MANAGING AND MONITORING OF PARKING LOT BY A VIDEO CAMERA

Tomislav Čaklović, Ivan Aleksi, Željko Hocenski
University of Osijek, Faculty of Electrical Engineering
Kneza Trpimira 2b, HR-31000 Osijek, Croatia

Summary:

In this work we present an automated parking lot system that is managed and monitored by a video camera. We used a web camera for monitoring and the PC with MATLAB environment for managing the parking lot model. Captured frame is first converted to the grayscale image and then to the binary image with Sobel edge detection method. Image is dilated and filled in order to get the cars in white pixels and the parking place as black pixels. To each parking place is dedicated a certain region in the image. Mean value of each region was used for indicator if the parking place is occupied or free. By using a camera as a sensor, relatively large parking area is covered with only one sensor. Proposed system is relatively cheap. It does not require major additional installations on the existing parking lot. Real-time experiment was made on a parking lot model. Experimentally observed execution time implies to a possible application on a real parking lot system.

1. INTRODUCTION

In this work we presented an automated system for managing and monitoring of parking lot by a video camera and a CPU, as it is shown in Fig. 1. Usually, surveillance systems used on parking lots are based on pure video recording. In order to fully use the camera capabilities, not just recording long video, a modern solution for parking lots management is presented. Camera-based sensor detects free parking places and informs the drivers at the parking lot entry about free parking place positions.

In this paper, free parking places are detected by using edge detection method, with indoor scenario considered only. More robust method for outdoor scenario is presented in [1] and [2].

Authors in [1] presented *ParkLotD* fuzzy c-means clustering method which is applicable for different weather and light conditions.

In this work we considered top-view camera. 2D Homography is used in [3] to get a pseudo-top-view image of a parking lot by using side-placed camera. Image is then processed as a top-view image of the parking place.

Proposed model of parking lot is described in Section 2. Section 3 deals with efficient method for detection of free parking places. Experimental results are given in Section 4. Section 5 concludes the paper.

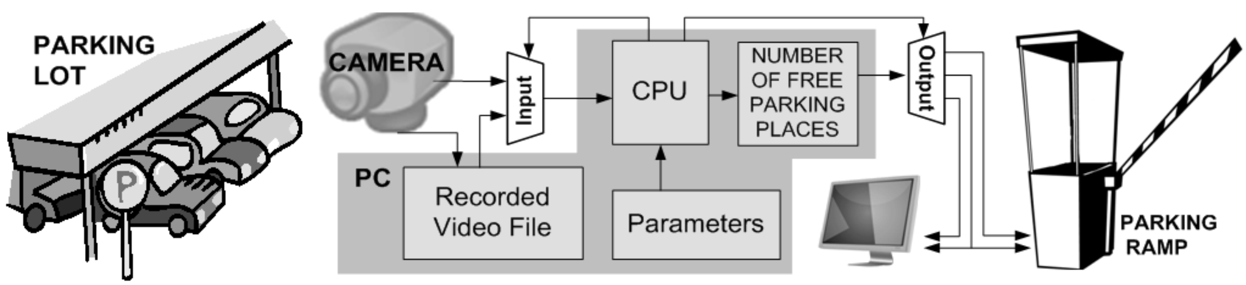


Fig. 1. Managing and monitoring of a parking lot system by a camera sensor and a PC.

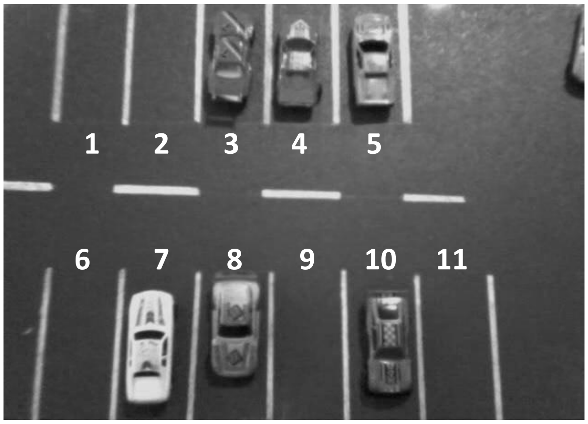


Fig 2. Parking lot model with ratio 1:64, with eleven parking places and six car parked.

