2011 IEEE International Conference on Robotics and Automation
Shanghai International Conference Center
May 9-13, 2011, Shanghai, China

Vehicle Detection from Aerial Imagery

Joshua Gleason, Ara V. Nefian, Xavier Bouyssounousse, Terry Fong and George Bebis

Abstract— **Vehicle detection from aerial images is becoming an increasingly important research topic in surveillance, traffic monitoring and military applications. The system described in this paper focuses on vehicle detection in rural environments and its applications to oil and gas pipeline threat detection. Automatic vehicle detection by unmanned aerial vehicles (UAV) will replace current pipeline patrol services that rely on pilot visual inspection of the pipeline from low altitude high risk flights that are often restricted by weather conditions. Our research compares a set of feature extraction methods applied for this specific task and four classification techniques. The best system achieves an average 85% vehicle detection rate and 1800 false alarms per flight hour over a large variety of areas including vegetation, rural roads and buildings, lakes and rivers collected during several day time illuminations and seasonal changes over one year.**

I. INTRODUCTION

Vehicles and heavy digging equipment in particular pose a potentially catastrophic threat to the vast network of oil and gas pipelines in rural areas. Current aerial patrol pilots determine these threats while maintaining the airplanes at a safe altitude above the ground. This task becomes particularly difficult in heavy weather conditions and often reduces the frequency of the surveillance flights.

The system described in this paper (Figure 1) is an attempt to allow unmanned airborne vehicles (UAV) flying at higher altitude to automatically detect ground vehicles in rural areas. Our approach uses optical images captured by a nadir looking commercial camera installed on the airplane wing and determines the vehicles location within each of the captured images. The main challenges of the system consist in dealing with 3D image orientation, image blur due to airplane vibration, variations in illumination conditions and seasonal changes.

There is a vast literature on vehicle detection from aerial imagery. Zhao and Nevatia [12] explore a car recognition method from low resolution aerial images. Hinz [6] discusses a vehicle detection system which attempts to match vehicles against a 3D-wireframe model in an adaptive “top-down” manner. Kim and Malik [7] introduce a faster 3D-model based detection using a probabilistic line feature grouping to increase performance and detection speed.

The vehicle detection system described in this paper uses nadir aerial images and compares the experimental results for several feature extraction techniques with strong discriminant power over vehicles and background, and a set of statistical classifiers including nearest neighbor, random forests and support vector machines. The method described in this paper analyzes each location in an image to determine the target presence. Due to the large number of analyzed location and real time requirements the method presented here starts with a fast detection stage that looks for man-made objects and rejects most of the background. The second stage of the algorithm refines the detection results using a binary classifier for vehicle and background.

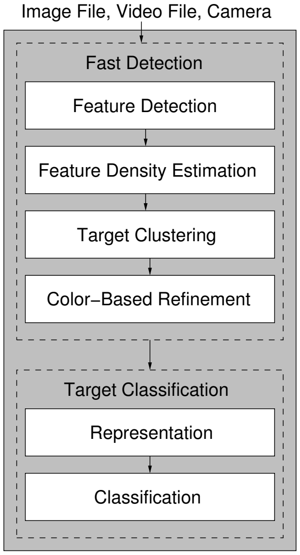


Fig. 1. The overall system.

The paper is organized as follows. Section II describes the fast detection stage, Section III describes the feature extraction and classification techniques, Section IV makes a quantitative comparison of the techniques, and finally Section V presents the conclusion of this work and gives directions for future research.

II. FAST DETECTION

The first stage of the algorithm inspects every image location at several scales and efficiently eliminates the large majority of the background areas. The algorithm begins by quickly detecting features using the Harris corner detector. Next, areas containing a high density of features are detected. The third step clusters heavily overlapping responses. In the final step, color-based properties are used to further refine the results.

