Using Color Profiles for Street Detection in Low-Altitude UAV Video

J. Candamo, R. Kasturi, D. Goldgof
Computer Science Department, University of South Florida, 4202 E. Fowler Ave., ENB 118,
Tampa, FL USA 33620-5399

ABSTRACT

This paper describes a vision-based street detection algorithm to be used by small autonomous aircraft in low-altitude urban surveillance. The algorithm uses Bayesian analysis to differentiate between street and background pixels. The color profile of edges on the detected street is used to represent objects with respect to their surroundings. These color profiles are used to improve street detection over time. Pixels that do not likely originate from the "true" street are excluded from the recurring Bayesian estimation in the video. Results are presented comparing to a previously published Unmanned Aerial Vehicle (UAV) road detection algorithm. Robust performance is demonstrated with urban surveillance scenes including UAV surveillance, police chases from helicopters, and traffic monitoring. The proposed method is shown to be robust to data uncertainty and has low sensitivity to the training dataset. Performance is computed using a challenging multi-site dataset that includes compression artifacts, poor resolution, and large variation of scene complexity.

Keywords: Unmanned Aerial Vehicles, UAV, Street Detection, Road Detection, Surveillance

1. INTRODUCTION

Unmanned Aerial Vehicles (UAV) and other small robots are increasingly being used in low-altitude urban environments as mobile reconnaissance platforms [1], especially in areas where there is significant safety risk to humans. Moving cameras and mobile surveillance platforms are yet to become an important player in transit surveillance of urban scenes. With much research and commercial interest in UAV and mobile surveillance, current solutions are not far from being usable as efficient surveillance platforms. As aerial surveillance has gained increased interest within the research community, authors have proposed techniques to detect low resolution vehicles [2] and buildings [3] from aerial images. Furthermore, researchers have already started looking at the problem of security-sensitive event detection using surveillance video from UAV [4].

This work focuses on the domain of low-altitude UAV urban surveillance, which is a relative new area of work typically oriented towards search and rescue and disaster response applications. Two key components of urban surveillance with extensive coverage over the last decade in the research literature applications are street [5] and vehicle detection [6]. However, it is unclear as to how reliable or feasible existing video processing methods are in highly unstructured domains, where training data is scarce and often unreliable. For the empirical analysis presented in this paper, another highly unstructured domain is considered: "Found Videos." Found videos is a term commonly used within the intelligence community for those videos that are freely available for download from the Internet. Fueled by the increasing popularity of websites for user-submitted videos, the internet has become a vast source of untapped information for intelligence and security applications. Driven by the public availability of found videos, the computer vision community is already active in areas such as found video categorization [7] and scene retrieval [8].

We propose the use of object boundaries and their surroundings to create a highly adaptable street detection system, able to cope with a great variety of scene content and quality. The notion of studying objects based on their contours is nothing new [9]. A common way to produce object boundaries is edge detection. Edge detection computes regions of abrupt change in low-level image features such as brightness and color. In general, traditional vision-based techniques that study object boundaries or edges leave aside contextual information. But, leveraging context to improve object detection performance is starting to motivate new directions of research [10]. In this work we use contextual information of edges within a street in order to improve street detection over time. The proposed method is compared to a previously published UAV street detection algorithm [11].

2. ALGORITHM OVERVIEW

First, a Bayes classifier is used to compute an initial street region. Recursive Bayesian estimation is used to incorporate temporal information into the classifier. Edge detection is used to identify boundaries of objects within the detected street. For each edge, the color profile (in HSV color space) is estimated. The color profile is a representation of the object (from which the edge originates) and its surroundings. This color profile is used to discriminate pixels originating from street markings, vehicles, and other objects that do not truly correspond to the street. Those pixels are excluded from the recurring Bayesian analysis; thereby, improving the reliability of the parameters used in the street classifier.

