Car Detection in Low Resolution Aerial Image *

Tao Zhao   Ram Nevatia
University of Southern California
Institute for Robotics and Intelligent Systems
Los Angeles CA 90089-0273
{taozhao|nevatia}@iris.usc.edu

Abstract

We present a system to detect passenger cars in aerial images where cars appear as small objects. We pose this as a 3D object recognition problem to account for the variation in viewpoint and the shadow. We started from psychological tests to find important features for human detection of cars. Based on these observations, we selected the boundary of the car body, the boundary of the front windshield, and the shadow as the features. Some of these features are affected by the intensity of the car and whether or not there is a shadow along it. This information is represented in the structure of the Bayesian network that we use to integrate all features. Experiments show very promising results even on some very challenging images.

1 Introduction

Vehicle detection in aerial images has important civilian and military uses, such as traffic surveillance, both for traffic information system or to gather traffic statistics for urban planning. It can also produce strong evidence for road detection [10]. It also provides a good test domain for methods of object detection in difficult situation that require integration of multiple cues. The aerial images we used are grayscale images taken mostly from a vertical or slightly oblique viewpoint. The length of a typical car is around 26 pixels in image. The camera calibration is known as well as the sunlight direction.

Detection from aerial image is easier than from detection from an arbitrary viewpoint in that the viewpoint is constrained. However, it is still not as easy as it may seem to be. Example images are shown in Fig.1. The main difficulties lie in the following:

- Although the viewpoint is constrained, there are still variations that make the cars look different.
- The image resolution is low so not many details are visible.
- Some cars are heavily interfered with the environment in the images, mostly tree branches. (Fig.1.b)

- Cars can be of any intensity in the image, from the darkest to the lightest. And some cars' intensity is very close to the road.
- The image quality varies. The brightness, contrast and sharpness of the images change due to factors including illumination, focusing and atmospheric turbulence.
- The expected features of a car differ with its intensity and the existence of shadow. For a simple example, whether or not the boundary of a gray car can be detected depends heavily on its shadow. (See 4.1 for more detail.)

We need to account for all these difficulties to get a reasonable good system.
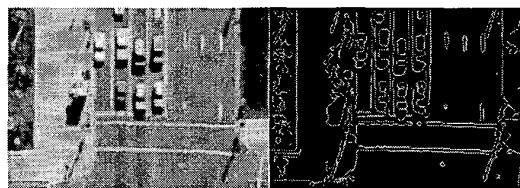
(a)
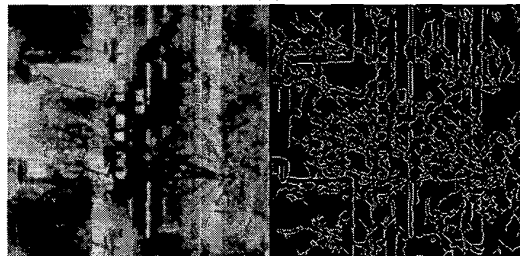
(b)

Figure 1: Two patches of images and their Canny results. (a) is clean but some edges are not all like our perception. (b) is heavily interfered with the tree branches and the cars are hardly visible in the edge map.

1.1 Related work

The detection of vehicles has been receiving attention in the computer vision community because vehicles are such a significant part of our life. A lot of car detection

710

work is done for a single viewpoint, such as [8] [11]. Some arbitrary viewpoint car detection [12] is done by detecting in a number of viewpoints and then combining the results.

There is not much literature on detecting vehicles in aerial images. In [1] [7], a vehicle is modeled as a rectangle of a range of sizes. Canny like edge detector is applied and GHT (Generalized Hough Transform) [1] or convolution with edge masks [7] are used to extract the four sides of the rectangular boundary. [5] uses average gray-level and average gradient level of the inside/outside/along the sides of the vehicle as features and learning the feature distribution for recognition. All of them treat vehicles as 2D objects and their primary evidence is the boundary of the car. This approach may be good for their data (Fort Hood, a military site) where the vehicles are mostly of dark color, but may have problems when applied on urban scenes and the performance may degenerate when the viewpoint changes. Besides, the shadow cue can not be utilized.

1.2 Our approach

We formulate the problem as a 3D object recognition problem to accommodate the change of viewpoint and make use of the shadow cues. A sketch of our approach is described below.