This work was not supported by PRCI

Joshua Gleason and George Bebis are with the University of Nevada Reno, gleaso22@gmail.com and bebis@cse.unr.edu

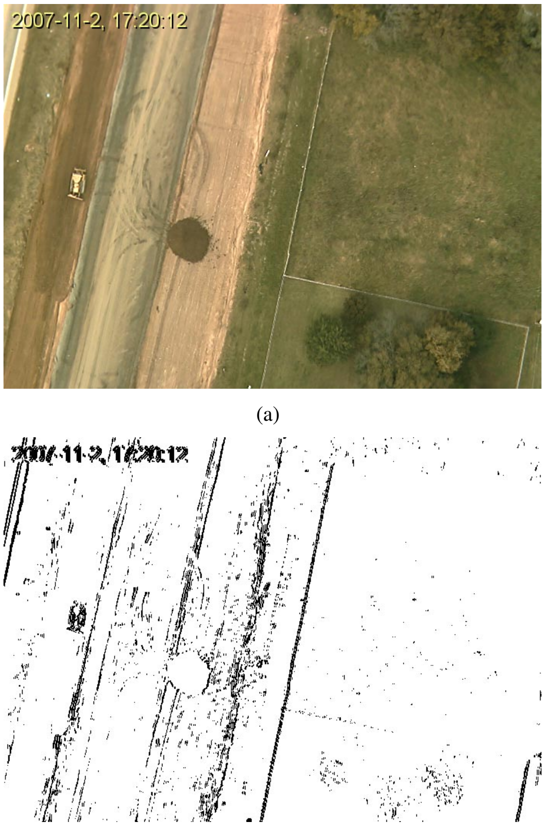
Ara Nefian is with the Carnegie Mellon University and NASA Ames Research Center, ara.nefian@nasa.gov

Xavier Bouyssounouse and Terry Fong are with the NASA Ames Research Center, xavier.bouyssounouse@nasa.gov and terry.fong@nasa.gov

978-1-61284-385-8/11/$26.00 ©2011 IEEE

A. Feature Detection

This stage relies on the observation that man-made objects and vehicles in particular have a large number of edges and corners compared with other natural objects (trees, hills, roads, forest, water flows). Image features based on edges detected using a Sobel operator represent a viable solution to detect a large number of man made objects and discriminate from background as shown in Figure 2. An improved alternative to Sobel edge detection is the use of Harris corner detection. Corners represent a better descriptor for vehicles and are able to reject background areas with large areas of random edge distribution.



(a)

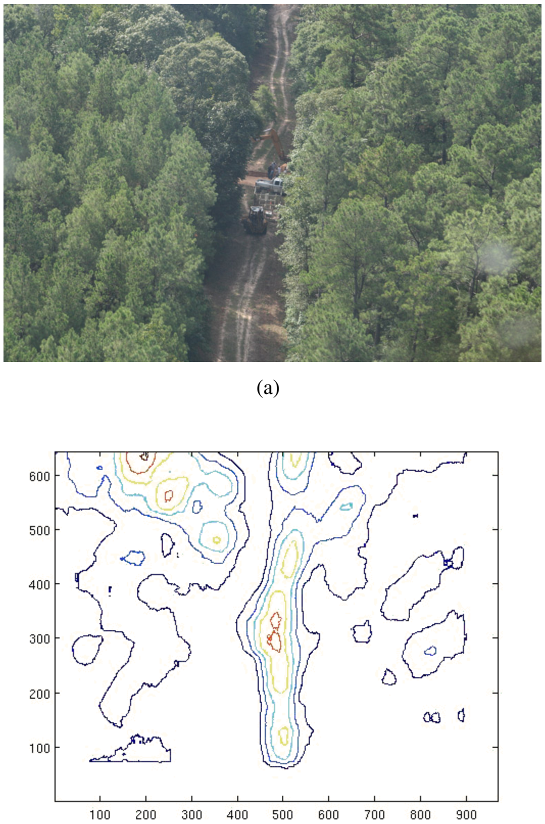
(b)

Fig. 2. (a) The original aerial image of a pipeline threat. (b) The results of the Sobel edge detection method. Detected edges are shown in black.

B. Feature Density Estimation

The next stage of our system involves the efficient detection of areas with high concentration of features. The algorithm searches through all rectangular windows of all aspect ratios and scales to determine those rectangles with feature density higher than a fixed threshold. The feature density score $Score_{feat}(x, y, w, h)$ for a particular rectangle with top left corner at position x, y in the image and of width w and height h is defined as $Score_{feat}(x, y, w, h) = \frac{S_{x,y,w,h}}{w \times h}$, where $S_{x,y,w,h}$ is the number of features found within the rectangle. An important aspect in the computation of the $Score_{feat}$ is maintaining a low computational complexity. This is accomplished by discarding all redundant computations in summing over features extracted in overlapping windows.

