Vacant Parking Space Detector for Outdoor Parking Lot by Using Surveillance Camera and FCM Classifier

H. Ichihashi, *Member, IEEE*, A. Notsu, K. Honda, *Member, IEEE*, T. Katada, M. Fujiyoshi

***Abstract*— The most prevailing approach now for parking lot vacancy detecting system is to use sensor-based techniques. The main impediments to the camera-based system in applying to parking lots on rooftop and outside building are the glaring sun light and dark shadows in the daytime, and low-light intensity and back-lighting in the nighttime. To date, no camera-based detecting systems for outdoor parking lots have been in practical use. A few engineering firms provide the camera-based system, which is only for underground and indoor parking lots.**

This paper reports on the new camera based system called ParkLotD for detecting vacancy/occupancy in parking lots. ParkLotD uses a classifier based on fuzzy c-means (FCM) clustering and hyper-parameter tuning by particle swarm optimization (PSO). The test result of the detection error rate for the indoor multi-story parking lot has improved by an order of magnitude compared to the current system based on the edge detection approach. ParkLotD demonstrates high detection performance and enables the camera-based system to achieve the practical use in outdoor parking lots.

I. Introduction

In parking areas such as those at shopping malls or train stations, drivers have difficulty to find vacant and convenient car parking rooms or bays. It is especially difficult when the parking lot is crowded or almost full.

There are four categories of parking lot guidance systems. The most prevailing approach now is to use sensor-based techniques. Counter-based systems use sensors to count the number of vehicles enter and exit a parking lot. Gate-arm counters and induction loop detectors located at the entrances and exits are used [1]. This system can give information on the total number of vacant spaces or rooms in a closed area. “Full/Spaces” sign on LED display panels at the entrance of the parking lot is the main output of the system and does not help much in guiding the driver to the exact location of the vacant spaces.

Wired sensor-based system uses detection sensors such as ultrasonic and infrared light sensors which are installed at all parking spaces or rooms in either outside or inside the buildings. The parking space occupancy information is then transmitted to display panels at the entrance of the parking lot. The display panels provide information, direction and guide the drivers to vacant parking spaces.

Sipark SSD multi-story car park guidance system by German company Siemens (http://www.mobility.siemens.com), Vipark parking monitoring and guidance system by Polish company Elitel (http://www.vipark.pl/parking_systems.php), and Identipark System by South African Nortech International Inc. (http://www.identipark.com/) are the examples of car park guidance systems with individual parking space detection. An ultrasonic sensor equipped with LED indicator is mounted above each parking space and detects whether the space is occupied or vacant. The system is designed to facilitate operation of underground and multi-story parking lot. LED indicator directs the driver to a vacant parking bay. LED display shows numbers of free spaces in the designated area. The operator is able to see at all times how many and which parking spaces are filled or available. Vehicle Sense, Inc. (http://www.vehiclesense.com/website/index.html) use wireless magnetic sensors to detect vehicles and relay the data in real-time to the appropriate office. SPIN (Street Parking Information Network) is for on-street parking (e.g. meters and pay-stations), and SmartLot is for off-street parking (e.g. parking garages).

These sensor-based systems, without regard to wired or wireless, are not well suited for vigilance or surveillance. Security cameras have been installed in many parking lots independently of the guidance system. So, if camera-based system with image processing techniques is deployed, both of the functions, i.e., vigilance and guidance, are realized simultaneously. The image-based video sensor techniques may become the most prominent approach with the spread of surveillance cameras. Images from the surveillance camera can provide status of every parking space at curbside, in lots and in garages.

Japanese firm Nichizo Tech, Inc. offers a camera-based system, in which three to six parking spaces in a bay of a parking lot are covered by a single camera. Detection algorithm is based on the edge detection technique.

Application of the camera-based system is usually limited to parking lots in the buildings due to the poor accuracy of the detection by camera-based system. The accuracy differs depending on weather and lighting conditions, and buildings and fences in the background. Under glaring sun light, vehicles, fences and buildings cast dark shadows on other vehicles and ground. Low-light intensity and back-lighting at night can also be impediments. The cameras are usually equipped with an auto iris system, which partially changes the brightness according to the different sunlights and back-lightings. Figs. 1-2 show example images in an indoor multi-story parking lot, where the lighting conditions of the background affect on the detection of edges.

S. Funck, *et al*. [2] proposed an image-based system with single image of a single camera. The image processing for

H. Ichihashi, A. Notsu, K. Honda are with the Department of Computer Science and Intelligent Systems, Osaka Prefecture University, 1-1 Gakuen-cho, Naka, Sakai, Osaka, 599-8531, Japan (phone: 72-254-9352; email: ichi@cs.osakafu-u.ac.jp), T. Katada is with Nichizo Tech, Inc. and M. Fujiyoshi is with Hitachi Zosen Corporation.

