Vehicle Detection in Aerial Surveillance Using Dynamic Bayesian Networks

Hsu-Yung Cheng, *Member, IEEE*, Chih-Chia Weng, and Yi-Ying Chen

***Abstract*—We present an automatic vehicle detection system for aerial surveillance in this paper. In this system, we escape from the stereotype and existing frameworks of vehicle detection in aerial surveillance, which are either region based or sliding window based. We design a pixelwise classification method for vehicle detection. The novelty lies in the fact that, in spite of performing pixelwise classification, relations among neighboring pixels in a region are preserved in the feature extraction process. We consider features including vehicle colors and local features. For vehicle color extraction, we utilize a color transform to separate vehicle colors and nonvehicle colors effectively. For edge detection, we apply moment preserving to adjust the thresholds of the Canny edge detector automatically, which increases the adaptability and the accuracy for detection in various aerial images. Afterward, a dynamic Bayesian network (DBN) is constructed for the classification purpose. We convert regional local features into quantitative observations that can be referenced when applying pixelwise classification via DBN. Experiments were conducted on a wide variety of aerial videos. The results demonstrate flexibility and good generalization abilities of the proposed method on a challenging data set with aerial surveillance images taken at different heights and under different camera angles.**

***Index Terms*—Aerial surveillance, dynamic Bayesian networks (DBNs), vehicle detection.**

I. Introduction

In recent years, the analysis of aerial videos taken from aerial vehicle has become an important issue [1]. These technologies have a variety of applications, such as military, police, and traffic management [2]–[4]. Compared with ground-plane surveillance systems, aerial surveillance is more suitable for monitoring fast-moving targets and covers a much larger spatial area. Therefore, aerial surveillance systems become an excellent supplement of ground-plane surveillance systems. One of the main topics in aerial image analysis is scene registration and alignment [1], [5], [6]. Another very important topic in intelligent aerial surveillance is vehicle detection and tracking [7]–[12]. The challenges of vehicle detection in aerial surveillance include camera motions such as panning, tilting, and rotation. In addition, airborne platforms at different heights result in different sizes of target objects.

Hinz and Baumgartner [7] utilized a hierarchical model that describes different levels of details of vehicle features. There is no specific vehicle models assumed, making the method flexible. However, their system would miss vehicles when the contrast is weak or when the influences of neighboring objects are present. Cheng and Butler [8] considered multiple clues and used a mixture of experts to merge the clues for vehicle detection in aerial images. They performed color segmentation via mean-shift algorithm and motion analysis via change detection. In addition, they presented a trainable sequential maximum *a posterior* method for multiscale analysis and enforcement of contextual information. However, the motion analysis algorithm applied in their system cannot deal with aforementioned camera motions and complex background changes. Moreover, in the information fusion step, their algorithm highly depends on the color segmentation results.

Lin *et al.* [9] proposed a method by subtracting background colors of each frame and then refined vehicle candidate regions by enforcing size constraints of vehicles. However, they assumed too many parameters such as the largest and smallest sizes of vehicles, and the height and the focus of the airborne camera. Assuming these parameters as known priors might not be realistic in real applications. In [11], the authors proposed a moving-vehicle detection method based on cascade classifiers. A large number of positive and negative training samples need to be collected for the training purpose. Moreover, multiscale sliding windows are generated at the detection stage. The main disadvantage of this method is that there are a lot of miss detections on rotated vehicles. Such results are not surprising from the experiences of face detection using cascade classifiers. If only frontal faces are trained, then faces with poses are easily missed. However, if faces with poses are added as positive samples, the number of false alarms would surge.

Choi and Yang [12] proposed a vehicle detection algorithm using the symmetric property of car shapes. However, this cue is prone to false detections such as symmetrical details of buildings or road markings. Therefore, they applied a log-polar histogram shape descriptor to verify the shape of the candidates. Unfortunately, the shape descriptor is obtained from a fixed vehicle model, making the algorithm inflexible. Moreover, similar to [8], the algorithm in [12] relied on mean-shift clustering algorithm for image color segmentation. The major drawback is that a vehicle tends to be separated as many regions since car roofs and windshields usually have different colors. Moreover, nearby vehicles might be clustered as one region if they have similar

Manuscript received August 01, 2010; revised May 11, 2011 and August 17, 2011; accepted October 05, 2011. Date of publication October 19, 2011; date of current version March 21, 2012. This work was supported in part by the National Science Council of Taiwan under Project 99-2628-E-008-098. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Mary Comer.