In this approach the efficient computation of the number of features is obtained using integral images [10]. An integral image $T(i, j)$ for a given interest point image $E(i, j)$ is an image where each pixel is computed as $T(i, j) = \sum_{u<i,v<j} E(u, v)$. Using an integral image the number of interest points $S_{x,y,w,h} = T(x + w, y + h) - T(x, y + h) - T(x + w, y) + T(x, y)$. This significantly reduces the computational complexity of the system and allows for real-time implementation. Figure 3 illustrates a typical aerial image with two targets and the contours of the feature density score. Red contours represent areas that are more likely to be targets while blue contours are areas less likely to contain targets.



(a)

(b)

Fig. 3. (a) The original aerial image of a pipeline threat. (b) The contours of the detection surface showing in red indicates likely target positions and blue indicates unlikely target positions. The yellow contours represent intermediate values between red and blue values.

C. Target Clustering

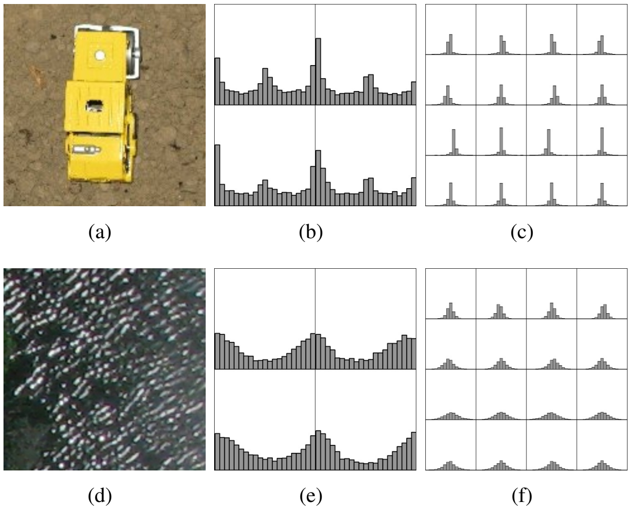
All windows for which the feature density score is above a fixed threshold are assigned to potential targets. As expected the system returns a large number of responses around actual targets. At this stage, the overlapping responses are grouped together and the overlapping detection are rejected using the following iterative method.

- **Step 1** Determine a set of overlapping windows.
- **Step 2** Determine the centroid rectangle using the average width, height and center position of all overlapping rectangles.
- **Step 3** Assign all rectangles that have an area of overlap with the centroid rectangle to the same class. The remaining rectangles are processed in step 1.
- **Step 4** If the norm of the center, width and height of the centroid rectangle at consecutive iterations falls below a fixed threshold the algorithm converges, otherwise go to step 1.

D. Color-based Detection Refinement

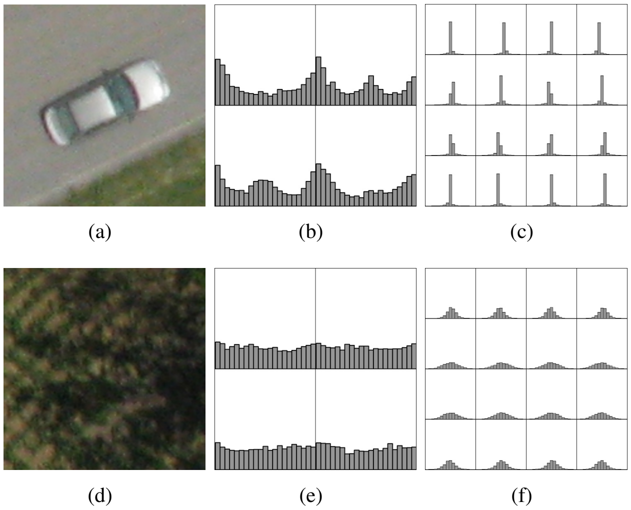
The target locations determined in the previous stages are refined to further reduce the false alarms using color information. A rectangular window is not a perfect fit for a vehicle and often a "correct" detection window contains background areas. On the other hand an "incorrect" detection window contains only background which often has a locally monochromatic distribution. The detection score used in this stage of the algorithm eliminates the background areas characterized by a monochromatic color distribution. The color score is given by $S_{color} = \max_F((\mu_r^F - \mu_r^B)^2, (\mu_g^F - \mu_g^B)^2, (\mu_b^F - \mu_b^B)^2)$ where F is a detection window, B is a background window that includes F and $\mu_r^F, \mu_g^F, \mu_b^F$ and $\mu_r^B, \mu_g^B, \mu_b^B$ are the mean of the R,G,B colors inside windows F and B respectively. In our experiments the background window was chosen to have twice the number of pixels of the detection window.

