2011 14th International IEEE Conference on
Intelligent Transportation Systems
Washington, DC, USA. October 5-7, 2011

Real-time Vehicle Detection and Tracking Using Stereo Vision and Multi-View AdaBoost

T. Kowsari* S.S. Beauchemin* J. Cho*

*Department of Computer Science
The University of Western Ontario, London, ON, N6A-5B7

Abstract

We propose a multi-layer, real-time vehicle detection and tracking system using stereo vision, multi-view AdaBoost detectors, and optical flow. By adopting a ground plane estimate extracted from stereo information, we generate a sparse set of hypotheses and apply trained AdaBoost classifiers in addition to fast disparity histogramming, for Hypothesis Verification (HV) purposes. Our tracking system employs one Kalman filter per detected vehicle and motion vectors from optical flow, as a means to increase its robustness. An acceptable detection rate with few false positives is obtained at 25 fps with generic hardware.

I. Introduction

Most Driving Assistance Systems (DAS) rely on the detection of relevant features in the immediate environment of the vehicle, such as other vehicles, pedestrians, lanes, traffic signs, and other potential obstacles [1]. Many driving assistance systems such as adaptive cruise control, collision warning, blind spot monitoring, and park assist rely on some form of obstacle detection mechanisms. While obstacle detection in general can be made simpler with the use of active sensors, the resulting cross-talk and noise from other vehicles can potentially deteriorate the robustness of such systems. Moreover, for large distances and increased resolution, passive sensing seems more applicable [2].

Recently, visible spectrum vision-based vehicle detection has attracted a lot of attention due to improved machine vision algorithms and the availability of low-cost high computational power [3]. Real-time detection is vital for DAS as road-based imagery is highly dynamic (a vehicle moving at a speed of $120\ \mathrm{km/h}$ changes its position by $33\ \mathrm{m}$ every second). While very dependable techniques and methods for object detection have appeared [4], most of them require high-cost, advanced hardware to be executed in real-time, if at all possible.

This contribution proposes a real-time, multi-layer vehicle detection system using stereo vision, optical flow, and a machine-learning classifier, and is structured as follows: Section II reviews the recent vehicle detection literature, Section III presents our main approach, while Sections IV, V, and VI describe Hypothesis Generation (HG), Hypothesis Verification (HV), and Tracking respectively. Section VII presents our experimental results while Section VIII offers a conclusion.

II. Related Literature

There exists a large body of research from the last two decades on DAS. With the goal of creating autonomous vehicles, many research groups have launched several projects in different aspects of DAS [5], [6], [7], [3].

Our focus is on detecting and tracking vehicles as imaged by passive sensors inside an instrumented vehicle. The vehicle detection stage is usually broken into three parts: Hypothesis Generation (HG), Hypothesis Verification (HV), and tracking. For HG, several contributions have proposed various methods, including the selective detection of vertical and horizontal edges [8], symmetry maps [9], stereo depth [10], and more, either in isolation or in combination.

In Region Of Interest (ROI) based HG, Cheng *et al.* use vanishing points obtained from the intersection of detected lanes [11] and achieved 20fps on conventional hardware. While lane-based vanishing point detection requires salient road markings, Sappa *et al.* computed ROIs with horizon lines obtained from stereo depth data [12]. In addition, Keller *et al.* proposed a similar algorithm but employ a B-Spline model for the road rather than a flat plane [13]. In both cases however, a lack of knowledge about obstacles on the road often leads to erroneous ground plane estimations. Conversely, samples of disparity-based obstacle detection methods are that of Jung *et al.* [14] and Mandelbaum *et al.* [15] in which the authors used disparity histogram peaks as evidence of obstacles.

The HV stage is often performed with either block matching [16] or appearance-based methods, which take features such as Scale Invariant Feature Transform (SIFT) [17], Principal Component Analysis (PCA) [18], summation of intensity or Gabor filtering [19], to train a classifier into verifying generated hypotheses. Different classifiers such as Support Vector Machines (SVMs) [19], neural networks [20], AdaBoost [21], and nearest neighbor [18] are often used in the literature.

III. Description of Approach

We propose a three-stage vehicle detection method which includes: Hypothesis Generation based on ground plane estimation, Hypothesis Verification with Haar-like features, an AdaBoost classifier, and disparity histogramming, followed by vehicle tracking using optical flow and Kalman filters.

