

Performance Analysis of a Simple Vehicle Detection Algorithm

H. Moon^a R. Chellappa^a A. Rosenfeld^{a,*}

^a*Center for Automation Research, University of Maryland, College Park, MD
20742-3275*

Abstract

We have performed an end-to-end analysis of a simple model-based vehicle detection algorithm for aerial parking lot images. We constructed a vehicle detection operator by combining four elongated edge operators designed to collect edge responses from the sides of a vehicle. We derived the detection and localization performance of this algorithm, and verified them by experiments. Performance degradation due to different camera angles and illuminations was also examined using simulated images. Another important aspect of performance characterization — whether and how much prior information about the scene improves performance — was also investigated. As a statistical diagnostic tool for the detection performance, a computational approach employing bootstrap was used.

Key words: Performance analysis, Vehicle detection, Aerial image, Bootstrap, Empirical evaluation

1 Introduction: Performance Characterization

The task of a computer vision algorithm can be specified in terms of two components: the range of images to be processed and the performance criterion that the algorithm should try to achieve. The algorithm can then be designed to handle every image in the given class, and at the same time, to optimize the specified criterion function.

* Corresponding author.

Email addresses: hankyu@cfar.umd.edu (H. Moon), rama@cfar.umd.edu (R. Chellappa), ar@cfar.umd.edu (A. Rosenfeld).

The performance of the algorithm is evaluated in terms of these components, using two different methodologies: theoretical formulation of the relation between the imaging conditions and the criterion function, and empirical evaluation using real or simulated imagery.

If there is an underlying model for the image and for the possible perturbations of the image that we may expect in a real environment, and also if the algorithm is based on some mathematical formulation, it may be possible to predict the performance theoretically. Haralick [15] has asserted that performance characterization of a vision algorithm has to do with establishing the random variations in the output data that result from the random variations and imperfections in the input data. However, a scene model is not always available, or is usually not complete; hence the error propagation computed using a theoretical model is usually a crude approximation.

Numerous imaging conditions can affect the performance of an algorithm, including scene noise and camera and lighting angles. If we can quantify these scene factors and if we have a very large set of images annotated with these parameters, experiments on this set of images would give a very informative performance measure for the algorithm. In reality, gathering data for such a complete and systematic evaluation is not usually possible.

Most algorithms are based on some model or an underlying theoretical framework; however, analytical tools for predicting the behavior of algorithms are not usually available. On the other hand, it is usually easy to generate simulated images and investigate algorithm performance using these images. If a perturbation model for the scene is available, we can generate perturbed images and conduct a systematic evaluation. While this kind of experiment is commonplace in computer vision, there have been objections to this method: the scene model as well as the perturbation model are far from realistic. This is true; however, as stated in [12], simulation is inevitable to verify the correctness of implementations, and also to analyze the behavior of algorithms under varying conditions for which real images are not available. Some scene environments are not easy to model, and it is usually impossible to acquire annotated ground truth data for all the relevant sets of parameters. In our analysis of the vehicle detection task, we followed the above guidelines.

The three most important characterizations of the task of vehicle detection are the detection, false alarm, and localization performances. There are numerous relevant scene factors, including resolution, noise, camera and lighting conditions, parking lot layout, etc. It is important to identify which parameters affect the criterion function significantly and which do not. We have found that the camera and illumination angles are the most crucial factors. We tested extreme values of these factors to verify the degradation of performance. With low illumination angles, the vehicle detector produces many

false alarms due to long shadows; also, very oblique camera angles contribute to a poor detection rate. Since these angles are the most crucial factors in algorithm performance, we can conjecture that some prior information about these parameters can reduce errors.

The noise level of the image, how the vehicles are positioned in the parking lot, and the colors of the vehicles also affect performance. We have designed a vehicle detector based on a simple vehicle model and Canny's formulation of edge detection, and have investigated these issues in a controlled environment using a parking lot scene simulator, as well as in a real environment. The utility of site information was investigated using real images with ground truthed vehicles and annotated parking lot orientations.

By using the local linearity of the vehicle operator response, statistical properties of its detection and localization performance were derived, and they were then verified by simulation. We can view this formulation as error propagation: via the geometry of the vehicle model, from the image domain to the detection probability and parameter space. This framework is one of the contributions of this work, which can be extended to more general problems.

As it turned out, the most serious problem in vehicle detection is the occurrence of many false alarms, due to spurious responses from adjacent vehicles, road structures, and buildings. We developed an empirical hypothesis test to remove false alarms and self-diagnose the detection and localization performance.

This paper is organized as follows: Section 2 introduces the task of vehicle detection (2.1) along with a review of related work (2.2). By extending Canny's formulation of step edge detection (2.3), we provide a solution to the problem of vehicle detection (2.4). The output of the algorithm for a small parking lot image is discussed in Section 2.5. A mathematical formulation is given in Sections 3.1 and 3.2. Performance evaluation using simulated images is discussed in Section 4. Experiments on real ground-truthed images are given in Section 5. Finally, a method of self-diagnosing the performance using bootstrap is introduced in Section 6.

2 The Vehicle Detection Algorithm

2.1 *The task of vehicle detection*

Aerial parking lot images usually have quality limitations due to the low resolution of the camera optics and atmospheric turbulence. The vehicles in the

images we have used to test our algorithm occupy about 7 by 17 pixels on the average, and have approximately rectangular shapes. The front and/or rear windshields are usually recognizable, depending on the color of the vehicle and lighting conditions.

Aerial vehicle images are generally simpler to model than other objects. While vehicles in a typical parking lot have different overall dimensions, shapes, relative positions of windshields, and colors, we can use a simple rectangular model for the vehicle boundaries. We also used some variations on this model to deal with camera and lighting conditions, as discussed in later sections.

2.2 Related work

Most of the work on vehicle detection or recognition [3] [11] [18] has been on ground images, mainly as preprocessing before tracking for surveillance or traffic applications. There have not been many papers on non-military vehicle detection in aerial imagery. The proposed method makes use of a similar vehicle model and essentially the same image features as in [9], where local edge detection, a generalized Hough transform (GHT), and “rubber-band matching” are performed. The GHT is employed to narrow the search space for a vehicle centroid using edge information. A rectangular band of fixed width is examined in every position where the GHT response is high. The number of edge pixels inside the band is taken to be the likelihood of the presence of a vehicle at that position. [4] investigates parameter adjustment of the method used in [9] using a Bayesian and Neyman-Pearson framework. [19] uses a machine learning approach employing a hierarchical vehicle and road structure model and a multi-layer perceptron for classifying vehicles and non-vehicles. While there has been much work on the detection of military vehicles using SAR [2] [9] or FLIR imagery taken either from the ground or from the air, these sensors don’t usually provide the same level of geometric image signatures as visible light images do.

The proposed work is performed in the context of site monitoring and surveillance, such as vehicle activity monitoring [9], and vehicle detection on roads [10]. This work exploits a site model (the positions and structures of buildings and roads) and a context model (regions where certain activity is more probable) to detect changes and activities. Some of the performance issues investigated in this paper, such as the performance degradation due to illumination and acquisition angle changes, or the role of site and context models, were considered in this previous work.