III. TARGET CLASSIFICATION


(a) (b) (c)

(d) (e) (f)

Fig. 4. Model vehicle (a), corresponding HoG features (b) and Gabor histograms (c). Water region in aerial image (d), corresponding HoG features (e) and Gabor histograms (f).


(a) (b) (c)

(d) (e) (f)

Fig. 5. Vehicle in aerial image(a), corresponding HoG features (b) and Gabor histograms (c). Vegetation region in aerial image (d), corresponding HoG features (e) and Gabor histograms (f).

The final stage of our cascade detection system is target classification (Figure 1). A binary classifier assigns each of the detection results of the previous stages into vehicle or background categories and further reduces the false alarm rate. This process begins by obtaining eight additional windows surrounding the initial detection location obtained from the previous stages. These neighboring windows are selected using a window displaced 25 pixels in the vertical and/or horizontal directions. All nine of the rectangular areas are then analyzed. If any one window around a detection result is classified as a target, then the entire area is detected as a target, otherwise it is classified as background.

From a computational stand point, this stage is significantly more complex for each window but analyzes a much smaller number of windows compared to the first stage of the algorithm. This section compares two feature extraction methods (Histogram of Oriented Gradients and Histogram of Gabor coefficients) and several classification techniques (nearest neighbor, decision trees, random trees and support vector machines) for the task of vehicle detection.

A. Histogram of Oriented Gradients

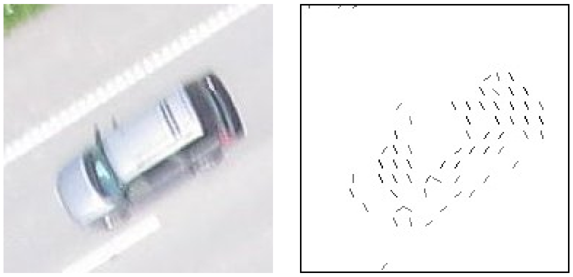


Fig. 6. Edge orientations above a particular threshold; it can be seen that the edge orientations have a perpendicular nature.

The feature extraction method used here is based on the Histogram of Oriented Gradients (HoG) approach proposed by Dalal and Triggs [4]. The classical implementation of HoG involves building histograms of orientation in small subsections of an image, then using these histograms as a feature vector. The method modifies these features by shifting the histogram in such a way that the largest bin is always located in the first position. When using a sufficient number of bins, this modification gave HoG a significant increase in performance and produced a more rotation tolerant feature set. Through preliminary testing the ideal number of bins was determined as 21.

It is important to note that the HoG features have a strong discriminatory power for vehicle versus rural background images. From an overhead image, a vehicle has two primary edge orientations; the first is with respect to the front and back of the vehicle, while the other is oriented with the sides. These two orientations are, for the most part, perpendicular, which gives a strong common feature among vehicles. After implementing this it can be seen in Figure 4-5 that there is a "W"-like shape to the vehicles, while the vegetation is flat, and the reflections of water have only one major direction giving a "U"-like shape to the histogram.

During experimentation it was determined that dividing the regions into four cells (2x2 cells) yielded the most accurate classification results. We believe that the reason this worked so well is because no matter the orientation of a vehicle, one of the four main corners will be located in each of the four cells. Because this method relies on the strong perpendicular edge orientations, having cells without parts of the vehicle in them can hinder the feature extraction. On the other hand, reducing the number of cells to one simply does not extract enough information for accurate classification. However, it is important to point out that using only one cell worked far better than using more than four. This is most likely because, although the histogram is general to the image, it still retains the information about the strong edge orientations, while having more than four sacrifices the rotation tolerance of the feature extraction.

B. Histogram of Gabor Coefficients

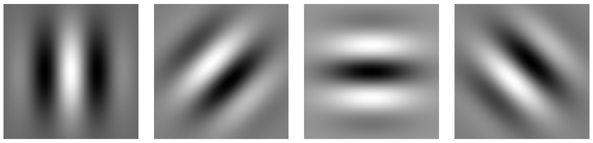


Fig. 7. Four of the sixteen Gabor kernels, all with different orientation and phase offset.