The authors are with the Department of Computer Science and Information Engineering, National Central University, Chungli 320, Taiwan (e-mail: chengsy@csie.ncu.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2011.2172798

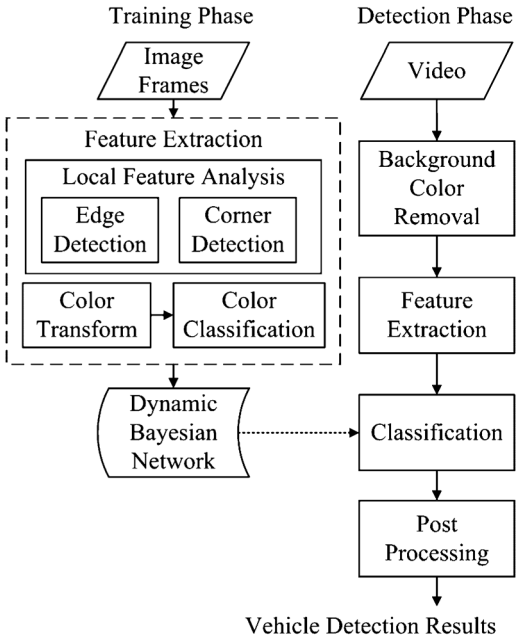


Fig. 1. Proposed system framework.

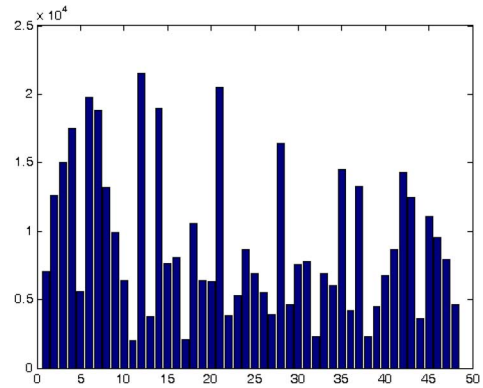


Fig. 2. Color histogram of a frame.

colors. The high computational complexity of mean-shift segmentation algorithm is another concern.

In this paper, we design a new vehicle detection framework that preserves the advantages of the existing works and avoids their drawbacks. The modules of the proposed system framework are illustrated in Fig. 1. The framework can be divided into the training phase and the detection phase. In the training phase, we extract multiple features including local edge and corner features, as well as vehicle colors to train a dynamic Bayesian network (DBN). In the detection phase, we first perform background color removal similar to the process proposed in [9]. Afterward, the same feature extraction procedure is performed as in the training phase. The extracted features serve as the evidence to infer the unknown state of the trained DBN, which indicates whether a pixel belongs to a vehicle or not. In this paper, we do not perform region-based classification, which would highly depend on results of color segmentation algorithms such as mean shift. There is no need to generate multiscale sliding windows either. The distinguishing feature of the proposed framework is that the detection task is based on pixelwise classification. However, the features are extracted in a neighborhood region of each pixel. Therefore, the extracted features comprise not only pixel-level information but also relationship among neighboring pixels in a region. Such design is more effective and efficient than region-based [8] [12] or multiscale sliding window detection methods [11]. The rest of this paper is organized as follows: Section II explains the proposed automatic vehicle detection mechanism in detail. Section III demonstrates and analyzes the experimental results. Finally, conclusions are made in Section IV.

II. Proposed Vehicle Detection Framework

Here, we elaborate each module of the proposed system framework in detail.

A. Background Color Removal

Since nonvehicle regions cover most parts of the entire scene in aerial images, we construct the color histogram of each frame and remove the colors that appear most frequently in the scene. Take Fig. 2 for example, the colors are quantized into 48 histogram bins. Among all histogram bins, the 12th, 21st, and 6th bins are the highest and are thus regarded as background colors and removed. These removed pixels do not need to be considered in subsequent detection processes. Performing background color removal cannot only reduce false alarms but also speed up the detection process.

B. Feature Extraction

Feature extraction is performed in both the training phase and the detection phase. We consider local features and color features in this paper.

1) Local Feature Analysis: Corners and edges are usually located in pixels with more information. We use the Harris corner detector [13] to detect corners. To detect edges, we apply moment-preserving thresholding method on the classical Canny edge detector [14] to select thresholds adaptively according to different scenes. In the Canny edge detector, there are two important thresholds, i.e., the lower threshold T_{low} and the higher threshold T_{high}. As the illumination in every aerial image differs, the desired thresholds vary and adaptive thresholds are required. The computation of Tsai's moment-preserving method [15] is deterministic without iterations for L-level thresholding with $L < 5$. Its derivation of thresholds is described as follows.

Let f be an image with n pixels and $f(x, y)$ denote the gray value at pixel (x, y). The ith moment m_i of f is defined as

$$m_i = \left(\frac{1}{n}\right) \sum_j n_j (z_j)^i = \sum_j p_j (z_j)^i, \qquad i = 1, 2, 3, \ldots \tag{1}$$

where n_j is the total number of pixels in image f with gray value z_j and $p_j = n_j/n$. For bilevel thresholding, we would like to select threshold T such that the first three moments of image f are preserved in the resulting bilevel image g. Let all the below-threshold gray values in f be replaced by z_0 and all the above-threshold gray values be replaced by z_1; we can solve for p_0 and p_1 based on the moment-preserving principle [15].

