Vision-Based Parking Lot Occupancy Evaluation System Using 2D Separable Discrete Wavelet Transform

Miron Kłosowski
Department of Microelectronic Systems, Gdańsk University of Technology,
Narutowicza 11/12, 80-233 Gdańsk, Poland,

Marek Wójcikowski*
Department of Microelectronic Systems, Gdańsk University of Technology,
Narutowicza 11/12, 80-233 Gdańsk, Poland, e-mail: wujek@ue.eti.pg.gda.pl, tel. +48 58 347 1974 (corresponding author).

Andrzej Czyżewski
Multimedia Systems Department, Gdańsk University of Technology,
Narutowicza 11/12, 80-233 Gdańsk, Poland

Abstract

A simple system for rough estimation of the occupancy of an ad-hoc organized parking lot is presented. A reasonably simple microprocessor hardware with a low resolution monochrome video camera observing the parking lot from the location high above the parking surface is capable of running the proposed 2-D separable discrete wavelet transform (DWT)-based algorithm, reporting the percentage of the observed parking area occupied by cars. A simple calibration is needed – the mask covering all the areas outside the parking lot must be prepared. The proposed system has been tested on the dedicated FPGA-based hardware in real conditions and proven immune to scene and light changes. As is discussed in the paper, it can be used in low-power wireless sensor networks.

***Keywords*—computer vision, discrete wavelet transforms, sensor systems and applications.**

1. Introduction

There are many car-park occupancy information systems reported in the literature, a good introduction can be found in the paper if Idris *et al.* [1]. Intelligent vision-based surveillance systems

* corresponding author

are becoming more attractive due to the development of image processing algorithms. The advantage of such systems are easy installation and value-added services. Smart Parking System is an example where vision-based system can be used for detecting occupancy of the parking lots or directing drivers to the vacant parking places, which is beneficial for car park operators, drivers and environment conservation. There are also great challenges of vision-based image processing, such as lighting changes, occlusions or shadows, especially for outdoor scenes. The images are usually obtained from a stationary camera observing a parking lot, installed high above the ground. In one of papers [2], object detection with adaptive background subtracting and salient motion detection is used to detect vehicles. The light-weight and efficient background modeling and foreground detection algorithm is described in [3]. Wu *et al.* [4] describe the Principal Component Analysis combined with wavelet transform used for feature extraction and Support Vector Machine for vehicle recognition. Segmentation by thresholding together with Sobel edge detection to eliminate shadows is shown in the paper of Bong *et al.* [5]. Tsai *et al.* [6] propose a color-based model to detect vehicle candidates and a Bayesian classifier to verify the detection of vehicles based on corners, edges, and wavelet features. A Bayesian hierarchical framework integrating the 3-D scene knowledge is presented in [7], utilizing the pixel-based car model and parking space model with the estimate of the lighting condition. Those methods often use complex algorithms for detecting and analyzing the observed scene. Some vision-based systems are, however, implemented on a FPGA-based hardware with limited resources (i.e. processing speed, memory, power supply, dimensions), such as self organizing sensor networks [8][9][10], what has a severe influence on the methods and algorithms. The survey of low-level image processing integrated circuits suitable for sensor networks can be found in [11].

A proper classification of the observed patterns by extracting features that are invariant to certain geometric transformations is a key to the successful image interpretation when the background subtraction algorithm is not used. This usually means that some kind of a Fourier Transform has to be implemented [12]. This paper presents a novel idea of using a wavelet transform-based algorithm to

estimate the overall percentage of the area occupied by vehicles at a car-park. The system is designed to a fast ad-hoc installation on temporary outdoor parking areas and it can be used for daytime parking lot occupancy evaluation. The goal of the system is to calculate roughly what percentage of the uniform parking area is occupied by the cars; the general idea of it is presented in Fig. 1. Our algorithm does not use background subtraction and it exhibits low computation complexity, hence it can work on a unsupervised vision-based system with limited hardware resources. Using the information from the system, an automatic control of parking traffic can be realized, i.e. cars can be directed to a less occupied part of the parking lot by automatic signs or other systems [13]. The 1D-DWT based algorithm used to the traffic jam detection is presented in [14]. An interesting use of discrete wavelet transform for digital image texture analysis is described in [15].