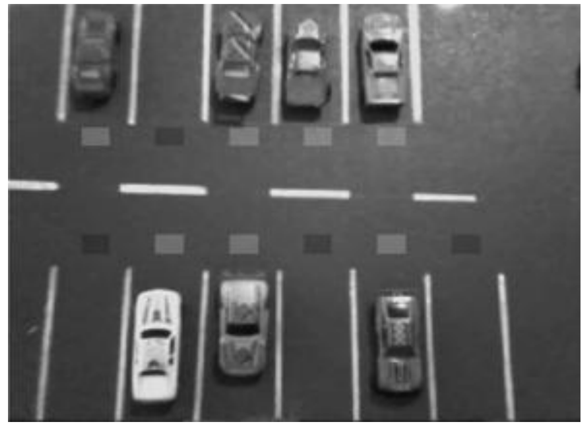


Fig 3. Parking lot model with indicators for free and occupied parking places.

2. PARKING LOT MODEL

Parking lot model considered in this work is a simple indoor parking garage, as it is illustrated in Fig. 2. Parking lot has available non-stop constant lightning source. As usual, camera-based systems can operate only in lightened environments. Camera is placed on top of the parking area. Camera used for this paper was the VGA web camera 640x480 resolution at a frame rate up to 30 frames per second. Parking lot model was made with small cars in ratio 1:64 in comparison to the real cars. Parking places are made with correspondence to the car size. To each parking place is assigned unique ID and certain image region. Image region consists of pixels assigned to a certain parking place. Parking place IDs are numbers from 1 to 11, as it is shown in Fig. 2. Image gathered from camera is processed by a CPU. At the output, CPU provides the information about the number of available free parking places. Information about free parking places can be used in several ways. It can be displayed with a lamp placed at the free parking place. It can be used for car entry allowance if there is a free parking place, at a parking ramp, on the parking lot entrance. Another interesting application would be for investigation purposes. When there is assumption that certain parking place was visited in the past and then the time log is required. Proposed system can be easily adjusted for time logging that indicates when the parking place was occupied or free.

Indicators of free parking places used in this work are placed near the parking place, as it is displayed in Fig. 3. In Fig. 3, light rectangle denotes an occupied parking place, while dark rectangle denotes a free parking place.

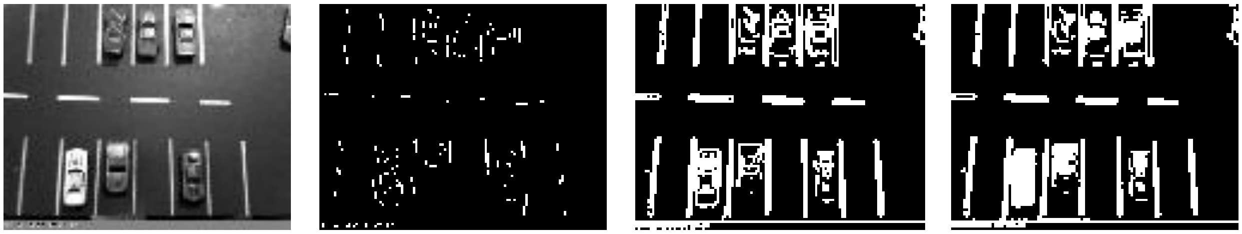
3. FREE PARKING PLACES DETECTION METHOD

This section describes the *free parking places detection method* on a parking lot system described in Section 2. Algorithm for this method is proposed in Fig. 5.