After obtaining p_0 and p_1, the desired threshold T is computed using

$$p_0 = (1/n)\sum_{1}^{T} n_j. \tag{2}$$

In order to detect edges, we use the gradient magnitude $G(x, y)$ of each pixel to replace the grayscale value $f(x, y)$ in Tsai's method. Then, the adaptive threshold found by (2) is used as the higher threshold T_{high} in the Canny edge detector. We set the lower threshold as $T_{\text{low}} = 0.1 \times (G_{\max} - G_{\min}) + G_{\min}$, where $G_{\max}$ and $G_{\min}$ represent the maximum and minimum gradient magnitudes in the image. Thresholds automatically and dynamically selected by our method give better performance on the edge detection. We will demonstrate the performance improvement on the edge detection with adaptive thresholds and the corresponding impact on final vehicle detection results in Section III.

2) Color Transform and Color Classification: In [16], the authors proposed a new color model to separate vehicle colors from nonvehicle colors effectively. This color model transforms (R, G, B) color components into the color domain (u, v), i.e.,

$$u_p = \frac{2Z_p - G_p - B_p}{Z_p} \tag{3}$$

$$v_p = \text{Max}\left\{\frac{B_p - G_p}{Z_p}, \frac{R_p - B_p}{Z_p}\right\} \tag{4}$$

where (R_p, G_p, B_p) is the R, G, and B color components of pixel p and $Z_p = (R_p + G_p + B_p)/3$. It has been shown in [16] that all the vehicle colors are concentrated in a much smaller area on the u–v plane than in other color spaces and are therefore easier to be separated from nonvehicle colors. Although the color transform proposed in [16] did not aim for aerial images, we have found that the separability property still presents in aerial images. As shown in Fig. 3, we can observe that vehicle colors and nonvehicle colors have less overlapping regions under the (u, v) color model. Therefore, we apply the color transform to obtain (u, v) components first and then use a support vector machine (SVM) to classify vehicle colors and nonvehicle colors. When performing SVM training and classification, we take a block of $n \times m$ pixels as a sample. More specifically, each feature vector is defined as $[u_1, v_1, \ldots, u_{n\times m}, v_{n\times m}]$. Notice that we do not perform vehicle color classification via SVM for blocks that do not contain any local features. Those blocks are taken as nonvehicle color areas.

As we mentioned in Section I, the features are extracted in a neighborhood region of each pixel in our framework. Considering an $N \times N$ neighborhood Λ_p of pixel p, as shown in Fig. 4, we extract five types of features, i.e., S, C, E, A, and Z, for the pixel. These features serve as the observations to infer the unknown state of a DBN, which will be elaborated in the next subsection. The first feature S denotes the percentage of pixels in Λ_p that are classified as vehicle colors by SVM, as defined in (5). Note that $N_{\text{Vehicle color}}$ denotes to the number of pixels in Λ_p that are classified as vehicle colors by SVM, i.e.,

$$S = \frac{N_{\text{Vehicle color}}}{N^2}. \tag{5}$$

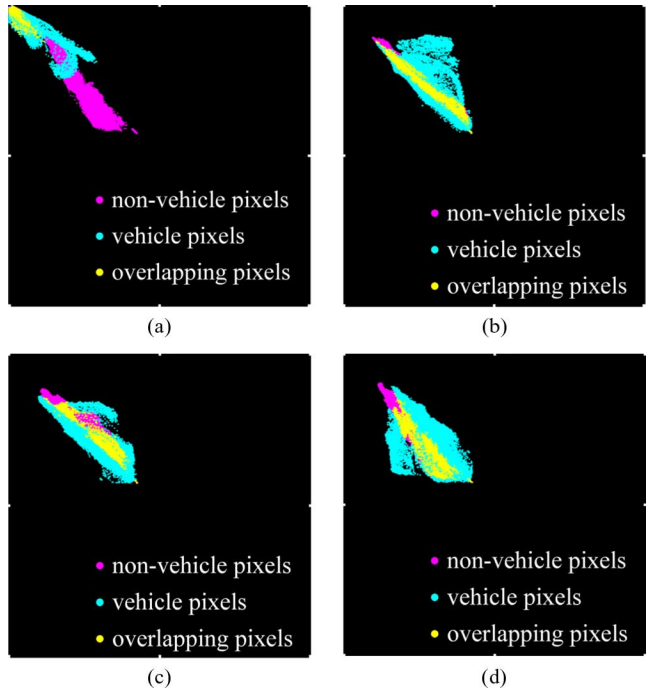


Fig. 3. Vehicle colors and nonvehicle colors in different color spaces. (a) u–v, (b) R–G, (c) G–B, and (d) B–R planes.

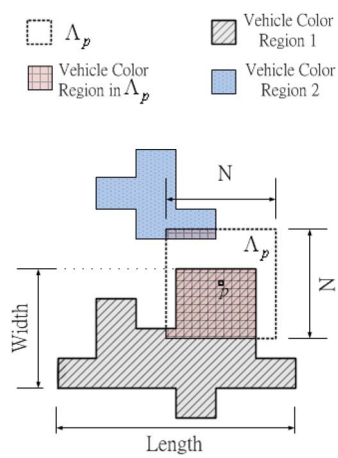


Fig. 4. Neighborhood region for feature extraction.