The other feature extraction algorithm was a bank of Gabor kernels (some examples in Figure 7) defined by Equation 1. The bank contained a total of sixteen filters constructed using combinations of four orientations and four phase offsets [11]. Although it was unclear as to how well these features would perform, because of the size of the filter bank this would perform with some tolerance for rotation. After filtering, a histogram of the filtered image was constructed which was used as the final feature vector for the region. The feature histograms shown in Figures 4-5 show how Gabor features differ between vehicles and vegetation. For the most part, the histograms of vehicles are concentrated in the center, while the histograms for the background are more spread out. Through preliminary testing, the ideal number of bins in the Gabor histogram was found to be 14.

$$
\begin{gathered}
g(x,y;\lambda,\theta,\psi,\sigma,\gamma) = \exp\left(-\frac{x'^2+\gamma^2 y'^2}{2\sigma^2}\right)\cos\left(2\pi\frac{x'}{\lambda}+\psi\right)\\
x' = x\cos\theta + y\sin\theta\\
y' = -x\sin\theta + y\cos\theta
\end{gathered}
\tag{1}
$$

C. Classifiers

The above features were tested using k-Nearest Neighbors (k-NN) [3], Random Forests (RF) [2], and Support Vector Machines (SVM) [5] classification techniques.

The kNN uses the training samples directly making for a very fast training time. Using these training vectors, the classification of an input vector is predicted by observing its *k nearest neighbors*, and returning the label which holds the majority among them [9]. In our experiments the best results were obtained for k=3.

The RF classifier builds a model based on a labeled training set. A random forest is composed of an assembly of unpruned Decision Trees [2] grown using random vectors. These vectors are generally sampled independently, with the same distribution for all trees. Random forests have also been shown to significantly outperform single Decision Trees and are considered among the most accurate general-purpose classifiers [1].

The SVM classifier determines a hyper plane which optimally separates samples [5]. The use of different kernels in the construction of the hyper plane can drastically affect the accuracy of an SVM. The best results were obtained when using a polynomial kernel [5].

IV. RESULTS

The data set was captured using a nadir looking Canon G9 camera with a resolution of 3000×4000 pixels flown at an average altitude of 500 feet.

A. Fast Detection Results

Our experimental results for fast detection were obtained using a data set consisting of 2000 images obtained from several flight hours over various terrain, and changing illumination conditions. Figure 8 illustrates the number of false alarms and the detection rate of the first stage of the threat detection algorithm (Section II). A correct detection was validated when we found at least 50% overlap between a manually selected rectangle and the detection rectangle computed automatically by our algorithm.

Figure 9 illustrates an example of the fast detection algorithm on a typical pipeline patrol image.

B. Classification Results

Our objective during these tests was to differentiate vehicles from the numerous background regions while also attempting to maintain all the correct detections of the first stage.

For training, two different data sets of vehicles were used. The first set was constructed using the true positives detected in the fast detection stage through approximately one hour of flight. The second set was created from indoor images of