2.3 Canny's formulation of edge detection

Our approach uses Canny's formulation [7],[8] of optimal edge detection. Canny observed that there is a natural trade-off between detection and localization performance with varying operator size. Given the step edge

$$s(x) = \begin{cases} 0 & \text{for } x < 0 \\ d & \text{for } x \geq 0 \end{cases}$$

with amplitude d and average noise amplitude n_0 , and the edge operator function f defined on $[-w, w]$, the signal-to-noise ratio (which dominates the detection performance) and the reciprocal of average localization error are given by

$$\text{SNR} = \frac{d \left| \int_{-\infty}^0 f(x) dx \right|}{n_0 \sqrt{\int_{-\infty}^{+\infty} f^2(x) dx}} \quad (1)$$

$$\text{Localization} = \frac{d |f'(0)|}{n_0 \sqrt{\int_{-w}^{+w} f'^2(x) dx}} \quad (2)$$

Using three criteria (maximizing both (1) and (4), and a single response for a single edge) applied to the task of edge detection under a step edge model, Canny derived a numerical solution to the optimal edge operator. Since this solution is not easy to manipulate geometrically, the first derivative of a Gaussian was suggested as a good approximation.

If we scale the operator f by ω :

$$f_\omega(x) = f(x/\omega),$$

the product of the SNR and the Localization is computed as

$$\text{SNR}_\omega \cdot \text{Localization}_\omega = \sqrt{\omega} \text{SNR} \cdot \frac{1}{\sqrt{\omega}} \text{Localization} = \text{SNR} \cdot \text{Localization}$$

Therefore, increasing the operator size results in better detection performance and poorer localization performance.

Canny suggested that a spatial operator elongated along the edge direction can improve both detection and localization to overcome the trade-off limitation.

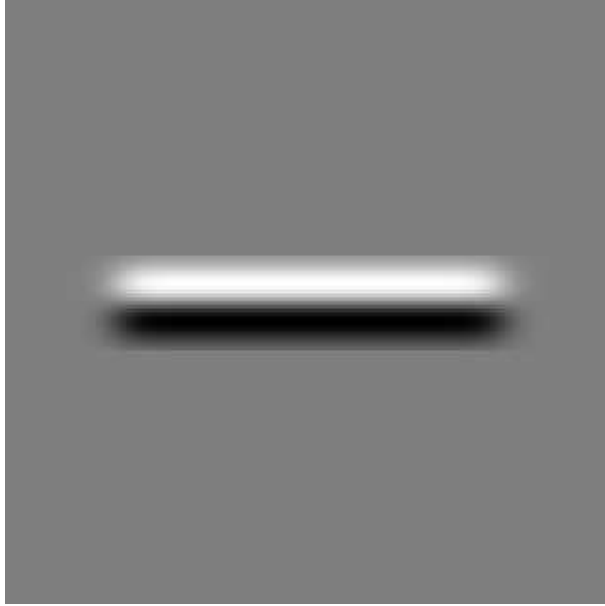


Fig. 1. Elongated edge mask: Elongated derivative-of-Gaussian edge operator used to collect edge responses from a side of a vehicle image

Given the spatial operator $f(x, y)$, if we scale it along the y direction:

$$f_l(x, y) = f(x, y/l)$$

we obtain $\text{SNR}_l = \sqrt{l}\text{SNR}$ and $\text{Localization}_l = \sqrt{l}\text{Localization}$; we have improvement in both terms.

We exploited this part of Canny's work in our application. For general edge detection applications, the width of the edge operator should be limited, so it can detect edge pixels correctly along possibly complicated edge contours. In our application, however, the vehicle model is rectangular, so we can employ an elongated edge mask, which is shown in Figure 1, to improve both detection and localization performance.

2.4 *The vehicle model and the operator for detection*

As mentioned previously, we assume that a vehicle in a parking lot gives rise to a two-dimensional rectangular shape in an aerial image. Since images can be taken from directions other than vertical, this model should be generalized to a parallelogram using orthogonal projection assumptions, which is reasonable for aerial images. Since it is hard to take all the possible vehicle shapes and colors into account, our algorithm depends primarily on the grey level difference along the vehicle's boundary, assuming that it is approximately a rectangle.

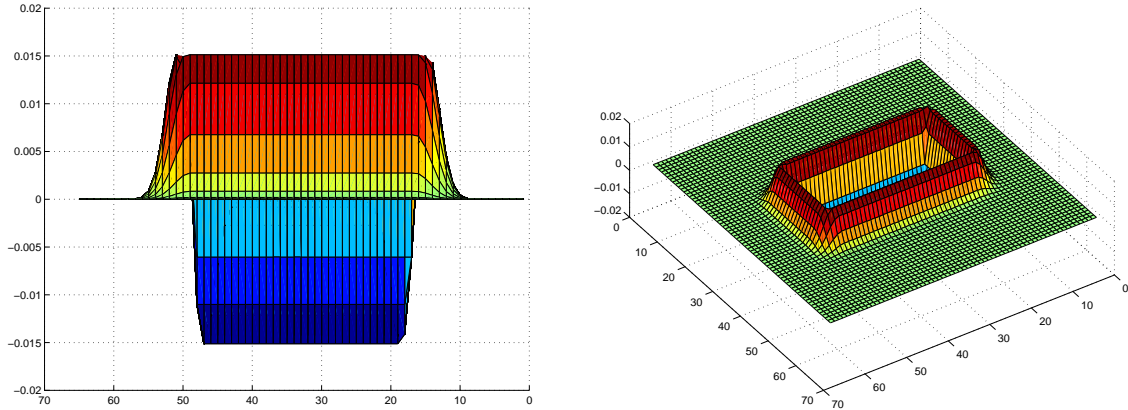


Fig. 2. Vehicle operator: The vehicle operator constructed using four elongated edge operators (as shown in Figure 1)

This simple model can pose serious problems when the image is taken with adverse camera and/or illumination parameters, as will be shown in subsequent sections. Moreover, the inner details of vehicles, which are not accounted for in our model, can affect detection and localization, even in a favorable environment.

Our approach uses a vehicle model similar to that in [9]; we have found that our approach is more robust with respect to noise. Edge detection, being a local operation, yields noisy results, and potentially useful edge information can be lost due to the thresholding of individual edge responses. Since our vehicle mask is a global operator, it has a much greater SNR, and all the edge information around the hypothetical vehicle boundary is preserved until the final decision is made.

The mask used for detecting vehicles consists of four elongated operators with first-derivative-of-Gaussian cross-sections. Each operator corresponds to one of the four sides of the vehicle. Responses are collected at the center of the set of operators, which corresponds to the vehicle center. By combining four operators into a rectangle, the steps of applying edge detection and a generalized Hough transform for finding rectangles are combined into a single process. The operator for detecting a vehicle is illustrated in Figure 2.

Parking lot scenes can have different camera parameters and vehicles can have different sizes. The detection algorithm should therefore try every possible hypothesis about the camera angle and the vehicle size if there is no prior information. Vehicle operators having given ranges of geometric parameters are generated and applied at each pixel of the image. The hypothesis giving maximum response is accepted and the corresponding response is registered at the pixel as the measure of the likelihood of a vehicle being present with the chosen parameters. If we know the value or range of values of a parameter,

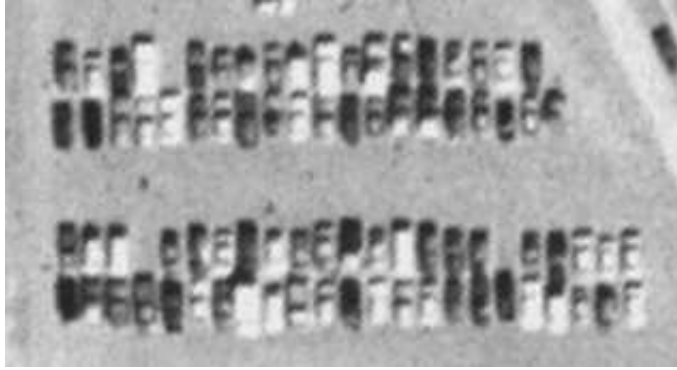


Fig. 3. Parking lot image

fewer hypotheses need to be tried. This operation is carried out at every pixel in the region of interest, or in the whole image, if no regions of interest are given. A typical response image after this step is shown in Figure 4.

