Outdoor Vacant Parking Space Detector for Improving Mobility in Smart Cities

Carmen Bravo, Nuria Sánchez*, Noa García, and José Manuel Menéndez

Grupo de Aplicación de Telecomunicaciones Visuales, Universidad Politécnica de Madrid
Av. Complutense, 30, Madrid - 28040, Spain
{cbh,nsa,ngd,jmm}@gatv.ssr.upm.es

Abstract. Difficulty faced by drivers in finding a parking space in either car parks or in the street is one of the common problems shared by all the big cities, most of the times leading to traffic congestion and driver frustration. Exploiting the capabilities that Computer Vision offers, an alternative to those ITS commercial solutions for parking space detection that rely on other sensors different from cameras is presented. The system is able to detect vacant spaces and classify them by the type of vehicle that could park in that area. First of all, an approximate inverse perspective transformation is applied for 2D to 3D reconstruction of parking area. In addition, feature analysis based on Pyramid Histogram of Oriented Gradients (PHOG) is carried out on every parking zone within the parking area. Experiments on real scenarios show the excellent capabilities of the proposed system with independence of camera orientation in the context.

Keywords: vehicle detection, video surveillance, outdoor parking.

1 Introduction

Large scale car parks with hundred/thousands of spaces are more and more common in cities nowadays. However, still very often, especially during busy hours, motorists have trouble parking cars driving around with no guarantee to park.

Increasingly gaining popularity, the vacant parking space detection has been implemented by using various technologies: RFID sensors [1], ultrasonic sensors [2][3], laser scanners [4][5], short-range radar networks [6], and those relying on computer vision [7-14]. With independence of the technology being used, parking guidance systems share the common objective of ensuring easy and time effective operation of parking usage.

Most of the existing solutions to detect parking availability in real environments, rely either on a network of active wireless sensors [15] or ultrasonic ones like the commercial products [16][17]. While they are able to detect in real-time where and when cars are parked, these solutions require a dedicated sensor (installed under the pavement surface or attached to a ceiling on its top) per each individual parking space to be monitored.

* Corresponding author.

L. Correia, L.P. Reis, and J. Cascalho (Eds.): EPIA 2013, LNAI 8154, pp. 30–41, 2013.

Computer vision based approaches have received particular interest for their application outdoor because of presence of surveillance cameras already installed in most of the parking lots. In this case, images from a single camera can be processed to analyze a wider area of the parking lot. This allows controlling several parking spaces using just one sensor, at the same time installation and maintenance costs associated to other types of sensors are reduced.

Although more sensitive to variations of lighting conditions, weather, shadow and occlusion effects, as well as to camera resolution and perspective distortion, detection methods based on cameras and video processing offer a lot of possibilities for creating simple, easy-to-use, and cost-effective outdoor parking spot detection solutions [13]. Based on this technique, many parking space detection algorithms have already been proposed.

Some of them [7][12], proposed the use of visual surveillance to detect and track vehicle movements in an area. Relying on a well-known background subtraction approach, system in [7] is able to recognize simple events in which vehicles can be involved at a parking lot ('stopped vehicle'), at the same time the position of the vehicle is identified in the image. This approach requires the real-time interpretation of image sequences. Other authors [9][10] apply also frame differencing followed by segmentation to locate vacant parking spaces. A different approach to differentiate whether a parking space is occupied or not relies on the classification of the state of a particular region of interest (ROI), i.e. determining whether the ROI is a part of a vehicle or ground surface respectively. In this regard, Dan [11] firstly characterizes each ROI by a feature vector whose components define the textures observed in that region of the image, to train later a general SVM classifier to detect the status of each single parking space. This approach, although more effective, still requires a big amount of training. Based also on ROI characterization, True [14] proposes the combination of color histogram and interest points in vehicles based on Harris corner detection, while Chen et al. [13] select edge density features for classification. Authors in [24] introduced a new descriptor, PHOG, which is based on the analysis of the histograms of oriented gradients in an image but weighting each edge orientation according to its magnitude, so that more edge information is taken into account. This allows a more flexible representation of the spatial layout of local image shape. Results show increased performance compared to other classifiers previously proposed.

