Video-based parking space detection

Marc Tschentscher, Marcel Neuhausen
Institute for Neural Computation, Ruhr-Universität Bochum
marc-philipp.tschentscher@ini.rub.de

Abstract: Finding a vacant parking lot in urban areas is time-consuming and, thus, not satisfying for potential visitors or customers. Efficient car-park routing systems could support drivers to get an optimal parking lot immediately. Current systems detecting vacant parking lots are either very expensive due to hardware requirement for each parking lot or do not provide a detailed occupancy map. In this paper, we propose a video-based system for low-cost vacant parking space detection. A wide-angle lens camera is used in combination with a desktop computer. Different feature extractors and machine learning algorithms were evaluated in order to retrieve accurate state information for each of the observed parking lots. We found a combination of feature extractors and classifiers which properly solved the given task. Our final system, incorporating temporal integration, reached an accuracy of 99.8 %.

1 Introduction

For vehicle drivers finding a vacant parking lot is a time-consuming and tedious task. Therefore a system to detect vacant spaces is desirable to route drivers efficiently to proper lots. By navigating drivers to parking lots close to their destination, such a routing system reduces the time spent for searching. In this manner it improves the convenience of car drivers.

Some systems have reached the market promising to support the driver by locating a vacant parking lot. On the one hand, a naive method is to evaluate the number of arriving and leaving cars within a certain area. This method is indeed very cheap but does not provide detailed information about positions of unoccupied lots. On the other hand, there are systems offering detailed occupancy information. Unfortunately, these systems are very expensive due to hardware requirements. They either need a magnetic sensor in the asphalt or a radar sensor above each lot. Typically, smartphone apps[1] can be used as interface for end users.

A video-based system offers a proper alternative to deal with the detection problem. Thereby it is possible to combine the advantages of the previously introduced methods, namely low-cost hardware requirements and providing detailed occupancy maps for parking areas. As we show in this paper, several image processing and machine learning algorithms already exist which can be employed to detect and classify vacant and occupied parking lots.

By using video-based systems several challenges occur – especially on outdoor car-parks. Different weather and lighting conditions or objects occluding some parking lots, might influence the accuracy of the results for the given task.

Some approaches to this task have already been proposed but are not in practical use, yet. True (2007) used color histograms to train a k-nearest neighbor algorithm as well as a support vector machine. In parallel, regions found by an interest point detector were classified. Wu et al. (2006) favored an advanced approach by regarding the surrounding parking lots and integrated them into the classification process. Ichihashi et al. (2010) proposed an approach based on fuzzy logic to classify single parking spaces.

[1] https://play.google.com/store/apps/details?id=com.parkmeright&hl=en

2 System Overview

In this paper we evaluate several combinations of feature extractors and machine learning algorithms to implement an observation system that classifies parking lots with a high reliability. All algorithms that are used in this context run on both step motion images and succeeding images. Latter is realized via temporal integration executed on the step motion images. We introduce the entire system in detail in the following sections.

Among others, for purpose of interchangeability of different feature extractors or machine learning algorithms, our system is separated into six modules which are specified in Sec. 4. A drafted overview therefor is given in Fig. 1.

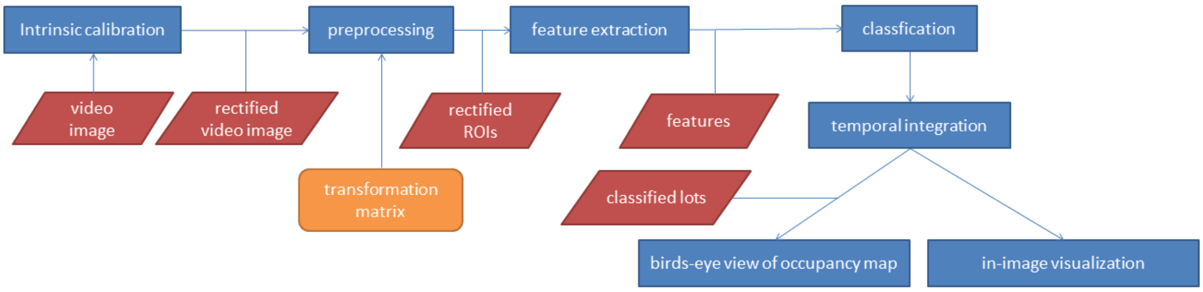

Figure 1: module overview

3 Hardware Setup

The software runs on a standard desktop PC. In order to cover a possibly large amount of parking spaces, a wide-angle lens camera of-the-shelf is used for recording. Like an observation camera, it is positioned in the back of the parking lots with a slight top view (Fig. 2). In our configuration this leads to a monitored area of 15 parking lots per camera. This fact also keeps the hardware and installation costs rather low, given that otherwise a sensor has to be applied to each lot.

