacant parking spots has attracted considerable interest from the research community during the last years. The interested reader will find a review on smart car park systems in [23].

The work by Huang and Wang proposes a general categorization of the existing vacant parking spot detection methods into two categories: car-driven and space-driven [18]. The former are based on car detection algorithms, and the detection of vacant parking spots

[1] The QuickSpotDB video database is available upon request to the corresponding author.

works by estimating the distances between detected cars. However, these methods present a major challenge, perspective distortion, which affects the cars detection performance and the scene knowledge estimation. In contrast, space-driven methods focus on detecting the available parking spaces in a scene, either comparing the surveillance videos with a empty background model, detecting the lines, or assuming homogeneous appearance in regulated parking lots.

Following this categorization, this section presents an up-to-date review of the state of the art on this topic.

### 2.1 Space-driven approaches

Most approaches found in the literature about vacant parking spot detection follow a space-driven approach. For instance, one of the earliest works on this topic detected the occupancy of parking spots by comparing each parking spot with a reference image of the empty parking spot, using vehicle to parking spot pixel area ratio to determine whether a car is parked or not [15]. In [31], the presence of parked vehicles was detected by performing quad-tree segmentation of parking spots, using the number of blocks resulting from the segmentation as an indicator of the presence of a car.

Later, Wu et al. addressed the problem by designing multistage approaches. In [54, 55], the authors proposed a 4-step approach that included (i) parking region detection, (ii) feature extraction based on Gaussian ground color model, (iii) multi-class SVM training and (iv) Markov Random Field conflict classification correction system between contiguous parking spaces. Another example of this type of approaches was presented by True [49], which relied on human-labeled parking space region extraction, followed by color histogram classification and vehicle feature detection consisting on the detection of interest points using Harris corner detection.

In [30], the authors developed a color-based approach to detect vacant parking spots. They generated an adaptive parking lot background model, obtaining the color of each individual parking spot from statistical analysis of video sequences of the parking lot under surveillance. Finally, luminance analysis was applied to detect and remove shadows.

Sastre et al. presented a methodology for computing 2D homographies to compute projective transformations and tested it to compound a pseudo-top-view images of a parking site [43]. They completed their design by extracting Gabor features from each parking spot and trained a support vector machine classifier to determine the occupancy of parking spots.

The work by Bong et al. presents a bi-stream vacant parking spot detection approach, in which one stream analyzes the number of pixels below a certain intensity threshold to determine the presence of a car in the parking spot, and the other stream eliminates false detections caused by shadows of vehicles using median filters and edge detection [7].

Other authors have addressed the problem of occlusions while evaluating the vacancy of parking regions [12]. In that work, the proposed approach was based on extracting regions of interest in a 3D model of the parking lot. A similar approach was presented in [11], in which vacant parking detection was based on building 3D volume models of parking spaces. By means of a vehicle detector, the authors could determine the occupancy of the parking lot based on a vehicle detector and the inferred volume of each spot.

In [9], a multi-camera outdoors vacant parking spot detection method was presented, based on modeling the color changes of the parking ground to determine the vacancy of a parking region. Moreover, authors dealt with perspective effects proposing two geometrical models to represent a parking region/space (ellipses and grids).

In [34], the authors proposed a method based on the region covariance matrices assuming prior knowledge indicating where the parking spaces/regions are. The distance metric between an occupied space and an empty space was then compared and threshold returning a binary decision. Authors presented a region of covariance approach for different lighting conditions, e.g. day and night.

Generally, the proposed approaches analysed parking lots either indoors or with certain space limitations. In contrast, in [21, 22] the authors proposed a method later applied in Tokyo underground parking lot in early October 2009. Their method consisted on the use of a FCM classifier based on a semi-hard clustering and a hyperparameter tuning by particle swarm optimization.

The use of visual texture descriptors for determining the status of parking spots was explored in [2]. The authors made a critical comparison of the performance of Local Binary Patterns and Local Phase Quantization texture descriptors on a collection of over 100k images parking spots, using Support Vector Machines for parking spot status classification.

In another recent work [32], Liu et al. implemented a low complexity parking spot status detection system based on the analysis of every parking spot in terms of edge and closed contour density, and foreground/background pixel ratio to determine the presence of a car.

The work by Tschentscher et al. proposes an evaluation of a pipeline composed of image feature extraction (such as color histograms, gradient histograms or Haar features), classifier systems (k-nearest neighbour, linear discriminant analysis and support vector machines) and exponential smoothing for temporal filtering to detect vacant parking spots [50].

In [13], Fabian proposes using a priori available information about the parking lot geometry and the general shape of common cars to obtain a reliable status of a parking space. Using a probabilistic car model and a physically based feature extraction using computational fluid dynamics, the author avoids the training phase as much as possible to reduce the time required to bring the system into a fully operational state.

Pursuing the same goal, Sevillano et al. [44] designed a training approach based on external vehicle image databases which reduced time-to-operation of vacant parking spot detection systems. In that work, moreover, the authors present a critical comparison between local and global visual features (either alone or fused at feature level) and different classifier systems applied to the task.

In [24], the authors proposed a vacant parking spot detection system based on training two-layered feed-forward neural networks with low level visual features extracted from parking spots (e.g. edge or color related features). To test the proposed system, a 24-hour recording of an outdoors parking lot was made. To address changes in illumination, the authors trained one network for daytime and another one for nighttime. In a subsequent paper, the same authors presented a deeper statistical analysis of the obtained vacant parking spot detection results [25].

Recently, in [38], the authors derived two features based on the dispersion of the histogram of the image subregions corresponding to each parking spot: average local entropy and the standard deviation of the average entropies. With those features, a SVM classifier was trained to determine the status of each spot.

### 2.2 Car-driven approaches

Among the car-driven approaches to vacant parking spot detection, the work by Choeychuen [10] proposed a background subtraction algorithm with adaptive background model to detect the existing cars. Once the cars were detected, a feature extraction module computed the masked-area and edge orientation histogram density (EOH), followed by a feature combination module. The approach ended with a decision-making module which thresholded the multi-feature pattern.

The work by Masmoudi et al. [36] presented an approach to deal with occlusions between neighboring parking spots. To that end, the authors proposed a surface based model for parking spot model extraction, performing vehicle tracking to detect vehicles entering or leaving parking spots. In a very recent work, the same authors made a step towards the real implementation of a video-based vacant parking spot detection system, proposing the architecture of a multi agent parking lots management system based on computer vision to detect and localise vacant parking spots at a city level and to provide drivers with relevant information in real time, as well as to detect anomalies in the behaviour of drivers through the parking site [37].

On the other hand, Wang et al. proposed a three-stage approach to video-based vacant parking spot detection [51]. In the first stage, foreground and background models were generated for determining parking spot status. Next, the authors computed the difference between successive video frames to determine whether a car is finally parked on a given spot. And finally, the final decision about parking spot status was taken by applying an adaptive decision threshold.

Another recent contribution is the work by Gálvez del Postigo et al. [16]. In their work, the authors estimate the dimensions of non-delimited free parking areas through a temporal analysis of the video frames to detect the occupancy variation of the parking areas, combining background subtraction and the creation of a transience map to detect the parking and leaving of vehicles.

An interesting car-based work related to parking lot surveillance is the one by Ng and Chua [39], which focuses on activity recognition in parking lots rather than in occupancy detection. Using Gaussian mixture models and object tracking, the authors extract spatio-temporal and contextual information of motion trajectories to classify events into a set of six predefined activities (e.g. a vehicle passes by the monitored parking area, a vehicle enters the monitored area and parks at an empty parking space, etc.), thus moving towards behavior analysis in this context.

### 2.3 Parking lot-driven approaches

In addition to the classic car-driven or space-driven classification of vacant parking spot detection techniques, Huang et al. propose an approach that differs with the previous ones in that it is parking lot-driven [19]. This means that the whole monitored parking site is modeled as a structure composed of multiple surfaces, which are analyzed within a Bayesian hierarchical framework following a bottom-up approach divided into three layers: (i) image observation, (ii) image content labeling and (iii) 3D scene modeling. In successive works, the authors propose different approaches to deal with several visual challenges such as luminance variations, shadow effect, perspective distortions and inter-occlusion among vehicles [17, 18], including night-time operation [20].

### 2.4 Connections to other topics

It is interesting to note that the vacant parking spot detection problem presents strong conceptual connections with another video-based surveillance application such as the detection of abandoned and stolen objects. In this case, the goal is detecting stationary blobs in street video sequences and classifying those blobs into abandoned object, stolen object or pedestrian. Thus, these algorithms could be adapted to perform vacant parking spot detection by associating the *car parking* action with *abandonment*, and *car unparking* with *stealing*. Examples of recent approaches in this field are the work by Ortego et al. [41], which presented a long term stationary object detection scheme using a spatio-temporal change detection approach based on filtering out image regions containing moving objects, and by detecting stationary objects by means of online clustering. In [27], edge-segments distributions across the image were used to segment it into background, foreground, and interesting new objects, using clustering to recover the unattended objects.

In particular, some works in this field are specifically oriented to detect long term parked cars, a problem that presents an even closer resemblance to vacant parking spot detection. An example of a recent approach on this topic is the work by Albiol et al. [1], in which parked cars are detected by an algorithm based on a temporal accumulation of static corners detected in certain spatial locations of the scene. The use of corner points, this time coupled with illumination adaptive template matching, was also the basis of the long term object tracking for parked vehicle detection in [14]. More recently, Carletti et al. [8] presented an approach to detect stopped vehicles by encoding spatio-temporal information in a heat map updated using a weighted moving average, that in turn is also employed to update the background model.

### 2.5 Contributions summary

What are the main contributions and novelties of the present work in comparison to the state of the art in video-based vacant parking spot detection? In this regard, it is important to state that QuickSpot is designed with an eye on its practical applicability. To that end, we have devised a strategy for simplifying its setup and minimize its time-to-operation through the training of the visual recognition stage of the system by means of an external database of images of objects of interest, which constitutes a characteristic novelty of our work.

Moreover, we have put special emphasis on evaluating the system real-time performance, an aspect that is analyzed in few works on this topic (e.g. [37]). Thus, to ensure that QuickSpot is capable of providing real time accurate information about the existence of vacant parking spots whatever the number of parking spots under monitoring, we have followed a car-driven approach combining background subtraction, object tracking and visual recognition (a strategy found in other papers in the literature, e.g. [16]). However, we have made a novel analysis of its computational complexity from two perspectives. Firstly, we have compared the ability of performing in real time of QuickSpot vs. an analogous space-driven approach. And secondly, we have evaluated the vacant parking spot detection accuracy of our proposal as a function of the video sampling rate in an attempt to optimize the accuracy-computational complexity trade-off.