A potential downside of these methods is the fact that they analyze the status of each parking space individually, requiring the segmentation in the image of every parking space manually, so that the assistance of a human operator is needed in any case. This is a drawback for the application of this kind of systems to monitor other types of parking areas, different from wide open parking lots, where parking spaces are usually not delimited by lines (e.g. side parking in streets). Additionally, most of the methods do not use to provide the mechanism for their adaptation to a broad set of scenarios each characterized by a different orientation of vehicles and zones of interest in the image due to variations of camera configuration during the installation process. Thus, to the author's knowledge, there is no camera-based solution robust and flexible enough for its consideration as part of a real deployment using surveillance cameras already installed at parking lots and streets.

With this motivation, a novel procedure of vacant parking space detection using a monocular camera is presented in this paper. It relies on the correlation of PHOG features [24] extracted both online and during the initial training phase, followed by the processing of a Dissimilarity-Graph which results as a dedicated geometrical car model slides across the zone. Finally, the distance between adjacent detected cars is used to classify a region as vacant parking spot.

Based on the appearances of the parking area and vehicles in each of the zones in the scene, a simple yet effective method to implement approximate inverse perspective transformation is proposed to reconstruct their 3D geometry beforehand. This approach is particularly used to fit in the image such a geometrical car model over the zone where vehicles are parked in order to find correspondences.

The rest of the paper is organized as follows: Theoretical basis of the system are discussed in Section 2. In Section 3, experimental setup is presented and results on different parking configurations are compared. Section 4 concludes with the summary and future work.

2 System Overview

Following the approach proposed by Pecharromán et al. in [12], our system is divided into independent modules as shown in Figure 1 which separate the low-level of processing, where context is modeled as well as target objects detected, from the high-level, where inference about activity in the parking lot is made.

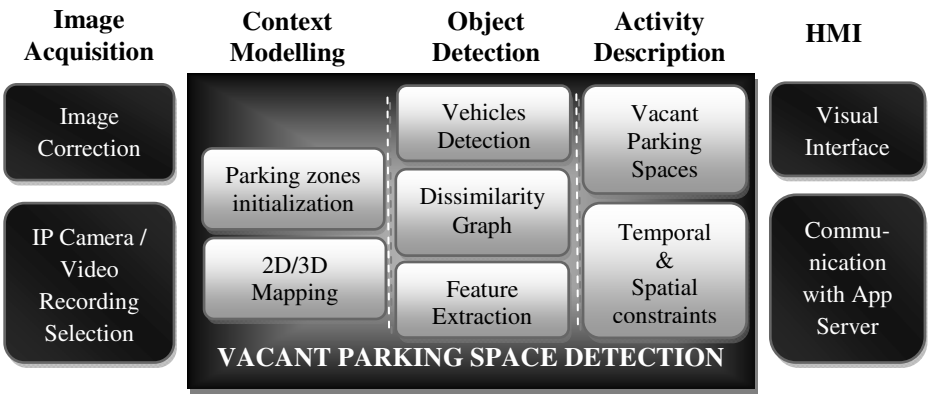

Fig. 1. System overview

First of all, the assistance of a human operator is required for the initialization of parking zones and the calibration of the contextual objects (both area and vehicles). Images from an IP Camera are processed following a bottom-up approach and using individual modules which provides the system with enough flexibility for future improvement. After image correction is applied, the vacant parking space detection starts. Once detection is completed, context-based spatio-temporal analysis is applied.

Adding intelligence in the last step aids the system to filter possible errors in the previous stage. Details of the methodology can be found in following Section. Finally, the system sends relevant information about vacant and occupied spaces in each zone to an App server in charge of providing a service for parking finding to drivers.

2.1 Unconstrained Characterization of Parking Areas

A stream of images is firstly provided to the system by an IP Camera located in the proper position covering its field of view the parking lot. Only in the case that the optic of the camera introduces aberrations (e.g. straight lines in the image become apparently curved), distortion must be corrected before the image can be processed [19].