Two thresholds are used to remove spurious responses and to declare that a vehicle has been detected: If we have an empty parking lot region next to a high-contrast vehicle, the response from one side of that vehicle is high enough at the empty spot to be accepted as indicating the presence of a vehicle. We filter out this kind of response by removing candidates that don't have enough votes from all four sides. This gives rise to our first threshold. This threshold should depend on scene factors such as noise level and contrast; however, we found that this parameter was not very sensitive, and it was fixed during most of our experiments.

If the sum of the responses of the four operators at a given point is above a certain value, we suspect that a vehicle may be present at that point and put it on the list of possible vehicle centroids. This step gives our second threshold.

Finally, since there can be spurious responses due to windshields or shadows, candidates that are inside other stronger candidates are discarded.

2.5 Algorithm output

We initially tested our algorithm on Fort Hood images [14]. Figure 3 shows a parking lot image and Figure 4 shows the responses of the algorithm to that image. In this example the algorithm was given information about the orientation of the parking lot. Darker pixels show positions where there is a greater likelihood of the presence of a vehicle. Comparing the image with the responses, we notice that most of the true vehicle centers give very high responses that survive after applying the second threshold.

Inevitably, higher-contrast vehicles, which are much brighter or darker than

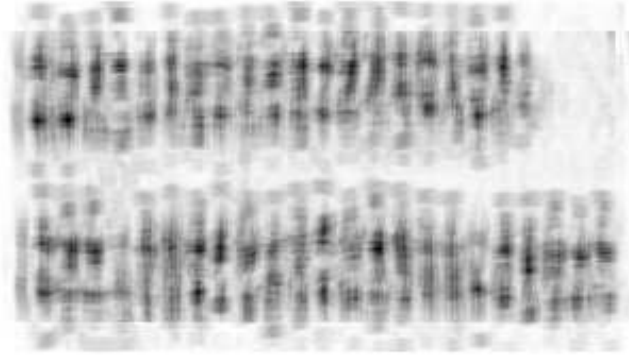


Fig. 4. Response image

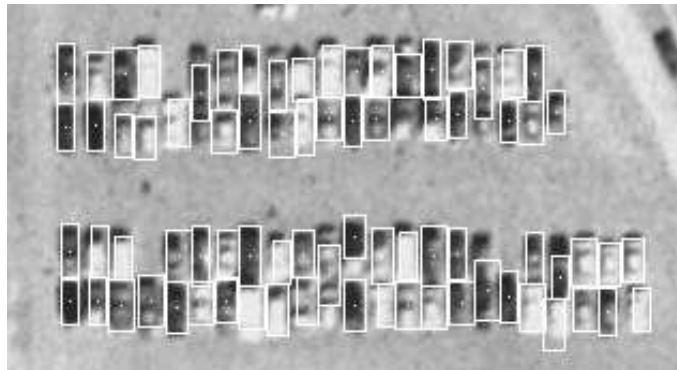


Fig. 5. Detected vehicles

the background, give higher responses. We can thus have more confidence as to the presence of a vehicle at such points. On the other hand, vehicles with smaller grey level differences from the background give low responses, due not only to the low contrast, but also to spurious responses from more prominent windshields. Low-contrast vehicles are also more susceptible to shadows, since the contrast of a shadow can dominate an otherwise weak response.

Figure 5 shows the final output of the vehicle detector, which gives the most probable vehicle dimensions as well as the locations of the centroids. The high-contrast vehicles again give more accurate estimates of the dimensions and centroids. The effect of contrast on the centroid estimation is not readily visible, and will be verified in a later section.

3 Characterization of detection and localization performance

3.1 Response profile

The process of applying operators having different geometric parameters can be viewed as one of parameter estimation. Let X , Θ , and S denote the posi-

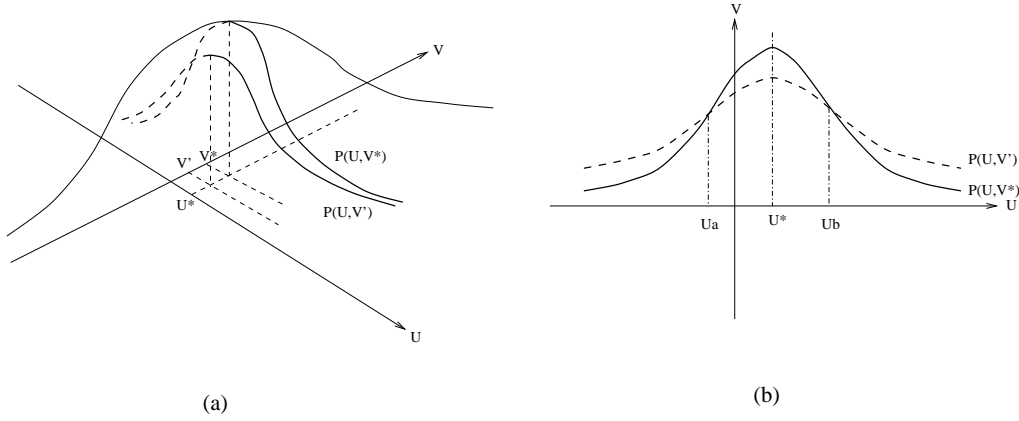


Fig. 6. Response profile and local linearity

tion, orientation, and scale parameters, respectively. Let $f(X, \Theta, S)(\cdot)$ be the operator having these parameters, and I be the image. The estimate is given by

$$(X^*, \Theta^*, S^*) = \arg \max_{(X, \Theta, S)} f(X, \Theta, S)(I)$$

If, for example, the orientation and scale parameters are known to be (Θ_0, S_0) , then the location estimate is

$$X^* = \arg \max_X f(X, \Theta_0, S_0)(I)$$

and the operator $f(X, \Theta_0, S_0)(\cdot)$ is a convolution.

Since the location parameter X is usually more informative than the other parameters, the two-dimensional response profile

$$P(f, I)(X) = \max_{(\Theta, S)} f(X, \Theta, S)(I)$$

is stored and used to characterize the localization performance.

We have found that the response profile $P(f, I)$, with or without prior information about (Θ, S) , is the same as the convolution response with the correctly matched operator.

$$\begin{aligned} P(f, I)(X) &= \max_{(\Theta, S)} f(X, \Theta, S)(I) \\ &= \max_{\Theta} f(X, \Theta, S_0)(I) \\ &= \max_S f(X, \Theta_0, S)(I) \\ &= f(X, \Theta_0, S_0)(I) \end{aligned}$$

for $X \in N(X_0)$

where $N(X_0)$ is a small neighborhood around the true centroid X_0 .

This local property doesn't hold strictly in the continuous domain; nevertheless, the response profile is usually well approximated locally by the spatial convolution. It holds in the discrete domain due to the quantization of the parameter values. Figure 6 shows this effect. Suppose that the response of a filter is given in terms of two parameters, U and V , as in Figure 6(a). The parameters (U^*, V^*) that give the maximum of $P(U, V)$, are chosen as the correct parameters. Consider the situations in which the correct value V^* is and is not known. Assume that we are only interested in computing the response profile for the parameter U . When we know the correct parameter value, the response profile for U is $P(U, V^*)$ (Figure 6(b)). If, on the other hand, we have no information about V , the response profile is computed as $\bar{P}(U) = \max_V P(U, V)$. Since the correct parameters are (U^*, V^*) , we have $\bar{P}(U^*) = P(U^*, V^*)$. If we move away from U^* , the function $\bar{P}(U)$ may pick up values from the other parameter value V . Let V' be the quantized parameter value closest to V^* . Then as in Figure 6(b), the profile function $P(U, V')$ is always less than or equal to $P(U, V^*)$ on some neighborhood $[U_a, U_b]$ of U^* , by the continuity of P . This should hold for other values of V . In other words, the response profile $\bar{P}(U) = \max_V P(U, V)$, without any prior information about V , always picks up values from $P(U, V^*)$, locally on $[U_a, U_b]$.

