

# Vacant Parking Space Detection in Static Images

Nicholas True

University of California, San Diego  
9500 Gilman Drive, La Jolla, CA 92093

ntrue@ucsd.edu

## Abstract

*Current parking space vacancy detection systems use simple trip sensors at the entry and exit points of parking lots. Unfortunately, this type of system fails when a vehicle takes up more than one spot or when a parking lot has different types of parking spaces. Therefore, I propose a camera-based system that would use computer vision algorithms for detecting vacant parking spaces. My algorithm uses a combination of car feature point detection and color histogram classification to detect vacant parking spaces in static overhead images.*

## 1. Introduction

Nobody enjoys circling parking lots looking for non-existent empty parking spaces. A lot of time and effort could be saved if information on parking space availability could be accessed by drivers via phone or with a vehicle's gps-map display. Unfortunately, current vacant parking space detection (VPSD) systems involve simple parking lot entrance/exit sensors which have no way of determining which types of parking spaces—student spots, faculty spots, etc.—are available. Furthermore, the current VPSD systems cannot determine when more than one parking space has been obstructed either by an oversized vehicle or by a poorly parked car.

Another possible solution to this problem is to put individual magnetic sensors under the asphalt at each parking space. This system is simply not practical due to the cost—roughly \$20 to \$60 per sensor not including installation labor costs which could easily double the final cost per space<sup>1</sup>.

An alternate approach for vacant parking space detection could use parking lot security cameras along with computer vision algorithms to make a more robust and flexible system. This type of system has been implemented with some success in [1] using a support vector machine to classify the

color histograms of parking spaces. In [1], a multi-class SVM was used to classify the state of three, contiguous parking spaces at a time. Markov Random Fields were then used to improve the final classification results by resolving conflicting classifications between two neighboring three-space patches.



Figure 1. Example of an overhead security camera image

Yet another approach to detecting parking vacant parking spaces is to use car feature detection, like in [2], to determine the existence and location of vehicles in a parking lot. The advantage of vehicle feature detection over color histogram classification is that feature detection won't get confused between vehicles whose color is very similar to possible pavement colors either due to extremely low light levels or due to the paint color of the vehicle itself.

In this paper I present and compare a color histogram VSPD system and a vehicle feature detection VPSD system. In the future, these different algorithms could be combined to create a VPSD system which is more accurate than any of its individual parts.

## 2. Approach

The algorithm breaks down into four stages: extraction of human-labeled parking space regions, pre-processing (converting to L\*a\*b\* color space and creating the color histograms) and color histogram classification, vehicle fea-

<sup>1</sup>[http://www.ssec.honeywell.com/magnetic/mark\\_det.html](http://www.ssec.honeywell.com/magnetic/mark_det.html)

ture detection (interest point detection and feature extraction), and combining the results to form the final classification. The first three stages are the focus of this paper as the last stage is still under development.

## 2.1. Parking Space Extraction

The first step in the proposed VPSD system is to capture a still, overhead image of the parking lot and extract the individual parking spaces. To increase the number of vehicles which can be seen in each picture, the camera will be angled by some arbitrary degree. This does make the classification process more complicated as the system is then forced to overcome problems of vehicle occlusion but the benefit is that fewer cameras will be required to cover the same area, resulting in lower system costs.

As the system will remain unchanged (no movement by the camera) from the time of installation onward, the system can utilize human labeled parking space regions of interest (ROIs). Automatic ROI labeling is possible but not necessary as there are usually not very many spaces to label in each image and this operation only happens when the system is first being installed. Using the parking space ROI labels, the system then extracts information from each individual parking space and passes it on to the color histogram classifier and the feature point detection algorithm.