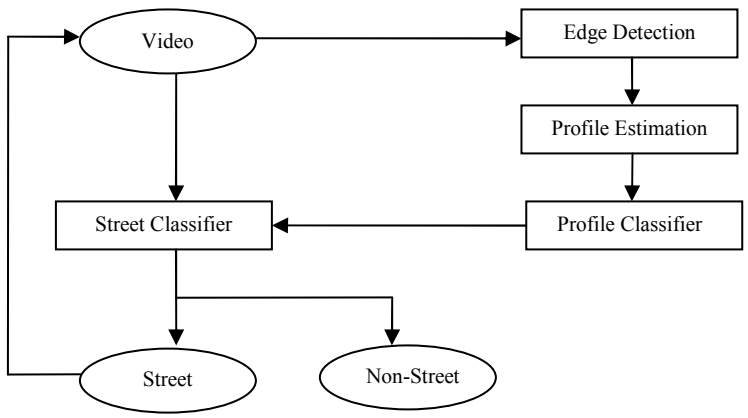


Figure 1. Street detection algorithm flowchart.

3. STREET DETECTION

3.1 Bayesian Analysis

Bayesian analysis is a technique that studies how to change existing beliefs based on new found evidence. This technique exploits the following mathematical rule, commonly known as Bayes rule:

$$P(s \mid c) = \frac{P(c \mid s)P(s)}{P(c)} \tag{1}$$

Where c and s are two (possibly dependant) random events. The posterior belief $P(c|s)$ is computed by multiplying the prior belief $P(s)$ by the likelihood $P(c| s)$, i.e. probability that event c will occur if s is true. Equation (1) can be viewed as changing the belief about hypothesis s in the light of the new evidence c. Bayes classifier has been shown to be an effective classification tool for street detection using UAV images [11], due to its robustness and fast computational speed.

3.2 Street Detection

At the pixel level, the problem of street detection can be described as follows: given a pixel with color c (a HSV triplet), we want to find the probability $P(s|c)$ that the pixel belongs to the street. Since the posterior belief $P(s|c)$ is difficult to compute directly, Bayes rule can be used instead. The prior distribution of a pixel belonging to the street $P(s)$ in an image can be computed somewhat reliably with training data. However, the prior distribution $P(c)$ of the color c can not. Nevertheless, we do not need to compute $P(s|c)$ since we are only interested in knowing if the pixel belongs to the street or not. This can be done by computing the likelihood ratio given by:

$$L(street \mid c) = \frac{P(c \mid s)P(s)}{P(c \mid n)P(n)} \tag{2}$$

Where s and n represent street and non-street respectively. $P(c|s)$ and $P(c|n)$ can be approximated by the normalized distances from each pixel color in an image to the street and the non-street respectively. The approximated likelihood function is given by:

$$L(street \mid c) = \left(\frac{\| c - \mu_s \| / | \Sigma_s |}{\| c - \mu_n \| / | \Sigma_n |} \right) \left(\frac{P(s)}{P(n)} \right) \quad \textbf{(3)}$$

Where μ and $\sum$ represents the color mean and covariance matrix respectively. Next, to binarize the likelihood map a local threshold [12][13] is used. The threshold is based on the local likelihood's mean m and standard deviation d:

$$T = m(1 - d / 2) \quad \textbf{(4)}$$

Once the likelihood map has been binarized, connected component analysis is used to remove noise, and a 2-pass hole filling is used to eliminate gaps in the street. However, if the difference between the filled and unfilled street is bigger than a proportion factor of the total image, filling is skipped in order to avoid merging unrelated regions.

3.3 Improving Street Detection Using Video

Sensitivity of Bayesian analysis to changes in prior distributions has been studied [14] by analyzing the difference between posterior means for different priors. Studies have shown that it is especially problematic to have little or erroneous information available to construct the priors. Thus, there is great motivation behind finding reliable priors. Only when the priors chosen are close enough to the "true prior," is the Bayes decision function best among all other estimators [15]. In a street detection problem the prior $P(s)$ can greatly vary depending on the content of the video. However, in general the street view in an image will not change significantly from frame to frame. Consequently, the temporal consistency in the street view can be exploited by updating the street distribution parameters in time, i.e. the likelihood classifier is improved by integrating new frames into the parameter estimation. Let regions labeled street over t frames be Gaussian distributions with mean μ_i and covariance matrix $\sum_i$, for i=1,..,t. The second moment for the street at the ith frame is:

$$E\left[X_i^2\right] = \Sigma_i + \vec{\mu}_i^{\,T} \vec{\mu}_i \quad \textbf{(5)}$$