Another novelty of this work is that QuickSpot includes a robust parking spot status decision scheme. This robustness is achieved thanks to two aspects: first, as it only makes sense to change the status of a parking spot to "occupied" if a car enters there, QuickSpot's recognition module allows to determine whether what has entered a particular parking spot is a car or a pedestrian. This is especially useful in parking lots that are frequently crossed by pedestrians, which is not a rare situation. And the second novel aspect that allows QuickSpot performing robust parking spot status decisions is the temporal hysteresis in toggling the status of a parking spot. This avoids changing the status in the case a car crosses through a parking spot while maneuvering to park in another spot. These two aspects constitute a novelty compared to other recent works that follow similar approaches [36].

A final contribution of this work is the creation and placement at the public's disposal of QuickSpotDB, an annotated video data set of an outdoor parking site. It is important to highlight that there exist very few publicly available datasets focused on this specific application, and as a consequence, researchers interested in testing their video-based vacant parking spot detectors must either create their own recordings (in fact, most works in this field have historically used proprietary recordings, e.g. [11, 24, 30, 38]) or resort to general purpose benchmark datasets as VIRAT [40], which requires a manual and burdensome annotation process. Moreover, the most relevant annotated datasets that has been made publicly available in recent times (the PKLot database [3]) consists of video frames sampled at a 5-minute rate, which makes them of little use for evaluating car-driven approaches nor real-time performance.

For this reason, the QuickSpotDB video database has been carefully prepared and contains several instances of parking and unparking cars, pedestrians passing by and strong illumination variations.

## 3 The QuickSpot vacant parking spot detector

QuickSpot is a natural extension of our previous approach to vacant parking spot detection based on video analytics [44]. In that work, we presented a space-driven approach to vacant parking spot detection that included a critical comparison between several local and global visual features and classifier systems.

Taking the conclusions drawn in that work as the starting point, this paper presents QuickSpot, a car-based vacant parking spot detection system designed as a background subtraction+object tracking+visual recognition pipeline. The main reason for relying on such approach is motivated by the desire to build a generic and simple framework easily adaptable to different scenarios that ensures real time operation. For instance, had we implemented a more complex pipeline by including specific detectors for different types of objects, this would not only lead to a less generic design, but it would also probably increase the overall computational complexity. Instead, we have preferred to simplify the design of the system and defer to a single visual recognition stage the identification of the moving objects in the scene.

Thus, QuickSpot goes beyond mere parking spot detection status, as it is also able to detect and recognize the presence of objects of interest, such as cars and pedestrians, in regions of interest of the scene. Moreover, QuickSpot includes background estimation and update, which allows to operate in varying lighting conditions during daytime. Finally, the inclusion of motion detection and object tracking capabilities enable QuickSpot to reduce the time devoted to the analysis of parking spots status, which notably reduces the overall computational complexity of the system.

To describe QuickSpot, we take an operational perspective. For this reason, Section 3.1 presents the functional block diagram of QuickSpot during its setup phase, and Section 3.2 describes the functional architecture of the system when it is being exploited.

### 3.1 Setup phase

During the setup phase, the QuickSpot system learns the appearance of occupied and vacant parking spots in the parking site under surveillance, as well as of the objects it will be capable of detecting, tracking and recognizing.

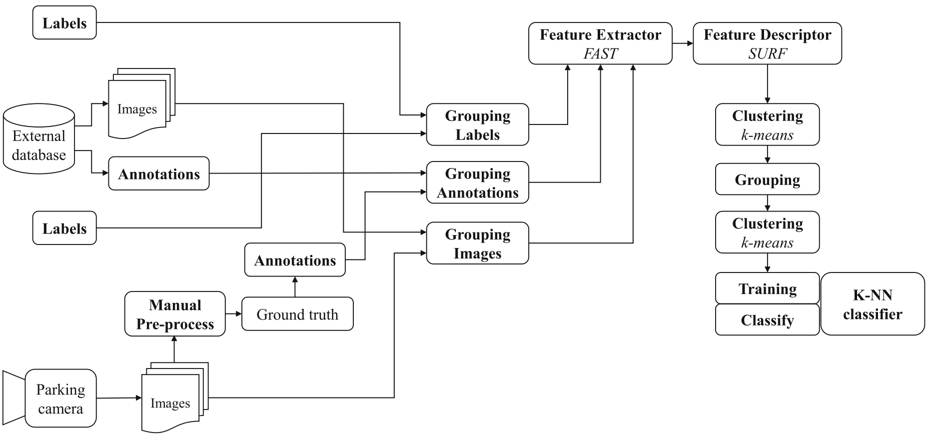



**Fig. 1** Functional block diagram of the QuickSpot system during the setup phase

The functional block diagram corresponding to the setup phase of QuickSpot is presented in Fig. 1. It can be broken down into two stages: training image data collection and annotation, and visual recognition pipeline configuration.

### *3.1.1 Training image data collection and annotation*

Our system contemplates two alternative sources of training data. The first alternative consists on gathering video footage from the parking site during the system setup. The advantage of this approach is that the system is trained with very similar data to that the data it will have to process during the system exploitation stage. We use this approach to collect asphalt parking spot images, so the accuracy of vacant parking spot detection is optimized.

If this approach was also followed to compile vehicles images, it would be more difficult to gather a significant amount of training data, especially if the traffic activity in the parking site under analysis is low. For this reason, QuickSpot also employs, as a second source of training image data, an external database of images of the types of vehicles we want QuickSpot to be able to detect, track and recognize.

Moreover, if the monitored parking site is often crossed by people walking, it is also necessary to include images of pedestrians in the external image database, which will make QuickSpot able to keep the status of a parking spot as vacant when a pedestrian stops in it or passes by.

Let us elaborate on several aspects of the external database training concept. First, this approach has the advantage of simplifying the compilation of many vehicle instances, thus making it easy to build a large and diverse training set with little effort, as it is relatively straightforward to obtain vehicle images from collections available online. Moreover, the use of a standard vehicle images databases simplifies the obtainment of images with different views, which enables the implementation of the QuickSpot system on scenarios with vehicles parked in diverse positions.

Second, at the time of designing the external image database, it is necessary to take into account the perspective under which vehicles are viewed given the characteristics of the parking site under monitoring and the position, elevation and angle of the camera. For instance, if the system is analyzing a parallel parking site –i.e. streetside parking– with cameras located on the sidewalks, the external database should contain car side views.

And third, it is important to consider that it is not likely that the images gathered from external databases match the visual perspective of the system cameras exactly. While this constitutes a challenge to the generalization ability of the classifier responsible for the recognition task, it also allows endowing the system with larger robustness against future changes in camera settings, like height and angle.

After collecting the training images, it is necessary to manually annotate them to guide the subsequent visual feature extraction process. The type of annotation depends on the training data source.

As for the images gathered from the parking site footage, the limits of each parking spot must be set. This quick and simple operation should be conducted on a single frame of the parking site given the stationarity of the camera. To deal with occlusions between parking spots, this annotation can be conducted either using a ground model or a surface model of parking spots [36], depending on the parking disposition.

As regards the external images database, the region occupied by the object of interest (i.e. car or pedestrian) in each image should be indicated. The process consists in delimiting each car with a mere couple of mouse clicks (i.e. it generally is a quick and non-fatiguing operation). It must be noticed that this must be done once for a given database, which can then be reused for the installation of other QuickSpot instances.

### *3.1.2 Visual recognition pipeline configuration*

To perform object detection and recognition, QuickSpot employs local visual features, locating salient points within the image regions delimited during the annotation process, and providing descriptors of their surrounding area.

To that end, features are first detected using the FAST corner detection algorithm [42], and then local Speeded Up Robust Features (SURF) descriptors are computed. Based on an integer approximation to the determinant of Hessian blob detector to build a local feature detector partially inspired in SIFT [33] and several times faster [5], SURF descriptors are invariant to image scale and moderate rotations (e.g. due to camera tilts), robust to changes in illumination, minor changes in viewpoint and partial occlusion, which makes them suitable for the problem at hand. Finally, so as to reduce the number of SURF descriptors per image, we have applied a 2-stage dimensionality reduction process based on k-means clustering with a reduction factor of 50 % per stage.

To determine whether a parking spot is occupied or vacant, QuickSpot employs a supervised classifier. In the current implementation, this classifier is the k-nearest neighbor algorithm, or k-NN. This choice is motivated by the results of our previous work [44], where

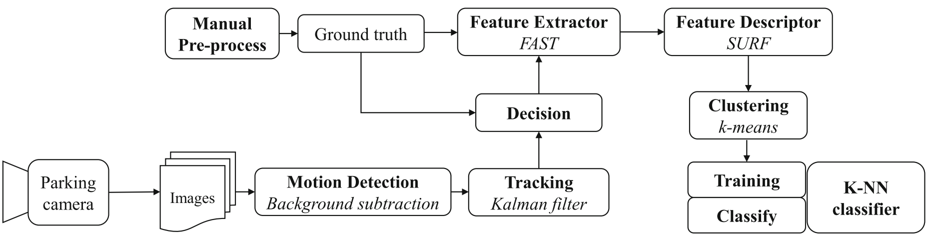



**Fig. 2** Functional block diagram of the QuickSpot system during the exploitation phase

we made a critical comparison involving different types of features and classifiers. Those experimental results proved that when using local visual features such as SURF for conducting recognition, the k-NN classifier outperformed other alternatives such as Support Vector Machines.

Being a lazy learner, the k-NN classifier needs no training process, so at this point it is only necessary to store a database consisting of the visual features gathered from the training images collection conveniently labeled.

## 3.2 Exploitation phase

During the exploitation phase, QuickSpot processes the video feed from the stationary camera monitoring the parking site to determine the status of the parking spots in the site under surveillance in real time.

The functional block diagram corresponding to this phase is depicted in Fig. 2, and Algorithm 1 presents an algorithmic description of the process. It can be decomposed into four sequential steps: initialization, motion detection, vehicle tracking and parking spot status decision.