978-1-4244-3597-5/09/$25.00 ©2009 IEEE

Fig. 1. Example images of the parking lot taken by Camera P10

Fig. 2. Example images of the parking lot taken by Camera P43

the vehicle classification works by constructing a reference image of the empty parking space given in the input image and then comparing those two. The occupancy estimate is determined by the vehicle to parking space pixel area ratio. N. True [3] proposed a camera-based system, which uses a combination of car feature point detection and color histogram classification. S. Liu [4] proposed an algorithm using the segmentation of parking space region of images and temporal differencing. Car Park Occupancy Information System (COINS) by D.B.L. Bong, K.C. Ting and K.C. Lai [5] is the camera-based system applied to outdoor parking lot. The system was tested using simulation model and also in real-case scenarios. The camera is positioned in facing the front or rear view of parking lots, acquiring a fixed scene all the time. The height of the camera must be enough to obtain a clear, unobstructed top view of the parking lots. The gray scale is used as the threshold to differentiate the presence of vehicle from vacancy. If the number of pixels whose gray scale value is larger than the threshold exceeds 10% , then the system will recognize that the space is occupied by a vehicle. In the COIN system, the Sobel edge detection is also used. If the percentage of total white pixels, i.e., the edge, exceeds 5% of total pixels, the system will recognize the parking lot as occupied, and vice versa for otherwise. The final decision is based on the exclusive OR of the two classification results with vacancy (1) and occupancy (0).

To date, due the the accuracy of the detectors, no camera-based systems for outdoor parking lot have been in practical use. This paper reports on the new car parking lot occupancy information system by using camera-based system. Digital images were extracted out from the video clip for image processing. A classifier based on fuzzy c-means clustering (FCMC) is employed and called as Park Lot Detector (ParkLotD). ParkLotD covers initial setup, i.e., 1) collecting training samples and adding class information to the sample data, 2) training the classifier, and 3) testing unknown new images transmitted from the camera.

The camera-based system is cost effective than sensor based system if single camera covers many parking spaces as in the parking lot on rooftop and outside building. The camera is in a fixed position and facing a fixed direction all the time. But, the camera is not necessarily facing front or rear view of the vehicles, thus a parking space on the image may not be rectangular.

We tested under uncontrolled luminance conditions with images captured during different hours such as sunny, cloudy, rainy, morning, afternoon, evening and night. In the nighttime or near-to-darkness, lightings are not expected to be fully available. The absence of enough lighting makes detection of vehicle almost impossible, but these circumstances are not avoided in the tests of ParkLotD.

The fuzzy c-means based classifier (FCMC) in [6], [7], [8] is employed in the ParkLotD system. FCMC consists of two phases. The first phase is an unsupervised clustering. The clustering is implemented by using the data from one class at a time. Semi-hard clustering approach is introduced to improve the convergence of the clustering algorithm. At the completion of the clustering, Mahalanobis distances between validation data and cluster centers are computed and fixed for parameter search. The best setting of the free parameters (i.e., hyperparameters) are chosen by the particle swarm optimization (PSO) [9], [10]. The distances are not computed for every evaluation of the fitness value of PSO, and this approach reduces the computational time.

The paper is organized as follows. Section II briefly reviews the FCMC, i.e., the semi-hard clustering algorithm and the parameter search by PSO. In Section III, the procedures of ParkLotD is detailed. Section IV reports on the performance of ParkLotD and Section V concludes the paper.

II. Semi-hard Clustering and FCM Classifier

In this section we briefly review the semi-hard clustering and FCM classifier. See [11] for more details. The first phase of FCMC is clustering or partitioning, which is modified from [12]. Let

$$D(x_k, v_i; S_i) = (x_k - v_i)^\top S_i^{-1}(x_k - v_i) \tag{1}$$

be squared Mahalanobis distance from $x_k \in \mathcal{R}^p$ to $v_i \in \mathcal{R}^p$, and let S_i be a covariance matrix of data samples of the i-th cluster as:

$$S_i = \frac{\sum_{k=1}^N u_{ki}(x_k - v_i)(x_k - v_i)^\top}{\sum_{k=1}^N u_{ki}}. \tag{2}$$

$$v_i = \frac{\sum_{k=1}^N u_{ki} x_k}{\sum_{k=1}^N u_{ki}} \tag{3}$$

is a cluster center and the mixing proportion of i-th cluster is

$$\alpha_i = \frac{\sum_{k=1}^N u_{ki}}{\sum_{j=1}^c \sum_{k=1}^N u_{kj}} = \frac{1}{N}\sum_{k=1}^N u_{ki}. \tag{4}$$

c denotes the number of clusters and N denotes the number of samples.

