World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:6, No:5, 2012

Detecting and Tracking Vehicles in Airborne Videos

Hsu-Yung Cheng and Chih-Chang Yu

***Abstract*—**In this work, we present an automatic vehicle detection system for airborne videos using combined features. We propose a pixel-wise classification method for vehicle detection using Dynamic Bayesian Networks. In spite of performing pixel-wise classification, relations among neighboring pixels in a region are preserved in the feature extraction process. The main novelty of the detection scheme is that the extracted combined features comprise not only pixel-level information but also region-level information. Afterwards, tracking is performed on the detected vehicles. Tracking is performed using efficient Kalman filter with dynamic particle sampling. Experiments were conducted on a wide variety of airborne videos. We do not assume prior information of camera heights, orientation, and target object sizes in the proposed framework. The results demonstrate flexibility and good generalization abilities of the proposed method on a challenging dataset.

***Keywords*—**Vehicle Detection, Airborne Video, Tracking, Dynamic Bayesian Networks

I. Introduction

DETECTING vehicles is an important part in airborne video analysis. The challenges of vehicle detection in airborne videos include camera motions such as panning, tilting and rotation. In addition, airborne platforms at different heights result in different sizes of target objects. Lin et al. [1] 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 focus of the airborne camera. Assuming these parameters as known priors might not be realistic in real applications. In [2], 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. Also, multi-scale 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. But if faces with poses are added as positive samples, the number of false alarms would surge. Choi and Yang [3] proposed a vehicle detection algorithm using the symmetric property of car shapes.

Hsu-Yung Cheng is with Department of Computer Science and Information Engineering, National Central University. No.300, Jhongda Rd., Jhongli City, Taoyuan County 32001, Taiwan. (Tel: +8864227151- 35306, Email: chengsy@csie.ncu.edu.tw)

Chih-Chang Yu is with Department of Computer Science and Information Engineering, Vanung University, Taiwan (Email: tacoyu@mail.vnu.edu.tw)

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 designed to be obtained from a fixed vehicle model, making the algorithm inflexible. Moreover, the algorithm in [3] 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. The high computational complexity of mean-shift segmentation algorithm is another concern.

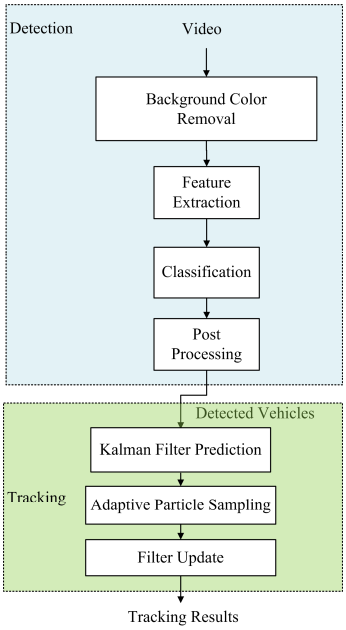


Fig. 1 Proposed system framework

In this work, we design a vehicle detection framework that preserves the advantages of the existing works and avoids their drawbacks. The contribution of the proposed framework is that the detection task is based on pixel-wise classification. However, the features are extracted in a neighborhood region of each pixel. Such design is more effective and efficient than region-based [3] or multi-scale sliding window detection methods [2]. The proposed system framework is illustrated in Fig. 1. 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 [1]. Afterwards, 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. Finally, post processing is performed to eliminate objects that are impossible to be vehicles after morphological operations. The size and aspect ratio constraints are applied in post processing. However, the constraints used here are very loose. The rest of this paper is organized as follows. Section 2 explains the proposed feature extraction process. Section 3 elaborates the pixel-level classification mechanism via DBN. Section explains the tracking process. Section 5 demonstrates and analyzes the experimental results. Section 6 concludes the paper.