The composite street region is defined as the joint distribution G_s. The composite street mean is given by:

$$\vec{\mu}_s = \sum_{i=1}^{t} w_i \vec{\mu}_i \quad \textbf{(6)}$$

Where w is a weighting function (w_i=1/t for i=1,…,t for an equal-weight joint distribution). Unlike the mean, the covariance does not combine linearly, but the second moments about the origin do [16]. The joint second moment is:

$$E\left[X_s^2\right] = \sum_{i=1}^{t} w_i E\left[X_i^2\right] \quad \textbf{(7)}$$

Consequently, the joint covariance matrix is:

$$\Sigma_s = E\left[X_s^2\right] - \vec{\mu}_s^{\,T} \vec{\mu}_s \quad \textbf{(8)}$$

After the street (G_s) parameters are found, we propose to use information of objects that do not belong to the street to further improve the reliability of these parameters over time. In the next section, we describe how color profiles of objects detected within the street can be used to update the estimation parameters.

4. DETECTING OBJECTS ON THE STREET

4.1 Edge Detection and Curve Fitting

From each video frame, an edge map is computed using Canny edge detector (Figure 2-a). Canny edge detector was chosen due to its robustness under most conditions (e.g. noise, scene complexity, etc) [17]. Edge detector reliability is a desirable property when dealing with possibly weak training data, such as one you would expect from highly uncertain domains like UAV surveillance. The edge detector uses adaptive thresholds as described in [18], and a constant width of the Gaussian mask (σ=1). A logical operation is used to keep only edges on the detected street (Figure 2-b,c).

(a) Edge Map. (b) Binary Street Map. (c) Reduced Edge Map.

Figure 2. (a) Edges resulting from UAV video shown in Figure 7-b. (b) Detected binary street map. (c) A logical operation using (a) and (b) discards all edges outside of the detected street.

In general, there is a relatively high contrast between common objects in street (e.g. vehicles, traffic signals, etc) and their background. Thus, it is assumed that all objects present in the street will have some corresponding edges in the reduced edge map (Figure 2-c). Often, edges originating from the street will be connected to edges originating from other objects. For separation purposes, edges are broken into smaller segments by discarding all edge pixels on a squared grid from further processing, i.e. all pixels along a grid such as the one shown in Figure 3 are discarded.

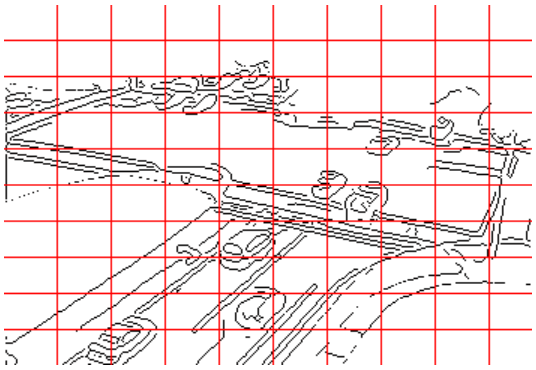
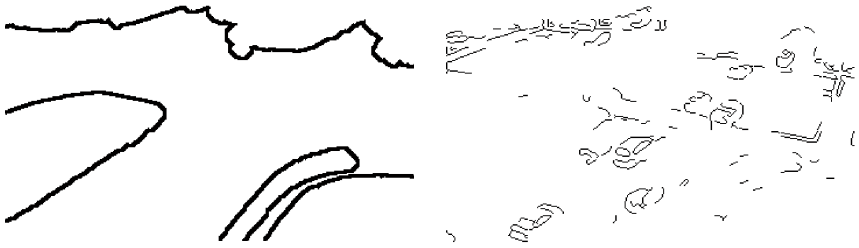


Figure 3. A separation grid is used to minimize connectedness of edges that originate from both the street and other objects.

Edges corresponding to the street boundary are discarded. The street boundary (Figure 4-a) is computed by subtracting the binary street (Figure 2-b) from an eroded street map. Additionally, small and linear-shaped edges are eliminated by using connected component analysis, size and eccentricity filters. In [19], a similar technique was used in order to keep only large linear-shaped objects in a wire detection algorithm for UAV urban surveillance. Linear shaped edges are likely to correspond to markings commonly found on the street.



(a) Street Boundary. (c) Candidates for Non-Street Edges.

Figure 4. Small, linear-shaped, and delimiting street edges are discarded to generate a first set of vehicle edge candidates.