2. Algorithm description

The proposed algorithm consists of the following steps:

Step 1- Image capture

In the first part of the algorithm, a single raw monochrome 256x256 pixels 8-bit image is captured from the camera and gamma normalized using square root compression.

Step 2 - Image transformation and masking

The input image is transformed using 2D separable discrete wavelet transform (DWT) with the Daubechies 4-coefficient wavelet. The input image should contain only the uniform areas where the cars can park, all other regions should be masked. The masking should be done after the DWT application, by writing down the value 128 to the masked pixels in all the scales of the output image from DWT transform, or by simply ignoring masked pixels in the subsequent steps of the algorithm. Masking the original input image would introduce some false detections at mask's borders. An input image, the full resulting DWT, 16 high frequency scales separated and the same scales masked by the ROI are presented in Fig. 2a-b, Fig. 3, 4 and 5 respectively.

Step 3 - Creation of the single image from several different scales

The scales from Step 2 are combined to create the single image, pixels of which are calculated as the linear combination of the absolute values of the pixels of the same coordinates, according to (1):

$$r_{xy} = \sum_{i=1}^{4}\sum_{j=1}^{4} c_{ij}\left| p_{ijxy} - 128 \right| \tag{1}$$

where:

r_{xy} – the intensity of the pixel in the position (x, y) of the resulting image (Fig. 2c);

c_{ij} – the (i, j) coefficient from the matrix **C**;

p_{ijxy} – the intensity of the pixel in the position (x, y) of the vertical scale i and horizontal scale j of the DWT result (resized to the original image size) (Fig. 5).

Step 4 - Thresholding

Conversion of the image acquired in the previous step to the binary image employing the chosen threshold value τ.

Step 5 - Quality improvement operations on the resulting binary image

The number of n_c closings and n_d dilations (morphological operations) are done. Resulting sample image is presented in Fig. 2d. The values in the matrix **C**, as well as the values of τ, n_c and n_d have been set using the procedure described below.

Step 6 - Pixel counting

The detected pixels are counted up and the parking lot percentage status is evaluated. If the perspective correction was not performed at the beginning of the image processing, it can now be done by generating weights for the pixels before counting (pixels representing distant objects are

assigned higher weights during the free place percentage calculation procedure). The acquired information about the percentage of the parking lot occupancy can be transmitted e.g. to parking guidance and information system using a wireless network.

3. Deriving the parameters of the algorithm

The set of *N* learning images from the cameras pointed to the various parking lots have been chosen and the ground truth images have been manually extracted (see Fig. 6 for the examples of the original and ground truth images). The set of several scenes with various light conditions have been selected for learning to obtain the parameter values that can be used in many different situations. Of course, using the pictures from similar scene for learning phase as for the target application would further improve the accuracy of the detection.

The set of 3 scenes with total of N=119 images has been evaluated by the algorithm described in section 2, returning the error ε (2):

$$\varepsilon(\mathbf{C}, n_c, n_d, \tau) = \sum_{i=1}^{N} (S_i(\mathbf{C}, n_c, n_d, \tau) - G_i)^2 \tag{2}$$

where:

n_c – the number of morphological closing operations;

n_d – the number of morphological dilation operations;

τ – the threshold value;

$S_i(\mathbf{C}, n_c, n_d, \tau)$ – the number of the pixels detected by the algorithm using the parameters: $\mathbf{C}$, n_c, n_d and τ for the i-th image;

G_i – the number of the pixels in the ground truth image derived from the i-th image.

Only the pixels inside the active mask are counted.

The learning images have been processed by the genetic algorithm, searching for the solution with the lowest ε criterion. The implementation of the genetic algorithm is based on C++ library *GAlib* using

Goldberg's method, with population size 150, crossover probability 0.6, mutation probability 0.001 and population replacement rate of 50%. The encoding of the parameters in the chromosome has been shown in Table 1.