II. Feature Extraction

We combine local features and color features for vehicle detection. Local features include corners and edges. We use Harris corner detector [4] to detect corners. To detect edges, we apply moment-preserving thresholding [5] method on classical Canny edge detector [6] to select thresholds adaptively for different scenes. For color features, we transform (R, G, B) color components to (u,v) color domain proposed in [7] to separate vehicle colors from non-vehicle colors effectively. It has been shown [7] in that vehicle colors and non-vehicle colors have less overlapping regions under the (u,v) color model. The color components are converted using Eq. (1) and (2) where $Z = (R+G+B)/3$.

$$u = \frac{2Z - G - B}{Z} \tag{1}$$

$$v = \mathrm{Max}\left\{\frac{B-G}{Z}, \frac{R-B}{Z}\right\} \tag{2}$$

We use a support vector machine (SVM) to classify vehicle colors and non-vehicle colors. When performing SVM training and classification, a block of $n \times m$ pixels is taken as a sample. More specifically, each feature vector is defined as $[u_1, v_1, ..., u_{n\times m}, v_{n\times m}]$. 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 non-vehicle color areas.

The features are extracted in a neighborhood region of each pixel in our framework. Considering an N × N neighborhood Λ_p of pixel p, we extract five types of features S, C, E, A, 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 Eq. (3). Note that $N_{Vehicle\,color}$ denotes to the number of pixels in Λ_p that are classified as vehicle colors by SVM.

$$S = \frac{N_{Vehicle\,color}}{N^2} \tag{3}$$

The features C and E are defined in Eq. (4) and Eq. (5), respectively.

$$C = \frac{N_{Corner}}{N^2} \tag{4}$$

$$E = \frac{N_{Edge}}{N^2} \tag{5}$$

Similarly, N_{Corner} denotes to the number of pixels in Λ_p that are detected as corners by 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 size of the connected vehicle-color region where the pixel p resides.

III. Detection via Classification

We perform pixel-wise classification for vehicle detection using Dynamic Bayesian Networks [8]. The design of the DBN model is illustrated in Fig. 2. The 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}. Also, at each time slice *t*, the 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 sub-section. 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 by providing the ground truth labeling of each pixel and its corresponding observed features from several training videos. In the detection phase, Bayesian rule is used to obtain the probability that a pixel belongs to a vehicle, as shown in Eq. (6).

$$\begin{aligned} &P(V_t \mid S_t, C_t, E_t, A_t, Z_t, V_{t-1}) \\ &= P(V_t \mid S_t)P(V_t \mid C_t)P(V_t \mid E_t)P(V_t \mid A_t)P(V_t \mid Z_t)P(V_t \mid V_{t-1})P(V_{t-1}) \end{aligned} \tag{6}$$

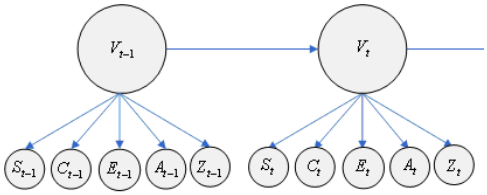


Fig. 2 DBN model for pixel-wise classification

The proposed vehicle detection framework can also utilize 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. We will compare the detection results using BN and DBN in the next section.

IV. Tracking Process

For a vehicle detected, the system initializes its system state $x_k = [u_k\ v_k\ \dot{u}_k\ \dot{v}_k\ a_k\ b_k]^T$ and an appearance model ξ_k for it. Commonly used appearance models are color values of the fitted ellipse (color matrices), and compact summarization of color distribution such as histograms or mixture of Gaussians.

The position (u_k, v_k) is coordinate of the centroid of an object in the image plane. The velocities $\dot{u}_k$ and $\dot{v}_k$ are initialized as zeros. The sizes (a_k, b_k) are the length of the major axis and the minor axis of the ellipse fitted on the vehicle. The measurement state is defined as $y_k = [u_k \; v_k \; a_k \; b_k]^T$. After Kalman Filter prediction, particles are generated if necessary. First, N_p random samples are generated in the four-tuple state space (u_k, v_k, a_k, b_k) around the point $(\hat{u}_{k|k-1}, \hat{v}_{k|k-1}, \hat{a}_{k|k-1}, \hat{b}_{k|k-1})$, where $\hat{u}_{k|k-1}$, $\hat{v}_{k|k-1}$, $\hat{a}_{k|k-1}$, and $\hat{b}_{k||k-1}$ are obtained from the predicted system state $\hat{x}_{k|k-1}$ of the target object at frame k. Then, each particle is associated with a weight. Finally, particles with higher weights are put into the measurement candidate list. After obtaining the measurements in the measurement candidate list, we utilize an enhanced probabilistic data association to update our filter. The details of particle sampling and enhanced probabilistic data association can be found in [9], [10].