Each remaining edge is represented by a curve, which is described by a polynomial:

$$fit(z) = a_0 + a_1 z + ... + a_n z^n \quad \textbf{(9)}$$

Polynomial fitting is done through regression for degrees 1 through 5, sequentially incrementing the degree of the polynomial while providing a sufficient error improvement over the last fit. The fit error is computed comparing the fit with respect of the originating pixels. After having a representation for all edges, the color profile is used to discard edges that most likely originated from the street.

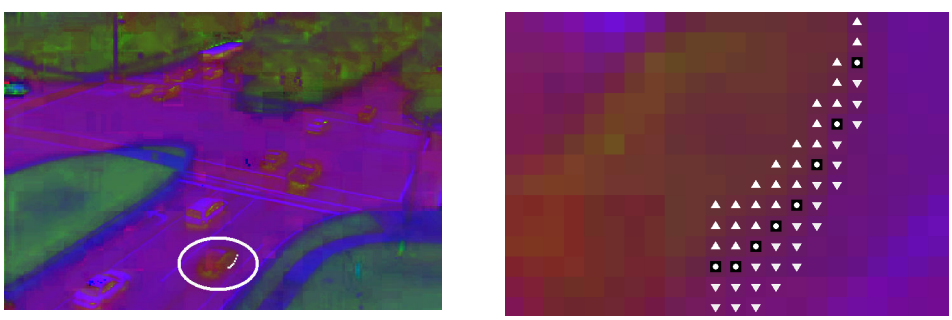
4.2 Color Profile Estimation

The color profile of a line represents the color surface of the line and both surrounding regions (regions at each side of the line). In the case of edges, the color profile is a representation of the color of the object it originates from and its surroundings. A profile model using a mixture of Gaussians has been previously used to detect wires [20] in UAV images. Gaussian mixture models are commonly used in computer vision, and they have proven particularly effective in image segmentation [21] and background modeling [22]. In this work, we use the online means approximation algorithm for profile estimation proposed in [19], modified to use curves instead of straight lines and color instead of gray levels.

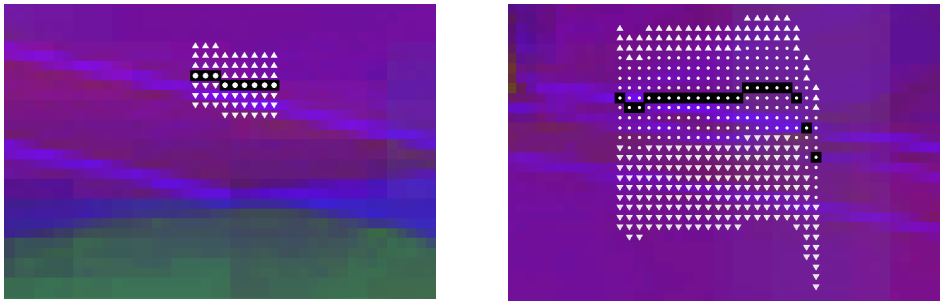
In general, the color profile of a line can be described by three regions. Each color region can be modeled by a multivariate Gaussian distribution. In other words, a color profile is composed of 3 regions. Each region can be represented by the color mean and covariance, and classified as street or non-street based on training data. Due to the broad nature of the problem, collecting training data for an entire visual scene containing all possible street and non-street regions is a very hard and perhaps impractical task. On the other hand, collecting a robust set of street and non-street pixels from traffic scenes is an easier task. In general, street regions in traffic scenes will consist of groups of low-saturated pixels clustered together. So, reducing the complexity of the problem allows the use of a simple K-Nearest Neighbors (KNN) pixel classifier. In our context, the KNN classifier classifies pixels based on the K closest training examples in the color space. However, spatial relationships are not used in a pixel classification strategy like this. So, pixel classification is done within a profile region. A street region is defined if the majority of pixels are classified as street pixels. And, street markings are defined as those profiles composed of edges surrounded by street regions. In Table 1, a KNN (K=3) street pixel classifier is shown to be relatively accurate (accuracy is the proportion of the total number of predictions that were correct), but not enough to be used as a stand alone street detector.

Table 1. Street profile classifier accuracy on the training data.