The genetic algorithm has been started 20 times and run for 200 generations in each run and then the best value has been saved. The genetic algorithm has found the following values of the parameters:

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 7 & 0 & 9 & 0 \\ 3 & 14 & 2 & 4 \\ 0 & 5 & 4 & 5 \end{bmatrix}, n_c = 1, n_d = 0, \tau = 200 \quad (3)$$

4. Hardware description

The proposed solution requires a reasonably simple microprocessor system with monochrome low resolution 256x256 pixels camera. For this purpose, a custom designed FPGA-based sensor network hardware has been used (Fig. 7), with 32-bit processor BA12 from Beyond Semiconductor (same class as Arm's ARM9™) and the peripherals connected to the Wishbone bus. The board has a CMOS camera MT9V111 from Micron with standard resolution 640x480 pixels, limited to 256x256 pixels. The processing time on the BA12 processor with 100MHz clock is approx. 60s, on a desktop PC with Intel's i7 processor was less than 0.01s.

5. Results and discussion

The algorithm's parameters have been derived according to the procedure described in Section 3. The algorithm has been tested using 163 images from test 13 scenes, different from the images used for learning. The examples of the images processed by the algorithm, the respective ground truth images and the detection results have been presented in Fig. 6.

Fig. 8 contains the quantitative evaluation of the selected scenes used for testing. All the results are

available online at: http://www.ue.eti.pg.gda.pl/parking. Some detection errors have been shown in Fig. 9. The three main causes of the false positives have been observed: the painted lines on the parking surface, the people, patches of snow at strong light and the objects very close to the masked regions. False negatives are mainly observed at low light. Nevertheless, for uniform parking surface and reasonable light, the detection results are satisfactory for rough evaluation of parking lot occupancy.

6. Conclusion

The system for parking lot occupancy evaluation has been implemented in a custom FPGA-based hardware and it was tested with the real images from various outdoor parking scenarios. The results of the algorithm's work with real images are very promising – an average error in pixel detection of 6.8% has been measured, at maximum error 25.8% for all 163 test images. The algorithm does not use any background subtraction, so it is not sensitive to overall scene changes, camera's vibrations or sudden light changes, but it requires the preparation of the mask, individual for each scene. Since the background update is not needed, the camera does not need to work continuously to update the background image, resulting in the reduction of the power consumption. Operation of the camera is needed only to capture a single frame from time to time, i.e. every 60 seconds. The experiments show, that the proposed solution is suitable for rough estimation of parking occupancy lot of an ad-hoc organized parking area in most situations.

Acknowledgments

This work was partially supported by the European Commission within FP7 project INDECT (Grant Agreement No. 218086). This research was performed using GAlib, a library of genetic algorithm components (http://lancet.mit.edu/ga/).

References

[1] M. Y. I. Idris, Y. Y. Leng, E. M. Tamil, N. M. Noor and Z. Razak, "Car Park System: A Review of Smart Parking System and its Technology", *Information Technology J.*, 8 (2), 101-113 (2009).

[2] A. Hampapur, L. Brown, J. Connell, A. Ekin, N. Haas, M. Lu, H. Merkl, and S. Pankanti, "Smart video surveillance: exploring the concept of multiscale spatiotemporal tracking", *IEEE Signal Process. Mag.*, 22 (2), 38-51 (2005).

[3] M. Casares, S. Velipasalar, and A. Pinto, "Light-weight salient foreground detection for embedded smart cameras", Computer Vision and Image Understanding, 114 (11), 1223-1237 (2010).

[4] J. Wu, X. Zhang, and J. Zhou, "Vehicle detection in static road images with PCA and wavelet-based classifier", *Proc. IEEE Intelligent Transportation Systems Conf.*, Oakland, CA, Aug. 25-29, 740-744 (2001).

[5] D. B. L. Bong, K. C. Ting, and K. C. Lai, "Integrated Approach in the Design of Car Park", *IAENG Int. J. Comput. Sci.*, 35 (1), 7-14 (2008).

[6] L. Tsai, J. Hsieh, and K. Fan, "Vehicle Detection Using Normalized Color and Edge Map", *IEEE Trans. Image Process.*, 16 (3), 850-864 (2007).

[7] C. Huang and S. Wang, "A Hierarchical Bayesian Generation Framework for Vacant Parking Space Detection", *IEEE Trans. Circuits Syst. Video Technol.*, 20 (12), 1770-1785 (2010).

[8] A. Gardel Vicente, I. Bravo Munoz, P. J. Molina, and J. L. L. Galilea, "Embedded Vision Modules for Tracking and Counting People," *IEEE Trans. Instrum. Meas*, 58 (9), 3004-3011 (2009).