## 2.2. Color Histograms

The color histogram classifier takes each individual parking space ROI and converts its color space from RGB to  $L^*a^*b^*$ . By throwing away the luminance channel ( $L^*$ ) and only keeping the chrominance channels ( $a^*b^*$ ), the images become (mostly) invariant to lighting (see [3] for more explanation). Afterwards, a color histogram—32 bins for the  $a^*$  channel and 32 bins for the  $b^*$  channel—is constructed for each parking space sub image. These color histograms are then classified either using a support vector machine or a k-nearest neighbor classifier.

## 2.3. k-Nearest Neighbor Classifier

The k-nearest neighbor (k-NN) classifier is a simple yet effective classification algorithm. The tradeoff for this algorithmic simplicity is that the classification runtime is very slow and continues to get slower as the size of training set is increased. However, it is important to note that as the size of the training set approaches infinity, the k-NN's error rate is guaranteed to be no worse than twice the Bayes error rate (the minimally achievable error rate) [4].

The algorithm works by comparing the data which you want to classify against all of the training data. The distance between the input and each instance of training data is computed and the  $k$  closest training points are recorded. The  $k$  data points then vote and the input is labeled with the most

frequently-occurring class. To prevent ties from occurring in the voting process,  $k$  is usually an odd number.

There are many different distance metrics which can be used by the k-NN classifier. The most commonly-known distance function is Euclidean Distance Metric where  $h_i$  and  $h_j$  are histograms with  $n$  bins:

$$D(h_i, h_j) = \sum_{k=1}^n \sqrt{(h_i(k) - h_j(k))^2} \quad (1)$$

Because color histograms are probability density functions, the Chi-Squared Distance, Equation (2), is more applicable to this problem. In fact, [5] showed that out of a number of non-parametric test statistics, Chi-Squared gets some of the best results meaning that it is one of the more accurate distance metrics for this type of problem.

$$\chi^2(h_i, h_j) = \frac{1}{2} \sum_{k=1}^n \frac{(h_i(k) - h_j(k))^2}{h_i(k) + h_j(k)} \quad (2)$$

Given two histograms with equal Euclidean Distance, Chi-squared distance's normalizing factor determines how significant that distance is given the relative percentage of elements in the bins compared to the total.

## 2.4. Support Vector Machine Classifier

An alternate way of classifying the parking space color histograms is to use a support vector machine. Developed by Vapnik, a support vector machine (SVM) is a sophisticated general classification algorithm which computes the maximum margin hyperplane separating the training data by class and uses it to classify the input data [6]. For this project, I used a binary SVM to classify parking spaces as vacant or occupied, based on the fact that vacant parking spaces all have a similar color. In [1], a multi-class SVM was used to classify the state of three, contiguous parking spaces at a time. Markov Random Fields were then used to improve the final classification results by resolving conflicting classifications between two neighboring three-space patches. to achieve good detection rates. Given the same number of training image regions, both methods achieve similar results.

## 2.5. Vehicle Feature Detection

Working in parallel to the color histogram classification, a car feature point detection algorithm takes each parking space sub image and classifies it as being vacant or occupied according to the number and types of features that are recognized.

The first step is to detect interest points in the given image using corner detection. The algorithm uses an implementation of Harris corner detection instead of Förstner Corner Detection as in [2] because the Förstner Operator

didn't detect many interest points in the training images. During training, the algorithm creates a vocabulary of vehicle features and another of non-vehicle features.



Figure 2. Interest point detection using Harris corner detection

The first phase in creating the vocabulary of features is to normalize the size of the parking space ROIs by up or down-scaling the ROI subimage according to the length of the longest ROI edge. This normalization is only an approximation because perspective effects and the camera angle relative to the ROI in question significantly effect the perceived dimensions of the parking space. However, with a large enough feature vocabulary, the slight differences in the scale of some features will not matter as the system will have ample examples from all of the feature sizes.