With the depth map we estimate the parameters of a plane which fits the data in a near rectangle at the bottom of the image where there is no visible obstacle. The absence of obstacles in this region is determined by the absence of peaks within the part of the disparity histogram corresponding to the ground plane near the vehicle. By using RANdom SAmple Consensus (RANSAC) in estimating the parameters of the ground plane, the effects of depth outliers are minimized. The

978-1-4577-2196-0/11/$26.00 ©2011 IEEE

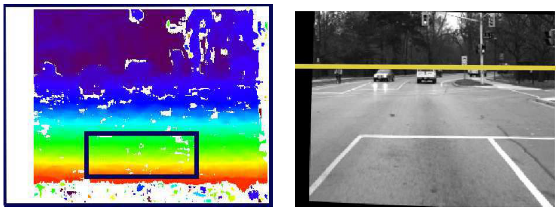

Fig. 1. **a) (left)**: *Color-coded stereo depth map and the region used for ground plane estimation* **b) (right)**: *Horizon line estimated from ground plane*

horizon line can in turn be estimated by intersecting the plane parallel to the ground plane and containing the focal point with the imaging plane of one of the sensors. The horizon line constitutes the basis for generating hypotheses. These are then tested with a set of AdaBoost classifiers trained with four different views of various vehicles from existing data sets. An integral disparity histogram is used to increase the robustness of the verification stage. Following these steps, optical flow data provides assistance in tracking sparse features on verified hypotheses. If the number of overlapping rectangles of similar scale containing detected vehicles exceeds a certain limit in a number of consecutive frames, they are then merged into a single rectangle and a Kalman filter is created to track the detected vehicle(s) within the extent of the merged rectangle.

Our main contribution is two-fold. First, the integration of horizon-based ROI generation together with multi-view vehicle detection allows it to execute very efficiently, since many hypotheses can be rejected outright, given their relative location with respect to the horizon line. In addition, the horizon detection stage can take advantage of the tracking data and the disparity histogram to verify the absence of obstacles at close range, yielding a robust horizon line estimate. Other similar techniques often do not take advantage of detected obstacles which may lead to erroneous ground plane estimation [13]. Second, the efficient integral disparity histogram-based hypothesis rejection removes a significant portion of false positives. Excellent results are obtained with AdaBoost classifiers trained on very small sets of images, resulting in a training time for each classifier under 5 minutes. The algorithm is discussed with greater detail in the next Sections.

IV. Hypothesis Generation

With rectified stereo images, finding disparities merely consists of a 1-D search with a block matching algorithm (our implementation uses the stereo routines from Version 2.2 of OpenCV). As expected, the presence of an obstacle in the image creates a peak in the disparity histogram. Considering this, an obstacle-free area from the bottom portion of the image may be determined and then used to estimate the position of the ground plane.

Using the disparities and the calibration parameters of the stereo system, the 3D positions of the image pixels in the camera coordinate system are computed. Assuming that the ground plane equation is of the form

$$ax + by + cz = d \tag{1}$$

where $\vec{n} = (a, b, c)$ is the unit normal vector to the plane, we pose

$$d = \frac{1}{\sqrt{a'^2 + b'^2 + c'^2}} \tag{2}$$

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = d \begin{bmatrix} a' \\ b' \\ c' \end{bmatrix} \tag{3}$$

With the coordinates of 3D points in the reference system of the left camera

$$(X_i, Y_i, Z_i) \tag{4}$$

we can write

$$\mathbf{Ax} = \mathbf{B} \tag{5}$$

and solve for $\mathbf{x}$ in the least-squares sense as

$$\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{B} \tag{6}$$

where

$$\mathbf{A} = \begin{bmatrix} X_1 & Y_1 & Z_1 \\ X_2 & Y_2 & Z_2 \\ \vdots & \vdots & \vdots \\ X_n & Y_n & Z_n \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} a' \\ b' \\ c' \end{bmatrix}$$

Often times the ground surface leads to inordinate amounts of outliers, due in part to a lack of texture from the pavement or other drivable surfaces. With the sensitivity of least-squares to outliers being known, we resort to the use of RANSAC in selecting the inliers and obtain a robust estimation of the ground plane coefficients, in the following way:

1) randomly select three points from the 3D points believed to be representative of the ground plane
2) compute the coefficients of the plane defined by the randomly selected points using (5)
3) count the points whose distance to the plane is less than a threshold ϵ
4) repeat these steps n times where n is sufficiently large[1]
5) among the n fits choose the largest inlier set which respect to ϵ and compute the coefficients of the ground plane this time using least-squares as in (6)

The plane parameters are averaged over a short period of time in order to stabilize them further. The coefficients of the plane are recomputed at each new stereo frame arrival. However, in cases when the number of depth values is low (poor texture, etc.) or other vision modules indicate the presence of a near obstacle, the coefficients of the ground plane are not recomputed, the previous parameters are used instead.

The horizon line is approximated by intersecting the plane parallel to that of the ground and passing through the focal point with the image plane of the sensor, and converting to image coordinates using calibration parameters. Figure 3 depicts the geometry involved in approximating the horizon line, while Figure 1 displays a sample disparity map and

[1] Choosing $n > 20$ does not significantly improve the number of inliers with respect to ϵ.