For updating u_{ki}, following rule is adopted [13], [6].

$$u_{ki} = \begin{cases} \beta & (i = \arg\min_{1\le j\le c} D(x_k, v_j; S_j) \\ & \quad +\log|S_j|), \\ \frac{1-\beta}{c-1} & (\text{otherwise}). \end{cases} \tag{5}$$

When $\beta = 1$, the algorithm produces crisp partition. When $\beta = 1/c$, all clusters overlap each other, i.e., there is no partitioning and all cluster centers overlap with the class mean. By increasing β from $1/c$ to 1, the partitioning gradually changes from a single cluster to crisp c clusters.

The clustering is done on a per class bases. The procedure is the repetition of update of S_i, v_i, α_i and u_{ki}. After partitioning data for all classes, the classification (i.e., the second phase) is performed by computing fuzzy memberships. Let π_q denote the mixing proportion of class q, i.e., the $a\ priori$ probability of class q. Let α_{qj} be α_i in (4) for cluster j of class q. The class membership of k-th data x_k to class q is computed as:

$$u^*_{qjk} = \frac{\alpha_{qj}|S_{qj}|^{-\frac{1}{\gamma}}}{(D(x_k, v_{qj}; S_{qj})/0.1 + \nu)^{\frac{1}{m}}}, \tag{6}$$

$$\tilde{u}_{qk} = \frac{\pi_q \sum_{j=1}^{c} u^*_{qjk}}{\sum_{s=1}^{Q} \pi_s \sum_{j=1}^{c} u^*_{sjk}}, \tag{7}$$

where c denotes the number of clusters of each class and Q denotes the number of classes. We selected the functional form of u^* based on the membership function derived from the generalized FCM objective function [6]. u^* is a modified and parameterized multi-variational version of Cauchy's weight function in M-estimation or of the probability density function (PDF) of Cauchy distribution.

One of the impediments for the classifier is the singularity of covariance matrices, which frequently occurs when feature dimension is relatively high and the number of samples in each cluster is small. So, the modification of covariance matrices in the mixture of probabilistic principal component analysis (MPCA) [14] and the character recognition [15] is applied in FCMC.

Initial locations of cluster centers or membership values are usually given randomly in the FCM clustering approach. Since the classification performance is sensitive to the initialization, our approach does not use random initialization. We set the number of clusters to 2 for each class. Let $f_k, k = 1, ..., n$ be PCA scores of the data set $D = (x_1, ..., x_n)^\top$ of a class. f_k are associated with the largest singular value of D.

Initial membership values of the two clusters of a class are given by

$$u_{k1} = \begin{cases} 1-\beta & (f_k \le 0) \\ \beta & (\text{otherwise}) \end{cases}$$

$$u_{k2} = \begin{cases} \beta & (f_k \le 0) \\ 1-\beta & (\text{otherwise}). \end{cases} \tag{8}$$

At the completion of clustering for all classes in the first phase of FCMC, we then compute Mahalanobis distance by (1) for all the samples in the validation sets. The second phase of FCMC is the supervised classification and the free parameters are selected to minimize error rate on the validation sets of 10-CV procedure. The distances by (1) for all the samples in validation sets are fixed and the best setting of the free parameters is searched by PSO. The Mahalanobis distances are not recomputed in the optimization procedure, since cluster center vectors are not changed. In [7], the change rates of center vectors are the free parameters, but the mixing proportions α_i are the parameters in this paper. In PSO, the search for the better positions of particles follows the rule as:

$$\begin{aligned} Para^{t+1} &= Para^t + Velo^{t+1}, \\ Velo^{t+1} &= w_0\ Velo^t + c_1\ Rand_1\ (pbest - Para^t) \\ &+ c_2\ Rand_2\ (gbest - Para^t), \end{aligned} \tag{9}$$

where $Para$ is the parameter vector (α_i, m, γ, ν of FCMC) denoting a position of a particle. $Velo$ is the vector of its velocity. It should be noted that $Rand$ is a diagonal matrix of random numbers chosen from the unit interval [0, 1]. w_0, c_1 and c_2 are scalar constants. $pbest$ and $gbest$ are the vectors of positions of pbest and gbest respectively. Though the update rules are written in vector-matrix form, they are the standard PSO rules. The best setting of the parameters is picked via 10-CV, which minimizes the error rate on the validation sets. We apply 10-CV for the parameter search by PSO. But, the error rate estimate of the final model on validation data in 10-CV will be biased. It is smaller than the true error rate, since the validation set is used to select the parameter values. To confirm the discriminative power of FCMC in terms of generalization ability, we adopt the 3-way data splits (3-WDS) approach. At every completion of the clustering for all folds of training sets in 10-CV procedure, the test set performance of the classifier are estimated using the following steps:

1. Choose optimal parameters by looking at the best performance on all validation sets of 10-CV.
2. Construct the classifier using the total training/validation set for the optimal choice of the tuned parameters.
3. Assess the test set accuracy by means of the independent test set.