Features C and E are defined, respectively, as

$$C = \frac{N_{\text{Corner}}}{N^2} \tag{6}$$

$$E = \frac{N_{\text{Edge}}}{N^2}. \tag{7}$$

Similarly, N_{Corner} denotes to the number of pixels in Λ_p that are detected as corners by the Harris corner detector, and N_{Edge} denotes the number of pixels in Λ_p that are detected as edges by the enhanced Canny edge detector. The pixels that are classified as vehicle colors are labeled as connected vehicle-color regions. The last two features A and Z are defined as the aspect ratio and the size of the connected vehicle-color region where the pixel p resides, as illustrated in Fig. 4. More specifically, $A = \text{Length}/\text{Width}$, and feature Z is the pixel count of "vehicle-color region 1" in Fig. 4.

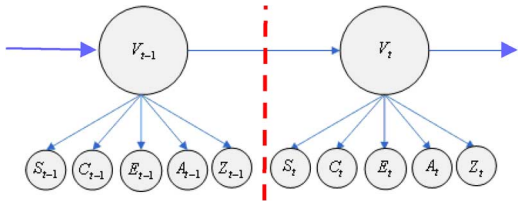


Fig. 5. DBN model for pixelwise classification.

C. DBN

We perform pixelwise classification for vehicle detection using DBNs [18]. The design of the DBN model is illustrated in Fig. 5. Node V_t indicates if a pixel belongs to a vehicle at time slice t. The state of V_t is dependent on the state of V_{t-1}. Moreover, at each time slice t, state V_t has influences on the observation nodes S_t, C_t, E_t, A_t, and Z_t. The observations are assumed to be independent of one another. The definitions of these observations are explained in the previous subsection. Discrete observation symbols are used in our system. We use K-means to cluster each observation into three clusters, i.e., we use three discrete symbols for each observation node. In the training stage, we obtain the conditional probability tables of the DBN model via expectation-maximization [18] algorithm by providing the ground-truth labeling of each pixel and its corresponding observed features from several training videos. In the detection phase, the Bayesian rule is used to obtain the probability that a pixel belongs to a vehicle, i.e.,

$$P(V_t|S_t, C_t, E_t, A_t, Z_t, V_{t-1}) = P(V_t|S_t)P(V_t|C_t) \times P(V_t|E_t)P(V_t|A_t)P(V_t|Z_t)P(V_t|V_{t-1})P(V_{t-1}). \quad (8)$$

The joint probability $P(V_t|S_t, C_t, E_t, A_t, Z_t, V_{t-1})$ is the probability that a pixel belongs to a vehicle pixel at time slice t given all the observations and the state of the previous time instance. According to the naive Bayesian rule of conditional probability, the desired joint probability can be factorized since all the observations are assumed to be independent. Term $P(V_t|S_t)$ is defined as the probability that a pixel belongs to a vehicle pixel at time slice t given observation S_t at time instance t [S is defined in (5)]. Terms $P(V_t|C_t)$, $P(V_t|E_t)$, $P(V_t|A_t)$, $P(V_t|Z_t)$, and $P(V_t|V_{t-1})$ are similarly defined.

The proposed vehicle detection framework can also utilize a Bayesian network (BN) to classify a pixel as a vehicle or non-vehicle pixel. When performing vehicle detection using BN, the structure of the BN is set as one time slice of the DBN model in Fig. 5. We will compare the detection results using BN and DBN in the next section.

D. Post Processing

We use morphological operations to enhance the detection mask and perform connected component labeling to get the vehicle objects. The size and the aspect ratio constraints are applied again after morphological operations in the postprocessing stage to eliminate objects that are impossible to be vehicles. However, the constraints used here are very loose.

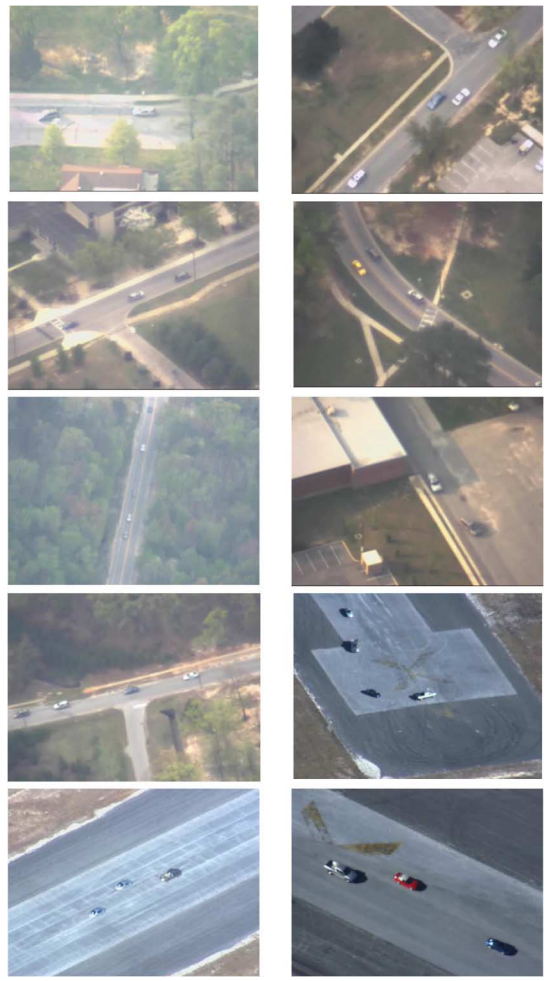


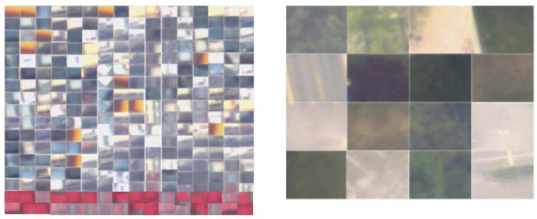
Fig. 6. Snapshots of the experimental videos.