Features	True Positive	False Positive	False negative	True Negative	Accuracy %
HSV triplet	2316	55	31	1043	97.50



(a) (b)



(c) (d)

Figure 5. (a) Vehicle edge (marked as white pixels) in HSV color image from the scene shown in Figure 7. (b) Resulting fitted curve pixels (shown as black squares), and corresponding profile regions representing the curve (white circles), above (white triangles), and below (white inverted triangles). (c-d) Examples of profiles from edges originating from the street.

4.3 Vehicle Pixel Detection

At this point, all remaining edges (Figure 6-b) potentially originated from non-street regions. Next, blobs are created using morphological dilation on the pixel map containing all pixels used in the computation of the color profiles (Figure 6-c). Since most objects found on the street are vehicles, a state-of-the-art vehicle pixel detector for traffic images [23] is used. The vehicle detector discriminates vehicle pixels from colored backgrounds only for those pixels corresponding to the computed blobs. Next, we describe how the vehicle pixels are used to update the parameters of the Bayes street classifier.

(a) Original Frame. (b) Detected Vehicle Edges. (c) Blobs.

Figure 6. Sample vehicle blob detection.

4.4 Improving Street Detection Using Vehicle Pixels

Similarly to the recursive Bayesian estimation approach described in Section 3.3, the street map is updated based on the vehicle pixels found within the street. However, in this case we are not combining two similar street maps. Instead, we are only incorporating the information of a bunch of pixels identified as not belonging to the street. In this case the weighting function becomes a learning rate [22]. We use a learning rate function based on the number of vehicle pixels found:

$$\alpha = f\left(\frac{pixels(non\text{-}street)}{pixels(street)}\right) \tag{10}$$

Where the function f is determined empirically, and ranges between 0 and 1. This learning rate determines the speed at which the distribution's parameters change. The estimation process would be similar to equations (5-8), but with only two Gaussians representing the street and non-street pixels instead of one Gaussian for each frame. Also, the weighting function is α for the street and (1- α) for the non-street pixels. The updated street region is now considered our best estimate for the street, and will be carried over to future frames in the recurring estimation process. Two sample improvements for street detection over time are shown in Figure 7, using Markovian recurring Bayesian estimation (memoryless system) with equal weights.

Figure 7. Street detection improvement over time. In this scene, the sky is segmented out from the street. Improvement is achieved by exploiting temporal consistency of the street in consecutive frames, and using learning rate functions to control the speed of change.

5. ALGORITHM PARAMETERS

All the algorithm's parameters are summarized in Table 2**.** The table consists of the parameter's name, short description, symbol, section in which they are described, and value after training. All parameters marked with an asterisks (*) were trained together.

Table 2. Algorithm parameter summary. Parameters marked with a (*) were trained together, all other were trained separately.

Parameter	Description	Symbol	Sect.	Value
Prior	Street and non-street Gaussians	$P(s)$, $P(n)$	3.2	0.32 , 0.68
Mean	Street and non-street Gaussians	M_s , μ_n	3.2	[0.280 0.137 0.563] [0.284 0.253 0.535]
Covariance	Street and non-street Gaussians	$\sum_s$, $\sum_n$	3.2	[0.056 -0.003 -0.001 -0.003 0.006 -0.0004 -0.001 -0.0004 0.0125] [0.027 -0.004 0.003 -0.004 0.034 -0.038 0.003 -0.0383 0.0848]
Likelihood threshold	Classifies likelihood map as street or not	T	3.2	Adaptive as in [12]
*Size filter threshold (regions)	Discard small regions (% image pixels)	-	3.2	1.5%
*Merging proportion factor	% of pixels in image	-	3.2	50%
Canny edge detector	Edge detector	-	4.1	Adaptive as in [18] and σ=1
*Component separation grid	Image is divided in 10^2 cells	-	4.1	10
*Size filter threshold (edges)	Discard small edges	-	4.1	8px
Eccentricity filter threshold	Discard linear shaped edges	-	4.1	0.99 as in [19]
*Learning rate modifier	rate of change based on vehicle pixels	f	4.4	$[\text{pixels(non-street)}/\text{pixels(street)}]^3$

6. DATASETS AND PERFORMANCE METRICS

Two test datasets are used: one with 15 minutes of UAV traffic monitoring videos and one with 40 sequences of traffic surveillance found videos. Found videos are those videos that can be freely downloaded from the Internet. The found videos data consists of helicopter police chases and overhead camera traffic monitoring scenes. The datasets are quite challenging, including scaling, compression artifacts, camera jitter, zoom, and a wide range of scene content and visual complexity. Manual ground truth was created for the street in selected frames. A set of images downloaded from the Internet was collected separately and used as training data. The training data consists of 16 traffic images, with 341196 manually marked pixel blobs for streets (32% of samples) and non-street regions (68% of samples). These pixels were used to train the Bayes street classifier (Section 3.2), and the nearest neighbor street profile classifier (Section 4.2).