In Figure 7 the responses of the vehicle operator are shown for different situations. In the left graph, the operator size, shape and orientation are fixed; in the middle, different sizes and shapes are tested; in the right, operators with different orientations are also tried. We can observe that the responses near the peaks are identical. Note also that when we have less prior information, the responses decrease slowly as we move away from the centroid, as pointed out above.

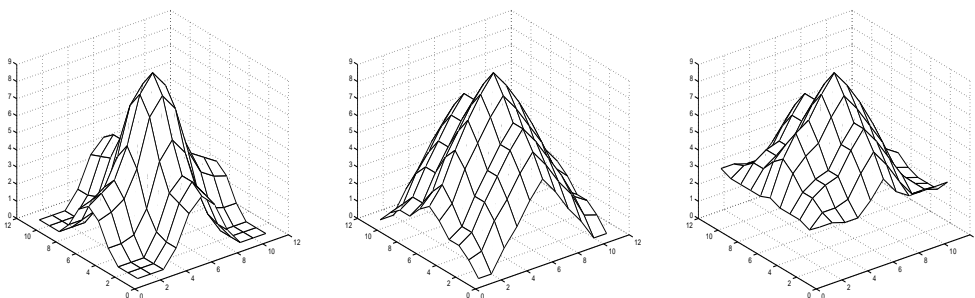


Fig. 7. Response profiles with different degree of prior information. Left: Fixed size and orientation. Middle: Varied size, fixed orientation. Right: Varied size and orientation.

This observation is very helpful, since the convolution profile is relatively simple to formulate because of its linearity and the known geometry of the op-

erator, while other response profiles are much more complicated. This local property is sufficient to get the probability density function of the response; even with a high level of Gaussian noise, the maximum response usually remains within a few pixels of the true centroid.

It is easy to compute the convolution profile in the spatial domain. The response profile $r(\mathbf{x})$, when the true position of the object centroid is the origin and the operator is positioned at \mathbf{x} , is given by

$$r(\mathbf{x}) = \int f(\mathbf{x} - \mathbf{u})I(\mathbf{u})d\mathbf{u}$$

where f is the vehicle operator and I is the ideal vehicle image. Since f consists of two pairs of elongated Gaussian operators, r is also the sum of the responses of I with four elongated operators.

3.2 Localization and detection performance

We are now able to formulate some of the statistical properties of the vehicle detection process — in particular, its detection probability and localization error. We assume that the errors can be modeled as additive Gaussian iid noise.

We can compute the probability density function of the responses at points around the true centroid. Since the filtering is locally linear, the responses are correlated Gaussian, and we can get the covariance matrix using the convolution profile. Let $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N)$ be points around the true centroid, including the centroid; $\underline{y} = (y_1, y_2, \dots, y_N)$ be the corresponding responses; and Σ be the covariance matrix. The ideal response profile $\underline{r}(\mathbf{x}) = (r(\mathbf{x}_1), r(\mathbf{x}_2), \dots, r(\mathbf{x}_N))$ has been calculated above.

The pdf of \underline{y} is given by

$$p_{\text{response}}(\underline{y}) = \frac{1}{(2\pi)^{\frac{N}{2}} \det \Sigma} \exp\left(-\frac{1}{2}(\underline{y} - \underline{r}(\mathbf{x}))^T \Sigma^{-1}(\underline{y} - \underline{r}(\mathbf{x}))\right) \quad (3)$$

The probability that the maximum occurs at position x , which gives the localization distribution, is given by

$$\begin{aligned} P_{\mathbf{X}_{max}}(\mathbf{x}_i) &= \text{Prob}(\text{maximum occurs at } \mathbf{x}_i) \\ &= \text{Prob}(y_1 \leq y_i, \dots, y_N \leq y_i) \end{aligned}$$

$$= \int \int^{y_i} \cdots \int^{y_i} P(y_1, \cdots, y_N) dy_N \cdots dy_1 dy_i$$

Note that the maximum response which is compared to the threshold is the largest order statistic $y_{(N)}$ from y_1, y_2, \dots, y_N , and the corresponding position $\mathbf{x}_{cent} = \mathbf{x}_i$ is marked as the centroid. The pdf of the maximum response is given by

$$\begin{aligned} P_{Y_{max}}(y) &= \sum_{i=1}^N P(y | \mathbf{X}_{max} = x_i) P_{\mathbf{X}_{max}}(\mathbf{x}_i) \\ &= \sum_{i=1}^N P_{y_i}(y) \int \int^{y_i} \cdots \int^{y_i} P(y_1, \cdots, y_N) dy_N \cdots dy_1 dy_i \end{aligned}$$

where $P_{y_i}(y)$ is the marginal distribution of y_i .

Based on the above, it is straightforward to predict the localization and detection performances under ideal assumptions: the operator is matched to the vehicle size, a step edge along the vehicle boundary, and the noise is iid Gaussian. The localization performance is represented by the pdf of the centroid estimate, $p_{centroid}$. The probability that the maximum response is less than the given threshold is the estimate of misdetection probability.

There is also the issue of the relationship between detection and localization, as in one dimension. We can show that the performance discussed in Section 2.2 also holds for the general case. This trade-off between detection and localization performance gives us a tool for optimizing the operator for a given performance requirement. For example, if the application demands accurate localization, we can make the operator narrower and sacrifice detection performance to some extent.

We conducted experiments to verify the derived theoretical performances. First, the probability distribution $p_{centroid}$ of the position estimate is computed by a randomized numerical integration algorithm [13] according to the above formulation. It is compared with empirical distributions obtained by the following experiments.

A rectangular shape with constant grey level 120 against a background with grey level 100 was generated. The easily available empirical response profile $r(\mathbf{x})$ was used to compute (3) and (4). This empirical profile is merely the response of the vehicle operator to the generated model vehicle image. It is perturbed by additive i.i.d. Gaussian noise with variance σ^2 . We generated many instances of images for a given σ and ran the detection algorithm on them to get the distributions of the centroid and maximum response. We tested three different noise levels: $\sigma = 12$, $\sigma = 16$, and $\sigma = 20$. 100,000 instances of

pixel number	33	34	35	36	37
$\sigma = 12$ theoretical	0.00001	0.04416	0.91166	0.04416	0.00001
$\sigma = 12$ empirical	0.00001	0.04329	0.91246	0.04424	0.00000
$\sigma = 16$ theoretical	0.00051	0.09781	0.80336	0.09781	0.00051
$\sigma = 16$ empirical	0.00047	0.09732	0.81841	0.09803	0.00044
$\sigma = 20$ theoretical	0.00330	0.14343	0.70654	0.14343	0.00330
$\sigma = 20$ empirical	0.00292	0.14337	0.70645	0.14242	0.00325

Table 1

Theoretical and empirical distributions of the centroid: The one-dimensional distribution of vehicle centroids is compared with the empirical histogram around the true centroid (pixel no. 35) using different operator widths ($\sigma = 12, 16, 20$)

perturbed vehicle images were used for each noise level. The two-dimensional empirical and theoretical distributions of the centroid of the rectangle around the true centroid ($\sigma = 20$) are compared in Figure 8. The empirical distribution of the centroid along the vehicle length direction and the corresponding theoretical distribution are summarized in Table 1; they closely match.