4 Modules

The specific modules are lined up in a feed-forward pipeline such that the output of a preceding module acts as input of the succeeding. Beforehand, the system has to be calibrated once by means of the extrinsic camera parameters, shown in Sec. 4.1, to map camera to world coordinates. Then, a preprocessing which is specified in Sec. 4.2 is executed to normalize all parking lots in the single images. Section 4.3 illustrates the different types of features that were calculated on the images and fed into each of the classifiers described in Sec. 4.4. Afterwards these results are smoothed within a temporal integration (Sec. 4.5). Finally the classification results are visualized in two individual modes (Sec. 4.6).

4.1 Intrinsic Calibration

Due to the wide-angle camera lens, the radial distortion on the video image must necessarily be reversed in order to extract feasible data out of the recorded video. This ensures to get the exact and undistorted positions of the lots and, hence, is indispensable for the preprocessing.

4.2 Preprocessing

For transferring a camera point into the world coordinate system, a *direct linear transformation* (Abdel-Aziz et al. (1971)) is employed which calculates the required transformation matrix. Therefore one has to provide at least four world coordinates and their corresponding points in the image by hand.

Since only single parking spaces are relevant for detection, each lot initially has to be marked manually by hand in the video image (Fig. 2). Such labeled *regions of interest* (ROIs), each representing a single lot, are extracted from the image. The remaining image can be neglected. Each extracted ROI is then rectified both to fit a common size and to disregard its surrounding. Figure 3 shows examples of such rectified ROIs.

Figure 2: labeled car park

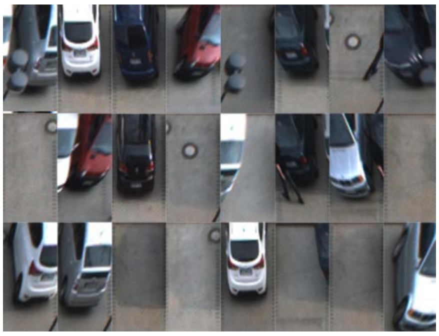

Figure 3: rectified ROIs

4.3 Feature extraction

For feature extraction various features are tested separately and in combination. The features we evaluated are described in detail in the paragraphs below. In particular we considered color histograms in three different color spaces, shown in Sec. 4.3.1. Furthermore edge features like gradient histograms (Sec. 4.3.2) and difference-of-Gaussian (doG) histograms, which are described in Sec. 4.3.3, were deployed. We also tested Haar-like features (Sec. 4.3.4).

4.3.1 Color histograms

Color histograms are one of the simplest ways to extract feature information from images. We developed an extractor, which is able to create multi-dimensional histograms with different resolutions per channel. We denote (4, 3, 2) as count of bits for each channel (4 bit for first channel, 3 bit for second channel and 2 bit for third channel).

Examples of such histograms are shown in Fig. 4. For the paper’s issue we concentrated on color histograms on RGB-, HSV-, and YUV-color spaces as they seem to be promising to us. Thereby it is important to distinguish between the asphalt color and the vehicles and, if applicable, to minimize the influence of brightness. The YUV color space possesses advantages with respect to the problems of brightness because of its explicit lighting channel Y. The HSV-color space parameterizes colors by their hue (H), saturation (S) and lighting (V) and so forms a cone in which grayish colors shape a closed region. The RGB-color space represents each color by its values of fundamental colors: red, green and blue.

4.3.2 Gradient histograms

On each ROI image the gradients were calculated in both x- and y-direction using a common Prewitt operator (Jain et al. (1995)). Based on the resulting edge image a histogram of angles is computed.

4.3.3 Difference-of-Gaussian histograms

Like gradient filters the difference-of-Gaussian method, described in Jähne (2002), also extracts edge information from ROI images. Its functionality is similar to a band pass filter.

Using a simple Gauss-filter, at first a highly smoothed image is generated from the source image which then is subtracted from the original. Figure 5 shows a thresholded example of the procedure. The histogram is then constructed via the distribution of gray scale values of the results.