In this work, we are only interested in detecting cars aligned with road direction. First the directions of the roads are estimated by clustering straight lines in an image considering the fact that most lines in urban images are aligned with the direction of roads. From a psychological test we performed and analogue to aerial image building detection system [4], we decide to use four sides of the boundary of the car; four sides of the front windshield; two sides of outer boundary of the shadow and the intensity of the shadow area (when exist) as features (Fig.3.b). With a generic model of a car, the expected features are predicted. The image features are computed at each pixel and verified with the expected ones. We observe that some of the feature distributions are affected by the intensity of the car and whether or not there is shadow along it. We embody this knowledge in the structure of the Bayesian network which is used to combine all features. The parameters of the Bayesian network are learnt from examples. Finally a decision of a car's existence is made using a Bayesian minimum risk classifier.

This paper is organized as follows: Section 2 outlines the psychological test we carried out mainly to discover how human recognizes cars in aerial images. Section 3 describes how the features are predicted and computed. Section 4 covers how the multiple features are combined with a Bayesians network. Section 5 presents the detection and post-processing. We show some results in Section 6, and finally reach the conclusion in Section 7.

2 A Psychological Test

The fact that human are very good at recognizing cars motivates us to do this informal psychological test to gain some insight on how humans achieve this capability. First we asked a number of testers to retrospect after seeing the aerial images about the factors that help them make the decision of the presence of a car. The factors most people mentioned are:

- The rectangular shape and size of the car.
- The layout of the visible windshields.
- The visible sides of the car when viewed obliquely.
- The shadow the car cast on the ground.
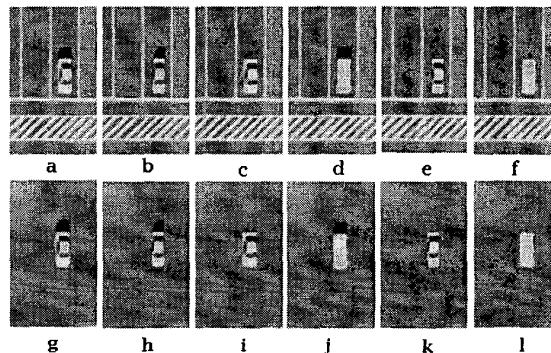- The road and other environmental evidences.

Figure 2: An example of the data used in psychological test. (a) original image of a car; (b) visible sides removed; (c) shadow removed; (d) windsheids removed; (e) both shadow and sides removed; (f) both shadow and windshields removed; (g)−(l) repeating (a)−(f) with car put in a neutral surrounding.

Then we listed the relative significance of the above factors by removing a factor from the images and have the testers re-classify what they see (Fig.2). The conclusions are the following:

- The rectangular or almost rectangular shape is the most important cue of a car.
- The layout of the windshields (frontal and rear ones) is an important factor for human detection.
- The car shadow, when it exists, can make the detection easier but it generally does not affect the decision.
- The environment also affects the detection. The presence of a parking lot, road or an assembly of cars is a strong supportive evidence that a rectangular object of appropriate size is a car although its other features are not salient.

711

3 Feature Extraction

3.1 Clustering of road directions

The vertical view aerial images of an urban area generally exhibit a few major directions. These directions are made by the parallel roads and the buildings and other structures aligned with them. These directions of interest (DOI) can be discovered from the images since a large number of straight lines are aligned with these directions. We obtain the DOI of a local part of a city (say 8 blocks square) by simply computing the histogram of the directions of the straight lines weighted by their length. The straight lines are fitted by LMS method from Canny edges. The histograms have sharp peaks at the major directions. Peaks above some threshold are declared to be DOIs of this local patch of image.

When the viewpoint is oblique and cannot be approximated with a vertical view, the image is re-projected onto a plane parallel to the imaging plane to remove the perspective effect on parallel lines. The image patch is rotated to make the DOI vertical in image. If one image has more than one DOI, we rotate it to form multiple images and then combine the result later. In fact, we rotate the image into twice the number of the DOIs to handle two-way traffic.

3.2 Features used for detection

From the psychological test and analysis, we decided to use the following features (Fig.3.b).

- The boundary of the car. The boundary of the car is mostly rectangular, but the two long sides may turn into curves under certain situations (see the difference of the two cars in Fig.3.c).

- The boundary of front windshield. We use only the front windshield because its shape, size and location in the car are relatively constant. It is always assumed to be rectangular.