**Algorithm 1** Algorithmic description of QuickSpot exploitation stage. The status of the parking spots is represented by vector **s**, and ParkingSpotsArea is a variable indicating the coordinates of all the parking spots.

```
Input: Video feed of the parking
Output: Updated parking spot status vector s
Data: ParkingSpotsArea

s = InitializeParkingSpotStatus;
toggle_counter = 0;
for frame = 1 ... end do
    Motion = BackgroundSubtraction (frame);
    if Motion then
        MotionPositions = Tracking (frame);
        Spots = IsInside (MotionPositions, ParkingSpotsArea);
        for spot = 1 ... NumberOf (Spots) do
            CroppedParkingSpot = Crop (Spots(spot), ParkingSpotsArea);
            Object = Recognize (CroppedParkingSpot);
            if Object == Car OR Object == Asphalt then
                toggle_counter(spot)++;
                if toggle_counter(spot) == 3 then
                    s = ToggleStatus (spot);
                    toggle_counter(spot) = 0;
                end
            end
        end
    end
    else
        UpdateBackgroundModel (frame);
    end
    return s;
end
```

### 3.2.1 *Initialization*

At the beginning of the exploitation phase, it is necessary to initialize QuickSpot by determining the status of each parking spot. This task can either be performed automatically by means of visual feature extraction and classification inside the limits of each parking spot, or by manually indicating the status of each spot at the starting time of operation.

### 3.2.2 Motion detection

In this stage, the system performs automatic detection of the moving objects in the input video to determine whether a change in the status of the parking spots is likely to occur.

The stationarity assumption of the camera allows the use of statical background modeling techniques for detecting moving objects. Then, by background subtraction, the system can detect foreground objects in the image.

In particular, a 3-way Gaussian mixture model (GMM) foreground detector compares a given video frame to a background model created upon 50 frames to determine which image pixels belong to the background and to the foreground, yielding a foreground binary mask [26]. The choice of GMM for detecting foreground objects is mainly motivated by its low computational complexity and by the fact that it successfully deals with gradual changes in illumination, which is the expectable situation for a system like QuickSpot that is continuously monitoring a parking site. Moreover, the typically low complexity of the scenes typically encountered in parking sites (in terms of moving objects speed, variability and density) makes GMM a suitable alternative to tackle this task.

It is well known that illumination variations pose a challenge to motion detection and tracking algorithms. For this reason, the next step intends to make our system robust to changes in scene illumination by performing a post-processing of the foreground mask yielded by the GMM foreground detector. This post-processing stage consists of opening and filling morphological operations applied to the mask to eliminate noise and to fill the holes of the remaining blobs.

Subsequently, blob analysis detects groups of connected pixels, which are likely to correspond to moving objects. The blob analysis is used to find such groups and compute their characteristics, such as area, centroid, and the bounding box. Figure 3 presents the motion detection with background subtraction algorithm flowchart.

Finally, the algorithm allows objects to transition from the foreground to the background. For example, a car entering a parking spot would initially be included in the foreground. However, once parked, it must eventually transition to the background, so that it can be detected when it unparks. To do so, the background model is regularly updated with the last 50 frames whenever no motion is detected. By doing so, any further movement of any parked or non-parked car is detected as a novelty.

### 3.2.3 Vehicle tracking

The detection of moving objects described in the previous section constitutes a reliable foundation for vehicle tracking.

Once a moving object is detected, we employ Kalman filtering to predict the location of each tracked object at each frame. Next, we determine the likelihood that each detected object is assigned to a track.

To assign object detections in the current frame to existing tracks, the Hungarian method (also known as Kuhn-Munkres or Munkres assignment algorithm) [28] is applied. The choice of this algorithm is motivated by its ability to solve the assignment problem in $O(k^3)$ time (where $k$ is the number of detections). To find the optimal assignment that yields the largest probability mass over all assignment probabilities, the Hungarian algorithm poses the assignment problem as a weighted bipartite matching problem.

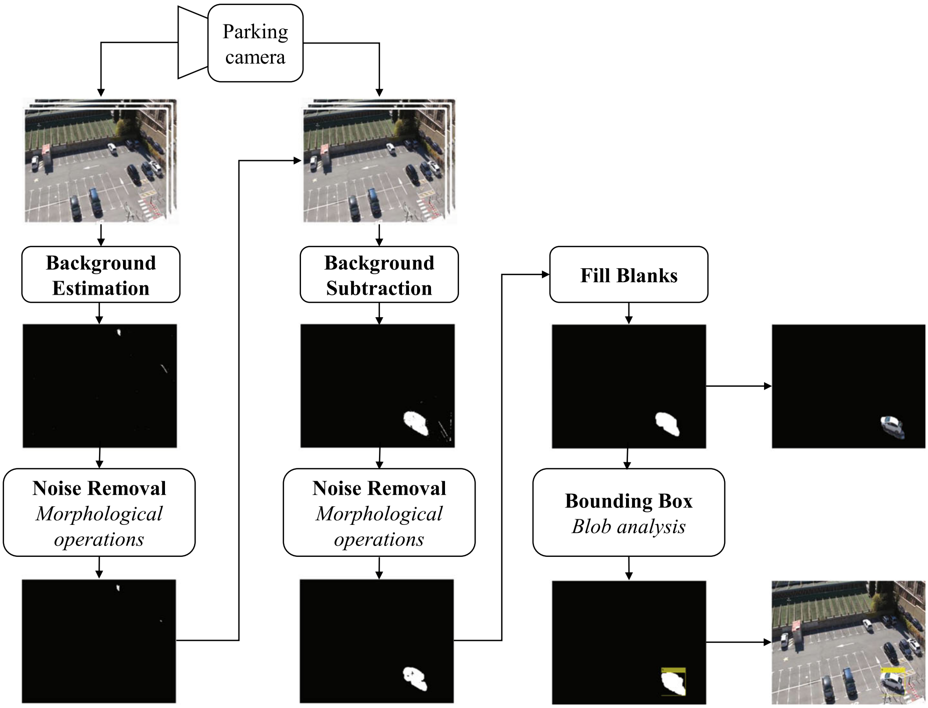



**Fig. 3** Flowchart of the motion detection with background subtraction algorithm

Once the assignment problem is solved, the assigned tracks are updated using the corresponding detections. The unassigned tracks are marked invisible, and an unassigned detection begins a new track. Each track keeps count of the number of consecutive frames where it remained unassigned. If the count exceeds a specified threshold, the system assumes that the moving object either left the field of view or stopped, and it deletes the track in consequence.

### *3.2.4 Parking spot status decision*

Robustness is one of the main issues of concern as regards the decision about parking spot status in real time. This means that it is important that whenever the system decides that the status of a parking spot has changed, this decision must be solid and well-grounded to avoid unnecessary and misleading status toggling. With this aim, QuickSpot implements a robust parking spot decision scheme based on visual recognition and a temporal hysteresis decision.

On one hand, the visual recognition module allows to change parking spot status only if motion in parking spots is caused by vehicles. This is especially useful in parking lots that are frequently crossed by pedestrians, which is not a rare situation. On the other hand, the temporal hysteresis decision is designed to only toggle the status of a parking spot if a consistent recognition result (i.e. vacant or occupied) is obtained during a temporal window of adjustable length. This avoids changing the status in the case a car crosses through a

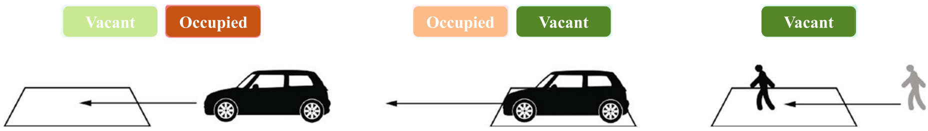



**Fig. 4** Parking spot status change cases

parking spot while maneuvering to park in another spot. The following paragraphs describe the aforementioned processes in more detail.

To detect changes in the status of the parking spots, QuickSpot evaluates the location of the moving objects. If their centroid is located inside one of the parking spots, the system performs visual feature extraction and classification. Otherwise, it skips these two latter steps and processes the next frame. Thanks to this, the overall computational complexity of QuickSpot is notably reduced.

In case the moving object is inside a parking spot, the object is cropped from the corresponding frame using the bounding box obtained from the object detection step. Next, visual features are extracted from the cropped image following the scheme described in Section 3.1.2. Subsequently, the detected object is classified as a car, asphalt or pedestrian using the k-nearest neighbor classifier.[2]

The final step of the system is to decide if the status of the parking spot must change or not, depending on the result of the recognition process (see Fig. 4). Changes in the parking spot status will only occur *i)* when asphalt is recognized inside a parking spot that was listed as occupied (thus, an 'occupied' to 'vacant' status change will take place), or *ii)* when a car is recognized inside a parking spot that was labeled as vacant (thus, a 'vacant' to 'occupied' status change will occur).

For decision stability reasons, a temporal hysteresis decision scheme is adopted. That is, parking spot status only change if the classifier yields the same recognition results for at least $N$ consecutive frames (in our experiments, $N$ was set to 3, which is equivalent to a 3 second temporal hysteresis). Moreover, thanks to the ability of QuickSpot to recognize pedestrians, it is possible to keep the status unchanged when pedestrians are detected inside parking spots, thus reducing occupancy false alarms and avoiding parking spot status toggling.

As an output, QuickSpot returns the camera video feed with a superimposed colored grid indicating which parking spots are vacant (in green) and occupied (in red), as depicted in Fig. 5.

## 4 Video data description

This section describes the video material employed in the development and evaluation of the QuickSpot system. First, we describe the video database created to this end (which we refer to as the QuickSpot database, or QuickSpotDB hereafter), including details about the recording and annotation processes. And second, we also present the external databases employed for training the classifier to recognize cars and pedestrians in the recordings.

[2]As for the configuration of the classifier, initial validation experiments (not reported here for the sake of brevity) allowed to observe that $k = 1$ nearest neighbor yielded the highest recognition accuracy. Moreover, the Euclidean distance is used to measure the similarity between patterns.

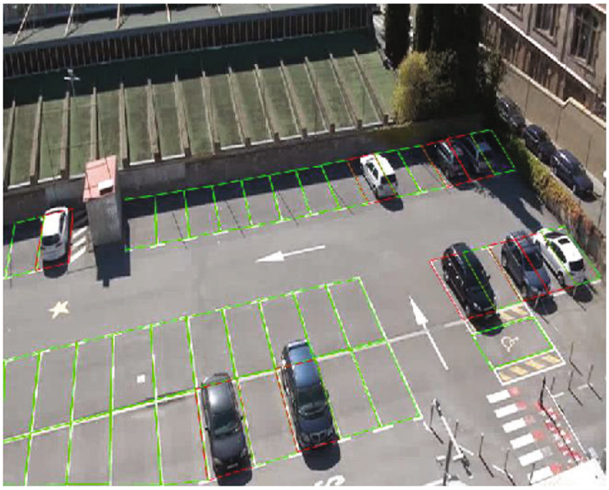


**Fig. 5** QuickSpot video feed output with the superimposed parking spot status color grid

### 4.1 The QuickSpotDB video database

Due to the scarcity of publicly available video material for the vacant parking spot detection application, we decided to create an annotated video database from scratch. To that end, we conducted a recording with a single stationary camera placed on top of one of the buildings of La Salle Campus Barcelona overlooking a private campus parking lot composed of 42 parking spots, as illustrated in Fig. 6.