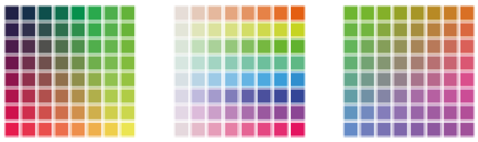

Figure 4: RGB, HSV and YUV-histogram

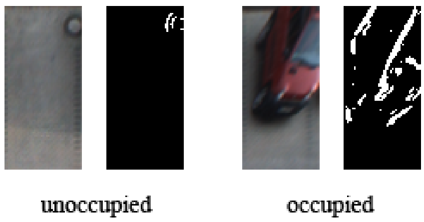

Figure 5: doG-images

4.3.4 Haar-like feature

Haar-like features (Viola et al. (2001)) are useful edge-filters (Fig. 6) which are calculated efficiently with the help of integral images. The algorithm, thereby, determines the sum of pixel values within a predefined window.

Existing Haar features can detect different orientations of edges. Vertical edges can be detected with features like the first and third whereas the second and fourth detect horizontal edges.

Figure 6: Haar features

4.4 Classification

In case of parking lot classification the algorithms have to perform a binary classification (occupied/vacant). Several classifiers were tested and compared to each other. To get significant results, we built a dataset, which contains approximately 10,000 samples. For our experiments, it was divided into two disjoint sets for training and testing, respectively. In particular we used a *k*-nearest neighbor (*k*-NN) (Sec. 4.4.1), linear discriminant analysis (LDA) as linear classifier, explained in Sec. 4.4.2, and a support vector machine (SVM) (Sec. 4.4.3).

4.4.1 *k*-nearest neighbor

The *k*-NN is among the simplest machine learning algorithms. Elements will be assigned to classes with smallest distance in relation to the *k* nearest training examples in feature space. In the training phase simply each training examples' multidimensional feature vector and its dedicated class label is stored. Afterwards an element is classified by a majority vote of its nearest neighbors.

4.4.2 Linear discriminant analysis

A method to train a linear classifier is the LDA (Hastie (2001)). The algorithm finds a linear combination for given feature vectors separating the labeled classes on the training set. The resulting combination can then be used to linearly classify elements and therefore fulfill real-time constraints.

4.4.3 Support vector machine

Amongst others, SVMs (Vapnik (2008)) are used to even classify non-linear separable data sets. Within the SVM, every object is represented as a multidimensional feature vector. During training, the feature vectors are transformed into a much higher dimensional space where the SVM is able to define a sectional hyper plane, which distinguishes the already classified training objects from each other with a possibly large "safety" margin. The hyper plane is then transformed back into the feature space and used for classification.

4.5 Temporal integration

For temporal integration exponential smoothing via an alpha beta filter was applied. Term (1) shows the filter's equation where α represents the learning rate and Y represents the particular class label (vacant/occupied) for each parking lot. If $\alpha = 1$, no new classification will affect the older for a specific parking space. In opposite, $\alpha = 0$ means no temporal integration. The term $f(t)$ is calculated in every time step or rather for every newer classification of a parking space respectively and, hence, leads to an temporal integrated probability of a vacant space.

$$f(t) = \alpha * Y_{t-1} + (1 - \alpha) * Y_t \qquad \textbf{(1)}$$

Though using this filter may lead to a time-shifted (i.e. slightly delayed) detection, this circumstance is insignificant for implementing it for car park routing systems.

4.6 Visualization

We implemented two different types of visualization. On the one hand, colored semitransparent layers are placed directly on the parking space in the video (Fig. 7). So, a green layer stands for a vacant parking lot and a red layer for an occupied one in turn. On the other hand, a virtual mapping of the world is computed. Using the world coordinates a true to scale parking area is drawn and occupied lots are indicated by a colored car pictogram (Fig. 8).