We are particularly interested in radial distortion, specifically that known as barrel distortion, where image magnification decreases with the distance from the optical axis. Based on Brown's model [20] in the XY system coordinate, if (x,y) is the distorted point, taking into account only the first order term of radial distortion, the corrected points (x',y') will be calculated as:

$$x' = x + k_1 \cdot x \cdot (x^2 + y^2), \qquad y' = y + k_1 \cdot y \cdot (x^2 + y^2), \tag{1}$$

being k_1 positive since the camera used presents barrel distortion.

Once the image has been corrected, the scene needs to be calibrated in order to reconstruct the 3D geometry of the scene, which provides orthogonal information besides the appearance of the object models of interest (in this case, parking area and car models within the zone the area is divided). Most of approaches for 3D reconstruction algorithms count on associating corresponding points in different views to retrieve depth, thus assuming Lambertian surface [18]. However, this is not true for a general application where usually just a sequence of single monocular images from a parking lot is available for its processing with no information about camera calibration parameters.

In this paper, a projective mapping from 3D-World coordinates to pixel coordinates on image plane assuming a Pinhole Camera Model [23] is obtained. The classic technique mostly used requires the use of a dedicated structured object [21]. Auto-Camera Calibration [22] is also possible, although it requires multiple unstructured images quite difficult to obtain when using static surveillance cameras.

The pinhole camera is the simplest, and the ideal, model of camera function. It is usually modeled by placing the image plane between the focal point of the camera and the object. This mapping of three dimensions onto two dimensions is called a perspective projection, being the relationship between World coordinates (X, Y, Z) and pixel coordinates (x, y) on image plane:

$$x = X * \frac{f}{Z}, \qquad y = Y * \frac{f}{Z} \tag{2}$$

where f is the focal length (see Figure 2).

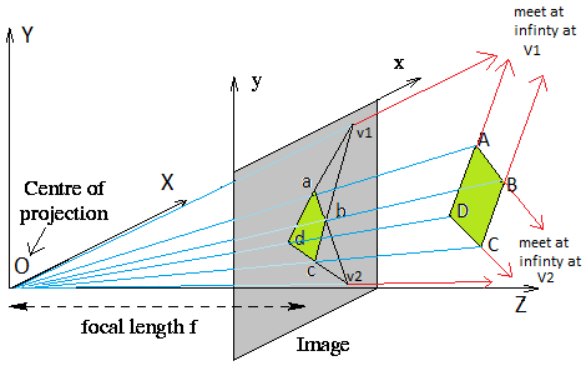

Fig. 2. 2D/3D Mapping using Pin Hole Camera Model

As shown in the Figure 2, suppose *ABCD* represents a rectangular ground plane, which projects polygon *abcd* on image. Although ***AD*** and ***BC*** are parallel, they intersect in infinity at a vanishing point which is projected on the image plane and can be found at the intersection of ***ad*** and ***bc*** marked as *v1* in Figure 2. Similarly, a second vanishing point *v2* can be found. If the distance between *a* and *A* is very small compared to distances *v1*-*V1* and *v2*-*V2*, the 3D point corresponding to *v1*, *V1*, can be approximated applying Equation (2). Similarly, *V2* can be obtained.

Vector algebra can be applied to move any point in 3D space towards the vanishing points at the same time the approximate inverse perspective projection of such point over 2D space can be obtained. Points *ABCD* can be moved proportionately toward or away from center of projection *O*, while their projection will still fall on *abcd* respectively. Direction perpendicular to the ground plane *ABCD* can be found easily by the cross product of the direction vectors along vanishing points *V1* and *V2*. This transformation has shown to be a good approximation for reconstructing object models in the scene, i.e. parking area model that covers the entire scene.

The last step in the initialization process is the establishment of each zone within the entire parking scene. Three points in the image must be selected manually clockwise. The first two points define the orientation of the car within the area, while the second and third point describes the direction of the global zone being extracted.

2.2 From Feature Extraction to the Detection of Vehicles in the Image

Once the system is initialized, a second module makes use of knowledge from context modeled in the first module to analyze features extracted from the zone of interest and thus detect vehicles in the image.