[9] Szczepański S., Wójcikowski M., Pankiewicz B., Kłosowski M. Żaglewski R., "FPGA and ASIC implementation of the algorithm for traffic monitoring in urban areas", *Bulletin of the Polish Academy of Sciences: Technical Sciences*, 59 (2), 137–140 (2011).

[10] M. Wójcikowski, R. Żaglewski, B. Pankiewicz, M. Kłosowski, and S. Szczepański, "Hardware-Software Implementation of a Sensor Network for City Traffic Monitoring Using the FPGA- and ASIC-Based Sensor Nodes", *Journal of Signal Processing Systems*, 71 (1), 57-73 (2013).

[11] Jendernalik W, Blakiewicz G, Handkiewicz A., and Melosik M., "Analogue CMOS ASICS in Image Processing Systems", Metrol. Meas. Syst., XX (4), 613–622 (2013).

[12] Hui S., Żak S.H., "Discrete Fourier transform based pattern classifiers", *Bulletin of the Polish Academy of Sciences: Technical Sciences*, 62 (1), 15–22 (2014).

[13] Y. Geng and C. G. Cassandras, "A New 'Smart Parking' System Based on Resource", *IEEE Tran. Intell. Transp. Syst*., 14 (3), 1129-1139 (2013).

[14] M. Klosowski, "Hardware accelerated implementation of wavelet transform for machine vision in road traffic monitoring system", 1st International Conference on Information Technology, IT 2008. DOI 10.1109/INFTECH.2008.4621686, 1-4 (2008).

[15] M. Kociołek, A. Materka, M. Strzelecki, and P. Szczypiński, "Discrete Wavelet Transform - Derived Features for Digital Image Texture Analysis", Proc. Int. Conf. Signals and Electronic Systems, 18-21 September 2001, Lodz, Poland, 111-116 (2001).

Figure captions

Fig. 1. The block diagram of the proposed system.

Fig. 2. Images illustrating selected steps of the algorithm, *(a)* - input image of the parking lot, (*b)* - region of interest of the input image, *(c)* - Combined 2D DWT scale images (result of the step 3), *(d)* - Previous image after applying the threshold and morphological operations (result of the step 4 and 5).

Fig. 3. Separable 2D DWT for image from Fig. 2a (scales surrounded by the rectangles are used in the following steps of the algorithm, see Fig. 4).

Fig. 4. Part of the separable 2D DWT for image from Fig. 2a, separated and scaled to the original image's size (16 high frequency scales).

Fig. 5. Part of the separable 2D DWT for image from Fig. 2a, separated, scaled to the original image size and masked (16 high frequency scales – result of the step 2 of the algorithm described in the text).

Fig. 6. Examples of detection results. Parking lot occupancy is calculated as the number of pixels classified as belonging to a car divided by the number of all non-masked pixels.

Fig. 7. The hardware used for the proposed algorithm verification in real conditions.

Fig. 8. Quantitative test results of the selected scenes. All results are available online at: http://www.ue.eti.pg.gda.pl/parking.

Fig. 9. The examples of detection errors caused by strong lines painted on parking surface.

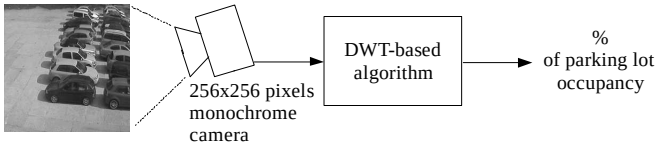
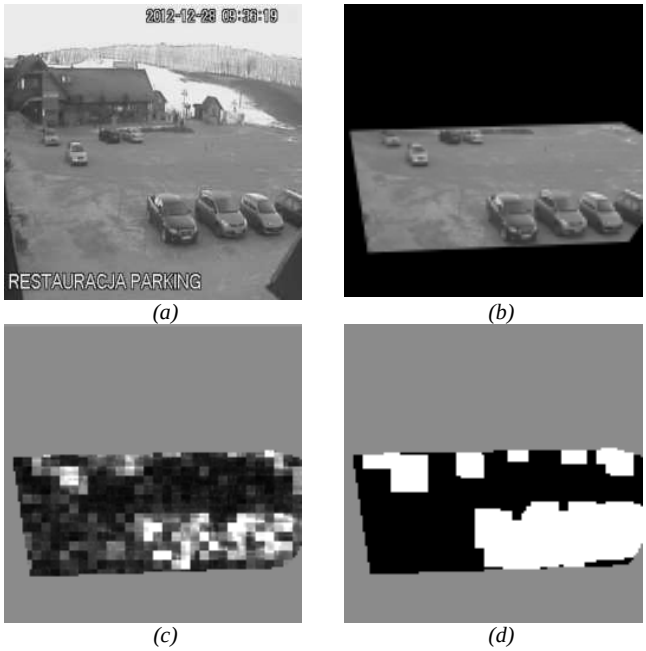