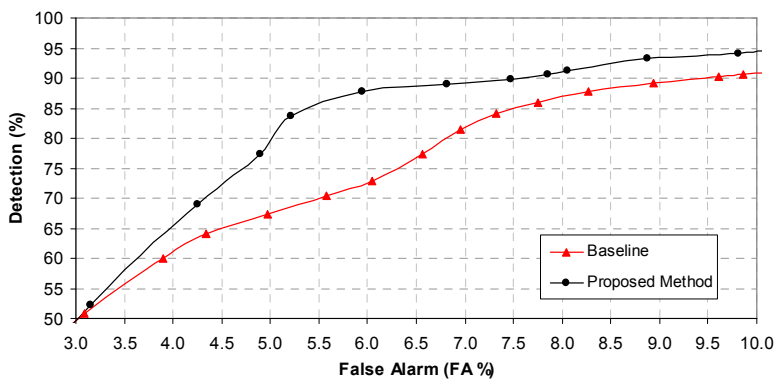
In the results shown in Section 7, detection and false alarm rates are computed using:

$$detection = \frac{correct\ pixels}{total\ street\ ground\ truth\ pixels} \tag{11}$$

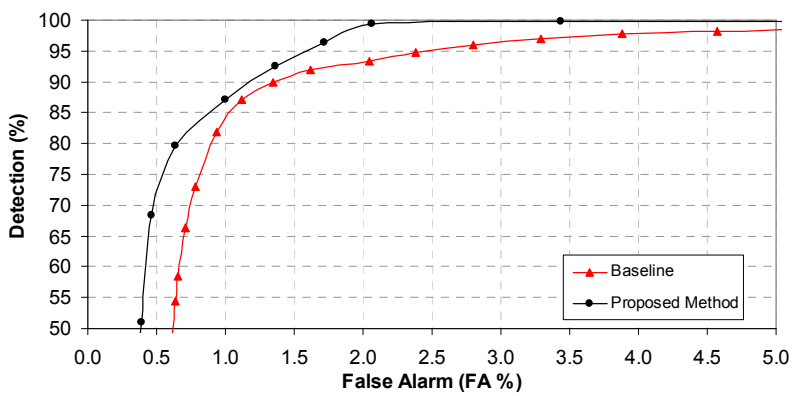
$$false\ alarm = \frac{false\ alarm\ pixels}{total\ image\ pixels} \tag{12}$$

7. RESULTS

Figure 8 shows the receiver operating characteristic curve (ROC) for the proposed street detection method with operationally relevant detection and false alarm (FA) rates. The ROC is generated by introducing a constant to (4). Performance is compared to the baseline algorithm described in [11], which relies in a traditional Bayes classifier. Clearly, as depicted in both charts in Figure 8, performance greatly varies depending on the scene content and quality of the data. Comparing performance on both of our datasets, i.e. UAV and found videos, classification drops over 60% based on the area under the ROC. This indicates a high level of scene content variation and video quality on both datasets. However, performance is still practical for operationally relevant scenarios across the entire data. The proposed street detection method shows a performance improvement over the baseline of 21% and 13% in the found videos and UAV dataset respectively.



(a) Performance on the found videos dataset.



(b) Performance on the UAV dataset.

Figure 8. Street detection's receiver operating characteristic (ROC) curve comparing to a baseline UAV road detection algorithm.

Figure 9 shows sample detection results using images with significantly different scene content. It is clear that these images have compression artifacts, different lighting effects, poor resolution, and a variety of traffic (i.e. number of vehicles). In the selected frames, consistent robust detection results are shown. As depicted in Figure 9, some miss detections occur where vehicles are present near the street boundaries. False alarms are likely to occur in low-saturated color regions, such as the grayish colored buildings in Figure 9-c.