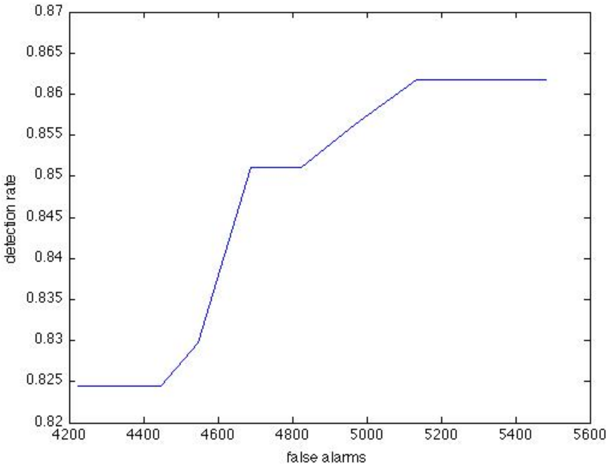


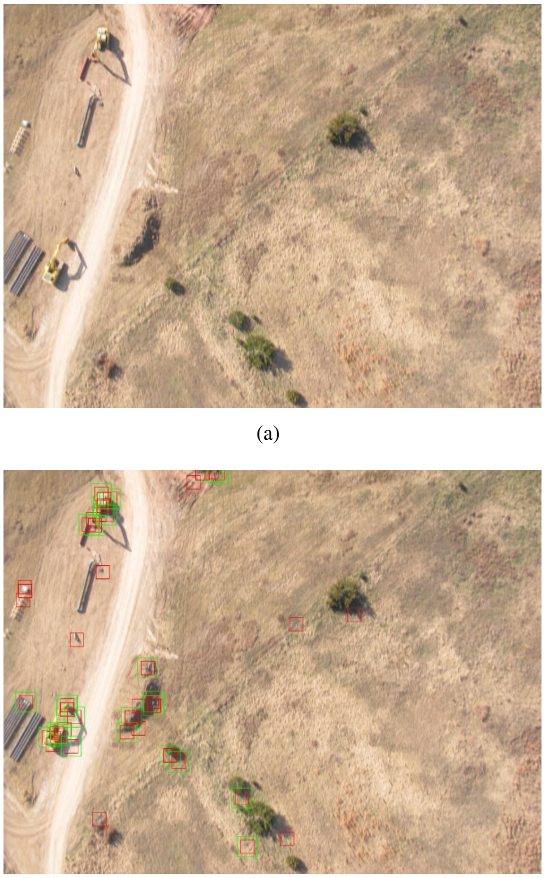
Fig. 8. ROC curve of the first detection stage.

model vehicles taken inside a sandbox. The training set based on the airborne images generated a high recognition rate if the testing and training images were captured under similar vibration and illumination conditions. However this is often not the case in real systems. The sandbox images captured under neutral conditions on the other hand, form a training set that is invariant to vibrations, blurring and illumination conditions. The trained models using these images will generate recognition rates that are more consistent over a large number of flight hours. The background data set was built from the false positives detected in the fast detection stage.

After the fast detection stage, a set of images taken over approximately one hour returned 4,963 possible threats. These regions of interest were hand separated into two classes; vehicles, and background. In this data set, there are 120 vehicles, and 4,843 background areas. The threshold of the fast detection algorithm was such that it resulted in an average detection rate of 85%. We believe that the performance on this test set gives a reasonably accurate estimate of the overall accuracy of the program.

Table I shows a summary of our results. As it can be seen, the top performing classifier is the Random Forest using Gabor features. This combination correctly classified 98.9% of vehicles and 61.9% of background, removing well over half of the false positives and keeping nearly all the true positives. It is also interesting to note that the Gabor histograms features had difficulties correctly classifying background containing vegetation; this is illustrated by the feature similarities in Figure 4-5. However, these features performed remarkably well against the reflective water. The rotation invariant HoG, on the other hand, performed in exactly the opposite manner, while it was easily able to discard the vegetation the reflective water gave it some trouble. Because HoG and Gabor histogram look at entirely different features they have unique strengths and weaknesses.

Another interesting result, is the performance of Gabor features when training with sandbox vehicles and the Random Forest classifier. Using an artificial model of a vehicle under different lighting conditions, Gabor was able to retain nearly all of the vehicles in the set while successfully reducing the false positives by half. This result was remarkable because it shows that the features work well across data sets. This is very important in real world application as seasons and weather can drastically affect the contents of these images. Showing that Gabor can successfully perform from plastic models, to an actual flight, enforces just how powerful it is.



(a)

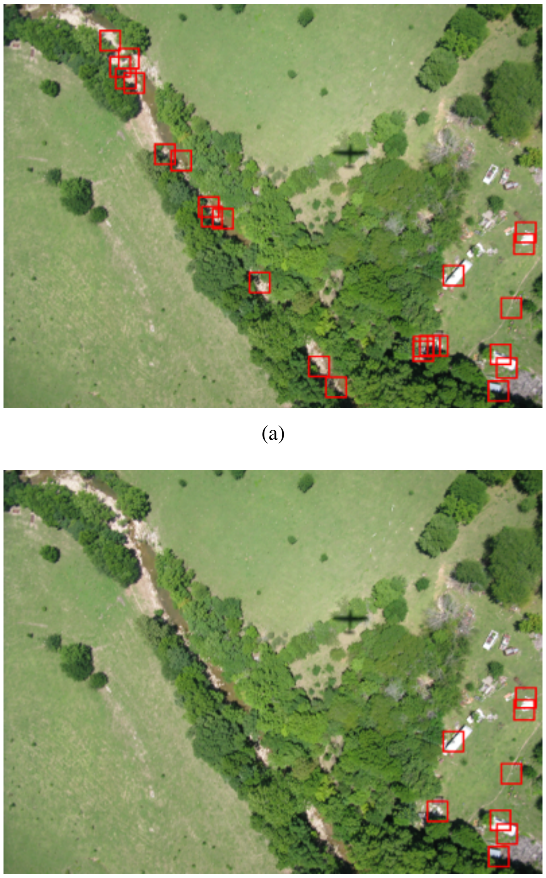
(b)

Fig. 9. (a) Original image. (b) Detected targets. Red rectangles are the results of the initial threat detection algorithm. Green rectangles are the final results.