(a) (b)

Fig. 7. Background color removal results.

III. Experimental Results

Experimental results are demonstrated here. To analyze the performance of the proposed system, various video sequences with different scenes and different filming altitudes are used. The experimental videos are displayed in Fig. 6. Note that it is infeasible to assume prior information of camera heights and target object sizes for this challenging data set. When performing background color removal, we quantize the color histogram bins as $16 \times 16 \times 16$. Colors corresponding to the first eight highest bins are regarded as background colors and removed from the scene. Fig. 7(a) displays an original image frame, and Fig. 7(b) displays the corresponding image after background removal.



(a) (b)

Fig. 8. Training images for vehicle color classification: (a) Positive samples and (b) negative samples.

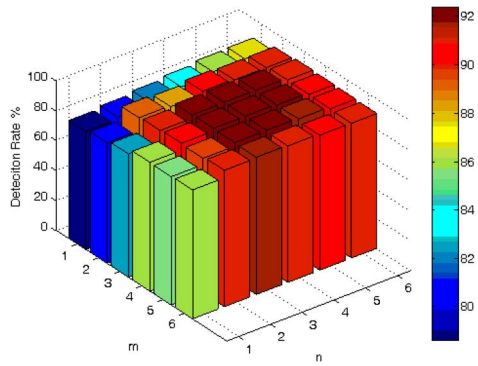


Fig. 9. Influence of different block sizes on the detection rate.

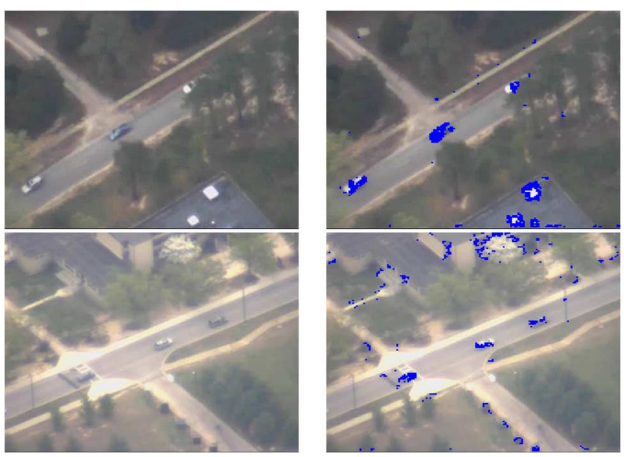


Fig. 10. Results of vehicle color classification.

When employing SVM, we need to select the block size$_{n\times m}$ to form a sample (see Fig. 8)<<AQ1>>. Fig. 9 displays the influence of different block sizes on the detection rate. According to the observation from Fig. 9, we take each 3×4 block to form a feature vector. The color of each pixel would be transformed to u and v color components using (3) and (4). More specifically, each feature vector has a dimension of 24. The training images used to train the SVM are displayed in Fig. 8. Notice that the blocks that do not contain any local features are taken as non-vehicle areas without the need of performing classification via SVM. Fig. 10 shows the results of color classification by SVM after background color removal and local feature analysis.

To obtain the conditional probability tables of the DBN, we select the training clips from the first six experimental videos displayed in Fig. 6. The remaining four videos are not involved in the training process. Each training clips contains 30 frames,

TABLE I
DETECTION ACCURACY USING DIFFERENT NEIGHBORHOOD SIZES

Size of Λ_p	Hit rate	Number of false positives per frame
5 × 5	70.91	0.523
7 × 7	92.31	0.278
9 × 9	87.06	0.281
11 × 11	82.35	0.401
13 × 13	75.58	0.415

(a) (b)

Fig. 11. Impact of the enhanced edge detector on vehicle detection results.

whose ground-truth vehicle positions are manually marked. That is, we use only 180 frames to train the DBN. To select the size of the neighborhood area for feature extraction, we list the detection accuracy using different neighborhood sizes in Table I. The detection accuracy is measured by the hit rate and the number of false positives per frame. There are a total of 225 025 frames in the data set. When calculating the detection accuracy, we perform evaluation every 100 frames. We can observe that the neighborhood area Λ_p with the size of 7×7 yields the best detection accuracy. Therefore, for the rest of the experiments, the size of the neighborhood area for extracting observations is set as 7×7.