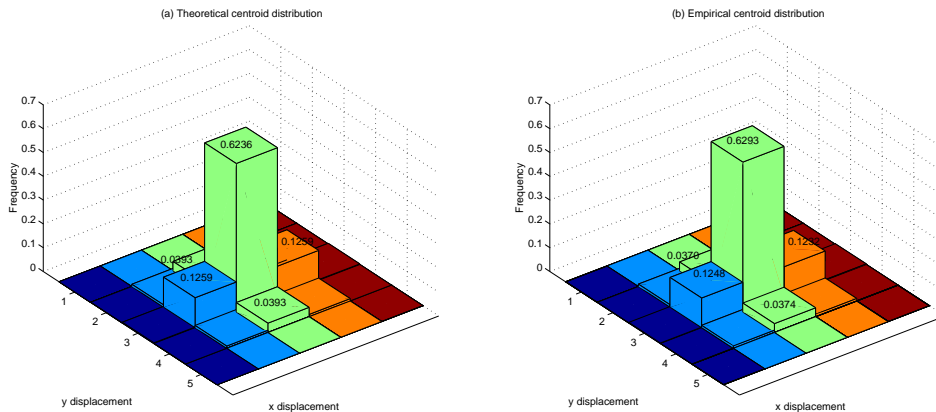


Fig. 8. Two-dimensional theoretical and empirical distributions of the centroid estimate. Left: Centroid distribution computed using equation (?). Right: Empirical centroid histogram.

4 Performance Evaluation Using a Parking Lot Scene Simulator

4.1 Simulated responses

As mentioned in the introduction, ideal performance evaluation requires all possible images to be tested. Even when there is a good analytical tool to predict it, performance measurement is not complete without extensive testing

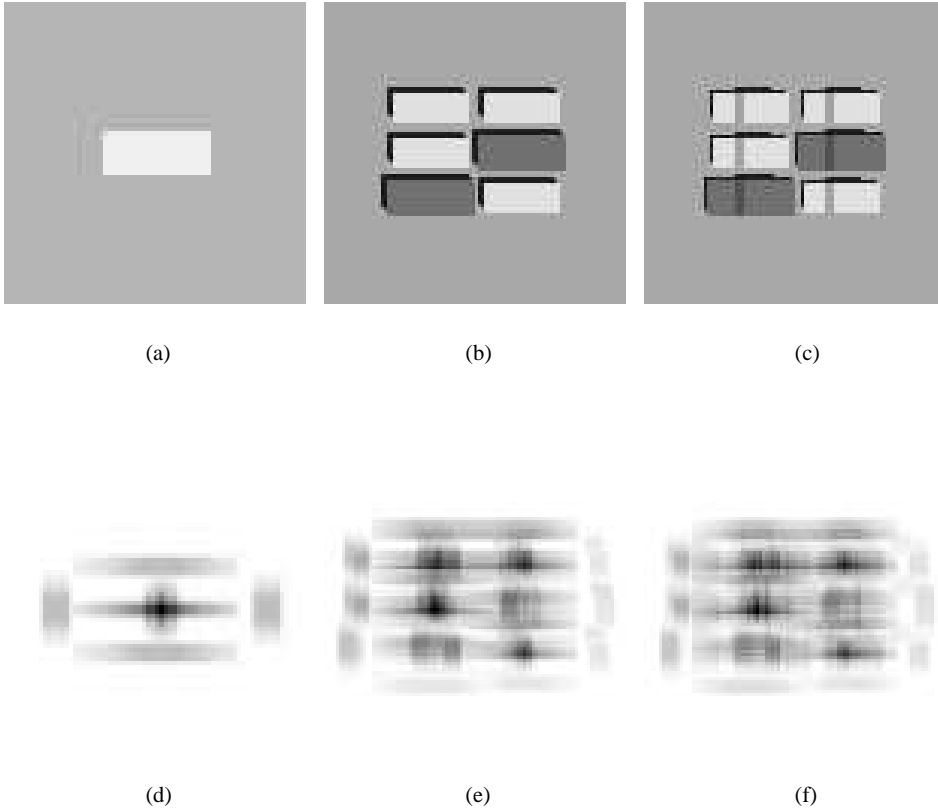


Fig. 9. Simulated vehicles and responses. Vehicle images generated by parking lot scene simulator: Model vehicle image(a) and the corresponding response(d); multiple vehicles with shadows(b) and response image(e); and multiple vehicles with windshields(c) and response image(f).

on images. As is the case in most domains, though, it would be very expensive to acquire large numbers of aerial images under varying conditions. Our parking lot image simulator allows us to manipulate not only noise and camera and illumination parameters, but also the sizes and colors of the vehicles and how they are positioned in the parking lot. It is important for performance characterization to verify whether or not the latter properties can significantly affect the performance.

Figures 9(a), 9(b) and 9(c) are images generated by our simulator. In Figures 9(a) and 9(b), the vehicles satisfy the simple rectangular box model employed in our vehicle detector, whereas Figure 9(c) uses more realistic vehicle shapes. Both images have two kinds of vehicles having different dimensions and colors.

Since the vehicle in Figure 9(a) satisfies our model exactly, the corresponding response image (Figure 9(d)) is the ideal output profile. Figures 9(e) and 9(f) are response images for Figures 9(b) and 9(c); the response patterns become more irregular from Figure 9(d) to Figure 9(e) to Figure 9(f).

It is interesting to compare Figure 9(f) with the real data in Figure 4. They

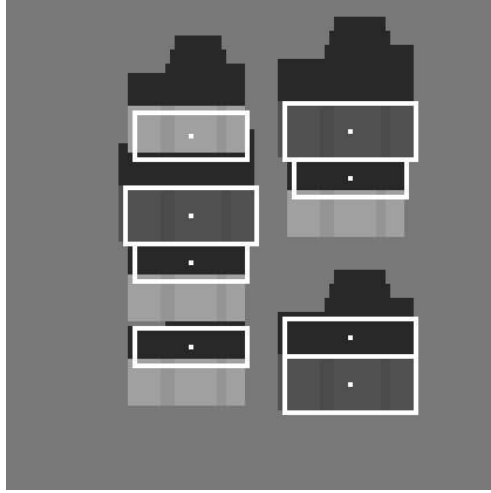


Fig. 10. Scene with long shadows. Since shadows typically have higher contrasts with the background than vehicles, they can give rise to serious false alarms.

have similar lighting conditions, and the response patterns look similar. The real image has some grainy noise, but its effect seems to be smoothed out by the Gaussian masking. The response pattern is more noticeably affected by the more structured responses from windshields and shadows, which can survive convolution with the operator.

4.2 Camera angle and illumination

Experiments with simulated illumination conditions showed that long shadows give large numbers of false alarms that are difficult to identify, since illumination perpendicular to the vehicles produces shadows having sizes close to that of the vehicles and contrast with the background higher than that of some of the vehicles. Figure 10 shows this effect.

Our vehicle model assumes that images are taken from close to the nadir view and doesn't take into account the three-dimensional structure of a vehicle. This two-dimensional model collapses as the depression angle of the camera decreases significantly from 90 degrees (Figure 11).

4.3 Ranges of centroid errors with camera and illumination parameters

We conducted another experiment to show how the centroid estimate is perturbed by varying lighting and camera parameters. There would be no serious localization error if our model were perfect. This experiment serves not only as a diagnostic of the current algorithm but also as a tool to verify whether an improvement in the model would produce better performance.