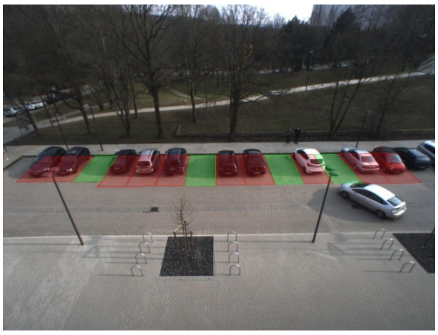

Figure 7: classified lots

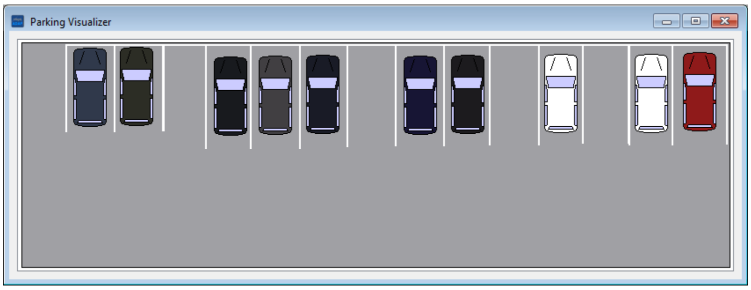

Figure 8: parking visualizer

5 Experiments

This section describes our experimental setup and the results obtained on classifying parking lots.

5.1 Setup

We built a dataset containing approximately 10,000 samples which was recorded on two different days with different lighting conditions. For our experiments we divided it into two disjoint sets for training and testing purpose, respectively.

In preliminary experiments we tested each feature extractor individually combined with each of the classifiers to evaluate the best features. Therefore we used different color spaces for color histograms. We also varied the color histogram's and doG's resolution. We found that color histogram and doG features outperformed the other features. The two best features were then used in combination for all further experiment.

The classifier parameters were chosen equally in all experiments. We employed k-NN with $k = 5$, using Euclidian distance measure and 50 prototypes per class. The LDA was regularized by adding $\hat{\sigma}^2 \in [10^{-6}, 1]$ to the diagonal elements of the empirical covariance matrix. Regularizing LDA can lead to better generalization (Hastie (2001)) and ensures numerical stability at the same time. During training the SVM we performed model selection with γ set between 0.1 and 1 and C between 0.1 and 1000. As kernel a radial basis function was employed.

Finally for the temporal filtering we varied $\alpha \in [0.7, 0.8]$ and a threshold above 0.8 considering *f(t)*.

5.2 Results

In our experiments we found different configuration yielding promising results. However, we observed misclassifications in non-standard situations, like cars exceeding their parking space boundaries or crossing pedestrians.

The combination of color- and doG-features is the only one performing well on non-standard situations described above but induces misclassifications on single images. Table 1 shows the results of these experiments. The highlighted cells represent the best feature configuration for each classifier. It is noteworthy that for all classifiers considered, the best results were obtained using the same color histogram-features but different doG-features.

Table 1: test error of color- and doG-histogram (in %)

	RGB (2, 2, 2)	RGB (3, 3, 3)	HSV (4, 0, 0)	HSV (3, 3, 0)	HSV (4, 4, 0)	HSV (3, 3, 2)	HSV (0, 3, 3)	YUV (2, 2, 2)	YUV (3, 1, 1)	YUV (0, 4, 4)	
LDA	4.39	5.08	4.21	4.19	4.37	5.08	4.63	4.40	4.40	4.72	DoG (3x3)
kNN	11.23	19.31	12.53	8.76	10.13	21.41	17.40	10.84	10.57	15.86	
SVM	6.23	6.50	7.37	5.25	2.91	5.10	5.94	6.39	4.90	9.18	
LDA	4.51	5.82	5.04	4.63	4.01	5.29	5.32	5.38	4.84	5.52	DoG (5x5)
kNN	9.36	12.18	9.83	8.23	7.04	10.54	9.80	9.53	9.21	13.87	
SVM	6.61	6.26	6.83	5.73	3.56	5.38	6.08	6.59	5.58	8.58	
LDA	4.54	5.70	4.90	4.10	3.90	4.60	4.46	4.60	4.40	6.95	DoG (7x7)
kNN	10.90	13.01	11.08	10.51	8.71	16.01	10.07	10.69	9.86	13.66	
SVM	5.25	6.20	5.76	5.82	3.42	4.46	5.79	6.06	4.90	7.84	
LDA	4.81	5.38	5.08	4.40	3.50	4.22	4.72	4.99	5.44	8.38	DoG (9x9)
kNN	19.75	18.50	18.74	18.44	13.96	20.46	15.30	19.39	17.55	21.77	
SVM	6.08	6.77	5.01	5.94	4.31	4.40	5.56	4.90	4.37	7.54	

Figure 9 shows an example of a car leaving a parking lot with the best setup found. Due to the temporal filtering applied the estimated occupancy changes with a slight delay. Note that the other parking lots are classified correctly.