Fig. 1.



(a) *(b)*

(c) *(d)*

Fig. 2.

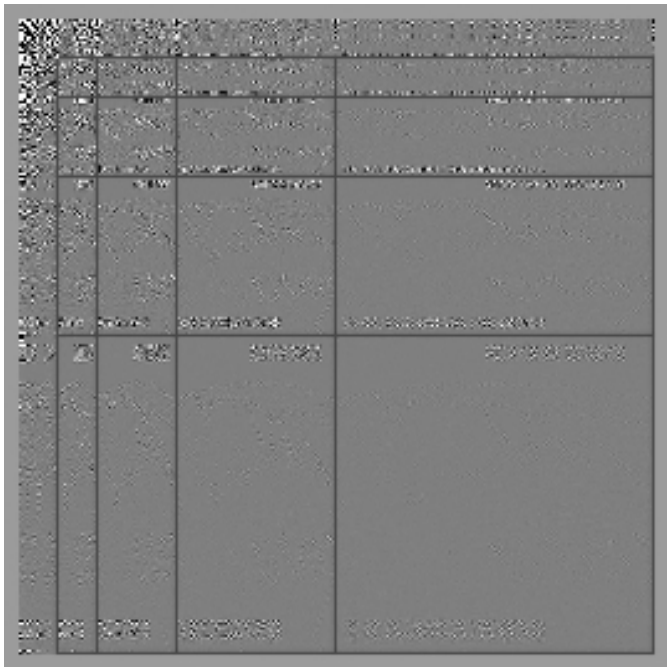


Fig. 3.

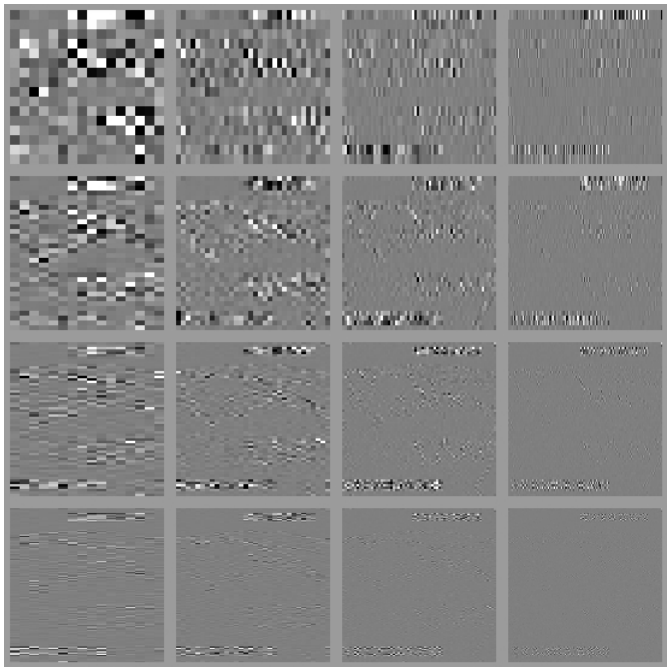


Fig. 4.