The impacts of the enhanced Canny edge detector on vehicle detection results can be observed in Fig. 11. Fig. 11(a) shows the results obtained using the traditional Canny edge detector with nonadaptive thresholds. Fig. 11(b) shows the detection results obtained using the enhanced Canny edge detector with moment-preserving threshold selection. Nonadaptive thresholds cannot adjust to different scenes and would therefore result in more misses or false alarms, as displayed in Fig. 11(a). To examine the necessity of the background removal process and the enhanced edge detector, we list the detection accuracy of four different scenarios in Table II. We can observe that the background removal process is important for reducing false positives and the enhanced edge detector is essential for increasing hit rates.

We compare different vehicle detection methods in Fig. 12. The moving-vehicle detection with road detection method in [9] requires setting a lot of parameters to enforce the size constraints

TABLE II
DETECTION ACCURACY OF FOUR DIFFERENT SCENARIOS

Scenarios	Hit rate	Number of false positives per frame
Without background removal and without enhanced edge detector	75.08	0.399
Without background removal but with enhanced edge detector	92.35	0.459
With background removal but without enhanced edge detector	74.96	0.297
With background removal and with enhanced edge detector	92.31	0.278

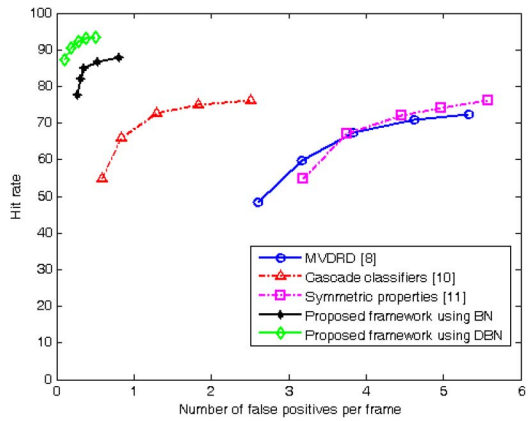
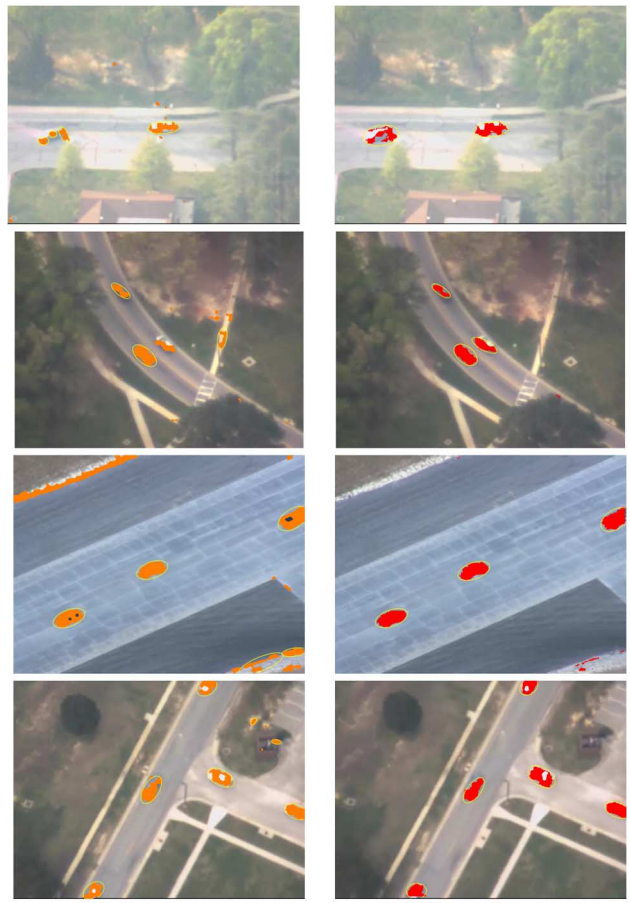


Fig. 12. Comparisons of different vehicle detection methods.



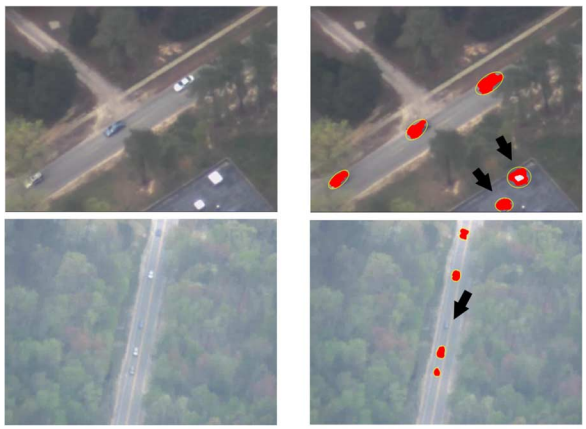
(a) (b)

Fig. 13. Vehicle detection results: (a) BNs and (b) DBNs.

in order to reduce false alarms. However, for the experimental data set, it is very difficult to select one set of parameters that suits all videos. Setting the parameters heuristically for the data set would result in low hit rate and high false positive numbers. The cascade classifiers used in [11] need to be trained by a large number of positive and negative training samples. The number of training samples required in [11] is much larger than the training samples used to train the SVM classifier in Fig. 8. The colors of the vehicles would not dramatically change due to the influence of the camera angles and heights. However, the entire appearance of the vehicle templates would vary a lot under different heights and camera angles. When training the cascade classifiers, the large variance in the appearance of the positive templates would decrease the hit rate and increase the number of false positives. Moreover, if the aspect ratio of the multiscale detection windows is fixed, large and rotated vehicles would be often missed. The symmetric property method proposed in [12] is prone to false detections such as symmetrical details of buildings or road markings. Moreover, the shape descriptor used to verify the shape of the candidates is obtained from a fixed vehicle model and is therefore not flexible. Moreover, in some of our experimental data, the vehicles are not completely symmetric due to the angle of the camera. Therefore, the method in [12] is not able to yield satisfactory results.