Support Vector Machines (SVMs) are the prominent approaches for solving classification problems. The performance of SVMs on the public domain benchmark datasets [16] was reported in [17]. The comparison was performed on an out-of-sample test set consisting of 1/3 of the data. The first 2/3 of the randomized data was reserved for training and/or cross-validation. The average test set performance and sample standard deviation were reported in [17].

In Table II, test set error rates of 100 runs of 3-WDS by FCMC with the semi-hard clustering algorithm, and 10 runs of 3-WDS by SVM-RBF are compared. Randomized test sets and 10-CV procedure were used. The regularization and

TABLE I
BENCHMARK DATA OF UCI [16]

	features (p)	objects (N)	classes (Q)
Iris	4	150	3
Breast	9	683	2
Iono	33	351	2
Liver	6	345	2
Pima	8	768	2
Sonar	60	208	2
Wine	13	178	3
Australia	14	690	2
German	24	1000	2
Heart	13	270	2

TABLE II
TEST SET PERFORMANCE OF FCMC, k-NN, AND SVM

	Test set error rate (%) trained by 10-CV				
	FCMC(α_i)			k-NN	SVM-RBF
	100 runs	t-test (k)	t-test (S)	100 runs	10 runs
Iris	3.04 ± 2.47	◯	-	5.68 ± 2.91	3.4 ± 3.4
Breast	2.99 ± 1.04	-	-	3.37 ± 0.10	3.6 ± 1.0
Iono	4.54 ± 1.54	⊙	-	14.86 ± 3.59	4.6 ± 1.7
Liver	31.50 ± 3.78	⊙	-	37.94 ± 4.37	29.6 ± 3.2
Pima	24.75 ± 2.11	-	♠	25.77 ± 2.60	22.7 ± 2.2
Sonar	16.39 ± 5.10	-	◯	15.80 ± 4.17	25.0 ± 6.6
Wine	1.86 ± 1.87	⊙	-	4.36 ± 2.89	2.2 ± 2.1
Australia	14.57 ± 2.54	-	-	14.39 ± 2.00	13.7 ± 1.8
German	24.59 ± 1.85	◯	-	27.83 ± 2.12	24.1 ± 1.4
Heart	16.90 ± 3.00	-	-	17.31 ± 3.57	15.3 ± 4.8

kernel parameters of the SVM classifier with RBF kernel were selected using the grid search method. The test sets and validation sets are randomized and not stratified. The results in "SVM-RBF" column are quoted from [17]. The optimal parameters and the classifier constructed from step 1 and 2 were used to assess the test set performance of the SVM classifier on 10 randomizations: for each randomization the first 2/3 of the data were used for training, while the remaining 1/3 was put aside for testing. In our test of FCMC and k-NN, we run the steps from step 1 to 3 for every randomized test set of 1/3. Since the optimum parameters may change depending on the training/validation sets, we strictly followed the procedure and repeated 100 times for performance evaluation. Table III shows the validation set performance of FCMC and SVM. FCMC outperforms SVM in terms of the validation errors. We set the number of clusters in FCMC to 2 for each class. We can use many more clusters, but they cause the over-fitting problem, since not only the cluster centers but also the covariance matrices have so many variables to be determined. As shown in Table III, most of the best β values lie between 0.6 and 0.9. This means that the proper number of clusters is between 1 and 2, since the resulting cluster is equivalent to a single cluster when $\beta = 0.5$ and when $\beta = 1.0$ we have crisp two clusters in the semi-hard clustering approach with c=2. We compared the performance of FCMC to k-NN and SVM with respect to a one-tailed two sample t-test ($p \leq 0.05$) in Table II. We just formally applied the two sample t-test. The results of these significance tests are given in "t-test (k)" and "t-test (S)" columns of Table II. ⊙ denotes that the error rates by

TABLE III
VALIDATION SET PERFORMANCE OF FCMC AND SVM

	Validation set error rate trained by 10-CV				
	r	β	FCMC (%)	t-test(S)	SVM (%)
Breast	0	0.6	2.50 ± 0	-	3
Iono	3	0.9	3.55 ± 0.08	-	4
Liver	5	0.9	25.75 ± 0.14	◯	28
Pima	3	0.8	22.56 ± 0.08	-	22
Sonar	8	1.0	10.31 ± 0.13	◯	23
Australia	2	1.0	12.70 ± 0.05	◯	14
German	0	0.9	22.92 ± 0.05	◯	24
Heart	0	0.6	14.15 ± 0.08	◯	17

FCMC is smaller than the other by more than 5% or less than half. FCMC outperforms k-NN, and is better on one data and worse on one data compared to SVM.

The overall results illustrate that the FCMC consistently yields good test set and validation set accuracies.

III. PARKING LOT VACANCY DETECTOR

In this section, the procedure of ParkLotD is detailed using the prototype system experimentally applied to a real outdoor parking lot. The images of the parking lot are automatically extracted from the surveillance camera installed in the parking lot. The real time images from the camera are transmitted to Nichizo Tech Inc. via the Internet and the selected pictures are stored in JPEG format of 704 × 576.