The parked cars were filmed from a semi-zenithal perspective. The recording was made on a sunny day of the month of July during the morning (from 9:00AM to 12:30PM) and

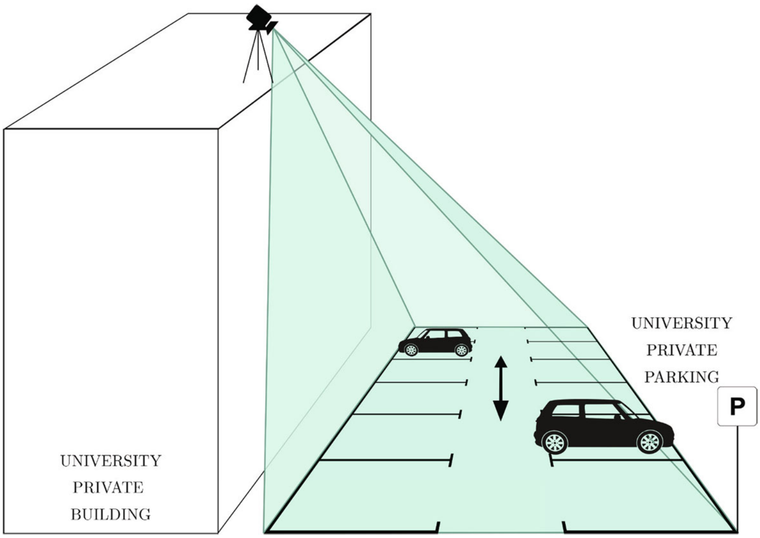
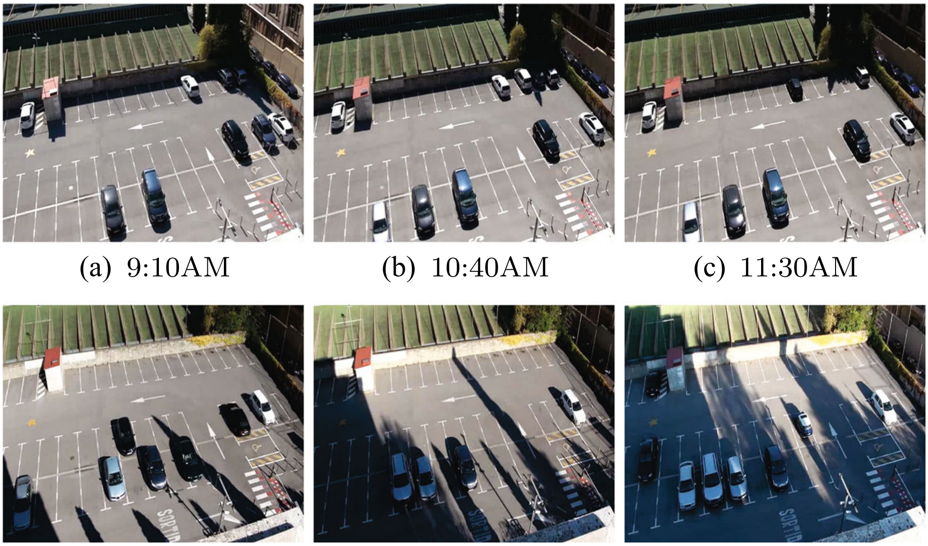



**Fig. 6** Settings of the recording location



(a) 9:10AM (b) 10:40AM (c) 11:30AM

(d) 5:00PM (e) 6:00PM (f) 6:30PM

**Fig. 7** Changes in illumination and moving shadows occurred during the recording

afternoon (from 4:00PM to 6:30PM). Hence, severe changes in illumination occur during the video sequence, which allows testing the QuickSpot system under varying lighting conditions. See Fig. 7 for several examples of video frames at different times of the day, and notice the differences in illumination and the presence of moving shadows cast by tall trees and nearby buildings.

Elaborating a bit on the illumination issue, it is important to notice that recording during the morning and the evening makes our database present the natural and gradual illumination changes occurring during daytime. Moreover, the fact that the monitored parking site is surrounded by trees and buildings introduces the additional difficulty of large shadows cast over the scene, a fact that is noticeable by comparing morning and evening frames (please compare Fig. 7a and f). This constitutes a challenging scenario in terms of testing the robustness of the system to illumination variations, as it has to deal with scenes with changing and highly non-uniform illumination patterns (i.e. parts of the scene are well illuminated while the rest are much darker). In quantitative terms, there is a relative decrease of average luminance of 11 % between morning and evening scenes.

As a result of the recording process, a total of twelve 30 minutes long videos were obtained. After an initial preview of the videos, we detected long periods with no vehicle nor pedestrian activity whatsoever. With the aim of reducing the database size so as to work with more manageable data, these long scenes were edited out, obtaining as a result 43 shorter video sequences (with a total length of 66 minutes) with the following specifications:

- Real resolution: 720x576 pixels.
- Display resolution: 720x405 pixels.
- Frame rate: 25 frames per second.
- Video encoding: MOV.
- Video size: from 5.35MB to 95.81MB.

**Table 1** Objects and events appearing in each one of the 43 video sequences of QuickSpotDB, with their corresponding length

| Video ID | Cars | Pedestrians | Parking events | Length (mm:ss) |
|---|---|---|---|---|
| 1 | 1 | 1 | - | 02:24 |
| 2 | 0 | 3 | - | 04:57 |
| 3 | 1 | 0 | - | 00:50 |
| 4 | 1 | 1 | Unparking (Spot # 28) at frame 2096 | 01:44 |
| 5 | 0 | 1 | Parking (Spot # 12) at frame 893 | 01:51 |
| 6 | 0 | 1 | - | 00:50 |
| 7 | 0 | 1 | - | 00:59 |
| 8 | 1 | 0 | Unparking (Spot # 13) at frame 375 | 00:34 |
| 9 | 1 | 1 | Unparking (Spot # 10) at frame 2049 | 01:31 |
| 10 | 1 | 1 | Unparking (Spot # 12) at frame 1680 | 01:21 |
| 11 | 1 | 1 | Parking (Spot # 9) at frame 643 | 02:09 |
| 12 | 1 | 3 | Parking (Spot # 22) at frame 775 | 01:21 |
| 13 | 2 | 3 | Parking (Spot # 13) at frame 900 / Unparking (Spot # 26) at frame 2798 | 02:29 |
| 14 | 1 | 3 | - | 01:25 |
| 15 | 1 | 1 | Parking (Spot # 3) at frame 681 | 01:32 |
| 16 | 1 | 0 | Parking (Spot # 32) at frame 626 | 00:34 |
| 17 | 1 | 3 | Unparking (Spot # 34) at frame 3827 | 03:05 |
| 18 | 1 | 1 | Parking (Spot # 40) at frame 520 | 01:31 |
| 19 | 1 | 3 | Unparking (Spot # 22) at frame 4240 | 03:00 |
| 20 | 1 | 1 | - | 00:36 |
| 21 | 1 | 0 | - | 00:18 |
| 22 | 1 | 0 | - | 00:11 |
| 23 | 1 | 0 | - | 00:34 |
| 24 | 1 | 0 | - | 00:34 |
| 25 | 1 | 0 | - | 01:19 |
| 26 | 1 | 0 | - | 00:13 |
| 27 | 1 | 0 | - | 00:29 |
| 28 | 1 | 0 | - | 00:28 |
| 29 | 0 | 3 | - | 00:16 |
| 30 | 1 | 2 | - | 01:18 |
| 31 | 1 | 0 | - | 00:14 |
| 32 | 2 | 6 | Unparking (Spot # 26) at frame 3293 / Parking (Spot # 35) at frame 4348 | 03:34 |
| 33 | 1 | 3 | - | 03:39 |
| 34 | 0 | 5 | - | 01:54 |
| 35 | 3 | 2 | Unparking (Spot # 22) at frame 2601 / Unparking (Spot # 40) at frame 5094 | 03:59 |
| 36 | 2 | 0 | - | 01:09 |
| 37 | 1 | 1 | Unparking (Spot # 28) at frame 2041 | 01:39 |
| 38 | 1 | 1 | Parking (Spot # 25) at frame 2124 | 03:34 |

**Table 1** (continued)

| Video ID | Cars | Pedestrians | Parking events | Length (mm:ss) |
|---|---|---|---|---|
| 39 | 1 | 1 | Parking (Spot # 17) at frame 787 | 02:09 |
| 40 | 1 | 0 | - | 00:19 |
| 41 | 2 | 2 | Parking (Spot # 1) at frame 2276 | 02:34 |
| 42 | 2 | 6 | Parking (Spot # 34) at frame 1260 | 02:49 |
| 43 | 1 | 0 | - | 00:29 |

– Video length: from 00:12 to 4:57 minutes.
– Average±standard deviation video length: 1:35±1:11 minutes.

Table 1 presents a detailed description of the complexity of QuickSpotDB, in terms of the objects and events appearing in each of the 43 video sequences of the database, as well as its length. In this sense, it is important to notice that our database is comparable in terms of number of video sequences, average clip length and average number of parking and unparking events per clip to standard video databases for surveillance applications. For instance, two scenes of the VIRAT database recorded in parking settings (scenes 0401 and 0502) contain 17 and 32 sequences, with average lengths of 1:56 and 1:42 minutes and around 1 parking and unparking event per clip, respectively [40].

## 4.2 External training database

As described in Section 3, one of the main features of QuickSpot is the possibility of training the classifier with an external image database of objects of interest.

As the parking site on which QuickSpot has been tested in this work is only used by cars, the external image database contains cars images. But if it was used by other types of vehicles (e.g. trucks, motorbikes, etc.), the external database should also contain images of these kinds of vehicles to make QuickSpot able to recognize them. Moreover, as the monitored parking site is often crossed by people walking, we have also included images of pedestrians in the external image database. The reason for doing so is to avoid false alarms as regards parking spots occupancy in case that, for some reason, a pedestrian stops inside a vacant parking spot.

It is important to note that the contents of this external image database must take into account the perspective under which objects are viewed given the characteristics of the parking site under monitoring and the position, elevation and angle of the camera.