Compared with these methods, the proposed vehicle detection framework does not depend on strict vehicle size or aspect ratio constraints. Instead, these constraints are observations that can be learned by BN or DBN. The training process does not require a large amount of training samples. The results demonstrate flexibility and good generalization ability on a wide variety of aerial surveillance scenes under different heights and camera angles. It can be expected that the performance of DBN is better than that of the BN. In Fig. 13, we display the detection results using BN and DBN [see Fig. 13(a) and (b), respectively]. The colored pixels are the ones that are classified as vehicle pixels by BN or DBN. The ellipses are the final vehicle detection results after performing postprocessing. DBN outperforms BN because it includes information along time. When observing detection results of consecutive frames, we also notice that the detection results via DBN are more stable. The reason is that, in aerial surveillance, the aircraft carrying the camera usually follows the vehicles on the ground, and therefore, the positions of the vehicles would not have dramatic changes in the scene even when the vehicles are moving in high speeds. Therefore, the information along the time contributed by $P(V_t|V_{t-1})$ helps stabilize the detection results in the DBN.

Fig. 14 shows some detection error cases. Fig. 14(a) display the original image frames, and Fig. 14(b) displays the detection results. The black arrows in Fig. 14(b) indicate the miss detection or false positive cases. In the first row of Fig. 14, the rectangular structures on the buildings are very similar to vehicles. Therefore, sometimes, these rectangular structures would be detected as vehicles incorrectly. In the second row of Fig. 14, the miss detection is caused by the low contrast and the small



(a) (b)

Fig. 14. Detection error cases.

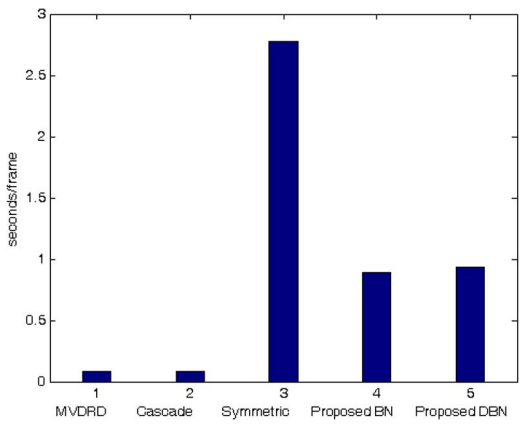


Fig. 15. Processing speeds of different methods.

size of the vehicle. However, other vehicles are successfully detected in this challenging setting.

Fig. 15 shows the average processing speeds of different vehicle detection methods. The experiments are conducted on a personal computer with 2.66-GB central processing unit and 4-G random access memory. The frame size of the experimental videos is 640 × 480 pixels. Although the proposed framework using BN and DBN cannot reach the frame rate of the surveillance videos, it is sufficient to perform vehicle detection every 50–100 frames. Tracking algorithms can be applied on the intermediate frames between two detection frames to track each individual vehicle. Therefore, high detection rate and low false alarm rate should be the primary considerations of designing detection methods given the condition that the execution time is reasonable.

IV. Conclusion

In this paper, we have proposed an automatic vehicle detection system for aerial surveillance that does not assume any prior information of camera heights, vehicle sizes, and aspect ratios. In this system, we have not performed region-based classification, which would highly depend on computational intensive color segmentation algorithms such as mean shift. We have not generated multiscale sliding windows that are not suitable for detecting rotated vehicles either. Instead, we have proposed a pixelwise classification method for the vehicle detection using DBNs. In spite of performing pixelwise classification, relations among neighboring pixels in a region are preserved in the feature extraction process. Therefore, the extracted features comprise not only pixel-level information but also region-level information. Since the colors of the vehicles would not dramatically change due to the influence of the camera angles and heights, we use only a small number of positive and negative samples to train the SVM for vehicle color classification. Moreover, the number of frames required to train the DBN is very small. Overall, the entire framework does not require a large amount of training samples. We have also applied moment preserving to enhance the Canny edge detector, which increases the adaptability and the accuracy for detection in various aerial images. The experimental results demonstrate flexibility and good generalization abilities of the proposed method on a challenging data set with aerial surveillance images taken at different heights and under different camera angles. For future work, performing vehicle tracking on the detected vehicles can further stabilize the detection results. Automatic vehicle detection and tracking could serve as the foundation for event analysis in intelligent aerial surveillance systems.