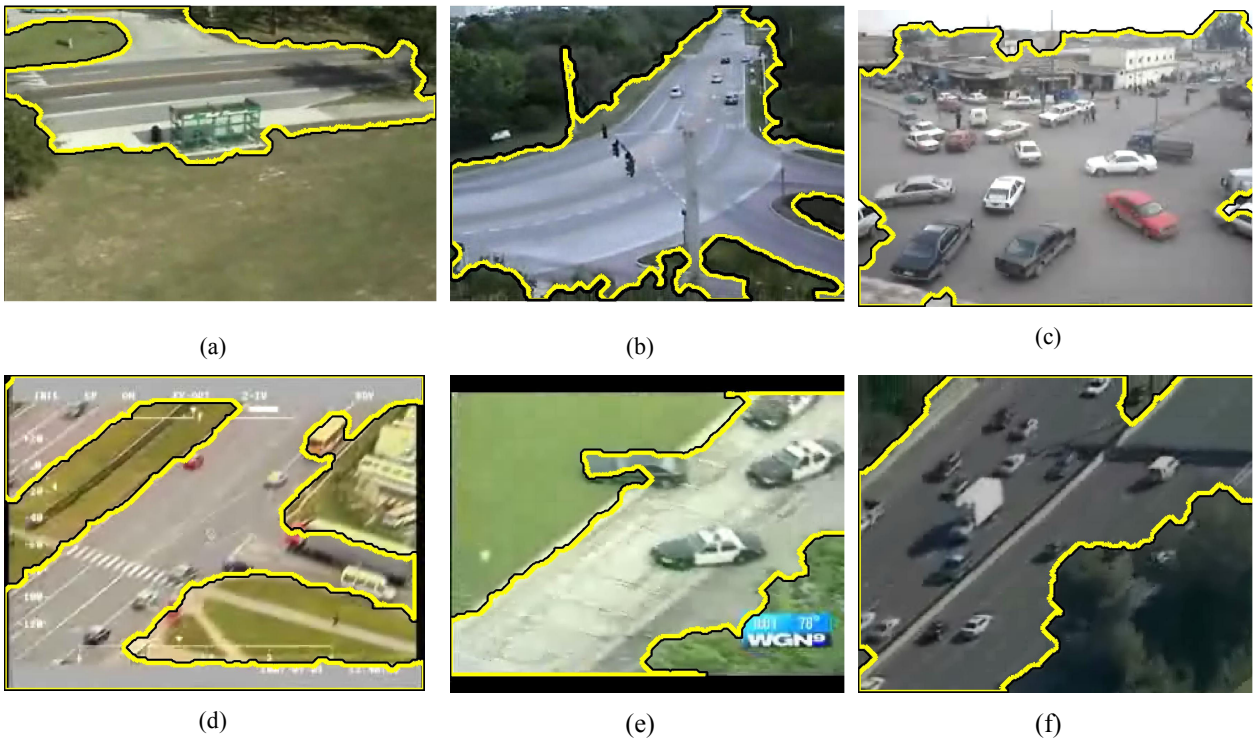


Figure 9. Sample street detection results. (a-b) Frames from UAV dataset. (c-f) Frames from found videos dataset.

8. CONCLUSIONS

As small unmanned aerial vehicles (UAV) are increasingly being used for surveillance applications, it is unclear how accurate and practical traditional vision-based algorithms might be. Especially uncertain are processing methods that require large amounts of training efforts and data. In this paper, a Bayes classifier is used to detect street regions. Vehicles are identified within the detected street and represented using color profiles. Color profiles are used as color descriptors for the object as well as its surroundings. These profiles are incorporated into a recurring Bayes estimation strategy that improves the street classifier over time, by taking into account vehicle pixels in the computation of classification parameters. The system has been successfully used with low quality videos from UAV traffic monitoring, helicopters, and overhead cameras. Encouraging results suggest significant reliability improvement compared to a non-recurring Bayesian estimation. Results show up to 21% area under the ROC improvement for operationally relevant detection and false alarm rates.

9. ACKNOWLEDGEMENTS

This work was partially supported by the Computational Tools for Discovery Thrust and by the Center of Urban Transportation and Research at University of South Florida under Grant BD549-49.