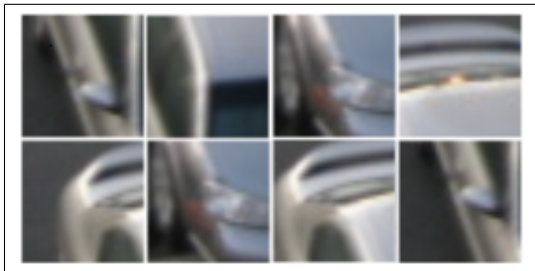


Figure 3. Sample feature blocks from one occupied parking space

The final step in creating the feature vocabulary is to create small 25x25 images centered at each feature point. These small images are features that the algorithm uses to detect vehicles in images. Because a lot of features tend to be 1-pixel-shifted variants of each other, the algorithm only keeps one feature image for every interest-point cluster within a 3 pixel range of each other. In [2], the feature vocabulary is then further divided into subsets of similar feature images. The existence or non-existence of these feature classes are then used to detect and locate the target object in the test image. This step will be incorporated into my feature detection algorithm in future stages of research.

During the classification phase, a new feature vocabulary is created for the image region begin tested. Each feature is

then compared to the feature vocabulary of the training set using Normalized Cross Correlation. The individual feature's class guess then 'votes' on the final classification of the target ROI (occupied or vacant). Because a car takes up most of a parking space, it may seem like it is unnecessary to have the detected features vote. However, objects like cups and oil stains can often be seen on the edges of a parking spots and this would throw off the system if the voting scheme was not used.

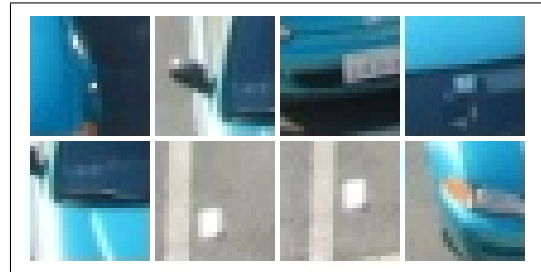


Figure 4. The two bottom middle images have a cup which was misclassified as a car feature because the cup lay within the parking space ROI.

### 3. Training and Testing

The training regimen involved a  $k$ -fold cross-validation test with  $k$  equal to five. This means that the training set was randomized and divided into five subsets. Four of the subsets were combined into a subtraining set and the remaining subset was then tested. This process was repeated for each of the five subsets and the results from each subset test were then averaged to get the final training results. This process was used on just the day images, just the night images, and the whole training set to get an idea of how effective the system was under those different conditions and whether or not lighting and time of day seemed to having a noticeable affect on the results.

For the final stage of testing, the different classifiers were trained on the whole training set and then given a single pass over the test set. The final test set contained no images that had ever been used for testing as this could corrupt the results and skew the analysis of the system's effectiveness.

### 4. Data and Results

The acquisition of both the training data and testing data used a Cannon PowerShot A540 hand-held camera using both the landscape setting and the night setting. In the cases where the landscape setting captured a completely black image due to poor lighting, I used the night setting. In an effort to reproduce the low resolution of many security cameras, all images were taken at 480x640 resolution. The set of training images was culled down to 29 images which varied widely in terms of lighting conditions and parking lot

locations. I then hand labeled a total of 758 parking space regions which had very little occlusion from neighboring vehicles and showed good park-jobs (no images of a car half in one spot, half in another).

The test set is comprised of 200 parking space regions. New pictures were taken of parking lots that had never been trained on and new pictures were taken of parking lots that had been used to create the training set. Just like the training set, the test set contains day, night, and dusk images. It is important to note that the test set only contains parking space ROIs with very minimal vehicle to vehicle occlusion as the project is still lacking algorithms which deal with this problem.

#### 4.1. Color Histogram Results

The following are the results of the color histogram classifier first using a k-nearest neighbor classifier and then using a support vector machine.

##### 4.1.1 Support Vector Machine Results

The following are the results for the SVM classifier on a test set of 127 occupied parking space regions and 63 vacant parking space regions.