Considering the configuration of the QuickSpot system in the experiments conducted in this work (see Fig. 6), back and side car images extracted from two publicly available image

(a) (b) (c) (d)

**Fig. 8** Samples of rear viewed cars from database Cars 1999 (Rear) 2

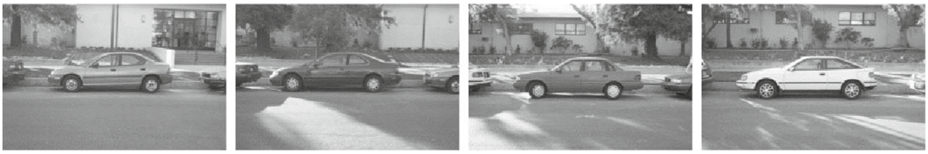



**Fig. 9** Samples of side viewed cars from database Caltech 101

collections are employed. First, we extracted images of cars seen from their rear from the Cars 1999 (Rear) 2 collection.[3] Figure 8 shows a few samples of the images contained in this database. As this collection lacks any type of annotation, it has been manually annotated for feature extraction, a process that took less than 5 minutes of labour.

The second public database we obtained cars images from is Caltech 101.[4] We took images of cars seen from their side. Figure 9 shows a few samples of the images contained in this collection, which contains annotations with some information about where the object is located in the image, which simplifies feature extraction.

Finally, pedestrian recognition is based on training the classifier using samples from the CBCL pedestrian database #1[5] (see Fig. 10 for a few samples). We selected images containing frontal and rear views of pedestrians. Each image was scaled to size 64x128 and aligned so that the person's body was in the center of the image; the height of these people is such that the distance from the shoulders to the feet is approximately 80 pixels. Hence, there is no need of annotations information for feature extraction.

### 4.3 Database annotation

The following step in the database creation process was annotating the video sequences that comprise QuickSpotDB, a task which was tackled using the open source ViPER[6] software tool. More specifically, we used the ViPER-GT (or ViPER Ground Truth Authoring Tool) to create metadata representing the *truth* of the events occurring in the video sequence, so the output of QuickSpot can be evaluated by direct comparison against this ground truth. In particular, we generated one ViPER XML ground truth file (XGTF) for each one of the 43 video sequences that comprise our database.

The first annotated objects were the parking spots appearing in the video sequence. Following the ViPER-GT annotation scheme, each parking spot was defined in terms of a single descriptor consisting of the following attributes: a unique numeric ID (ranging from 0 to 41, as illustrated in Fig. 11a), a static polygon indicating its position in the image, and two dynamic Boolean indicators telling whether the parking spot was occupied or not on every frame of the video file (see Fig. 11b).

The second type of objects we needed to annotate were moving cars and pedestrians appearing in the image, which were also associated to a unique descriptor containing an identifier and also a dynamic bounding box indicating their location in the image, as illustrated in Fig. 12a. However, due to their non-static nature, the location of cars and

[3] Available online at http://www.vision.caltech.edu/html-files/archive.html (last accessed in May 2016).

[4] Available online at http://www.vision.caltech.edu/Image_Datasets/Caltech101 (last accessed in May 2016).

[5] Available online at http://cbcl.mit.edu/software-datasets/PedestrianData.html (last accessed in May 2016).

[6] Available online at http://viper-toolkit.sourceforge.net/ (last accessed in May 2016).

**Fig. 10** Samples of pedestrian images from the CBCL pedestrian database #1

pedestrians had to be updated on a framewise basis. To simplify the tracking of the position of moving objects in the image, we employed the interpolation option of ViPER-GT, which approximates the real path of the object by a concatenation of straight segments, as illustrated in Fig. 12b. Last but not least, it must be noted that only moving objects were annotated, as parked cars were considered as part of the background.

## 5 Experiments

This section describes several experiments designed to evaluate different aspects of QuickSpot's performance. These experiments were conducted on the QuickSpotDB video database described in Section 4.2.

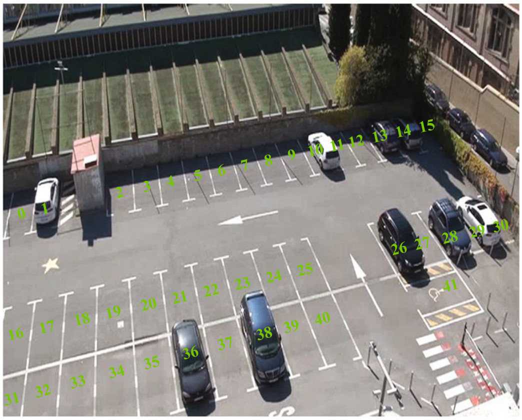
Throughout this section, we will first describe the preparation of the video data. Secondly, we present the results of the foreground mask post-processing implemented to cope with varying illumination conditions. Thirdly, we evaluate the vacant parking spot detection accuracy of QuickSpot. Fourthly, we compare the computational complexity of our proposal against a space-driven vacant parking spot detector counterpart of QuickSpot. And finally, we present an indirect accuracy comparison between QuickSpot and other twenty state-of-the-art systems for vacant parking spot detection.

All these experiments have been conducted under Matlab R2013a on a MacBook Pro with a 2.4 GHz Intel i5 processor and 8GB RAM.

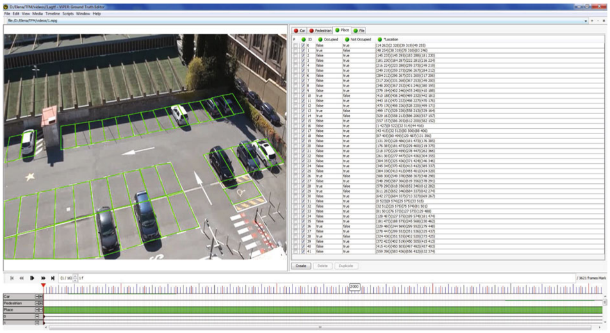
### 5.1 Video data preparation

It is important to notice that parking and unparking events typically occur at slow speed. For this reason, it is not necessary to process each frame of the video feed of the surveillance camera. For this reason, the video databases were downsampled, which is a typical strategy for data reduction applied in video-based vacant parking spot detection (see e.g. [24, 38]).

To that end, we defined a downsampling factor $d$, which indicates that $d$ out of every $f_r$ frames were selected for testing the system, where $f_r$ is the video sequence frame rate (in the QuickSpotDB database, $f_r = 25$). In other words, we select one frame every $d$ seconds and discard the rest, thus processing $\frac{1}{d}$ frames per second. Quite obviously, the total number of processed frames and of samples of parking spots depends on the value of $d$. Table 2




(a)



(b)

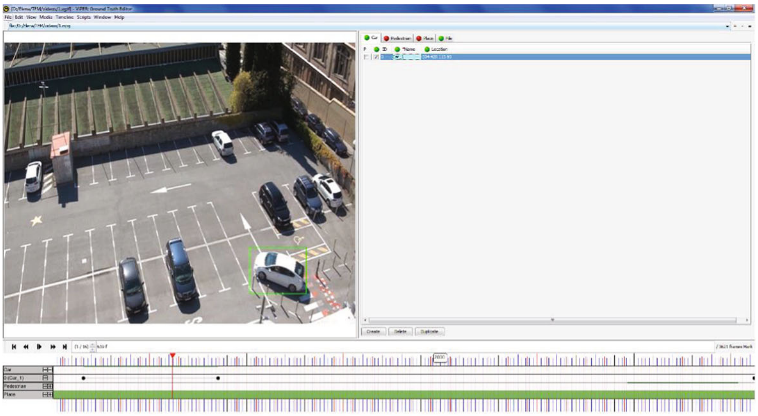
**Fig. 11** Parking spot annotation **a** numeric IDs, and **b** annotation scheme in ViPER

shows the number of frames and parking spot samples corresponding to the video sequences comprising the QuickSpotDB database for values of $d$ ranging from 1 to 10.

At this point, it is important to recall that QuickSpot relies on motion detection and object tracking to detect that a moving object has entered a parking spot, to then determine (via visual feature extraction and recognition) whether it is occupied or vacant. Therefore, it seems logical that its accuracy will be affected by the downsampling, as the more frames we discard, the more difficult the tracking of objects in the scene. For this reason, the following experiment is oriented to measure the accuracy of QuickSpot with different downsampling factors.

## 5.2 Parking spot detection accuracy

In this experiment, we evaluate the accuracy of QuickSpot at determining the status of the parking spots in the parking site for different values of the downsampling factor $d$, analyzing how the former depends on the latter. Then, we present a detailed study of the performance of the system, in an attempt to uncover the reasons behind wrong parking spot status detections.



(a)

(b)

**Fig. 12** **a** Car annotation, and **b** interpolation of the trajectory of a car in ViPER-GT

For starters, let us define the metric employed to measure the accuracy of the system. Of course, it is necessary to compare the status of each parking spot output by QuickSpot against the corresponding status as annotated in the ground truth. If both status match in a certain frame, the status detection is deemed as correct.

**Table 2** Number of frames and samples of parking spots as a function of the downsampling factor $d$ for the QuickSpotDB video database

| | $d = 1$ | $d = 2$ | $d = 5$ | $d = 7$ | $d = 10$ |
|---|---|---|---|---|---|
| Number of frames | 3969 | 1984 | 793 | 567 | 396 |
| Parking spot samples | 166698 | 83328 | 33306 | 23814 | 16632 |

If this idea is extrapolated to a certain span of frames, the resulting accuracy measure becomes the Average Detection Accuracy (ADA) expressed in (1), which is the parking spot status detection accuracy metric employed in this work.

$$\mathrm{ADA} = \frac{1}{K}\sum_{s=1}^{K}\sum_{f=1}^{M}\frac{\mathrm{match}(\mathrm{QS}_{s,f}, \mathrm{GT}_{s,f})}{M} \tag{1}$$

where $K$ is the number of parking spots in the parking site, $M$ is the number of frames under analysis, $\mathrm{QS}_{s,f}$ is the status of spot $s$ in frame $f$ output by QuickSpot, $\mathrm{GT}_{s,f}$ is the status of spot $s$ in frame $f$ annotated in the ground truth, and match$(a, b)$ is a function that equals 1 if $a$ equals $b$, and 0 otherwise. Figure 13 illustrates the computation of this metric on the status of a particular parking spot over a 10 frames span, resulting in a 0.7 Average Detection Accuracy score.

In the first part of this experiment, we have evaluated the accuracy of QuickSpot for different values of the downsampling factor $d$.

To that end, we have downsampled the QuickSpotDB video database using the values of $d$ presented in Table 2 (i.e. $d = 1, 2, 5, 7$ and 10), and computed the ADA metric in each case. The results are presented in Fig. 14.