By using MATLAB's function *mmreader* video is loaded from an *avi* file. After video is loaded, frames are accessed by using MATLAB command *read*. When frame is loaded, it is transferred from the RGB color map to the GRAY color map image, as it is illustrated in Fig.5. (a). Image is then stretched in order to get a clearer image and to better interpret the content of the image. MATLAB commands used for those calculations are: *rgb2gray*, *stretchlim* and *imadjust*. Increase in the number of pixels is required since the image is gathered from a low resolution web camera.

Grayscale image is transferred to the binary image by using the Sobel edge detection method, with MATLAB command *edge*. The Sobel method in MATLAB finds edges using the Sobel approximation to the derivative. It returns edges at those points where the gradient of input image is maximum. The *edge* command is used to determine a threshold level. After threshold detection, *edge* command is used in order to find the edges in the grayscale image. Finally, binary image has 1's where the function finds edges and 0's elsewhere. Sobel edge detection resulting image is proposed in Fig. 5.(b).

Binary image contains white lines that correspond to the high contrast in grayscale image. Those white lines don't truly follow the outer edge of an object which should be segmented. When compared to original image, lines that are drawn around the object



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

Fig. 4. Parking lot with six cars parked; (a) gray scaled image; (b) binary image after Sobel edge detection; (c) binary image dilated; (d) binary image after filling isolated regions with white pixels.

contain empty space. Those empty spaces will disappear if the binary image is dilated by using the elements of linear structure *strells*, which can be created with MATLAB *strel* command. Image is dilated by using *imdilate* command. We used four *strells* described in (Eq.1), which resulted with dilated image proposed in Fig. 5.(c).

$$s_0 = [\,1\ \ 1\ \ 1\,],\quad s_{45} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},\qquad s_{90} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix},\quad s_{135} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}. \tag{1}$$

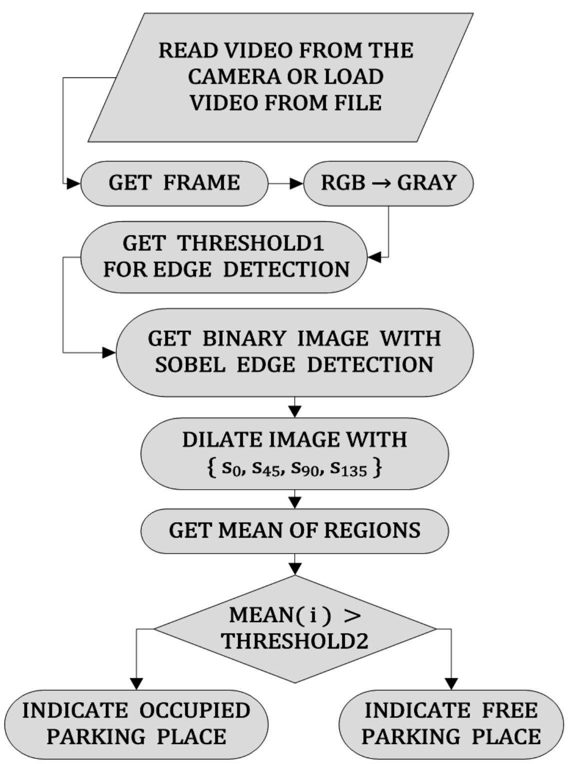


Fig. 5. The algorithm for free parking places detection method.

Dilated image has better outer borders of the object that is segmented, but still there exists some empty spaces inside the object being segmented. Black empty spaces within the white object borders are filled with white pixels by using MATLAB command *imfill*. Resulting image, after filling with white pixels, is proposed in Fig. 5.(d).

Resulting image is then divided into regions. Pixels from each region correspond to parking place positions. Parking place are predefined. For each region mean value is calculated by using MATLAB's command *mean2*. Mean values of each region are compared with the *mean threshold* value, which is in this case 0. If the mean value of the region is larger than the mean threshold value, the parking place is indicated as occupied parking place; otherwise it is indicated as a free parking place.