First of all, a moving cuboidal car appearance model is placed over the parking zone in 3D coordinate system to create the database of training samples of occupied spaces. Two kinds of features were considered for feature analysis: Edge Density [13] and Pyramidal Histogram of Orientation Gradients (PHOG) [24].

Once the system has been trained, the cuboidal car model is moved again on the parking zone and new features are again extracted in order to compare them with

training samples. This analysis results in a Dissimilarity Graph which is subsequently processed in order to discriminate vacant spots from the occupied ones in the parking lot being monitored.

Edge Density Features

To extract these features, each side of the car model needs firstly to be divided into blocks as shown in Figure 3. The edge density of a block region is given by:

$$\text{Edge density} = \frac{1}{\text{Block Area}} \sqrt{\sum_{block} \left(g_v^2 + g_h^2 \right)} \tag{3}$$

where g_v and g_h are the vertical and horizontal responses to the Sobel operator at each pixel within the block.

Original test scenario

Car model on test vehicle sample

Fig. 3. Edge density feature extraction

Only the most visible sides of the car model, divided into smaller blocks, are used for calculating the corresponding edge density feature vector.

In order to find correspondences between test and training samples, a Covariance distance metric is used as proposed in [25]. This dissimilarity measurement is based on the correlation among the edge density corresponding to different blocks from the car model:

$$d(C_{test}, C_{train}) = \sqrt{\sum_{i=1}^{n} \ln^2 \lambda_i (C_{test}, C_{train})} \tag{4}$$

where $\boldsymbol{C_{train}}$ is the covariance matrix of the feature vectors in the training set and $\boldsymbol{C_{test}}$ is an approximated covariance matrix calculated from a single test sample as indicated in equation (4). Finally, λ_i represents their i^{th} Eigen-value.

$$C_{test} = (x - m_i) \cdot (x - m_i)^t \tag{5}$$

$\boldsymbol{m_i}$ is the mean of the feature vectors in the training set, and $\boldsymbol{x}$ is the test feature vector where the distance obtained is added to similarity vector. The similarity vector is processed as described in the next section.

PHOG Features

PHOG features have been described in [24] by Bosch et al. These features represent an image by its local shape and the spatial layout of the shape. The descriptor consists of a histogram of orientation gradients over each image block, at each resolution level. The PHOG method uses *L* levels, which determine the number of blocks taken into account for analysis. In each level, the image is divided into 2^L blocks for each dimension. The local shape is captured by the distribution over edge orientations within a region *(L=0),* while the spatial layout by tiling the image into regions at multiple resolutions *(L>0).*

Two different PHOG approaches are tested. In the first one, the descriptor is computed by using edge orientation $\Theta(x,y)$ over each image block, while in the second approach, the contribution of each edge orientation is weighted according to its magnitude $m(x,y)$. Magnitude and orientation of the gradient on a pixel (x,y) are calculated as:

$$\mathrm{m}(\mathrm{x},\mathrm{y}) = \sqrt{g_{\mathrm{x}}(\mathrm{x},\mathrm{y})^2 + g_{\mathrm{y}}(\mathrm{x},\mathrm{y})^2} \tag{6}$$

$$\theta(\mathrm{x},\mathrm{y}) = \arctan\left(\frac{g_{\mathrm{x}}(\mathrm{x},\mathrm{y})}{g_{\mathrm{y}}(\mathrm{x},\mathrm{y})}\right) \tag{7}$$

where $g_x(x,y)$ and $g_y(x,y)$ are image gradient along x and y directions, respectively.

The resulting PHOG vector for each level is concatenated in order to obtain a final feature vector. A $L=3$ has been used in our experiments.

In this case, a Chi-square distance χ^2 between a single test sample vector, $\boldsymbol{X_{test}}$, and a single PHOG feature vector of the training set, $\boldsymbol{X_{train}}$, is used for analyzing similarity:

$$\chi^2 = \sum_{i=1}^{n} \frac{(X_{test,i} - X_{train,i})^2}{(X_{test,i} + X_{train,i})}, \tag{8}$$

denoting as i the i^{th} element of the feature vector of length n.