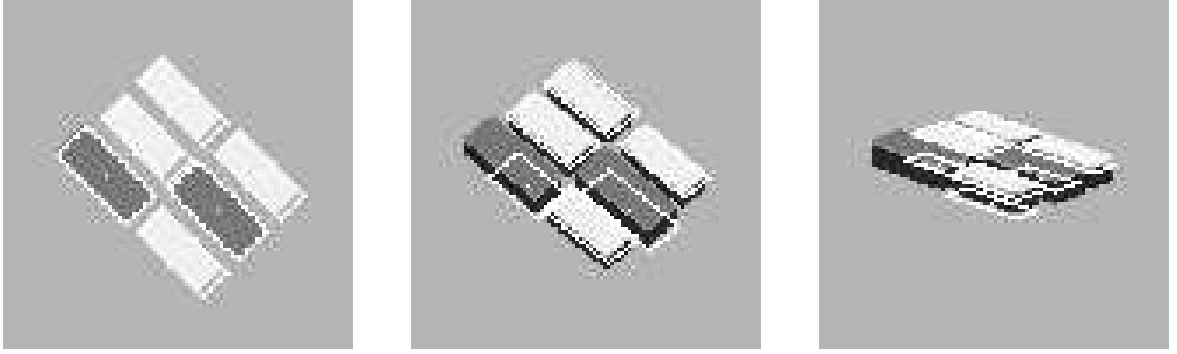


Fig. 11. Scenes obtained from different oblique angles. As the camera angle deviates from the (vertical camera) model, false detections due to the three-dimensional structure of the vehicle appear.

We generated 256 images of a single vehicle with different camera and illumination parameters covering four horizontal angles ($0^\circ, 45^\circ, 90^\circ, 135^\circ$) and four vertical angles ($15^\circ, 40^\circ, 65^\circ, 90^\circ$) (Figure 12). The horizontal angles were chosen from only half of the range 0° to 360° to take advantage of symmetry.

Figure 13 is the output, which shows that the centroid estimates have a sharp peak at the true center, but some estimates are outside the vehicle.

5 Experiments on Ground-Truthed Data

5.1 Contribution of prior information

We ran our algorithm on a large set of vehicle images to investigate its performance on real images. Our data consisted of 15 images from the Fort Hood Image Set [14]. The corresponding ground truth images [14] were used to facilitate the process of counting the vehicles and computing the localization errors. Each parking lot and road on the images was assigned one of the orientations $0^\circ, 45^\circ, 90^\circ, or 135^\circ$.

To compare the performances of the algorithm when varying amounts of prior information are available, the roads and parking lots were annotated with their orientations.

Since we have ground truth images with marked vehicles, we can evaluate detection and localization performance by comparing the detected vehicles with the ground truth vehicles. A target vehicle (TV) is a ground truth vehicle in an image, and a detected vehicle (DV) is a vehicle detected by the algorithm. If there is any DV centroid within five pixels of a TV centroid, we claim a

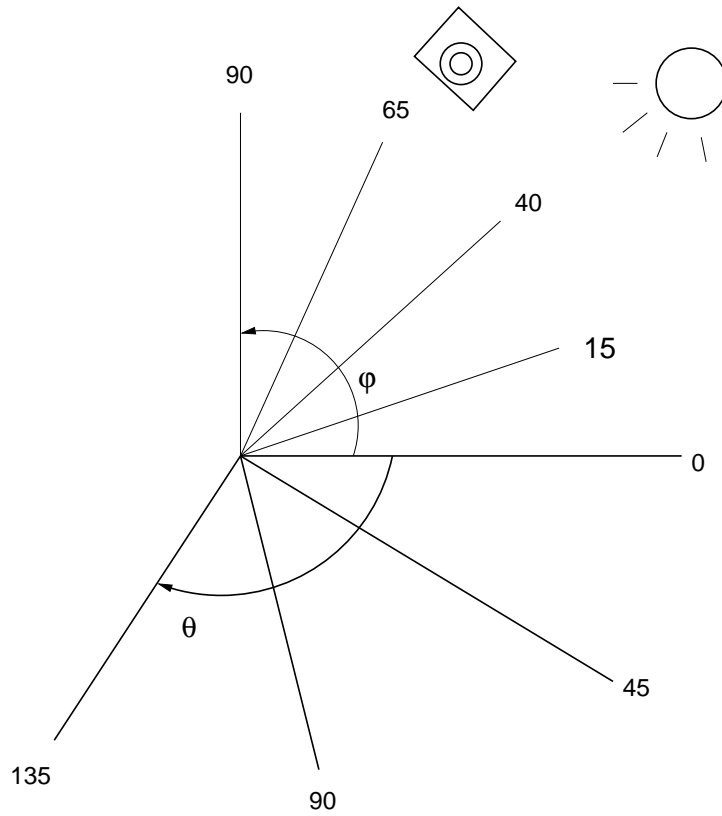


Fig. 12. Horizontal (θ) and vertical (ϕ) angles for camera and illumination.

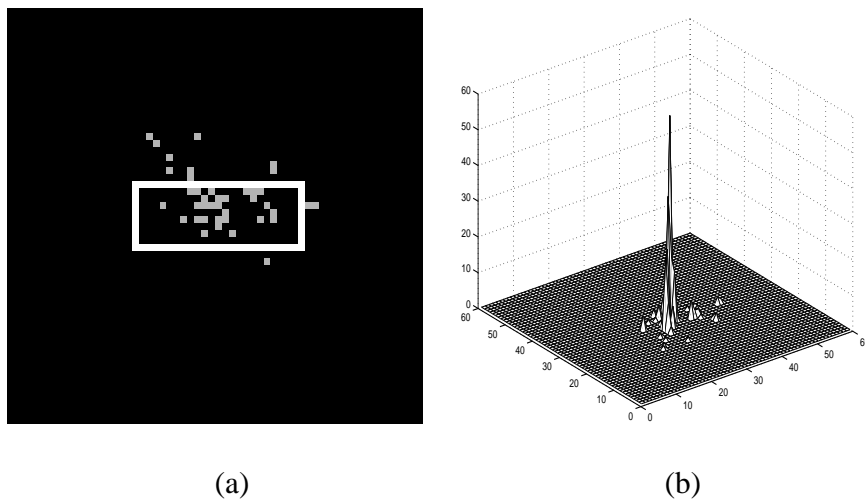


Fig. 13. Centroid perturbation due to illumination and camera angle changes
 detected target vehicle (DTV). If any TV is missing from the DTVs, it is
 declared as a missed detection (MD). A false alarm (FA) is a detected vehicle

which is not a DTV. A DV which belongs to some DTV is called a correctly detected vehicle (CDD). Evidently

$$TV = DTV + MD$$

$$DV = CDD + FA$$

We first tested our algorithm on one image; the output is shown in Figure 14. In this case, site information was used: the vehicle operator was applied only to regions where vehicles were expected to be present, and according to the annotated parking lot orientation. For this image, the results were $TV = 566$, $MD = 100$, $DV = 569$; and $FA = 83$, hence the missed detection rate was 17.7% and the false alarm rate was 14.6%. Figure 15 shows ROC plots of correct detection rate versus false alarm rate for this image. The curves show erratic behavior at both ends due to the removal of spurious responses: some candidate vehicles that survived the threshold were removed if they were inside more prominent vehicles. The top curve shows the performance of the algorithm when we have orientation information. When we do not have this information, but only know where the parking lots are, the performance deteriorated, as expected, as shown in the middle curve. When we have no prior information, the algorithm must look for a vehicle at every pixel in the image and try every possible orientation; its performance is then much worse, as shown in the bottom curve.

The utility of prior information for detection and localization performance was tested using all 15 images in the database. The results are summarized in Table 2. It appears that prior information does not improve localization performance.