As expected, the ADA metric value decreases with $d$, as the more frames we discard, the harder it is to track cars and pedestrians, which has a negative effect on the system accuracy. In the case of the QuickSpotDB database, ADA falls more than 10 % between $d = 1$ and $d = 10$. In any case, even analyzing one frame every 5 seconds, we attain accuracies over 0.9.

In the scope of our analysis, the optimal performance is attained with a downsampling factor of $d = 1$, when QuickSpot attains an Average Detection Accuracy of 0.988 on the

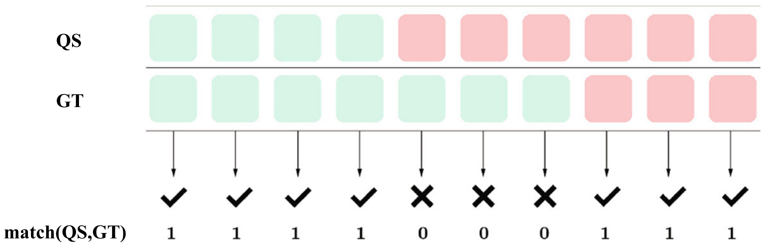



**Fig. 13** Example of computation of the parking spot status detection accuracy metric over a 10 frame span. *Green* and *red squares* represent the 'vacant' and 'occupied' parking spot status

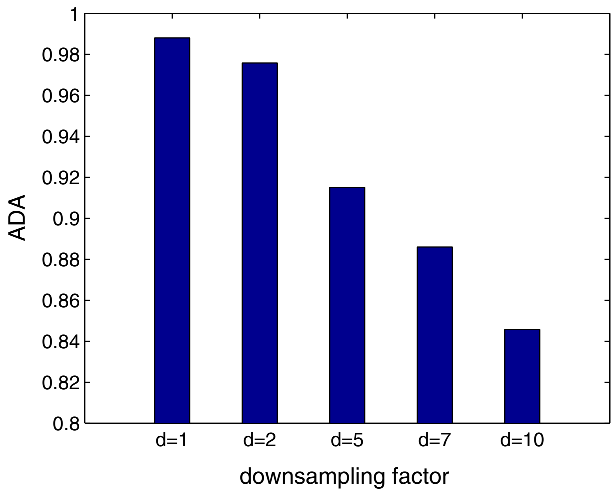



**Fig. 14** Evolution of the value of the average detection accuracy (ADA) metric as a function of the downsampling factor $d$ on the QuickSpotDB database

QuickSpotDB database. This performance is comparable, when not higher, to other recent video-based vacant parking spot detectors that report accuracies between 0.84 and 0.99 (e.g. see [2, 24, 38, 51], and Section 5.5 for an extensive indirect comparison between methods).

The results of this experiment confirm that there exists a trade-off between the downsampling factor and the required accuracy of the system. In this sense, it is necessary to ascertain to which extent downsampling affects the moving object tracking process, which ultimately has an effect on the ADA score. Our experimental results suggest that, given the relatively slow speed of parking and unparking events, processing one frame per second is sufficient for accurate tracking and parking spot status detection, as the ADA score obtained for $d = 1$ indicates.

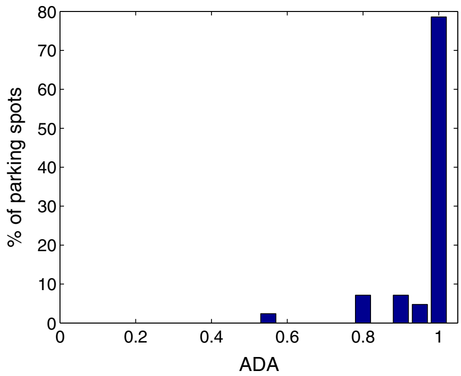
In the second part of this experiment, we have analyzed in detail the performance of the system when a downsampling factor of $d = 1$ is used.

If the ADA metric is broken down per parking spot, we obtain values over 0.9 for 90 % of the parking spots, with only one of the 42 spots on the QuickSpotDB database with an ADA score below 0.8, as depicted in the histogram of Fig. 15.

If very good, QuickSpot's accuracy in detecting vacant parking spots is not perfect. To understand the reasons why, we have analyzed the situations in which the system fails, comparing the parking spot status detected by QuickSpot and that annotated in the ground truth. For a better understanding, we have plotted the $\mathrm{QS}_{s,f}$, $\mathrm{GT}_{s,f}$ and $\mathrm{match}(\mathrm{QS}_{s,f}, \mathrm{GT}_{s,f})$ signals (see Fig. 16) for a few parking spots in which typical errors were detected.

In some cases, the decrease in accuracy is merely due to the fact that the system is evaluated framewise. Thus, if the system detects a change in parking spot status a few frames earlier or later than annotated in the ground truth, this gives rise to a lower ADA value, as $\mathrm{QS}_{s,f}$ and $\mathrm{GT}_{s,f}$ are not coincident for a few frames. In Fig. 16a, this situation is illustrated in the monitoring of the status of parking spot #13 (see Fig. 11a for the reference of the parking spot numbers), which goes from 'vacant' to 'occupied' and to 'vacant' again. QuickSpot correctly detects these status transitions, but with a few frames of difference compared to

**Fig. 15** Histogram of percentage of parking spots vs. ADA metric scores



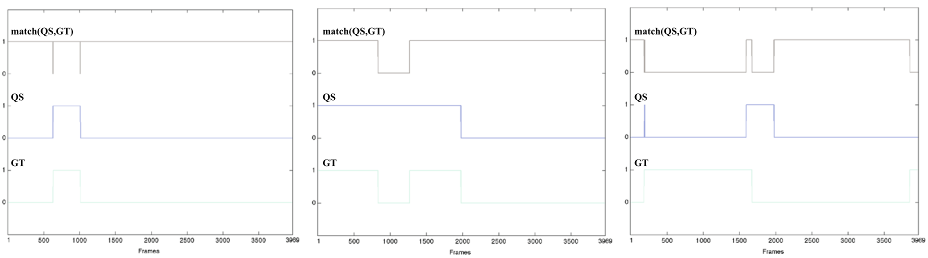

the ground truth, which is not critical from an operational perspective but causes a decrease in the ADA measure.

Another source of parking spot status detection errors are misclassifications, which can be caused by several reasons. One of the causes is the presence of severe shadows that make feature extraction difficult if not impossible when a dark car is present, as occurred in parking spot #14 (see Fig. 16b), in which the system fails to detect an 'occupied' to 'vacant' transition around frame 1000, although it correctly detects a further status change around frame 2000 when a clearer car enters the parking spot.

A more critical situation is caused by a severe mismatch between the actual car views and those contained in the external training database. Notice, for instance, that the view of the cars in the row that is closer to the camera is rather zenithal, so the camera angle is quite different to the back and side car views contained in the database. As a consequence, the classifier struggles to recognize cars parked in these parking spots. An example is parking spot #35, as illustrated in Fig. 16c. However, this problem could be alleviated by designing a more complete external training image database, or one with car views that closely match those present in the scene.

## 5.3 Robustness to varying illumination conditions

As described in Section 3.2.2, QuickSpot tries to gain robustness against varying illumination conditions by means of post-processing the foreground mask that is obtained from the




(a) Parking spot #13 (b) Parking spot #14 (c) Parking spot #35

**Fig. 16** Parking spot status detection errors caused by **a** temporal misalignment, **b** severe shadows, and **c** angle view

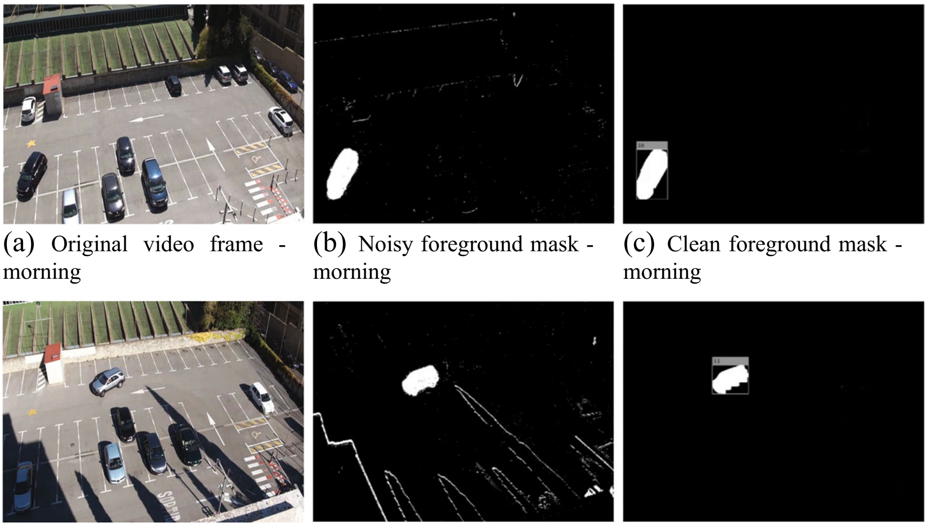
GMM foreground detector. Such post-processing is based on denoising via morphological operations, as changes in illumination appear as noisy pixels on the foreground mask.

In this section, we illustrate the performance of this post-processing on the QuickSpotDB database, which shows strong illumination variations due to the fact it was recorded during the morning and the afternoon, thus showing the natural changes in illumination that occur during a whole day.

As illustrated in Fig. 17, this post-processing is capable of eliminating the effect of severe shadows in the scene. On the top row of Fig. 17, it can be observed that the morning illumination conditions give rise to slight shadows that appear on the upper and central areas of the foreground mask (Fig. 17b), which are successfully eliminated by post-processing (Fig. 17c).

The illumination conditions variation becomes more evident during the afternoon (bottom row of Fig. 17). It can be observed that the shadows cast by the buildings and trees surrounding the parking site generate a very noisy foreground mask (see Fig. 17e). Morphological post-processing, however, manages to remove the noise caused by illumination variations, yielding the foreground mask of Fig. 17f. To sum up, this post-processing allows making motion detection and object tracking highly robust to illumination variations naturally occurring during the day.

To illustrate the performance of the system under different illumination conditions from a quantitative standpoint, Table 3 presents the ADA score attained by QuickSpot split by morning and afternoon recordings of our video database, corresponding to the illumination conditions depicted on the top and bottom rows of Fig. 17. By doing so, we can provide a view on how the changes in illumination affect the accuracy of the system. It can be observed that consistent accuracies close to or higher than 0.98 are obtained, which reveals the robustness of our system at the time of performing under different illumination conditions scenarios.