At every model location, the concatenated PHOG feature is compared with the training samples according to equation (8) and the minimum distance is added to the similarity vector. This similarity vector is processed as described in the following section in order to locate vacant parking spaces in the scene.

2.3 Vacant Space Detection

Occupied spaces by vehicles are used as training samples, and a similarity vector $\boldsymbol{D_{car}}$ represents similarity with the car training samples at each location. When represented as a graph, the x-axis represents every location at which the car model is placed and the y-axis contains the similarity measured (Figure 4). $\boldsymbol{D_{car}}$ has lower values in regions where cars are located.

Figure 4 shows what is known as Dissimilarity Graph, i.e. graphical representations of Chi-Square Distance between test and training samples when car model is displaced from beginning to end of a parking zone. The use of gradients orientations in the zone requires that differences between consecutive maximums and minimum must be greater than certain threshold to be detected as vehicle in the image. It can be shown that the incorporation of magnitude information produces a graph in which cars can be classified with an automatic absolute threshold.

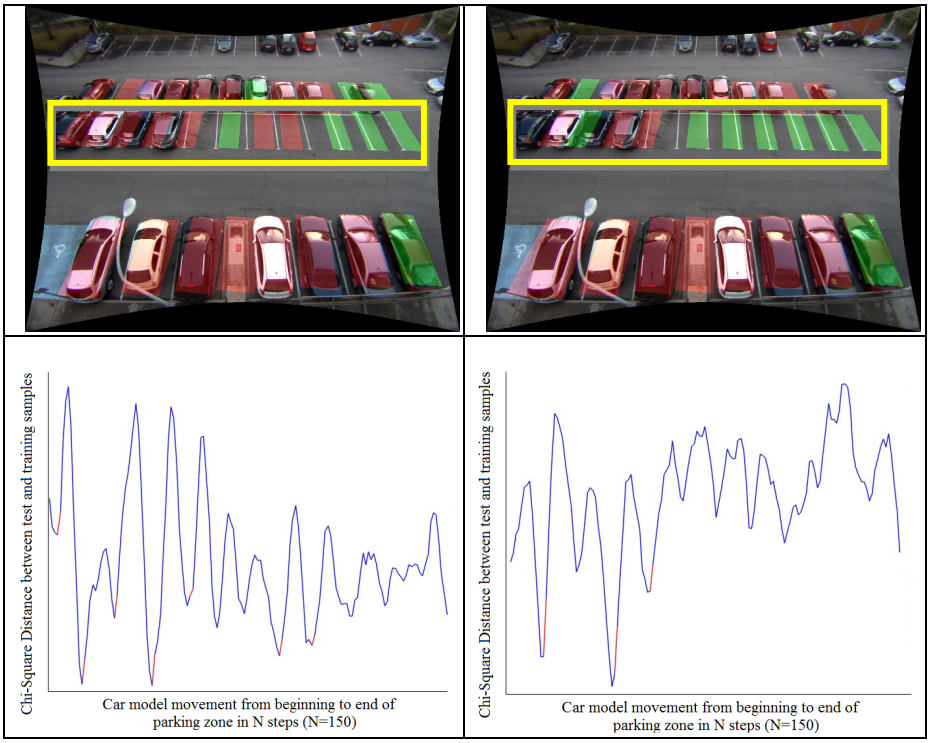

Fig. 4. Comparison between Dissimilarity-Graph using PHOG features, using edge orientations only (left) and edge orientations weighted by magnitude (right) for the second zone in the image. Marked in red those locations detected as cars in the image.