- The outer boundary of the shadow area when shadow exists. The shadow is an important evidence to differentiate cars from other planar rectangular structures. In the case of very oblique sun angle, this cue will not be used since the shadow boundaries are far from the car and less reliable.

- (optional) The intensity of the shadow area when shadow exists. It is optional because it is expensive to compute when the area is large and its contribution to detection is not as significant as others.

As can be seen, the features we use are mostly gradient features. To extract this information, two methods can be used - symbolic edges (e.g. extracted by Canny edge detector) [1] or responses of gradient filters of certain shapes [7]. We chose the latter due to the following reasons. Firstly, due to the small size of car and interference in image, edges of the car may be lost or become fragmented therefore cannot be extracted robustly (see Fig.1). Secondly, some features are not always linear. Thirdly, the response of gradient filter has better resolution than the binary valued edge detector.

Therefore the features are represented as thin gradient (horizontal or vertical) masks. These masks are convolved with the images to get the value of corresponding features. The convolution masks are generated by the feature prediction module described below.
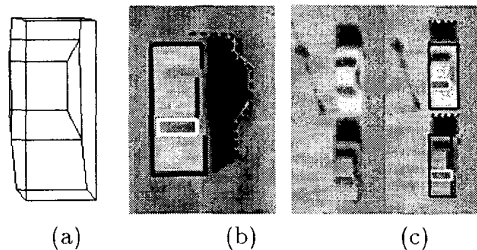

(a) (b) (c)

Figure 3: (a) Wire-frame car model. (b) All features used for detection, including boundary of the car body (in dark line), boundary of the front windshield (in white line), outer boundary of shadow area (in dotted line) and the intensity of shadow area (dark area). (c) Predicted features overlaid on the image.

3.3 Model-based feature prediction

To detect cars with the above features, we need to know the shape (for two sides of car body boundary only), the location relative to the center, and the expected strength of the features. Therefore, we use a coarse generic car model to predict the above information to verify in the input image. The generic model includes a wire-frame geometrical model and a surface reflectance model.

We use a generic wire-frame car model [6] coas shown in Fig.3.a to predict the 2D location of edges of the car. This is done by projection to the ground plane. The cars are far away from the camera and the depth of the cars in the view direction is small, so the perspective effect is hardly visible. Therefore, we use a scaled orthographic camera model.

We use a subset of the edges in the wire-frame model as features. Frontal windshield and shadow are approximately planar so there is no ambiguity about their boundary. But for the boundary of the car body itself, it is a little more complicated. For each of the four sides, there are always two edges (upper and lower) that may appear in the image. We assume that only one significant edge per side will appear in the image. For the case where the lower edge is occluded, the upper one is used. But for the case where both edges are visible, we choose the edge along which the two intensities have greater difference. For example, if the intensity difference of the hood and the front of the car is less than the intensity difference of

the front and the ground (or shadow if there is shadow along it), the lower front edge will be chosen. Therefore, we need a reflectance model of the car to determine the intensity of the planes of the car body in the image under the known illumination. We use the following modified Lambertian model.

$$I = I_{amb} + I_{spec}$$

$$I_{amb} = \begin{cases} k_a(a_{others} + a_{ground}) & \text{if } \hat{S} \cdot \hat{N} < 0; \\ k_a(a_{others} + a_{ground}\hat{S} \cdot \hat{N}) & \text{if } \hat{S} \cdot \hat{N} >= 0. \end{cases}$$

$$I_{spec} = -k_s s \hat{S} \cdot \hat{N}$$

where $\hat{N}$ is the surface unit normal, $\hat{S}$ is the unit normal of sun direction, s is sun light intensity, a_{ground}, a_{others} are ambient light intensity from the ground and other places respectively, k_d is reflectance to directional light, k_a is reflectance to ambient light.

The empirical modification is intended to take into account the light reflected by the road affecting the sides of the car. The parameters k_a, k_s are obtained directly by measuring the images for each car intensity, and s, a_{ground} and a_{others} are averaged from examples. Since we don't need very accurate values, the simple model works quite well. We are not modeling highlights on the car because they are sensitive to the subtle curvature on the car.