Linear Kernel	Num. Support Vectors	Rate
All Images	513	89.5%
Day Images	341	68.5%
Night Images	143	87.0%

Table 1. SVM Results (Linear Kernel)

RBF Kernel	Num. Support Vectors	Rate
All Images	390	94.0%
Day Images	298	88.0%
Night Images	105	91.0%

Table 2. SVM Results (Radial Basis Function Kernel)

One interesting thing to note is that in the training phase of the project, the k-NN classifier using the Chi-Squared distance metric consistently outperformed the SVM with a linear kernel by 2%-5%. As the final set of tests show, the SVM with either kernel achieved superior results to the k-NN classifier. Furthermore, the SVM has a significantly faster run time than the k-NN classifier.

##### 4.1.2 k-Nearest Neighbor Results

The following are the results for the k-NN (where  $k=3$ ) classifier on a test set of 127 occupied parking space regions and 63 vacant parking space regions.

Euclidean Distance	Rate
All Images	87.0%
Day Images	78.0%
Night Images	86.5%

Table 3. k-NN Results (Euclidean Dist.)

Chi-Squared Distance	Rate
All Images	89.0%
Day Images	81.0%
Night Images	86.5%

Table 4. k-NN Results (Chi-Squared Dist.)

It was previously mentioned that for the k-NN classifier, Chi-Squared Distance was a better distance metric than Euclidean Distance and the results show this to be true.

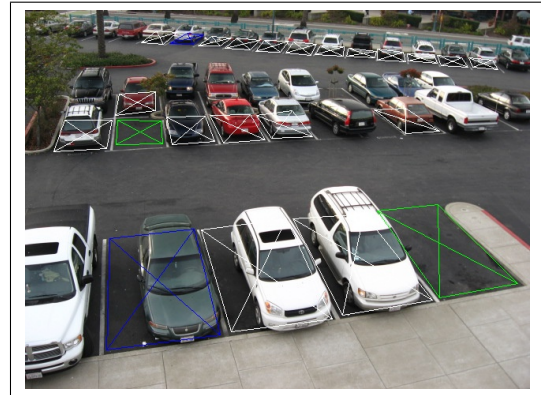


Figure 5. Results for k-NN using Chi-Squared distance. Note that *white*=occupied classification, *green*=vacant classification, and *blue*=misclassified.

As one can see, there are two misclassified spaces in Figure 5. A likely reason for the misclassification of the car on the bottom left is that the classifier 'confused' the dark color of the car as being the darkish gray color of the pavement. While the misclassified space in the top of the image is in fact light grey in color, most of the parking space ROI encloses the shadowy bottom of the vehicle which could be the reason for the misclassification.

#### 4.2. Vehicle Feature Detection Results

The following are the results of the feature detection algorithm used to detect the presence or absence of a vehicle in a given parking space. The test was conducted on a set of 127 occupied parking space regions and 63 vacant parking space regions.

Currently, the major drawback of the feature detection system is that it is *very* slow (it takes roughly twelve hours to classify 115 parking space ROIs with a training set of 758 images). Much of slow down is due to inefficiencies

Normalized Cross Correlation	Rate
All Images	49.0%
Only Images with Features	85.2%

Table 5. Vehicle Feature Detection Results

in the system's implementation. Just like k-nearest neighbor, this algorithm slows down as the size of the training set increase. This is because the feature detection algorithm iterates over the training vocabulary each time it classifies a feature image.

Another problem with the vehicle feature detector is that there were a number of cases where the Harris corner detection algorithm failed to generate any interest points. This means that the feature detection algorithm effectively fails at that point. However, in these cases where the algorithm fails to give a classification, the results from the color histogram classifier can be used as a substitute. This means that feature detection is still a very viable vacant parking space detection algorithm.



Figure 6. Example of features from an empty space being classified as being from an occupied space.



Figure 7. Example of features from an empty space being classified as being from an occupied space.



Figure 8. Example of features from an occupied space being classified as being from an empty space.