V. Experiments

Various video sequences with different scenes and different filming altitudes are used. It is infeasible to assume prior information of camera heights and target object sizes for our challenging dataset. There are total 225025 frames in the dataset. When calculating the detection accuracy, we perform evaluation every 100 frames. When employing SVM, we need to select the block size $n \times m$ to form a sample. We take each 3 × 4 block to form a feature vector for better detection results. For the observed feature of dynamic Bayesian networks, to select the size of the neighborhood area for feature extraction, we plot the detection accuracy using different neighborhood sizes in Fig. 3. The detection accuracy is measured by the hit rate and number of false positives per frame. We can observe that the neighborhood aera Λ_p with size of 7×7 yields the best detection accuracy.

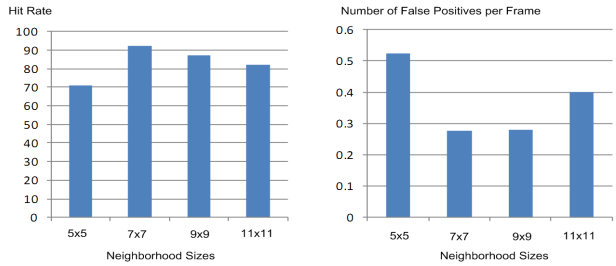
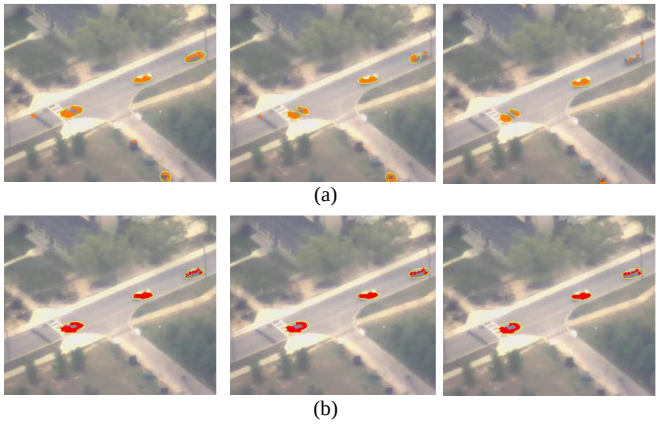


Fig. 3 Hit rates and number of false positives per frame for different neighbor sizes

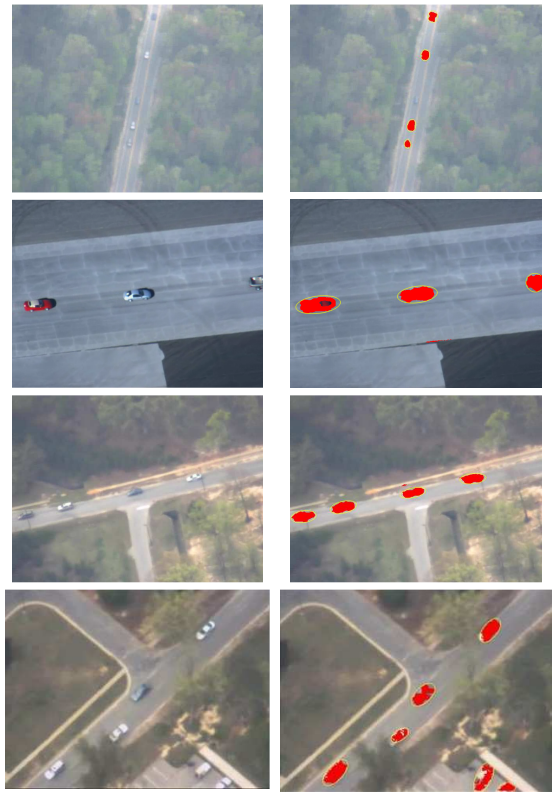
In Fig. 4 we display the detection results using BN (Fig. 4 (a)) and DBN (Fig. 4 (b)). The colored pixels are the ones that are classified as vehicle pixels by BN or DBN. The ellipses are the final vehicle detection results after performing post processing. DBN outperforms BN because it includes information along time. When observing detection results of consecutive frames, we also notice that the detection results via DBN are more stable.