We suspect that the degradation of detection performance is due mainly to spurious responses from neighboring vehicles. If the operator orientation is matched to the true orientation of the vehicles, the responses are highly accumulated at the true centroids of the vehicles. If, on the other hand, the algorithm tries every possible orientation, the operator picks up responses from the vehicle boundary and also possibly from inner intensity changes of adjacent vehicles. As illustrated in Figure 7, the simulated response profile without orientation information has a less sharp peak than the response profile with prior orientation information. However, it was observed that in each case the response profiles close to the centroid were identical. This explains why localization works no better when we have prior information. We can conjecture that the slightly better localization performance for the no-prior case, as opposed to the detection performance, is due to the fact that only prominent vehicles giving high responses survive the thresholding stage.



Fig. 14. Parking lot image and detected vehicles. Parking lot orientation information is incorporated in this experiment.

6 Bootstrap Diagnosis of Detection Performance

Since our algorithm depends only on the grey level difference between the vehicle and the background, some patterns on the parking lot such as dividing lines and oil spots give rise to false alarms. If we could design a tool to take into account the structural information of a vehicle image, it could comple-

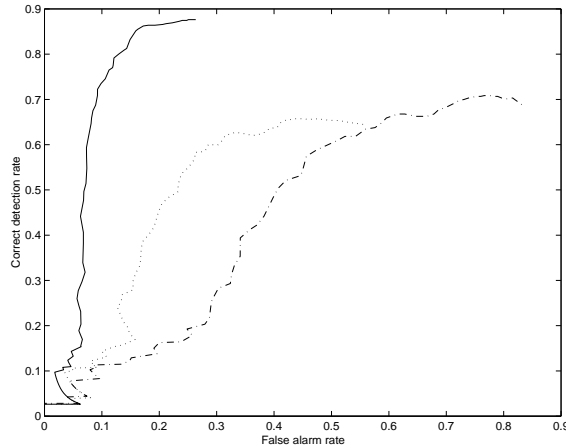


Fig. 15. ROC plots of detection performance. Top curve: parking lot orientation information provided. Middle curve: region of interest (parking lot area) provided. Bottom curve: no prior information provided.

Mask orientation	without prior	with prior
Missed detection (%) (MD/TV)	36.3 (1843/5073)	20.1 (1021/5073)
False alarm (%) (FA/DV)	54.6 (4077/7469)	21.8 (1176/5405)
Localization error (pixels)	1.963	2.109

Table 2
Detection and localization performances

ment our edge-based method. It would also be helpful, from the viewpoint of performance characterization, to have a statistical confidence measure for the detection.

We applied an empirical bootstrap method to produce statistical assessments of our estimates. Using bootstrap to characterize the performance of edge detection has been discussed in [16]; we adopted a similar approach in our work. Our method is designed in the form of statistical hypothesis testing. Let us assume that the algorithm has declared a vehicle present in the image with estimated centroid coordinates and dimensions. We would like to have a confidence measure of how reliable this decision is. If it turns out to be unreliable with respect to some criterion, we reject the decision. The null hypothesis is chosen to be the one favoring the original decision.

Assuming that the null hypothesis is correct, any perturbation of the image which respects the vehicle hypothesis should give estimates of location and dimensions which are very close to the original ones. Perturbation which preserves the structure of the null hypothesis is done by separately switching

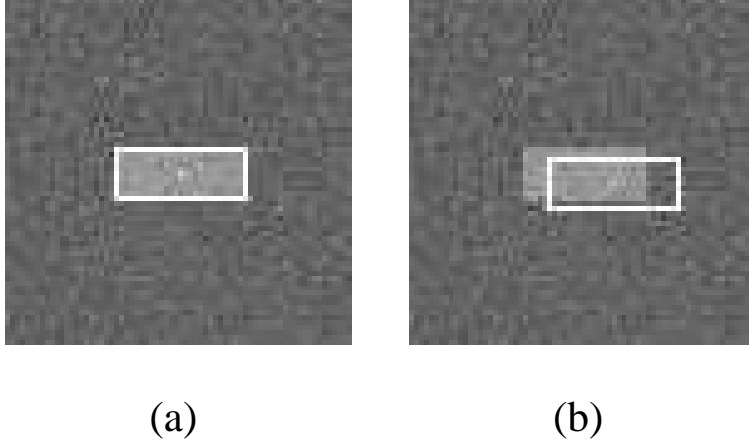


Fig. 16. Images with correct and wrong vehicle hypotheses

every pixel inside and outside the hypothetical vehicle. The new position of the given pixel is chosen randomly with replacement. The detection algorithm is applied to this image and new estimates are computed.

We repeat this process a designated number of times and compute the required statistic for testing: the variance of the centroid estimates. A small variance implies that the location and dimension estimates are coherent over the switching of pixels; the vehicle hypothesis is correct. If the variance is bigger than the threshold, we have to reject the null hypothesis. We compute the distribution of the statistic under the null hypothesis and test the hypothesis using the following bootstrap scheme.

Let the test statistic be $T = T(X^{*1}, \dots, X^{*b})$ where the X^{*i} 's are the perturbed images. Let T^* denote the bootstrapped test statistic. We compute the bootstrap replications T^{*1}, \dots, T^{*B} by generating $b \cdot B$ perturbed images

$$X^{*1}, \dots, X^{*b}, X^{*(b+1)}, \dots, X^{*(2b)}, \dots, X^{*(B-1)b+1}, \dots, X^{*Bb}$$

After computing T^{*1}, \dots, T^{*B} , we can get the threshold t for the decision by

$$\alpha = \text{Prob}_{\text{Null}}(T^* \geq t) = \#\{T^* \geq t\}/B$$

where $1 - \alpha$ is the confidence level.

We experimented in an artificial domain using images such as those shown in Figure 16, using the centroid covariance as the test statistic. Figure 16(a) shows an image with a correct vehicle hypothesis and Figure 16(b) is the same image with an intentionally perturbed vehicle hypothesis. The first image is used to generate the bootstrapped distribution of T^* . The numbers of bootstrap samples b and B were chosen empirically as $b = 20$ and $B = 150$.

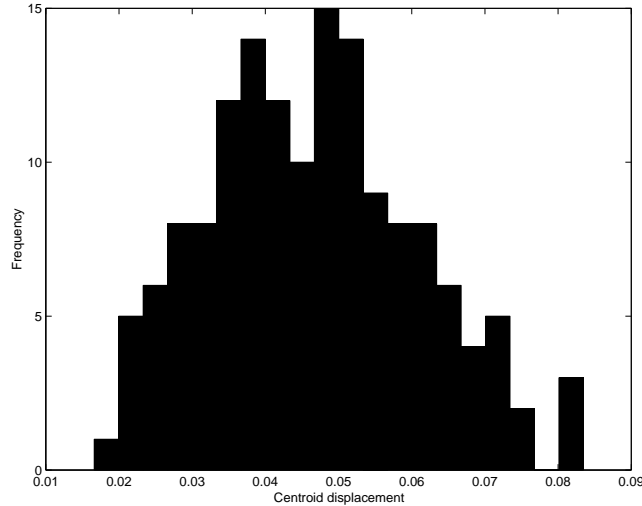


Fig. 17. Distribution of the centroid variance under the correct hypothesis. The test statistic is compared with the percentile of this distribution to decide whether the hypothesis is correct or not.

Vehicle number	0	1	2	3	4	5	6	7	8
Centroid variance	0.05	0.24	1.53	0.62	3.62	10.3	6.09	4.18	12.4

Table 3

Centroid variances for each hypothesis: We discard hypothetical vehicles with large centroid variances (vehicles 4,5,6,7,8).