3 Experiments

In order to analyze the performance of our system for both kinds of features we have tested the proposed methods using several video sequences. The sequences were captured using two IP cameras: (1) an AXIS-211 CCD camera with 640 × 480 resolution and frame rate up to 30 fps; (2) AXIS P3346/-VE with 2048 × 1536 resolution and frame rate up to 30 fps. Once the radial distortion is corrected, the image is resized to 640 × 480. Two sceneries were tested. Camera (1) monitors Parking C while camera (2) monitors Parking B. The training dataset was constructed from a group of 15 static

images. Each zone within the parking is trained individually. For the car dataset, around 30 samples of every side of the single-car model were extracted. Parking C was tested with frames covering around 6 hours of one day. In contrast, Parking B was tested also with two sequences with frames that cover up to 6 hours. The sequences were recorded at 10 meters of height from a building in front of the parking scene on days with different lighting and weather conditions. The experiments were carried out on a PC with a 3.4 GHz i7 processor and 8GB RAM.

Accuracy (ratio of samples that the system has been able to classify correctly among all samples) is the main indicator used to evaluate performance. Results shown in Table 1 compare results of the proposed approach with the edge density one, where only edge density features were used; the proposed approach providing clearly better performance. Visual results from both parking areas are shown in Figures 5 and 6.

Table 1. Rates using our method based on PHOG features and those relying on edge density features

	Accuracy		
	Parking B Cloudy day	Parking B Sunny day	Parking C
PHOG-based	0. 937	0.821	0.878
Edge density	0.658	0.523	0.853

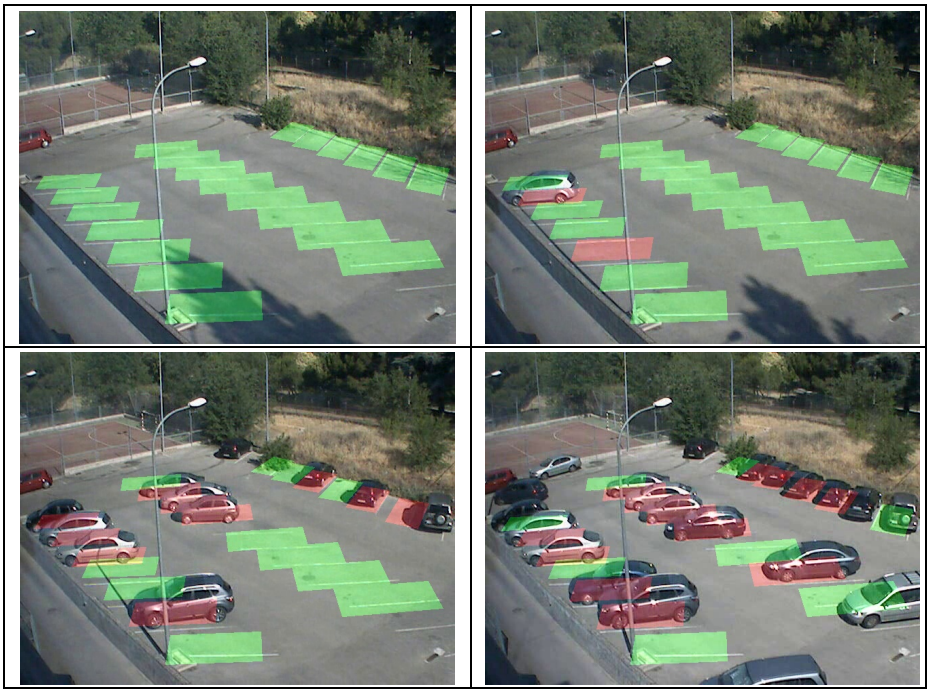

Fig. 5. Vacant space detection visual results at 'Parking C' context at UPM's premises