References

[1] R. Kumar, H. Sawhney, S. Samarasekera, S. Hsu, T. Hai, G. Yanlin, K. Hanna, A. Pope, R. Wildes, D. Hirvonen, M. Hansen, and P. Burt, "Aerial video surveillance and exploitation," *Proc. IEEE*, vol. 89, no. 10, pp. 1518–1539, 2001.

[2] I. Emst, S. Sujew, K. U. Thiessenhusen, M. Hetscher, S. Rabmann, and M. Ruhe, "LUMOS—Airbome traffic monitoring system," in *Proc. IEEE Intell. Transp. Syst.*, Oct. 2003, vol. 1, pp. 753–759.

[3] L. D. Chou, J. Y. Yang, Y. C. Hsieh, D. C. Chang, and C. F. Tung, "Intersection-based routing protocol for VANETs," *Wirel. Pers. Commun.*, vol. 60, no. 1, pp. 105–124, Sep. 2011.

[4] S. Srinivasan, H. Latchman, J. Shea, T. Wong, and J. McNair, "Airborne traffic surveillance systems: Video surveillance of highway traffic," in *Proc. ACM 2nd Int. Workshop Video Surveillance Sens. Netw.*, 2004, pp. 131–135.

[5] A. C. Shastry and R. A. Schowengerdt, "Airborne video registration and traffic-flow parameter estimation," *IEEE Trans. Intell. Transp. Syst.*, vol. 6, no. 4, pp. 391–405, Dec. 2005.

[6] H. Cheng and J. Wus, "Adaptive region of interest estimation for aerial surveillance video," in *Proc. IEEE Int. Conf. Image Process.*, 2005, vol. 3, pp. 860–863.

[7] S. Hinz and A. Baumgartner, "Vehicle detection in aerial images using generic features, grouping, and context," in *Proc. DAGM-Symp.*, Sep. 2001, vol. 2191, Lecture Notes in Computer Science, pp. 45–52.

[8] H. Cheng and D. Butler, "Segmentation of aerial surveillance video using a mixture of experts," in *Proc. IEEE Digit. Imaging Comput.—Tech. Appl.*, 2005, p. 66.

[9] R. Lin, X. Cao, Y. Xu, C. Wu, and H. Qiao, "Airborne moving vehicle detection for urban traffic surveillance," in *Proc. 11th Int. IEEE Conf. Intell. Transp. Syst.*, Oct. 2008, pp. 163–167.

[10] L. Hong, Y. Ruan, W. Li, D. Wicker, and J. Layne, "Energy-based video tracking using joint target density processing with an application to unmanned aerial vehicle surveillance," *IET Comput. Vis.*, vol. 2, no. 1, pp. 1–12, 2008.

[11] R. Lin, X. Cao, Y. Xu, C. Wu, and H. Qiao, "Airborne moving vehicle detection for video surveillance of urban traffic," in *Proc. IEEE Intell. Veh. Symp.*, 2009, pp. 203–208.

[12] J. Y. Choi and Y. K. Yang, "Vehicle detection from aerial images using local shape information," *Adv. Image Video Technol.*, vol. 5414, Lecture Notes in Computer Science, pp. 227–236, Jan. 2009.

[13] C. G. Harris and M. J. Stephens, "A combined corner and edge detector," in *Proc. 4th Alvey Vis. Conf.*, 1988, pp. 147–151.

[14] J. F. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.

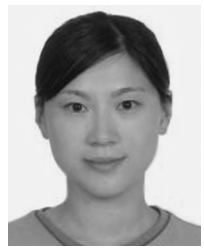
[15] W. H. Tsai, "Moment-preserving thresholding: A new approach," *Comput. Vis. Graph., Image Process.*, vol. 29, no. 3, pp. 377–393, 1985.
[16] L. W. Tsai, J. W. Hsieh, and K. C. Fan, "Vehicle detection using normalized color and edge map," *IEEE Trans. Image Process.*, vol. 16, no. 3, pp. 850–864, Mar. 2007.
[17] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge, U.K.: Cambridge Univ. Press, 2000.
[18] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 2003.

Hsu-Yung Cheng (M'08) was born in Taipei, Taiwan. She received the B.S. degree in computer science and information engineering and M.S. degree from the National Chiao Tung University, Hsinchu, Taiwan, in 2000 and 2002, respectively, and the Ph.D. degree in electrical engineering from the University of Washington, Seattle, in 2008.

Since 2008, he has been an Assistant Professor with the Department of Computer Science and Information Engineering, National Central University, Chung-li, Taiwan. Her research interest includes image and video analysis and intelligent systems.

Chih-Chia Weng was born in Taiwan. He received the B.S. degree in computer science and information engineering and the M.S. degree from Chung Cheng Institute of Technology of National Defense University, Taoyuan, Taiwan. He is currently working toward the Ph.D. degree in the Department of Computer Science and Information Engineering, National Central University, Chungli, Taiwan.

His research interest includes image processing and machine intelligence.



Yi-Ying Chen was born in Taiwan. She received the B.S. degree in computer science from the National Taipei University of Education, Taipei, Taiwan, and the M.S. degree from the National Central University, Chungli, Taiwan.

Her research interest includes signal and image processing.