The empirical distribution is shown in Figure 17. The threshold corresponding to level $\alpha = 0.05$ is found to be around 0.071. The hypothesis in Figure 16(b) is discarded by this threshold.

We tested this method in the real domain using a small image chip from the Ft. Hood Image Set, shown in Figure 18(a). A total of nine vehicles were detected as shown in Figure 18(b), marked with hypotheses numbered from 0 to 8. We applied the method twenty times; Figure 18(c) is one instance of a bootstrapped image. We kept track of each vehicle hypothesis in Figure 18(b); the final centroid standard deviations are given in Table 3. The correct hypotheses (vehicles 0 to 3) have small values compared to values from wrong hypotheses (vehicles 4 to 8). This experiment again reveals the weakness of our vehicle model, since the vehicles with multiple grey levels (vehicles 2 and 3), which are not accounted for in our simple rectangle model, have relatively large variances compared with vehicles 0 and 1 which are almost single-tone.

Determining the threshold for the decision is not easy, since the empirical distribution of variance depends on the grey level difference between the vehicle and the background. Vehicles with multiple tones should also be accounted for, in order for the assessment to be precise. These issues will be the subject of future work.

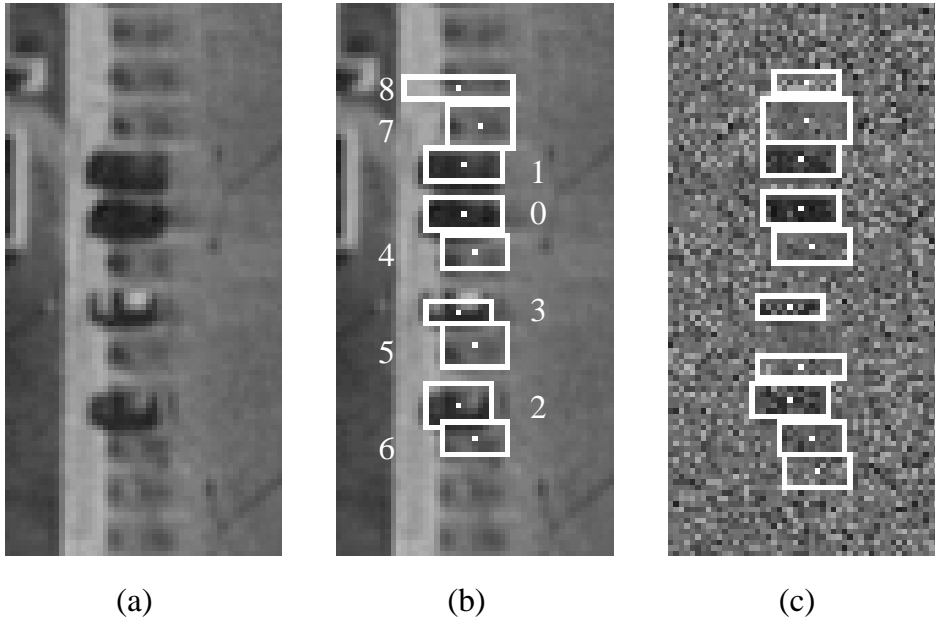


Fig. 18. Bootstrapping with a real image. (a) Cluttered parking lot image. (b) Initial vehicle detection result. (c) True vehicles show more consistent detection after bootstrapped perturbation.

7 Conclusion

The performance of a simple model-based vehicle detection algorithm under a wide range of operating environments has been investigated using mathematical analysis and empirical evaluation. It was verified that low illumination and/or acquisition angle contribute to poor detection and localization, and that site information can improve them. We derived statistical properties of the detection and localization performances in the presence of noise. The trade-off between detection and localization performances can be exploited to meet the requirements of given applications. Since our vehicle operator can be generalized for detecting shapes other than rectangles, this framework can be directly applied to more general shape detection problems. Through our analysis, the use of a more sophisticated vehicle model was suggested for performance improvement in adverse environments. Statistical validation using bootstrap was found to be effective in removing false alarms. However, since this method also depends on the vehicle model, further improvement can be made by using more sophisticated models.

References

- [1] R. Nevatia and K.R. Babu, "Linear feature extraction and description," *Computer Graphics and Image Processing*, Vol. 13, pp. 257-269, 1980.

- [2] A. Banerjee, P. Burlina, and R. Chellappa, "Adaptive target detection in foliage-penetrating SAR images using alpha-stable models," *IEEE Transactions on Image Processing*, Vol. 8, pp. 1823-1831, 1999.
- [3] M. Betke, E. Haritaoglu, and L.S. Davis, "Real-time multiple vehicle detection and tracking from a moving vehicle," *Machine Vision and Applications*, Vol. 12, pp. 69-83, 2000.
- [4] P. Burlina, V. Parameswaran, and R. Chellappa, "Sensitivity analysis and learning strategies for context-based vehicle detection algorithms," *Proceedings of the 1997 DARPA Image Understanding Workshop*, pp.577-584, 1997.
- [5] P. Burlina, R. Chellappa, and C.L. Lin, "A spectral attentional mechanism tuned to object configurations," *IEEE Transactions on Image Processing*, Vol. 6, pp. 1117-1128, 1997.
- [6] R. Chellappa, P. Burlina, L.S. Davis, and A. Rosenfeld, "SAR/EO vehicular activity analysis system guided by temporal and contextual information," *Proceedings of the 1994 ARPA Image Understanding Workshop*, pp. 615-620, 1998.
- [7] J. Canny, "Finding edges and lines in images," MIT AI TR-720, 1983.
- [8] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 8, pp.679-698, 1986.
- [9] R. Chellappa, et al., "An integrated system for site model supported monitoring of transportation activities in aerial images," *Proceedings of the 1996 ARPA Image Understanding Workshop*, pp. 275-304, 1996.
- [10] R. Chellappa et al., "Site model supported monitoring of aerial images," *Proceedings of CVPR*, pp. 694-699, 1994.
- [11] J.M. Ferryman, A.D. Worrall, G.D. Sullivan, and K.D. Baker, "A generic deformable model for vehicle recognition" *Proceedings of the British Machine Vision Conference*, pp. 127-136, 1995.
- [12] W. Foerstner, "10 pros and cons against performance characterization of vision algorithms," *Proceedings of the Workshop on Performance Characterization of Vision Algorithms*, Cambridge, MA, 1996.
- [13] G. P. Lepage, "VEGAS: An adaptive multidimensional integration program," Publication CLNS-80/447, Cornell University, 1980.
- [14] G. Liu et al., "Ground-truthing the vehicles in the vertical-view Ft. Hood images," Tech. Rep., Dept. of Electrical Engineering, University of Washington, 1997.
- [15] R.M. Haralick, "Performance characterization protocol in computer vision," *Proceedings of the 1994 ARPA Image Understanding Workshop*, pp.667-673, 1994.

- [16] K. Cho, P. Meer, and J. Cabrera, "Performance assessment through Bootstrap," Tech. Rep., Dept. of Electrical and Computer Engineering, Rutgers University, 1995
- [17] H. Moon, R. Chellappa, and A. Rosenfeld, "Performance analysis of a simple vehicle detection algorithm," Proceedings of the Fedlab Symposium, pp. 249-253, 1999.
- [18] A. N. Rajagopalan, P. Burlina, and R. Chellappa. "Higher order statistical learning for vehicle detection in images." In Proceedings of the International Conference on Computer Vision, pp. 1204-1209, 1999.
- [19] R. Ruskone, L. Guigues, S. Airault, and O. Jamet, "Vehicle detection on aerial images: A structural approach," Proceedings of the International Conference on Pattern Recognition, pp. 900-904, 1996.