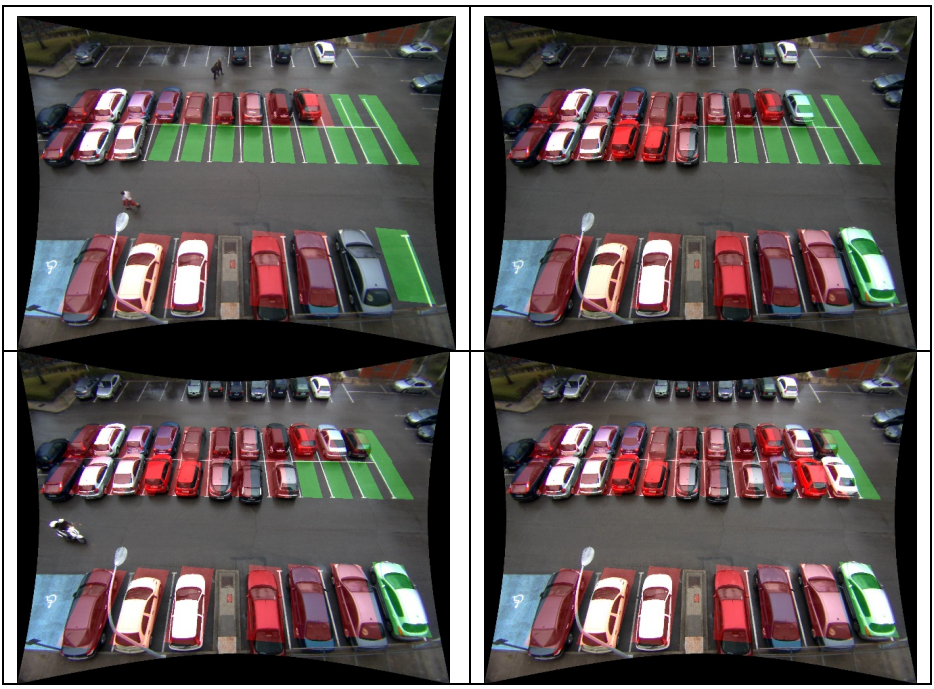

Fig. 6. Vacant space detection visual results at 'Parking B' context at UPM's premises

The computational cost for the proposed method depends directly on the number of sides of the car model to be studied. Under these conditions, and using a system like the specified above, one parking space for which only one side is extracted takes up to 2 seconds to be fully classified.

4 Conclusions and Future Work

In this paper, a system for scene reconstruction as well as vacant space detection in a parking lot is proposed. Experiments performed in real parking scenarios show good detection results achieved with the proposed method.

However, further work is needed to make system, on one hand, more robust to possible errors due to manual configuration during initialization process. Tests carried out on two different parking scenes demonstrate that the higher the number of points taken in each direction the better. The selection of points delimiting the whole parking area contributes to the reduction of the extrapolation error.

Besides, if the number of vehicle training samples is reduced too much, the dissimilarity graph is deteriorated and the detection is harder to achieve. It seems to be necessary the introduction of a process able to update progressively the vehicle database that will be used during the classification process always that a new possible candidate is available for testing. In any case, more tests on other parking scenes seem

to be needed before conclusions on the suitable minimum number of initial training samples that need to be extracted.

Now, a first stable version has been achieved. Current work at different lighting conditions shows that the proposed method can be used with good accuracy under these circumstances although further testing and adjustment is needed in this sense. Improvements should be made in order to achieve higher rates at several extreme weather conditions, in presence of shadows projected either by cars in the vicinity of a parking space or by buildings closed to the parking area.

On one hand, the incorporation of an adaptive process for establishing the threshold used during classification of vehicles that lead to a more automatic system is also foreseen. On the other hand, temporal analysis and the incorporation of more information about context around occupied spaces should filter, at a higher-level in the processing chain, some of the errors obtained due to misclassification. For instance, in some parking lots the white painted layout present on the parking floor could be used for a precise analysis of each parking space locally. It would probably help calibrating the camera. Finally, the use of a multi-camera system would enhance detection resulting after applying fusion at a higher level of abstraction.

The final objective is to increase the robustness of the proposed system in a broad set of outdoor parking spaces, not only in parking lots but also in the streets.

Acknowledgements. This work has been performed in the framework of the WePark and FOTsis projects. WePark has received research funding from the Spanish Industry, Tourism and Commerce Council and ERDF funds (European Regional Development Fund) under the Project reference TSI-020100-2011-57. The FOTsis project is partly financed by the European Commision in the context of the 7th Framework Programme (FP7) for Research and Technological Development (Grant Agreement no. 270447).