(a) Original video frame - morning

(b) Noisy foreground mask - morning

(c) Clean foreground mask - morning

(d) Original video frame - afternoon

(e) Noisy foreground mask - afternoon

(f) Clean foreground mask - afternoon

**Fig. 17** Foreground mask post-processing to eliminate illumination variation effects on background subtraction: **a-c** morning example, **d-f** afternoon example

**Table 3** Average detection accuracy (ADA) of the QuickSpot system split by morning and afternoon recordings

| Hour range | ADA score |
|---|---|
| 9:00AM to 12:30PM | 0.978 |
| 4:00PM to 6:30PM | 0.992 |

## 5.4 Computational complexity analysis

One of the most appealing aspects of car-driven vacant parking spot detectors is that their computational complexity can be optimized, especially when compared to space-driven approaches. In fact, the main reason for endowing QuickSpot with object detection and tracking capabilities is reducing time complexity, as it allows performing the visual feature extraction and recognition only on those parking spots liable to status change.

In this experiment, we evaluate the computational complexity of QuickSpot, analyzing its ability to monitor in real time the status of the parking spots in the parking site under surveillance depending on the downsampling factor $d$, and comparing it against a space-driven version of QuickSpot inspired in our previous work [44]. To that end, we evaluate the computational complexity of both the car-driven and the space-driven versions of QuickSpot in terms of the CPU time required to process a single frame (denoted as $T_{frame}$).

The values of the $T_{frame}$ parameter (measured in seconds) of every frame was computed, and Table 4 shows the minimum, maximum, average±standard deviation of $T_{frame}$ corresponding to QuickSpot and to its space-driven counterpart.

It can be observed that QuickSpot is capable of processing each frame in approximately a tenth of a second in average. Thus, it is able to satisfy the real time constraint by a wide margin even if the downsampling factor is set to $d = 1$, thus being able to provide real time information about the status of the 42 parking spots of the parking site under analysis.

In contrast, the space-driven version of QuickSpot needs nearly five seconds to process each frame in average (i.e. nearly 50 times slower than QuickSpot), as its complexity scales with the number of parking spots in the area under surveillance. The reason is that it needs to perform visual feature extraction and classification on each one of the 42 parking spots in the scene in order to process the whole frame, which increases its computational complexity to the point of requiring using higher values of the downsampling factor $d$ (e.g. it would not meet the real time constraint unless $d > 7$).

Of course, the video sequences could be further downsampled to allow the space-driven approach perform in real time, as using for instance a downsampling factor of $d = 10$ would still make it possible to detect most parking events. However, it is our goal to highlight here that space-driven strategies suffer from a trade-off between the downsample rate and the

**Table 4** Time required for processing a single frame ($T_{frame}$, measured in seconds) by QuickSpot and space-driven QuickSpot

| $T_{frame}$ (sec.) | QuickSpot | Space-driven QuickSpot |
|---|---|---|
| avg ± std | 0.0996 ± 0.0668 | 4.5232 ± 0.2744 |
| min | 0.0624 | 3.9792 |
| max | 0.9746 | 6.9337 |

number of parking spots in the area under analysis, whereas the computational complexity of QuickSpot is nearly invariant to this latter factor because it only analyzes those spots in which objects enter.

### 5.5 Comparison with other approaches

As mentioned before, there are no publicly available annotated video databases designed for the vacant video parking spot detection problem. Recently, Almeida et al. published the PKLot database [3], but it consists of images sampled at a 5 minutes rate, which makes it of little use to evaluate video-based approaches. Thus, the lack of a common database and of open source code entails a difficulty at the time of comparing different works in this field.

In spite of that, to provide a reader with a broad view of the performance of the state-of-the-art techniques, in this section we present an indirect performance comparison between QuickSpot and the reviewed literature in terms of the type of approach, the achieved accuracy, and the number of parking spots on which each method was evaluated, as well as the sampling rate of the database employed (when this information was available). Table 5 presents the results of this comparison, in which the compared systems are ordered depending on their approach (car-driven, space-driven, or parking lot-driven) and then by their reported accuracy.

**Table 5** Performance comparison between state-of-the-art techniques in vacant parking spot detection (N/A means "not available", indicating that the information was not explicitly reported)

| System | Type | # of parking spots | Sampling rate | ADA |
|---|---|---|---|---|
| Tschentscher et al. [50] | Space-driven | 10000 | N/A | 0,998 |
| Jermsurawong et al. [24] | Space-driven | 17640 | 10 min | 0,997 |
| Almeida et al. [3] | Space-driven | 211776 | 5 min | 0,996 |
| Fabian [13] | Space-driven | ≥16000 | N/A | 0,995 |
| Liu [31] | Space-driven | 284 | 5 sec | 0,989 |
| Mateus et al. [38] | Space-driven | 17474 | N/A | 0,986 |
| Sastre et al. [43] | Space-driven | 12150 | 1 min | 0,978 |
| Chen et al. [9] | Space-driven | 1373 | Real time | 0,975 |
| MacDonell and Lobo [34] | Space-driven | 335 | N/A | 0,955 |
| True [49] | Space-driven | 200 | N/A | 0,94 |
| Wu et al. [55] | Space-driven | 1100 | N/A | 0,935 |
| Bong et al. [7] | Space-driven | 80 | 2 sec | 0,93 |
| Funck et al. [15] | Space-driven | N/A | N/A | 0,9 |
| Delibaltov et al. [11] | Space-driven | N/A | N/A | 0,88 |
| QuickSpot | Car-driven | 166698 | 1 sec | 0,988 |
| Ichihashi et al. [21] | Car-driven | 54000 | N/A | 0,98 |
| Wang et al. [51] | Car-driven | N/A | N/A | 0,975 |
| Masmoudi et al. [37] | Car-driven | N/A | N/A | 0,916 |
| Huang et al. [19] | Parking-lot driven | 93456 | N/A | 0,995 |
| Huang et al. [18] | Parking-lot driven | 14766 | 5 min | 0,995 |
| Huang et al. [17] | Parking-lot driven | 2600 | 3 sec | 0,975 |

It can be observed that most compared systems consistently detect the correct status of parking spots with an accuracy higher than 0.9. However, there exists a large disparity among the characteristics of the databases (in terms of size and sampling rate), which makes further comparisons at least difficult.

As regards the performance of QuickSpot, it can be observed that it is one of the systems that has been tested on a higher number of parking spots, besides obtaining the highest accuracy among the car-driven approaches reported in the literature.

## 6 Conclusions

In this work, we have highlighted the relevance of smart parking management systems as a key step towards Smart Transportation in urban environments. More specifically, we have focused on investigating how video data gathered by existing infrastructures such as the deployed CCTV camera networks can be exploited to provide drivers with accurate and real time information about the existence of vacant on-street parking spots in a specific area.

To that end, we have designed a computer vision solution called QuickSpot that, using a single stationary camera, determines the status of the parking spots in an area under surveillance with an extremely high accuracy, comparable to that of the most recent works in the field. The proposed system builds on well-established visual feature extraction and classification approaches to perform parking spot status detection, including also motion detection and object tracking capabilities to reduce its computational complexity with a view on the scalability of the proposed solution.

From an experimental viewpoint, we have conducted a study to analyze the existing trade-off between the downsampling factor of the video feed and QuickSpot's accuracy and computational complexity. In this sense, we have observed that the greater the downsampling, the easier it will be to satisfy the real-time constraint, but this affects accuracy negatively. However, the motion detection and object tracking capabilities of QuickSpot allow to reduce its complexity to the point of being able to process 1 frame per second (which can be considered as real time in a parking application) with a detection accuracy close to 99 %. Moreover, it is important to notice that these results have been obtained on a video database that presents heavy changes in illumination conditions, which highlights the robustness of our proposal.

Another relevant aspect of this work has been the creation of QuickSpotDB, a brand new annotated video database for the outdoor parking spot status detection problem, which is at disposal of any interested researcher. The mentioned database allows the analysis of thousands of parking spot samples under different daytime illumination conditions. Moreover, the annotations of moving objects (cars and pedestrians) allows using the database with purposes beyond parking spot detection status.

In this sense, the current version of QuickSpot exploits its motion detection and object tracking capabilities for detecting when a parking spot status change is likely to occur. However, we believe that knowing what types of objects are moving across a parking site is the first necessary step to enable expanding QuickSpot in the future towards applications like behavior analysis or incident detection in parking areas.

Furthermore, we plan on evolving the recognition capabilities to make QuickSpot not only detect whether a parking spot is occupied or vacant, but to also recognize which type of vehicle is parked, which can open a promising research line for smarter parking management systems capable of detecting whether a vehicle is parked in an allowed area or not.

Moreover, we plan testing QuickSpot on video recordings with different parking dispositions and weather conditions other than those of the QuickSpotDB database. However, we understand that the modular approach taken in the implementation of QuickSpot (i.e. the use of SURF descriptors for visual recognition, which are robust to illumination changes, the post-processing of the foreground mask at the motion detection stage, and the flexibility to use ground or surface parking spot models for handling inter-space occlusions) would allow attaining good performance in different scenarios.

Finally, it is important to highlight that the ultimate challenge of any video-based vacant parking spot detection system is round-the-clock (i.e. 24 hours) operation. As the lack of night footage has not made possible to test QuickSpot in night conditions, we aim at creating an annotated night video database filmed with night vision cameras. From that point, we will continue our research towards the adaptation of QuickSpot algorithms to the characteristics of night-time video. In this sense, we foresee a dual approach to tackle round-the-clock vacant parking spot detection, i.e. a day-time and a night-time configuration.