4. EXPERIMENTAL RESULTS

Method proposed in Section 3 was implemented in MATLAB. *On-line* version was getting frames from the camera, while the *off-line* version was getting frames from the recorded video database. Results are measured with P4 CPU @ 3GHz, 3GB RAM and 32-bit Windows XP.

On-line version of implemented method convinced us that the proposed method works in practice. Our implementation was applied on the real-time system with 11 parking places. Resulting accuracy was 100% for detecting free parking places. Processing time for one frame was on average 350(ms). Processing speed was good enough to meet the system requirements. In the worst case, a car driver requires several seconds to park his car, while processing time was shorter than half of the second. Thus, any change on the parking places can be observed and indicated on time.

Second experiment was done *off-line*, with recorded video. We measured execution times for processing variable number of parking places. Intention of this experiment was to see how the

Tab. 1. Free parking places detection execution time with variable total number of parking places.

Parking places	t_{min} [ms]	t_{avr} [ms]	t_{max} [ms]
1	203	350	578
2	172	348	594
3	172	341	563
4	203	347	578
5	188	354	563
6	203	354	609
7	188	349	609
8	188	347	563
9	203	351	578
10	188	347	563
11	188	363	578
Average	190	350	579

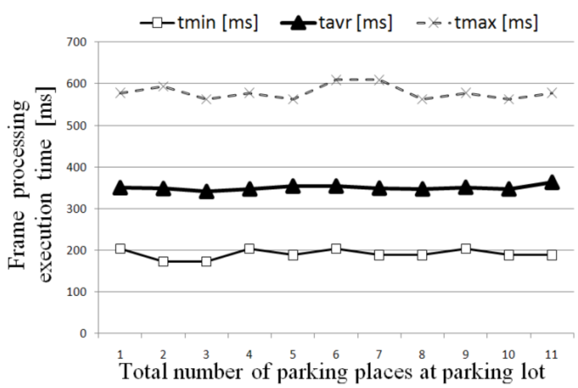


Fig. 6. Execution time for processing video with 414 frames with variable frame steps.

variable number of parking places affects the execution time. Results of the experiment are proposed in table 1, and are shown in Fig.6.

From results shown in Fig. 6., it is obvious that with increasing number of parking places there is no impact on the execution time. Reasons for stochastic fluctuations in minimal t_{min}, average t_{avr} and maximal t_{max} execution times are due to the following reason. Differences between minimum and maximum execution time is caused by using the *imfill* function. It finds isolated black pixel regions that are bounded with white pixels. Area of isolated black regions varies between frames and therefore varies the time for executing the *imfill* function.

5. CONCLUSION AND DISCUSSION

In this paper we presented a modern solution for management and monitoring of parking lot system by camera. We used the 640x480 camera, PC with MATLAB environment, and the parking lot model. The accuracy of proposed *free parking places detection method* was 100%. Maximum execution time 609(ms) and average execution time 350(ms) are below the time needed for drivers to park their car. Our results have shown that the method is applicable for practical use. Major con of proposed system is that camera requires constant light source in the parking lot environment. Camera sensor can replace many sensors that are usually in use and it reduces the maintaining time. Proposed system is a low-cost management and monitoring system since it has small number of parts and is not hard to implement in practice.

REFERENCES

[1] H. Ichihashi, A. Notsu, K. Honda, T. Katada, M. Fujiyoshi, *Vacant parking space detector for outdoor parking lot by using surveillance camera and FCM classifier*, *IEEE International Conference on Fuzzy Systems*, p. 127-134, August 2009.

[2] S.F. Lin, Y.Y. Chen, S.C. Liu, *A Vision-Based Parking Lot Management System*, 2006 IEEE International Conference on Systems, Man, and Cybernetics, vol. 6, p. 2897-2902, October 2006.

[3] R.J.L. Sastre, P.G. Jimenez, F.J. Acevedo, S.M. Bascon, *Computer Algebra Algorithms Applied to Computer Vision in a Parking Management System*, IEEE International Symposium on Industrial Electronics, p. 1675-1680, June 2007.