4 Multi-feature Integration

4.1 Parameterization of the features

The distributions of some of the features are influenced by the factors including the intensity of the car and the shadow along them, as can be seen in Fig.4. More specifically, the values of four boundary edges of the car body are affected by both the car intensity i and the shadow along that edge s; the two short vertical edges of the front windshield are affected by both i and s since they are close to the boundary; the two horizontal edges of the front windshield are affected by i; and the shadow features are not affected by either s or i. The factors are quantized into discreet values for the simplicity of processing in the future. s is quantized into *no shadow*, *thin shadow*, and *wide shadow*, and i is quantized into *dark colored*, *gray colored* and *light colored*. These dependencies are important information that should be taken advantage of. We will show next how they are used to decide the structure of the Bayesian network.

4.2 Integration through a Bayesian network

Considering all the available evidence, we may have as many as 11 features to use. We need to combine these features to get a final decision (probability) of a car's existence. Bayesian networks (BN) provide an optimal way to integrate multiple cues for a decision if the conditional distributions are known [9]. They have been used in various applications in computer vision research and shown

Figure 4: An image patch and its X/Y gradient maps. (in gradient maps, mid gray - 0, white - positive, black - negative) Expected feature values vary with the intensity of the car and shadow. Comparing the bottom three cars, all the features of the white car are clear; the top and right boundaries of the black car are invisible; the left boundary of the gray car is faint; the windshields of the black car and white car exhibit different sign in gradient maps.

promising performance [3]. We use a BN to integrate all available evidences with the structure shown in Fig.5. The factors i and s are made parent nodes of some of the evidence nodes according to the parameterization above. The posterior probability $P(car|F, i, s)$ of the node "car" (where $F = [f_1, f_2, f_n]$ is all available features) is the one to be evaluated. The values of the evidence nodes are measured from the image. The parameter node s is computed with the model, view angle and sun angle. The parameter node i is obtained by getting the median of the intensities of a small region around the pixel. Each of the evidence nodes has a conditional probability table (CPT) associated with it which is indexed by the value combination of its parents and itself. For example, the CPT of the node BU is in the form of $P(f_{BU}|car, i, s)$.
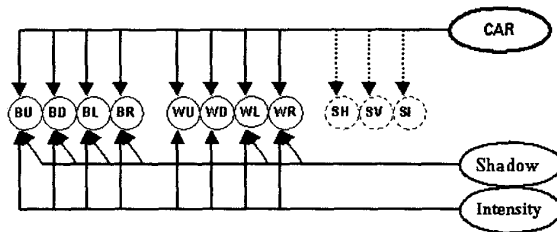

Figure 5: The Bayesian network used for detection. Dotted line shows not always available. B = body boundary / W = front windshield / S = shadow / U = rear / D = front / L = left / R = right / I = intensity. Evidences in dotted circles are used only when available.

It should be noted that the BN assumes conditional independency. In our case, although we introduced two parameter nodes to decrease the correlation of the evidence nodes, the correlation due to other factors still remains. However, [2] shows the reason why simple Bayesian classifier shows good result even in the presence of correlation of features which also applies here.

4.3 Handcrafted BN parameters

Having the network structure, human knowledge and experience can be used to fill in the parameters. An example is shown in Tab.1. The value of the features is quantized to binary (exists or not) using thresholds because it is hard for human to deal with continuous value directly. The parameters are designed considering the following empirical guidelines as well as measuring some examples in images:

- For *light* and *gray* cars, the probability to detect an edge along a *shadow* is higher than *no shadow* (s is quantized into *shadow* and *no shadow* in handcrafted BN);

- For *dark colored* cars, the probability to detect an edge along a *shadow* is much lower than *no shadow*;

- For *dark colored* cars, the probability to detect the windshield boundary edges is low;

- *Light colored* cars has higher probability to have a body boundary / windshield boundary edge detected than *gray colored* cars;

- Short edges are more likely to be made noisy than long edges;

- The left/right edges are more likely to be made noisy than front/rear edges.

car	i	s	$P(f_{BU}|car,i,s)$	$P(\overline{f_{BU}}|car,i,s)$
$\overline{car}$	-	-	0.05	0.95
car	dark	no	0.8	0.2
car	dark	thin	0.4	0.6
car	dark	wide	0.15	0.85
car	gray	no	0.6	0.4
car	gray	thin	0.7	0.3
car	gray	wide	0.8	0.2
car	light	no	0.8	0.2
car	light	thin	0.9	0.1
car	light	wide	0.95	0.05

Table 1: A hand crafted CPT for front boundary of car body.

4.4 Learning the parameters

Handcrafting the parameters is limited in the following aspects.

- Humans' qualitative experience may not reflect every part of the problem domain.