## References


1. Albiol A, Sanchis L, Albiol A, Mossi JM (2011) Detection of parked vehicles using spatiotemporal maps. IEEE Trans Intell Transp Syst 12(4):1277–1291
2. Almeida P, Oliveira LS, Silva E, Britto A, Koerich A (2013) Parking space detection using textural descriptors. In: Proceedings of IEEE international conference on systems, man, and cybernetics, pp 3603–3608
3. Almeida P, Oliveira LS, Britto A, Silva E, Koerich A (2015) PKLot – A robust dataset for parking lot classification. Expert Syst Appl 42:4937–4949
4. Antoniou C, Balakrishna R, Koutsopoulos HN (2011) Synthesis of emerging data collection technologies and their impact on traffic management applications. Eur Transp Res Rev 3:139–148
5. Bay H, Tuytelaars T, Van Gool L (2006) SURF: speeded up robust features. In: Proceedings of ECCV, pp 404–417
6. Bellucci P, Peruzzi L, Zambon G (2015) The LIFE DYNAMAP project: towards the future of real time noise mapping. In: Proceedings of international congress on sound and vibration
7. Bong DBL, Ting KC, Lai KC (2008) Integrated approach in the design of car park occupancy information system (COINS). IAENG Int J Comput Sci 35(1):7–14
8. Carletti V, Foggia P, Greco A, Saggese A, Vento M (2015) Automatic detection of long term parked cars. In: Proceedings of IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp 1–6
9. Chen LC, Hsieh JW, Lai WR, Wu CX, Chen SY (2010) Vision-based vehicle surveillance and parking lot management using multiple cameras. In: Proceedings of IIH-MSP, pp 631–634
10. Choeychuen K (2012) Available car parking space detection from webcam by using adaptive mixing features. In: Proceedings of JCSSE, pp 12–16
11. Delibaltov D, Wu W, Loce RP, Bernal EA (2013) Parking lot occupancy determination from lamp post camera images. In: Proceedings of IEEE conference on intelligent transportation systems
12. Fabian T (2008) An algorithm for parking lot occupation detection. In: Proceedings of CISIM, pp 165–170
13. Fabian T (2013) Parking lot occupancy detection using computational fluid dynamics. In: Proceedings of CORES, pp 733–742
14. Fan Q, Pankanti S, Brown L. (2014) Long-term object tracking for parked vehicle detection. In: Proceedings of IEEE international conference on advanced video and signal based surveillance (AVSS), pp 223–229
15. Funck S, Mohler N, Oertel W (2004) Determining car-park occupancy from single images. In: Proceedings IEEE intelligent vehicles symposium, pp 325–328
16. Gálvez del Postigo C, Torres J, Menéndez JM (2015) Vacant parking area estimation through background subtraction and transience map analysis. IET Intell Transp Syst 9(9):835–841
17. Huang CC, Wang SJ, Chang YJ, Chen T (2008) A Bayesian hierarchical detection framework for parking space detection. In: Proceedings of ICASSP, pp 2097–2100

18. Huang CC, Wang SJ (2010) A hierarchical Bayesian generation framework for vacant parking space detection. IEEE Trans Circuits Syst Video Technol 20(12):1770–1785
19. Huang CC, Dai YS, Wang SJ (2012) A surface-based vacant space detection for an intelligent parking lot. In: Proceedings of international conference on ITS telecommunications (ITST), pp 284–288
20. Huang CC, Tai YS, Wang SJ (2013) Vacant parking space detection based on plane-based Bayesian hierarchical framework. IEEE Trans Circuits Syst Video Technol 23(9):1598–1610
21. Ichihashi H, Notsu A, Honda K, Katada T, Fujiyoshi M (2009) Vacant parking space detector for outdoor parking lot by using surveillance camera and FCM classifier. In: Proceedings of FUZZ, pp 127–134
22. Ichihashi H, Katada T, Fujiyoshi M, Notsu A, Honda K (2010) Improvement in the performance of camera based vehicle detector for parking lot. In: Proceedings of FUZZ, pp 1–7
23. Idris M, Leng Y, Tamil EM, Noor NM, Razak Z (2009) Car park system: a review of smart parking system and its technology. Inf Technol J 8(2):101–113
24. Jermsurawong J, Ahsan U, Haidar A, Haiwei D, Mavridis N (2012) Car parking vacancy detection and its application in 24-hour statistical analysis. In: Proceedings of international conference on frontiers of information technology, pp 84–90
25. Jermsurawong J, Ahsan U, Haidar A, Haiwei D, Mavridis N (2014) One-day long statistical analysis of parking demand by using single-camera vacancy detection. J Transp Syst Eng Inf Technol 14(2):33–44
26. Kaewtrakulpong P, Bowden R (2001) An improved adaptive background mixture model for realtime tracking with shadow detection. In: Proceedings of AVBS
27. Kim J, Ramirez Rivera A, Ryu B, Ahn K, Chae O (2014) Unattended object detection based on edge-segment distributions. In: Proceedings of IEEE international conference on advanced video and signal based surveillance (AVSS), pp 283–288
28. Kuhn H (1955) The Hungarian method for the assignment problem. Nav Res Log 2:83–97
29. Li Y, Chen D (2015) A learning-based comprehensive evaluation model for traffic data quality in intelligent transportation systems. Multimed Tools Appl. doi:10.1007/s11042-015-2676-4
30. Lin SF, Chen YY, Liu SC (2006) A vision-based parking lot management system. In: Proceedings of IEEE international conference on systems, Man and Cybernetics, vol 4, pp 2897–2902
31. Liu S (2005) Robust vehicle detection from parking lot images. Technical Report no. ECE-2005-06, Boston University, Department of electrical and computer engineering
32. Liu J, Mohandes M, Deriche M (2013) A multi-classifier image based vacant parking detection system. In: Proceedings of IEEE 20th international conference on electronics, circuits, and systems, pp 933–936
33. Lowe DG (1999) Object recognition from local scale-invariant features. In: Proceedings of ICCV, vol 2, pp 1150–1157
34. MacDonell A, Lobo J (2011) Visual sensor for smart parking. Boston University. Technical Report No. ECE-2011-01
35. Martchouk M, Mannering F, Bullock D (2011) Analysis of freeway travel time variability using Bluetooth detection. J Transp Eng 137(10):697–704
36. Masmoudi I, Wali A, Alimi AM (2014) Parking spaces modelling for inter spaces occlusion handling. Proc WSCG
37. Masmoudi I, Wali A, Jamoussi A, Alimi AM (2016) Multi agent parking lots modelling for anomalies detection while parking. IET Comput Vis
38. Mateus PA, Maldonado EO, Nino CL (2015) Surveillance and management of parking spaces using computer vision. In: Proceedings of symposium on signal processing, images and computer vision, pp 53–59
39. Ng LL, Chua HS (2012) Vision-based activities recognition by trajectory analysis for parking lot surveillance. In: Proceedings ICCAS, pp 137–142
40. Oh S, Hoogs A, Perera A, Cuntoor N, Chen CC, Lee JT, Mukherjee S, Aggarwal JK, Lee H, Davis L, Swears E, Wang X, Ji Q, Reddy K, Shah M, Vondrick C, Pirsiavash H, Ramanan D, Yuen J, Torralba A, Song B, Fong A, Roy-Chowdhury A, Desai MA (2011) Large-scale benchmark dataset for event recognition in surveillance video. In: Proceedings of CVPR
41. Ortego D, SanMiguel JC, Martínez JM (2015) Long-Term Stationary object detection based on Spatio-Temporal change detection. IEEE Signal Process Lett 22(12):2368–2372
42. Rosten E, Porter R, Drummond T (2010) Faster and better: a machine learning approach to corner detection. IEEE Trans Pattern Anal Mach Intell 32:105–119
43. Sastre RL, Jiménez GP, Acevedo FJ, Bascón MS (2007) Computer algebra algorithms applied to computer vision in a parking management system. In: Proceedings of IEEE international symposium on industrial electronics, pp 1675–1680
44. Sevillano X, Màrmol E, Fernández-Arguedas V (2014) Towards smart traffic management systems: vacant on-street parking spot detection based on video analytics. In: Proceedings of international conference on information fusion, pp 1–8

45. Sevillano X, Socoró JC, Alías F, Bellucci P, Peruzzi L, Radaelli S, Coppi P, Nencini L, Cerniglia A, Bisceglie A, Benocci R, Zambon G (2016) DYNAMAP - Development of low cost sensors networks for real time noise mapping. Noise Mapp 3:172–189
46. Schaller Consulting, Curbing Cars: Shopping, Parking and Pedestrian Space in SoHo, Transportation Alternatives Reports, http://transalt.org/sites/default/files/news/reports/soho_curbing_cars.pdf (2006)
47. Shi X, Zhao N, Xia Y (2014) Detection and classification of traffic lights for automated setup of road surveillance systems. Multimed Tools Appl. doi:10.1007/s11042-014-2343-1
48. Tang Y, Zhang C, Gu R, Li P, Yang B (2015) Vehicle detection and recognition for intelligent traffic surveillance system. Multimed Tools Appl. doi:10.1007/s11042-015-2520-x
49. True N (2007) Vacant parking space detection in static images. University of California, San Diego
50. Tschentscher M, Neuhausen M, Koch C, König M, Salmen J, Schlipsing M (2013) Comparing image features and machine learning algorithms for real-time parking space classification. In: Proceedings of ASCE international workshop on computing in civil engineering
51. Wang Y, Hu Y, Hu X, Zhao Y (2015) Vision-based method for parking space surveillance and parking lot management. In: Proceedings of international conference on image and graphics, pp 516–528
52. Wen Y, Lu Y, Yan J, Zhou Z, von Deneen KM, Shi P (2011) An algorithm for license plate recognition applied to intelligent transportation system. IEEE Trans Intell Transp Syst 12(3):830–845
53. Wolff J, Heuer J, Haibin T, Weinmann G, Voit M, Hartmann S (2006) Parking monitor system based on magnetic field sensors. In: Proceedings of IEEE intelligent transportation systems conference, pp 1275–1279
54. Wu Q, Zhang Y (2006) Parking lots space detection. Mach Learn Fall
55. Wu Q, Huang C, Wang S, Chiu W, Chen T (2007) Robust parking space detection considering inter-space correlation. In: Proceedings of IEEE international conference on multimedia and expo, pp 659–662
56. Yao Y, Wang K, Xiong G (2013) Embedded technology and algorithm for video-based vehicle queue length detection. In: Proceedings of IEEE SOLI, pp 45–50
57. Zou Y, Shi G, Shi H, Zhao H (2010) Traffic incident classification at intersections based on image sequences by HMM-SVM classifiers. Multimed Tools Appl 52:133–145
58. Zhang X, Hu S, Xie J, Zheng S (2013) A novel traffic congestion detection system with feature fusion based track initiation technique. In: Proceedings international conference on information fusion
59. Zhong C, Huang X, Arisona SM, Schmitt G, Batty M (2014) Inferring building functions from a probabilistic model using public transportation data. Computers Environment and Urban Systems 48:124–137

**Elena Màrmol** (M.Sc.) is a junior researcher in the Media Technologies Research Group at La Salle - Universitat Ramon Llull, Barcelona. She holds a degree in Audiovisual Communications by Blanquerna (Universitat Ramon Llull), a B.Sc. in Audiovisual Systems Engineering and a M.Sc. in Telecommunications Engineering by La Salle. Her research interests are focused on computer vision and image processing applied to the development of mobility solutions in urban environments.

**Xavier Sevillano** (PhD) is a senior researcher and associate professor in the Media Technologies Research Group at La Salle - Universitat Ramon Llull, Barcelona. In the past, he was a Visiting Research Fellow at Queen Mary, University of London. He has authored or co-authored over 40 publications in international journals and conferences. His scientific work focuses on computer vision, image processing, and multimedia data analysis, indexing and mining.