TABLE I

PERCENTAGE OF VEHICLE AND BACKGROUND IMAGES CORRECTLY CLASSIFIED USING SEVERAL FEATURE EXTRACTION AND CLASSIFICATION TECHNIQUES.

		Indoor image vehicles			Aerial image vehicles		
		KNN	RTrees	SVM	KNN	RTrees	SVM
Gabor	Vehicles	98.3	98.3	87.5	100	98.9	100
	Backgnd	38.1	47.3	51.8	24.2	61.9	32.4
HoG	Vehicles	81.7	87.5	75.0	83.3	85.7	93.3
	Backgnd	71.6	58.4	75.4	76.7	66.6	61.4



(a)

(b)

Fig. 10. Example of classification on an image, (a) shows the areas discovered to be potential threats by the fast detection stage. Image (b) shows which how the classifier further reduced these results.

V. CONCLUSIONS AND FUTURE WORK

A. Conclusions

This paper introduced a real-time system for vehicle detection in rural environments from aerial images. Our approach consists of a cascade detection algorithm with the first stage serving as a fast detection solution that rejects most of the background and selects patterns corresponding to man made objects. The patterns selected by this stage are further refined in the second stage using image classification techniques. Our experiments for this stage compared four classification methods (KNN, SVM, decision trees and random trees) and two feature extraction techniques (histogram of gradients and Gabor coefficients). Our system achieves best overall results using Gabor derived histograms and random trees classifiers. The system presented in this paper is able to quickly process real data captured by the pipeline patrol airplanes which will enable pipeline threat detection to be performed automatically by UAVs.

B. Future Work

Our future research will be conducted primarily towards improving the accuracy of the classification stage while preserving the real time requirements of the system. We expect that multi-class classifiers that include background categories such as vegetation, lakes, buildings and roads will improve the accuracy of the current system. Various feature extraction techniques will be investigated to further improve the image descriptors of the above classes.

To improve the fast detection stage, we will be experimenting with a Harris-Laplace detector [8] due to its scale invariant nature; we believe this may be an improvement over the Harris detector currently being used.

REFERENCES

[1] Gérard Biau, Luc Devroye, and Gábor Lugosi. Consistency of random forests and other averaging classifiers. *J. Mach. Learn. Res.*, 9:2015–2033, 2008.

[2] Leo Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[3] T. Cover and P. Hart. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21 – 27, jan. 1967.

[4] Navneet Dalal and Bill Triggs. Histogram of Oriented Gradients for Human Detection. *IEEE Computer Society on Computer Vision and Pattern Recognition(CVPR'05)*, 2005.

[5] M.A. Hearst, S.T. Dumais, E. Osman, J. Platt, and B. Scholkopf. Support vector machines. *Intelligent Systems and their Applications, IEEE*, 13(4):18 –28, jul. 1998.

[6] Stefan Hinz. Detection of Vehicles and Vehicle Queues in High Resolution Aerial Images. In *Photogrammetrie - Fernerkunding - Geoinformation (PFG) 3*, pages 203–215, 2004.

[7] ZuWhan Kim and Jitendra Malik. Fast Vehicle Detection with Probabilistic Feature Grouping and its Application to Vehicle Tracking. *Computer Vision, IEEE International Conference on*, 1:524, 2003.

[8] Krystian Mikolajczyk and Cordelia Schmid. Scale and Affine Invariant Interest Point Detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

[9] Uday Rajanna, Ali Erol, and George Bebis. A Comparative Study on Feature Extraction for Fingerprint Classification and Performance Improvements Using Rank-Level Fusion. *Pattern Analysis and Applications*, 13:263–272, 2010.

[10] Paul Viola and Michael J. Jones. Robust Real-Time Face Detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[11] Sun Zehang, George Bebis, and Ronald Miller. Monocular Precrash Vehicle Detection: Features and Classifiers. *IEEE Transactions on Image Processing*, 15(7):2019–2034, July 2006.

[12] Tao Zhao and Ram Nevatia. Car Detection in Low Resolution Aerial Images. In *Image and Vision Computing*, pages 710–717, 2001.