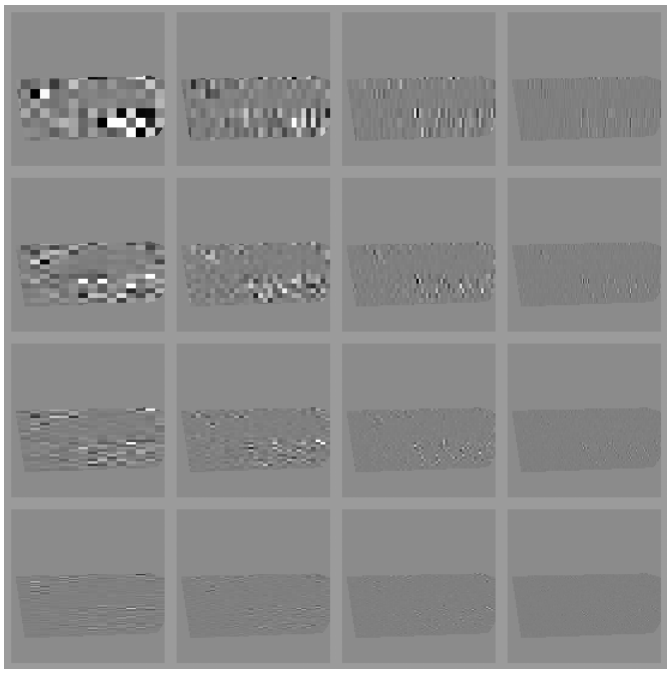


Fig. 5.

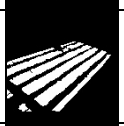
Input image	Detection result	Ground truth	Parking lot occupancy (occupied area)
			Detection result=36.3% Ground truth=36.0% |Error|=0.3%
			Detection result=23.5% Ground truth=22.9% |Error|=0.7%
			Detection result=31.9% Ground truth=27.7% |Error|=4.3%
			Detection result=54.1% Ground truth=60.3% |Error|=6.2%
			Detection result=32.6% Ground truth=35.8% |Error|=3.2%
			Detection result=13.7% Ground truth=16.5% |Error|=2.8%
			Detection result=68.6% Ground truth=58.7% |Error|=9.9%
			Detection result=12.1% Ground truth=8.3% |Error|=3.8%
			Detection result=48.2% Ground truth=41.7% |Error|=6.4%
			Detection result=58.1% Ground truth=48.2% |Error|=9.9%
			Detection result=36.5% Ground truth=20.5% |Error|=16.0%
			Detection result=36.3% Ground truth=41.8% |Error|=5.5%
			Detection result=34.9% Ground truth=38.6% |Error|=3.7%

Fig. 6.

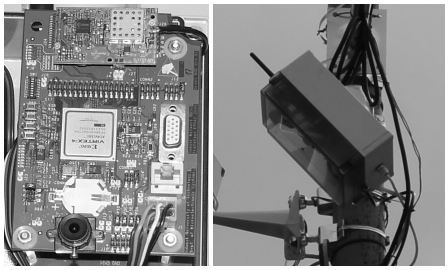


Fig. 7.

Scene name	Example image	Number of images in the scene	Quantitative evaluation of the scene
PG_parking		14	Min. error=0.2% Max. error=5.1% Average error=1.6%
UNM_Arena		12	Min. error=1.3% Max. error=12.4% Average error=6.5%
yt03		28	Min. error=0.0% Max. error=12.7% Average error=4.1%
yt07		30	Min. error=0.3% Max. error=25.8% Average error=7.7%
yt08		18	Min. error=6.7% Max. error=17.2% Average error=12.8%
yt12		14	Min. error=1.7% Max. error=9.4% Average error=5.4%

Fig. 8.

(a) (b)

Fig. 9.

Table 1. Parameter encoding used in the genetic algorithm. All values are represented as unsigned integers.

Parameter	$\mathbf{C}_{00}$	$\mathbf{C}_{01}$	$\mathbf{C}_{02}$	$\mathbf{C}_{03}$	$\mathbf{C}_{10}$	$\mathbf{C}_{11}$	$\mathbf{C}_{12}$	$\mathbf{C}_{13}$	$\mathbf{C}_{20}$	$\mathbf{C}_{21}$	$\mathbf{C}_{22}$	$\mathbf{C}_{23}$	$\mathbf{C}_{30}$	$\mathbf{C}_{31}$	$\mathbf{C}_{32}$	$\mathbf{C}_{33}$	n_c	n_d	τ
Min value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Max value	15	15	15	7	15	15	15	7	15	15	15	7	7	7	7	7	7	7	255
No. of bits	4	4	4	3	4	4	4	3	4	4	4	3	3	3	3	3	3	3	8