After examining these misclassification examples from the feature detection algorithm it is quite apparent that these features are somewhat ambiguous as to which class they belong to. The problem lies in the fact that the system is trained on regions and not just objects. This means that everything within an occupied parking space training image will get labeled as a feature of occupied spaces. The same problem exists for the training of vacant parking spaces.

During the classification phase, these misclassified training features throw off the final classification.

A solution to this problem is to be a lot more sensitive in the training phase: use smaller parking space ROIs which don't encompass the edges of parking spaces which is where most of the misclassified features are derived from. Of course the trade off is that this would probably require more human intervention to help weed out some of the bad training features, increasing the training time.

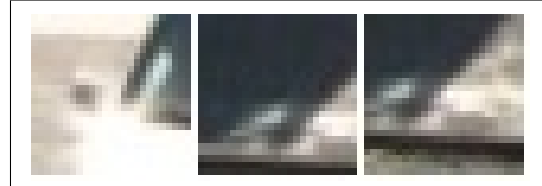


Figure 9. Example of a correctly-classified set of empty parking space features.



Figure 10. Example of a correctly-classified set of occupied parking space features.

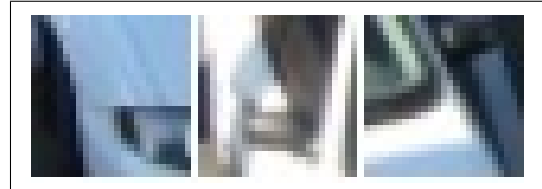


Figure 11. Example of a correctly-classified set of occupied parking space features.

## 5. Conclusion

This paper presents a novel computer-vision based parking space vacancy detection system. Using a combination of color histogram classification and vehicle feature detection, the system achieves good detection results. One major weakness of the current algorithm is that it can't accurately detect the state of parking spaces which are slightly or mostly occluded by objects such as other vehicles. While this shortcoming doesn't prevent the algorithm from working, it does mean that any real world implementation of the system would require more cameras to cover all the parking spaces in ways that won't see occlusion. This of course would increase the overall cost, thus, making it desirable to find a solution to this problem. Future work could fold in real-time video feeds and use vehicle tracking to and from parking spots to help overcome the problem of occlusion.

The original goal of this project was to create a vacant parking space detection system which utilized different fea-



ture detection algorithms—in this case color histogram classification and vehicle feature detection—in concert with each other to create a classifier which was more accurate than any of the individual classification algorithms. Unfortunately, there wasn't enough time to create a method for combining the results. Accordingly, this paper is more of an evaluation of the different individual parking space detection algorithms. However, future work could be applied to using boosting as one possible way of joining the different feature classifiers into one cohesive classification system.

## 6. Acknowledgements

Thanks to Serge and the WI07 cse190 class for the help and suggestions and a special thanks to the over-crowded parking situation at UCSD for the motivation!

## 7. References

- [1] Wu, Q. and Zhang, Y. (2006) *Parking Lots Space Detection*, Machine Learning, Fall 2006, Carnegie Mellon University.
- [2] Agarwal, S. and Roth, D. (2002) *Learning a Sparse Representation for Object Detection*, 7th European conference on Computer Vision. Vol. 2353 2002.
- [3] Marshall, P. (2005) *Understanding Colour*, <http://photography.about.com/library/weekly/aa031703b.htm>
- [4] Cover, T. and Hart, P. (1967) *Nearest neighbor pattern classification*, IEEE Transactions on Information Theory. Vol. 13 1967.
- [5] Puzicha, J., Hofmann, T., and Buhmann, J. (1997) *Non-parametric Similarity Measures for Unsupervised Texture Segmentation and Image Retrieval*, IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 1997.
- [6] Vapnik, V. (1995) *The Nature of Statistical Learning Theory*, Springer, 1995.
- [7] Feng, X., Fang, J., and Qui, G. (2003) *Color Photo Categorization Using Compressed Histograms and Support Vector Machines*, Image Processing, Vol. 3, 2003.