- It is not easy to transform humans' subjective qualitative experience into objective numerical quantities for computation.

- Since humans are not good at dealing with complex numerical relationships, such as distributions, the quantization from continuous distribution to binary value loses information that could have been utilized.

- Studying a large number of examples is a tedious job requiring the familiarity with vision algorithms.

Therefore, learning the parameters of the BN from examples by the computer will be ideal. In this way, we can model the sensory data with much finer resolution (we are using 64 quantization levels). A common way to represent a distribution is to fit it with some known parametric distribution forms, among which Gaussian density is the most commonly used. But here, some of the distributions display multiple modes, thus non-Gaussian. It can be modeled as a mixture of Gaussians, but for simplicity, we use non-parametric technique instead. We tried KNN (k-nearest neighbor) and Parzen Window with Gaussian kernel. The results showed the latter outperformed the former a little.

Since there are no hidden nodes in the network, learning the CPT requires just calculating the histogram. By the conditional independence assumption, the CPT of each evidence node can be learnt independently. We collected around 200 cars manually from 25 patches of images for training as positive samples and other parts of the images are served as negative samples. Only the car boundary is specified on the images and the locations of other evidences are computed by the feature prediction module. Fig.6 shows some of the learnt CPTs; the distribution contains the guidelines of human experience as well as other aspects not noticed by human observer.

We use it as human knowledge at first, that the value of the features are affected by the factors i and s. With the learnt distributions, we proved that the parameterization is efficient by showing Kullback-Leibler divergence of $P(f_j|car)$ and $P(f_i|factor,car)$ is large, where $factor = i, s$.
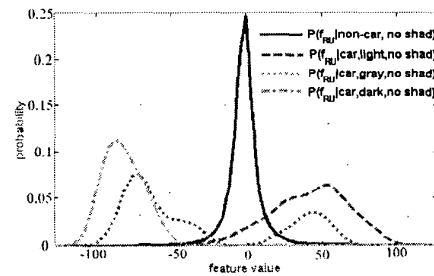

Figure 6: Learnt CPT of the front boundary of car body when there is no shadow along it ($P(f_{RU}|car,i,s = noshadow)$). The distribution exhibits large difference for different i.

5 Detection and Post-processing
5.1 Detection

After the BN is constructed and learnt, we can use it to detect cars in other images. First the available feature maps are computed. The values of the features are

retrieved from corresponding feature maps at the locations computed by the feature prediction module. And then the car existence $P(car|F,i,s)$ at each image pixel is taken to be the maximum of the probability of all sizes (*mini, compact, full-size,* and *luxury*).

$$P(car|F,i,s) = \alpha(\Pi_{j=1}^{n}P(f_j|car,i,s))P(car)$$

$$P(\overline{car}|F,i,s) = 1 - P(car|F,i,s)$$

$$\alpha = \frac{1}{(\Pi_{j=1}^{n}P(f_j|car,i,s))P(car)+\Pi_{j=1}^{n}P(f_j|\overline{car},i,s))P(\overline{car})}$$

A minimum risk classifier instead of a minimum error rate classifier is used to make the final decision. It is more suitable here for the user can conveniently specify their preference on false alarms or mis-detections by a risk matrix C:

$$C = \begin{pmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \end{pmatrix}$$

where 0 means $\overline{car}$ and 1 means *car*. c_{01} means the cost to misclassify a *car* to be a $\overline{car}$, and so on.

The decision rule for classifying a car in minimum error rate classifier $P(car|F,i,s) > P(\overline{car}|F,i,s)$ is replaced by $R(car|F,i,s) < R(\overline{car}|F,i,s)$ where R is the expected risk. i and s are omitted in the derivation below.

$$R(car|F) < R(\overline{car}|F)$$
$$\Longrightarrow$$
$$P(F|car)P(car)(c_{01} - c_{11}) > P(F|\overline{car})P(\overline{car})(c_{10} - c_{00})$$
$$\Longrightarrow$$
$$P(F|car)P'(car) > P(F|\overline{car})P'(\overline{car})$$

where $P'(car) = \frac{P(car)(c_{01}-c11)}{P(car)(c_{01}-c11)+P(\overline{car})(c_{10}-c_{00})}$, and $P'(\overline{car}) = 1 - P'(car)$.

It is equivalent to adjusting the prior probability $P(car)$ and $P(\overline{car})$. In our case, c_{00} and c_{11} are set to 0 and c_{01} is set to 1. c_{10} is the only free parameter for the user to specify.