Step0 is the computer program for specifying the following information. Number of parking spaces covered by each camera. Size of the vehicle/space images, which are cut out and extracted from the pictures and thinned. Dimensionality of image data compressed by Principal Component Analysis (PCA). The dimensionality equals the number of PC basis vectors.

One of the pictures specified is displayed by Step0 as in Fig. 3. Each parking space is specified by clicking three points (i.e., the three corners of a parallelogram) as writing the letter 'L' (top left and top center in Fig. 3). The numbers shown in Fig. 3 illustrate the order of clicking, but actually they do not appear on the screen. The image of a vehicle becomes right side left as in Fig. 3 top right, if one clicks from the number 10 to the number 12 in Fig. 3 bottom left. The area from the top of the windshield of a vehicle down to the bottom of the bumper including head lights and front grille is the recommended area for cutting out a space of the lot on the image. This procedure is repeated up to the number of parking spaces. Many pixels of the image are eliminated (thinned) and the images are transformed into squares of the same size of 32 × 32.

Step1 is the program for computing PCA basis vectors from the selected images and saving it for Step2.

Step2 is the graphic user interface as shown in Fig. 5 for collecting training data and training the classifier. One can modify the location of a parking space by clicking [Modify CUT] button. The picture as shown in Fig. 4 appears and the area specified for modification is depicted by white parallelogram. Then one can click three corners of a new

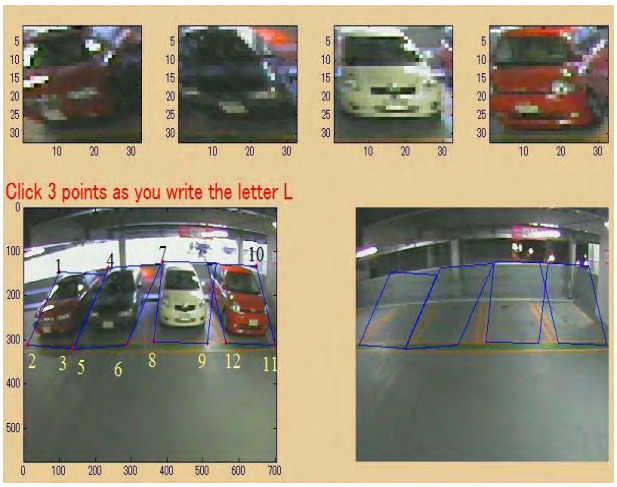

Fig. 3. Illustration of procedure for cutting out parking space images

rectangle in the same manner as in Step0. The old locations are replaced by the new ones.

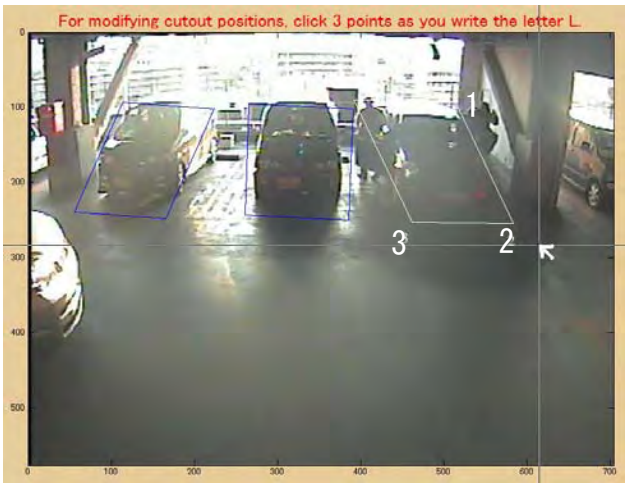

Fig. 4. Modifying cut positions

For collecting the training data, one can click [SKIP] button on the screen shown in Fig. 5, then the detection results on the first picture are depicted by red or white square at each center of the vehicle/space images. The red square denotes occupancy and the white square denotes vacancy. If a classification decision depicted by a square is incorrect, by clicking [VACAN/OCCU] button, followed by a click near the square mark, the color of the square is changed (i.e., the class information is changed for the training data). Then by clicking [ADD D] button, the training data is added to the training set. The RGB image is thinned and transformed into HSV and then into gray scale image. The gray scale image data is transformed into a vector form of small dimension (e.g., 50) by multiplying PC basis vectors. If the button is falsely clicked, one can click [DELETE] button to delete the data from the training set. For deleting multiple data, [DELETE] button is used multiple times. [SKIP] button is only for reviewing the classification decision on the next picture, and no training data are added to the training set. [Prev] is for testing previous pictures. By clicking [SKIP 50] and [SKIP-50] one skips 50 pictures forward and backward respectively.