(a)

(b)

Fig. 4 Detection results using (a) BN and (b) DBN.



(a) (b)

Fig. 5 Detection results of different scenes with various camera heights and angles: (a) Original image frames (b) Detected vehicles

In Fig.5, we show selected detection results of surveillance scenes at different camera heights and angles. The sizes and the orientation of the vehicles vary a lot in different scenes. We can observe that the experimental results demonstrate flexibility and good generalization abilities of the proposed method. For tracking experiments, we perform detection every 2 seconds and use the tracking algorithm to track the detected vehicles. The tracking accuracy is 96.52%.

VI. Conclusion

An automatic vehicle detection system for airborne videos using combined features is proposed in this work. We consider

features including vehicle colors and local features. For vehicle color extraction, we utilize a color transform to separate vehicle colors and non-vehicle colors effectively. We use Harris corner detector to detect corners. For edges, we apply moment-preserving thresholding method on classical Canny edge detector to select thresholds adaptively for different scenes. 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 do not perform region-based classification, which would highly depend on computational intensive color segmentation algorithms such as mean-shift. We do not generate multi-scale sliding windows that are not suitable for detecting rotated vehicles, either. We design a pixel-wise classification method for vehicle detection. The novelty lies in that in spite of performing pixel-wise classification, relations among neighboring pixels in a region are preserved in the feature extraction process. The system does not assume any prior information of camera heights, vehicle sizes, and aspect ratios. Performing vehicle tracking on the detected vehicles further stabilizes the detection results. Tracking via dynamic particle sampling is effective on the detected vehicles.

REFERENCES

[1] R. Lin, X. Cao, Y. Xu, C. Wu, and H. Qiao, "Airborne moving vehicle detection for urban traffic surveillance," *Proceedings of the 11th International IEEE Conference on Intelligent Transportation Systems*, Oct. 2008, pp. 163-167.

[2] R. Lin, X. Cao, Y. Xu, C. Wu, and H. Qiao, "Airborne moving vehicle detection for video surveillance of urban traffic," *IEEE Intelligent Vehicles Symposium*, 2009, pp. 203-208.

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

[4] C.G. Harris and M.J. Stephens, "A combined corner and edge detector," *Proceedings of the 4th Alvey Vision Conference*, 1988, p.147-151.

[5] W.H. Tsai, "Moment-preserving thresholding: a new approach," *Computer Vision Graphics, and Image Processing*, vol. 29, no. 3, pp. 377-393, 1985.

[6] J.F.Canny, "A Computational Approach to Edge Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*," vol. 8, no. 6, pp. 679-698, 1986.

[7] L.W. Tsai, J.W. Hsieh, and K.C. Fan, "Vehicle detection using normalized color and edge map," *IEEE Trans. on Image Processing*, vol. 16, no. 3, 2007.

[8] S. Russell, P. Norvig, "Artificial intelligence: a modern approach (second edition)," *Prentice Hall*, 2003.

[9] H. Y. Cheng and J. N. Hwang, "Multiple-target tracking for crossroad traffic utilizing modified probabilistic data association," *2007 IEEE International Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Honolulu, Hawaii, (2007).

[10] H. Y. Cheng and J. N. Hwang, "Adaptive particle sampling and adaptive appearance for multiple video object tracking," *Signal Processing*, vol. 89, no. 9, pp. 1844-1849, Sep. 2009.