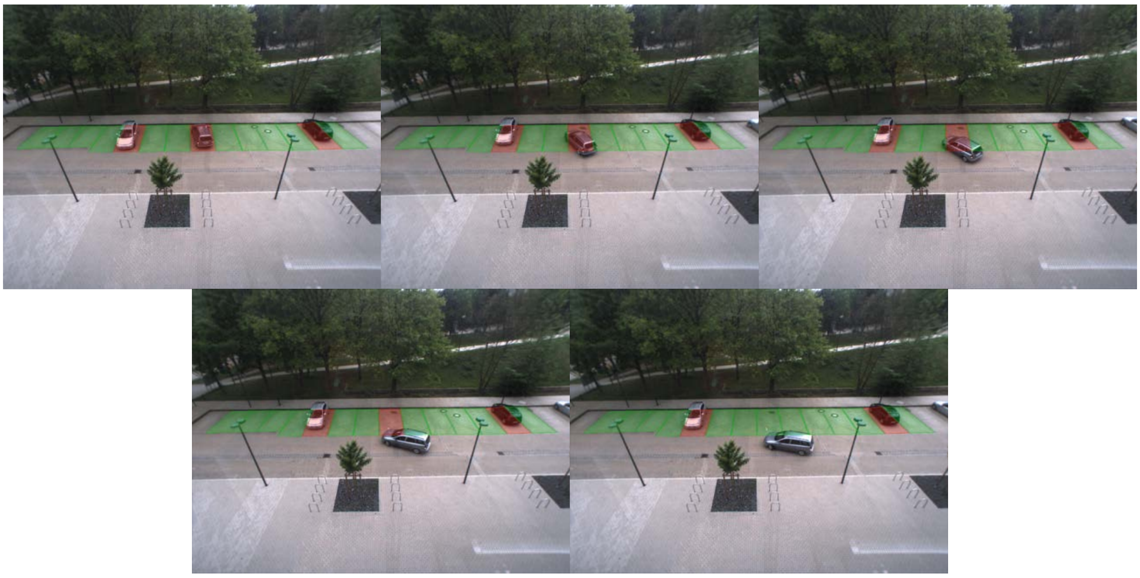

Figure 9: leaving a parking lot (result with temporal integration)

6 Conclusion

In this paper we developed a promising video-based system for vacant parking space detection which can be adopted by a car-park routing system to navigate drivers to a comfortable parking space. We evaluated different combinations of image features and machine learning algorithms. Our final system relies on color histograms and difference of Gaussian features, an SVM classifier, and exponential smoothing for temporal integration. This system reached an accuracy of 99.8 % on classifying parking situations. Furthermore, we achieved real-time speed for all six modules and the system as a whole.

Even though our system is tested only on outside parking lots, it is also imaginable to use it in parking garages. In this case on the one hand more cameras are needed since the cameras must be positioned closer to the parking spaces due to the lower suitable space and on the other hand the lightning conditions have to be adjusted.

Improvements can be achieved by minimizing influences of adjacent cars overlaying the labeled area due to the camera perspective. Further experiments should be done on other car-parks featuring different visual properties.

7 References

Abdel-Aziz, Y.I., Karara, H.M. (1971). Direct linear transformation from comparator coordinates into object-space coordinates in close-range photogrammetry. In: Proceedings of the ASP/UI Symposium on Close-Range Photogrammetry. American Society of Photogrammetry, Pages 1-18

Hastie, T., Friedman, J., Tibshirani, R. (2001). The elements of statistical learning. Springer Series in Statistics.

Ichihashi, H., Katada, T., Fujiyoshi, M., Notsu, A. and Honda, K. (2010). Improvement in the performance of camera based vehicle detector for parking lot. In: Proceedings of the IEEE International Conference on Fuzzy Systems. Pages 1-7

Jähne, B. (2002). Digitale Bildverarbeitung. Springer Verlag

Jain, R., Kasturi, R., Schunck, B. G. (1995). Machine Vision. McGraw-Hill, Inc.

True, N. (2007). Vacant parking space detection in static images. University of California, San Diego

Vapnik, V. N. (2008): Statistical Learning Theory. Wiley-Interscience

Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Vol. 1, Pages 511-518

Wu, Q. and Zhang, Y. (2006). Parking lots space detection. Carnegie-Mellon University: School of Computer Science / Machine Learning [Format: PDF, Date: 04/26/2012, URL: http://www.cs.cmu.edu/~epxing/Class/10701-06f/project-reports]