After collecting about 500 training data, the training of classifier is carried out by clicking [LEARN] button on the screen shown in Fig. 5. Then, the classifier can detect vacancy or occupancy of the parking spaces almost correctly. So, one can review the classification by clicking [SKIP] button or [AUTO] button. By [AUTO] button pictures are shown one after another. When finding incorrect classification, one can stop by clicking [AUTO] button again and goes back by clicking [PREV] button to the picture for changing the class information and adding a datum to the training set. The procedures of collecting training data and [LEARN] are repeated until almost all parking spaces are correctly classified.

Examples of the pictures obtained from a rooftop parking lot are shown in Figs. 5-10. The images of vehicles and parking spaces recovered from data compressed by PCA are given in Fig. 11. The recovered images give us an intuitive illustration of the data used for the detection. As the figures typically show, the extracted images of parking spaces and vehicles have different brightness. Various kinds of edges are detected from both occupied and vacant areas. It is apparent that the gray scale threshold approach and edge detection scheme for classification of vacancy and occupancy do not function under these ill conditioned circumstances.

IV. Classification Performance of ParkLotD

The captured images of a parking lot were transmitted from the site to the office of Nichizo Tech, Inc. via the Internet. 2000 images (pictures) of the lot were randomly selected

Fig. 5. Rooftop parking lot in the sunny morning. Red and white squares indicate classification results, i.e., occupied and vacant respectively. Each vehicle casts a long shade on the ground.

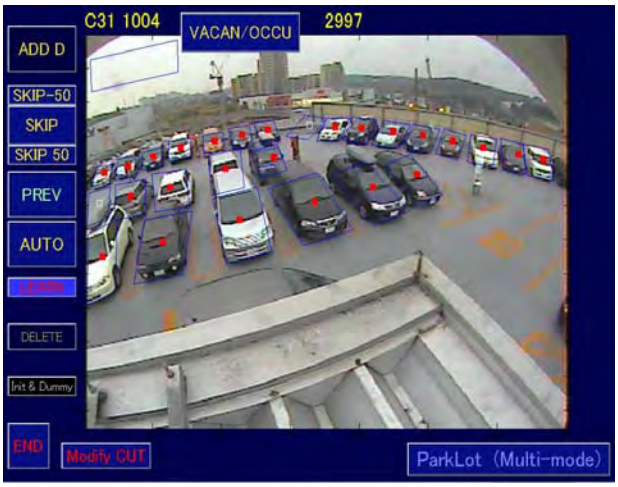

Fig. 6. The lot in the cloudy day.

Fig. 8. The lot in the midnight. Vehicles in the background are barely visible.

Fig. 7. The lot in the evening. The lighting in the background intrudes in the lot.

Fig. 9. The lot in the heavy rain. Raindrops obstruct the view from the camera.

from the collected images in a period of two weeks and used for the performance test. Training data were selected from the images of the first week by following the procedure stated in Section III. The 50 PC basis vectors were computed using the first 300 pictures. First, every 50 pictures were selected from the first half 1,000 pictures, which were taken in the first week. The training samples are the image data of the parking spaces. 27 parking spaces were cut out from each picture as in Figs. 5-10. The RGB image was thinned and transformed into HSV and then into gray scale image. The collected 540 images (20 pictures) were compressed by multiplying PC basis vectors (i.e., inner product) and the obtained 540 feature vectors of 50 dimension were used for the training of the FCM classifier. Then all 27,000 images of vehicles and spaces from the first 1,000 pictures were tested and reviewed by the GUI of ParkLotD. If a miss-classified space was found then the class information was corrected on the screen by clicking near the red or white square mark. If the same space was miss-classified in successive pictures and no apparent difference in the space or weather condition appeared, the pictures were ignored and not added to the training set. The procedure was repeated up to the 1,000th picture. At the completion of this procedure, the FCM classifier was trained again. Then the classifier performance was tested by using the test set of 27,000 data with correct class information. The test set was made up of all the latter half 1,000 pictures taken in the second week.

We tested the performance of FCM classifier with several options.

(1) The first test was done without PCA data compression. Each vehicle or vacant space is extracted and thinned as a 16×16 square image, i.e., a feature vector of 256 dimension. The number of training data was 3,915 (i.e., 145 pictures). The data were selected by the GUI of ParkLotD. The result

Fig. 10. The lot partly in the shadow of a billboard.

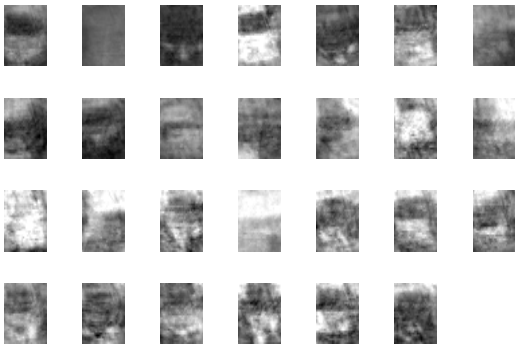

Fig. 11. Recovered images of the lot partly in the shadow of a billboard in Fig. 10.

is shown in Table IV.