5.2 Post-processing

Generally a number of pixels around the center of a car will all have a high probability value, thus many will be classified as cars. Besides, the coincidental alignment of boundary lines and shadow lines as well as the coincidental alignment of features of adjacent cars may also create high probability spots which are close to the true center of the car. (See Fig.7.a for example.) Overlap resolution is needed to remove the redundant results.

First we find the connected regions in the probability map. Then for each connected component, the accumulated probability and weighed centroid are computed. We assume that the true detections have higher accumulated probability than false alarms around it, which in most time is true in our experiments. The connected components are sorted by their accumulated probability. Valid detections are chosen from the front of the queue. For

each connected region, if it does not overlap with any of the previous chosen valid detections, it is identified as a valid detection, otherwise discarded.
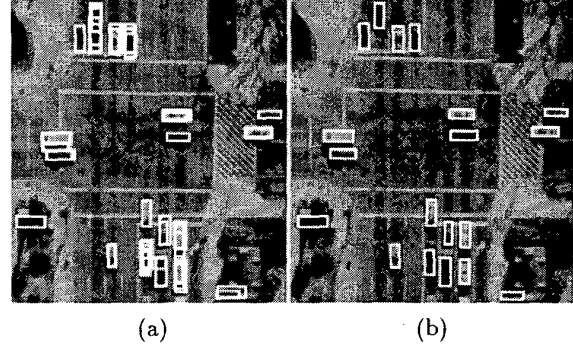

(a)   (b)

Figure 7: Before(a) and after(b) overlap resolution.

As mentioned in [7], false alarms will appear when the sunlight creates shadow with the similar width as a car. It may also happen in our system especially when the car features are not perfect to suppress it in the previous step. This is also taken care of in our overlap resolution. A result example is shown in Fig.7.b.

6 Result and Discussion
6.1 Result

We tested our system on 12 image patches of a Washington DC image set containing 320 cars, with different sun angles, view angles, image quality and against different backgrounds (Fig.9). Some of the images present very difficult conditions due to cars hidden in tree branches (Fig.9.c,h), shadow having similar width as a car (Fig.9.a) and oblique viewpoint of some images (Fig.9.i). With all these difficulties, our detector still showed very promising results.

We find that most cars with good features have been detected, while some of the difficult ones also got detected with appropriate choice of the c_{10} value. Although aimed at only detecting regular passenger cars, it also detected some other vehicles (vans, SUVs) sharing similar feature placement. Both the mis-detection rate and the false alarm rate of *dark colored* cars are higher than others because under most situations they don't have as salient features as *light* or *gray colored* ones. Most false alarms result from the coincidental alignment of rectangular shape and other lines of structures in buildings, foliage of trees or road markings.

For a detector, there is always a tradeoff between false alarm rate and mis-detection rate. ROC (receiver-operating-characteristic) curve is drawn in Fig.8. Besides the regular ROC curve, ROC curve not considering the off-road part is also given because in some applications such as vehicle counting, when the coarse knowledge about location the road is known, we are especially inter-

715

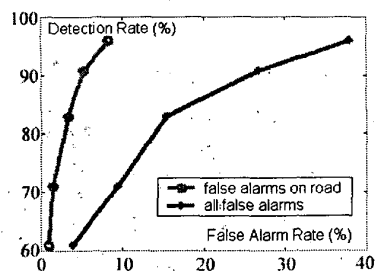ested in the false alarm rate on the road.

Figure 8: ROC curve.

6.2 Computation

The computation time is proportional to the number of pixels processed. Convolution takes the greatest share of time. We use convolution decomposition to accelerate the computation by separating the gradient and the shape of the feature. And since the convolution is with binary mask, all the multiplication operations are replaced by integer logic operations. Evaluation of the probability of car at each pixel also takes a big share of time. We also used a simple preliminary screening procedure to cut down 80% of the evaluation without noticeably affecting the result. After all these techniques, a $1000 * 870$ image takes about 30 seconds for one DOI on a PII $400MHz$ PC. All the operations are carried out to each pixel of the image, so it is suitable for highly parallel processing.