REFERENCES

[1] R.R. Murphy, "Trial by fire [rescue robots]," *IEEE Robotics Automation Magazine,* vol. 11, no. 3, pp. 50-61, 2004.
[2] T. Zhao and R. Nevatia, "Car Detection in Low Resolution Aerial Images," *Image and Vision Computing*, no. 8, pp. 693-703, 2003.

[3] Z.W. Kim and R. Nevatia, "Automatic description of complex buildings from multiple images," *Computer Vision and Image Understanding*, vol. 96, no. 1, pp. 60-95, 2004.
[4] G. Medioni, R. Nevatia, and I. Cohen, "Event Detection and Analysis from Video Streams," *IEEE Trans. Pattern Analysis and Machine Intelligence*, pp. 873-889, 2001.
[5] J. Huang, B. Kong, B. Li, and F. Zheng, "A New Method of Unstructured Road Detection Based on HSV Color Space and Road Features," *IEEE Int. Conf. Information Acquisition*, pp. 596-601, 2007.
[6] V. Kastrinaki, M. Zervakis, and K. Kalaitzakis, "A survey of video processing techniques for traffic applications," *Image and Vision Computing*, vol. 21, no. 4, pp. 359-381, 2003.
[7] G. Schindler, L. Zitnick, and M. Brown, "Internet video category recognition," *IEEE Computer Vision and Pattern Recognition Workshops*, 23-28, pp. 1-7, 2008.
[8] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," *Int. Conf. on Computer Vision*, pp. 1470-1477, 2003.
[9] D.R. Martin, C.C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 5, pp.530-549, 2004.
[10] S.J. McKenna and H. Nait-Charif, "Learning spatial context from tracking using penalized likelihoods," *Seventeenth Int. Conf. Pattern Recognition*, vol. 4, pp.138-141, 2004.
[11] E. Frew, T. McGee, Z. Kim, X. Xiao, S. Jackson, M. Morimoto, S. Rathinam, J. Padial, and R. Sengupta, "Vision-based Road Following Using a Small Autonomous Aircraft," *Proc. IEEE Aerospace Conference*, vol. 5, pp. 3006-3015, 2004.
[12] W. Niblack, "*A*n Introduction to Digital Image Processing," *Englewood Cliff, NI: Prentice Hall*, pp. 115-116, 1986.
[13] O.D. Trier and T. Taxt, "Evaluation of Binarization Methods for Document Images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 17, no. 3, pp. 312-315, 1995.
[14] S.D. Hill and J.C. Spall, "Sensitivity of a Bayesian analysis to the prior distribution," *IEEE Trans. Systems, Man and Cybernetics*, vol. 24, no. 2, pp. 216-221, 1994.
[15] G.C. Canavos, "Bayesian estimation: A sensitivity analysis," *NASA Langley Research Center*, vol. 22, no. 3, pp. 543-552, 2006.
[16] M.E. Johnson, "Multivariate Statistical Simulation," *Wiley Series in Probability and Mathematical Statistics*, New York, 1987.
[17] K. Bowyer, C. Kranenburg, and S. Dougherty, "Edge detector evaluation using empirical ROC curves," *IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 354-359, 1999.
[18] J. Lu, J. Ren, Y. Lu, X. Yuan, and C. Wang, "A Modified Canny Algorithm for Detecting Sky-Sea Line in Infrared Images," *Syst. Design App.*, vol. 2, pp. 289-294, 2006.
[19] J. Candamo, D. Goldgof, and R. Kasturi, "Detecting Wires in Clutter Using a Profile Gaussian Mixture Model," *Trans. Aerospace and Electronic Systems, In Review,* 2009.
[20] J. Candamo and D. Goldgof, "Wire detection in low-altitude, urban, and low-quality video frames," *Nineteenth Int. Conf. Pattern Recognition*, pp. 1-4, 2008.
[21] C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Blobworld: image segmentation using expectation-maximization and its application to image querying," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 8, pp. 1026-1038, 2002.
[22] C. Stauffer and W.E.L. Grimson, "Adaptive Background Mixture Models for Real-Time Tracking," *IEEE Conf. Computer Vision and Pattern Recognition*, 1999.
[23] 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, pp. 850-864, 2007.