(2) The second test was done with PCA data set compression. Each vehicle or vacant space is extracted and thinned as a 32×32 square image, i.e., a vector of 1,024 dimension. The image was transformed into a feature vector of 50 dimension by PC basis vectors. The number of training data was 3,537 (i.e., 131 pictures). The result is shown in Table V. With the PCA data compression, the memory requirement is smaller than (1), though the improvement of error rate is slight. This is important since a single computer controles many cameras e.g., 20 and 50.

(3) The third test was also done with PCA data set compression as in (2). A variable representing day or night decided by the brightness of a part of sky image shown in Fig. 5 upper left, and a variable of parking space number are added to the vector. The number of training data was 3,672 (i.e., 136 pictures). The result is shown in Table VI.

(4) The fourth test was also done with PCA data set compression as in (2). A part of sky image shown in Fig. 5 upper left was concatenated with each image of vehicle or vacant space and thinned as a 32×63 rectangular image, i.e., a vector of 2,048 dimension, which was transformed into a feature vector of 50 dimension by PC basis vectors. The number of training data was 3,807 (i.e., 141 pictures). The data were selected by the GUI of ParkLotD. The result is shown in Table VII.

(5) The fifth test was also done with PCA data set compression as in (2). By the brightness of a part of sky image shown in Fig. 5 upper left, day and night were distinguished. The threshold value was 210. Parking spaces are divided into two groups, i.e., one is for the foreground (#1, ..., #6) and the second row (#7, ..., #11), and the other is for the background (#12, ... #27). Mode-1 is for the 1st group in the nighttime and mode-2 is for the 2nd group in the nighttime. Mode-3 is for the 1st group in the daytime and Mode-4 is for the 2nd group in the daytime. The training samples from these four modes are used separately, though the procedure for collecting training samples is the same as in (1), ..., (4). Detection of vacancy/occupancy is based on the respective modes with different parameter values of cluster centers, covariance matrices and other parameters such as the fuzzifiers of FCM. The number of training data was 2,997 (111 pictures). The data were selected from the first week's 27,000 images (1,000 pictures). The result is shown in Table VIII.

Table IX shows the result when additional 600 pictures obtained in the third week were used. 1,056 images (44 pictures) were selected and added to the training set. No samples from the second week (from the latter half 27,000 images) were used for training. The respective error rates of the four modes are also reported in the table. The daytime error rate on the background parking spaces (#12, ... #27, i.e., Mode-4) is less than 1 %, though the rate on Mode-3, i.e., the parking spaces in the foreground (#1, ..., #6) and their behind (#7, ..., #11) in the daytime, is 2.26 %. The number of training data, which was selected by the GUI of ParkLotD was 4,185 (i.e., 155 pictures). This result suggests that the generalization capability of FCM classifier is greatly improved by increasing the training data from a different week.

The last test was conducted at an indoor multi-story parking lot, which has 167 parking spaces for vehicles and 49 cameras are installed. Camera images from all the 49 cameras

TABLE IV
CLASSIFICATION ERROR RATE OF FCMC (1) WITHOUT PCA.

Test data	Error rate (%)	(Daytime)	(Nghttime)
First week	0.0140	0.0094	0.0266
Second week	0.0457	0.0259	0.0986

TABLE V
CLASSIFICATION ERROR RATE OF FCMC (2) WITH PCA DATA COMPRESSION.

Test data	Error rate (%)	(Daytime)	(Nghttime)
First week	0.0088	0.0048	0.0200
Second week	0.0408	0.0265	0.0790

TABLE VI

CLASSIFICATION ERROR RATE OF FCMC (3) WITH PCA DATA COMPRESSION AND ADDITIONAL TWO FEATURES.

Test data	Error rate (%)	(Daytime)	(Nghttime)
First week	0.0153	0.0049	0.0445
Second week	0.0555	0.0336	0.1137

TABLE VII

CLASSIFICATION ERROR RATE OF FCMC (4) WITH CONCATENATION OF VEHICLE IMAGE AND A PART OF SKY IMAGE.

Test data	Error rate (%)	(Daytime)	(Nghttime)
First week	0.0089	0.0055	0.0183
Second week	0.0438	0.0317	0.0760

TABLE VIII

CLASSIFICATION ERROR RATE OF FCMC (5) : MULTI-MODE.

Test data	Error rate (%)	(Daytime)	(Nghttime)
First week	0.0041	0.0028	0.0068
Second week	0.0276	0.0247	0.0315

TABLE IX

CLASSIFICATION ERROR RATE OF FCMC (5) : MULTI-MODE WITH ADDITIONAL TRAINING DATA.