7 Conclusion

We have described a car detection system for aerial images. Different from pervious work, we can deal with vertical and slightly oblique viewpoints since we formulate it as a 3D object recognition problem. We analyze shadows explicitly to make them a useful cue for detection instead of a source of problems as has been the case in some previous work [7]. We used the response of gradient mask filters as feature to account for the low resolution and noise in aerial image and this makes it more robust than using edge detector. We introduced car intensity i and shadow s as parameters and they were proven to be effective. A Bayesian network is used to combine multiple features with learning and the only hand given parameter is to make a balance over false alarms and misdetections. Most importantly, in testing we found it gave very promising result even in very difficult situations.

Machine learning is gaining popularity in computer vision community. In this work, we showed a good example on how human knowledge and learning can be balanced. The generic car model and the parameterization of the features which are difficult if not impossible to learn from statistical data are introduced as prior knowledge and the parameters of the Bayesian network are learnt from examples. We believe that similar approaches can also be

useful for other object detection and recognition tasks.

References

[1] Burlina, P., Parameswaran, V., Chellappa, R., Sensitivity Analysis and Learning Strategies for Context-Based Vehicle Detection Algorithms, *DARPA IU Workshop 97* pp. 577-584, 1997.

[2] Domingos, P., Pazzani, M. Beyond independence: Conditions for the optimality of the simple Bayesian classifier. *In Proceedings of the 13rd ICML*, 1996.

[3] ZuWhan Kim and Ramakant Nevatia, Uncertain Reasoning and Learning for Feature Grouping, *CVIU*, Vol. 76, No. 3, pp 278-288, December, 1999.

[4] Chungan Lin and Ramakant Nevatia, Building Detection and Description from a Single Intensity Image, *CVIU(72)*, No. 2, November 1998, pp. 101-121.

[5] Gang Liu, Lixin Gong, Robert M. Haralick, Vehicle Detection in Aerial Imagery and Performance Evaluation, *Submitted for publication.*

[6] D. Koller, K. Daniilidis, and H. H. Nagel, Model-Based Object Tracking in Monocular Image Sequences of Road Traffic Scenes, *IJCV*, 10:3 (1993) 257-281.

[7] Moon, H., Chellappa, R., Rosenfeld, A., Performance Analysis of a Simple Vehicle Detection Algorithm, *ARL Federated Laboratory Symposium*, 1999.

[8] Papageorgiou, C. and T. Poggio, A Trainable System for Object Detection, *IJCV*, 38, 1, 15-33, 2000.

[9] Pearl, J, *Probabilistic reasoning in intelligent systems: networks of plausible inference*, San Francisco: Morgan Kaufmann.

[10] K. Price, Urban Street Grid Description and Verification, *WACV00*, 2000.

[11] A. Rajagopalan, P. Burlina and R. Chellappa, Higher Order Statistical Learning for Vehicle Detection in Images, *ICCV*, 1999.