References

1. Swedberg, C.: SF Uses Wireless Sensors to Help Manage Parking. RFID Journal (2007)
2. Degerman, P., Pohl, J., Sethson, M.: Hough transform for parking space estimation using long range ultrasonic sensors. SAE Paper. Document Number: 2006-01-0810 (2006)
3. Satonaka, H., Okuda, M., Hayasaka, S., Endo, T., Tanaka, Y., Yoshida, T.: Development of parking space detection using an ultrasonic sensor. In: 13th World Congress on Intelligent Transportation Systems and Services (2006)
4. Jung, H.G., Cho, Y.H., Yoon, P.J., Kim, J.: Integrated side/rear safety system. In: 11th European Automotive Congress (2007)
5. Schanz, A., Spieker, A., Kuhnert, D.: Autonomous parking in subterranean garages: a look at the position estimation. In: IEEE Intelligent Vehicle Symposium, pp. 253–258 (2003)
6. Gorner, S., Rohling, H.: Parking lot detection with 24 GHz radar sensor. In: 3rd International Workshop on Intelligent Transportation (2006)
7. Foresti, G.L., Micheloni, C., Snidaro, L.: Event classification for automatic visual-based surveillance of parking lots. In: 17th International Conference on Pattern Recognition, vol. 3, pp. 314–317 (2004)
8. Wang, X.G., Hanson, A.R.: Parking lot analysis and visualization from aerial images. In: 4th IEEE Workshop Applications of Computer Vision, pp. 36–41 (1998)

9. Lee, C.H., Wen, M.G., Han, C.C., Kou, D.C.: An automatic monitoring approach for unsupervised parking lots in outdoors. In: 39th Annual International Carnahan Conference, pp. 271–274 (2005)
10. Masaki, I.: Machine-vision systems for intelligent transportation systems. In: IEEE Conference on Intelligent Transportation System, vol. 13(6), pp. 24–31 (1998)
11. Dan, N.: Parking management system and method. US Patent, Pub. No.: 20030144890A1 (2003)
12. Pecharromán, A., Sánchez, N., Torres, J., Menéndez, J.M.: Real-Time Incidents Detection in the Highways of the Future. In: 15th Portuguese Conference on Artificial Intelligence, EPIA 2011, Lisbon, pp. 108–121 (2011)
13. Chen, L., Hsieh, J., Lai, W., Wu, C., Chen, S.: Vision-Based Vehicle Surveillance and Parking Lot Management Using Multiple Cameras. In: 6th International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Washington, DC, pp. 631–634 (2010)
14. True, N.: Vacant Parking Space Detection in Static Images, Projects in Vision & Learning, University of California (2007)
15. SFPark project, `http://sfpark.org/` (accessed May 2013)
16. SiPark SSD car park guidance system, Siemens AG (2011)
17. IdentiPark, Nortech Internacional (2013)
18. Kang, S.B., Weiss, R.: Can We Calibrate a Camera Using an Image of a Flat, Textureless Lambertian Surface? In: Vernon, D. (ed.) ECCV 2000, Part II. LNCS, vol. 1843, pp. 640–653. Springer, Heidelberg (2000)
19. Torres, J., Menendez, J.M.: A practical algorithm to correct geometrical distortion of image acquisition cameras. In: IEEE International Conference on Image Processing, vol. 4, pp. 2451–2454 (2004)
20. Brown, D.C.: Decentering distortion of lenses. In: Photogrommetric Eng. Remore Sensing, pp. 444–462 (1966)
21. Zhang, Z.: A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(11), 1330–1334 (2000)
22. Faugeras, O.D., Luong, Q.-T., Maybank, S.J.: Camera Self-Calibration: Theory and Experiments. In: 2nd European Conference on Computer Vision, pp. 321–334. Springer, London (1992)
23. Hartley, R., Zisserman, A.: Multiple View Geometry in computer vision. Cambridge University Press, Cambridge (2003)
24. Bosch, A., Zisserman, A., Munoz, X.: Representing shape with a spatial pyramid kernel. In: 6th ACM International Conference on Image and Video Retrieval, pp. 401–408. ACM, New York (2007)
25. Förstner, W., Moonen, B.: A metric for covariance matrices. Technical Report, Department of Geodesy and Geoinformatics, Stuttgart University (1999)