Test data	Error rate (%)	(Daytime)	(Nghttime)
First week	0.0031	0.0014	0.0066
Second week	0.0204	0.0142	0.0318
Mode-1	-	-	0.0358
Mode-2	-	-	0.0308
Mode-3	-	0.0226	-
Mode-4	-	0.0088	-

are processed by a personal computer. The test results are shown in Table X. The detection error rate of occupancy state by our new system is smaller than the current system by an order of magnitude. The current system extracts edges from the vehicle image including front grill, head lights and license plate. The edges are not correctly detected due to glaring sun light and dark shadows in the daytime, and low-light intensity and back-lighting in the nighttime. Our new system ParkLotD overcomes this deficiency.

V. CONCLUSION

We demonstrated an efficient system for parking lot vacancy/occupancy detection, which achieves the practical use of the camera based system for outdoor parking lots. One of the burdens for installing the detector to a real parking lot is to collect the training samples with correct class information. ParkLotD provides an efficient method of selecting training samples from large amount of image data, which are obtained from cameras set in the parking lot.

The accuracy of detection may deteriorate if a large part of a vehicle is hidden behind an another vehicle in the foreground. Our future research work will be focused on this subject.

ACKNOWLEDGMENT

This work was supported by the Ministry of Education, Culture, Sports, Science and Technology, Japan, under Grant-in-Aid for Scientific Research (#20500210)

REFERENCES

[1] T. Ristola, "Parking guidance system in Tapiola," *Proc. of IEE Conf. Road Traffic Monitoring*, pp.195, 1992.

[2] S. Funck, N. Mohler and W. Oertel, "Determining car-park occupancy from single images," *Proc. of IEEE Sym. Intelligent Vehicles*, pp. 325-328, 2004.

[3] N. True, "Vacant Parking Space Detection in Static Images," Projects in Vision & Learning, University of California, 2007 [Online]. Available: http://www.cs.ucsd.edu/classes/wi07/cse190-a/reports /ntrue.pdf.

[4] S. Liu, "Robust vehicle detection from parking lot images," Tech. Rep. 2005-06, Boston University, Dept. of Electr. and Comp. Eng., Dec. 2005.

[5] D. B. L. Bong and K. C. Ting and K. C. Lai, "Integrated approach in the design of car park occupancy information system (COINS)," *IAENG International Journal of Computer Science*, vol. 35, no.1, pp. 7-14, 2008.

[6] S. Miyamoto, H. Ichihashi, and K. Honda, *Algorithms for Fuzzy Clustering, Methods in c-Means Clustering with Applications*, Springer-Verlag, Berlin, 2008.

[7] H. Ichihashi, K. Honda, A. Notsu and K. Ohta, "Fuzzy *c*-means classifier with particle swarm optimization," *Proc. of the IEEE International Conference on Fuzzy System, World Congress on Computational Intelligence*, Hong Kong, China, pp.207-215, 2006.

[8] H. Ichihashi, K. Honda, A. Notsu and E. Miyamoto, "FCM classifier for high-dimensional data," *Proc. of 2008 IEEE International Conference on Fuzzy System, World Congress on Computational Intelligence*, Hong Kong, China, June 1-6, 200-206, 2008. (Best Paper Award)

[9] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," *Proc. of the 6th International Symposium on Micro Machine and Human Science*, Nagoya, Japan, pp. 39-43, 1995.

[10] J. Kennedy and R. C. Engelbrech, "Particle swarm optimization ," *Proc of the IEEE International Conference on Neural Networks*, Piscataway, NJ, vol. 4, pp. 1942-1948, 1995.

[11] H. Ichihashi, A. Notsu and K. Honda, "Triplet of FCM classifiers," *Proc. of 2009 IEEE International Conference on Fuzzy System*, Jeju, Korea, Aug. 20-24, 2009.

[12] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, 1981.

[13] H. Ichihashi, K. Honda and A. Notsu, "Postsupervised hard c-means classifier," *Lecture Notes in Artificial Intelligence*, Springer, vol. 4259, pp. 918-927, 2006.

[14] M.E. Tipping and C.M. Bishop, "Mixtures of probabilistic principal component analyzers," *Neural Computation*, vol.11, pp.443-482, 1999.

[15] F. Sun, S. Omachi, and H. Aso, "Precise selection of candidates for hand written character recognition," *IEICE Trans. Information and Systems*, vol.E79-D, no.3, pp.510-515, 1996.

[16] A. Asuncion and D.J. Newman, UCI Repository of machine learning databases [http://www.ics. uci.edu/˜ mlearn/MLRepository.html]. Irvine, CA: University of California, Dept. of Information and Computer Science, 2007.

[17] T. V. Gestel et al., "Benchmarking least squares support vector machine classifiers," *Machine Learning*, vol. 54, pp. 5-32, 2004.

TABLE X

DETECTION ERROR RATE (%) TESTED AT AN INDOOR MULTI-STORY PARKING LOT

	ParkLotD	Current system
occupied	0.21	2.99
vacant	0.41	0.74