AI

[12] Henry Schneiderman and Takeo Kanade, A Statistical Method for 3D Object Detection Applied to Faces and Cars, *CVPR*, 2000.
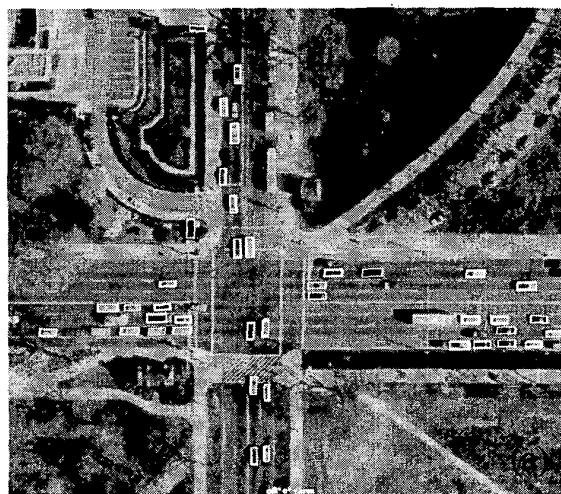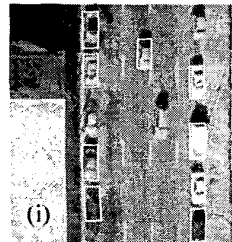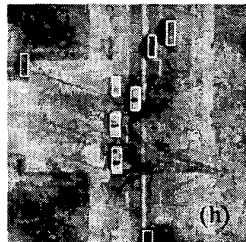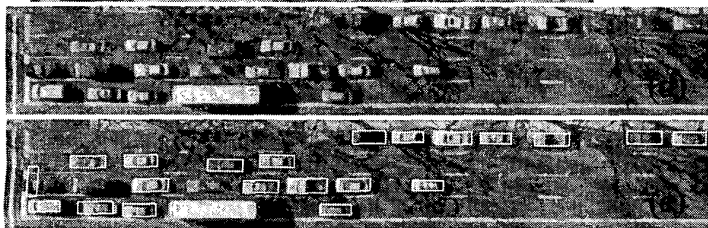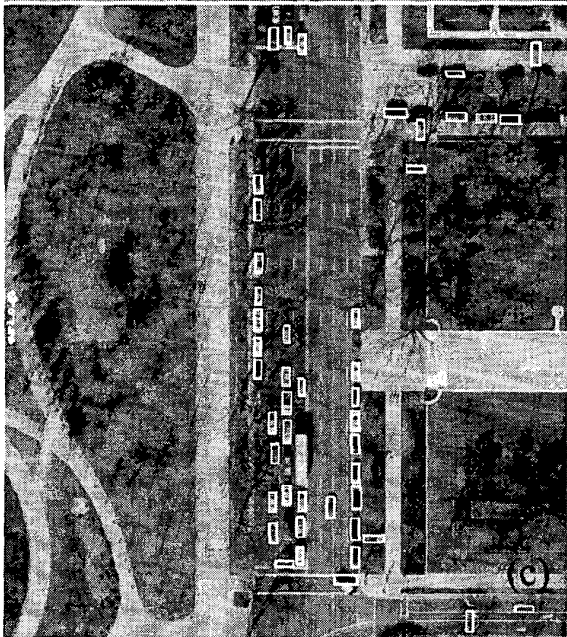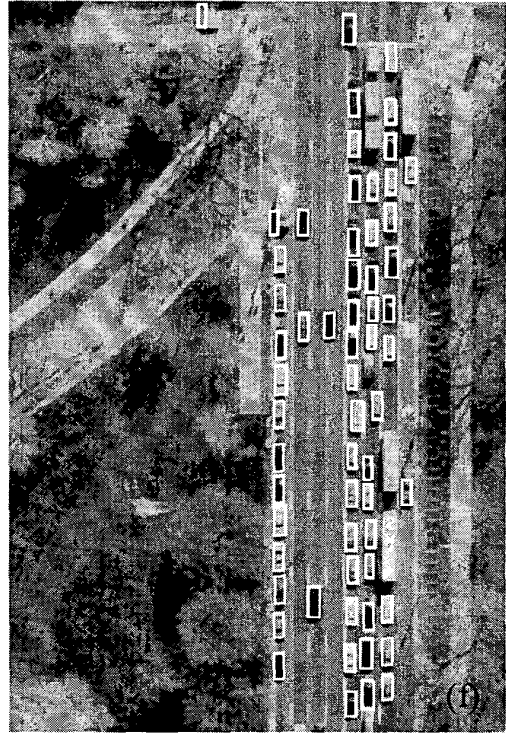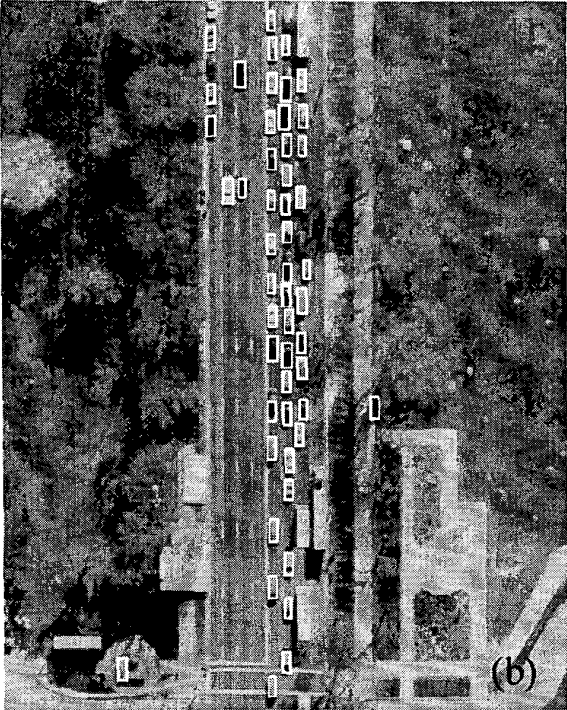
Figure 9: Detection results. (d)(e) are the close-up (rotated to fit in page) of (c); (h) is the results of Fig.1.b.