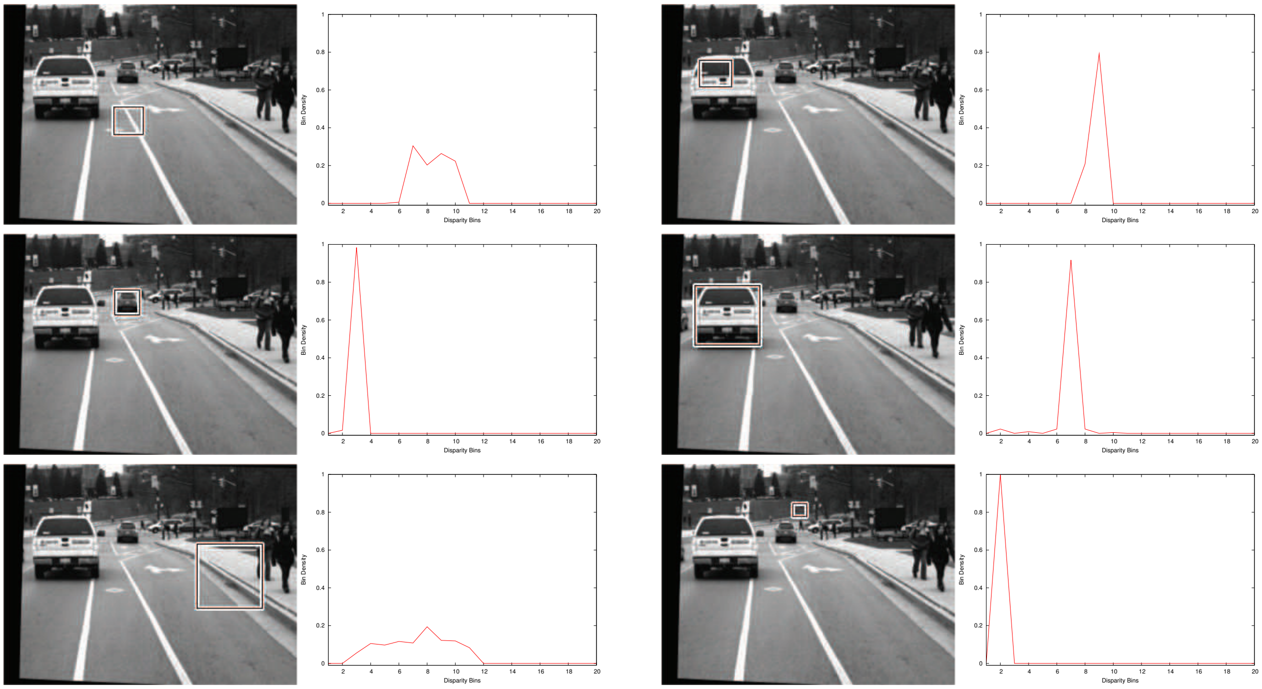

Fig. 2. *Histogram of depth map depicted for different rectangular areas. High peaks constitute evidence of presence of an obstacle in that area. The position of the peak defines acceptable scale ranges.*

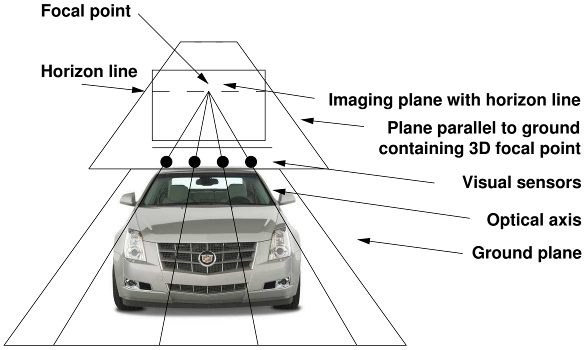

Fig. 3. *The geometry of estimating the horizon line in camera coordinates*

estimated horizon line, used in our hypothesis generation mechanism.

We generate hypotheses as rectangular shapes in which vehicles may be found. Consider a hypothesis

$$H(x, y, w, h) \tag{7}$$

where x and y are the image coordinates of the top-left corner of the rectangle and w and h are the width and height of the rectangle forming the hypothesis.

The method to generate hypotheses first postulates that, since the sensors are located on the roof of the vehicle, every other vehicle with similar (or smaller) height will appear under the horizon line in the image. Consequently, the imaging of taller vehicles such as transport trucks and buses will include image sections that are above the imaged horizon line.

The fact that the horizon line is parallel to the detected ground plane does not signify that the resulting imaged horizon line is parallel to the x-axis of the image (due to the roll of the vehicle, for instance). Vehicular acceleration and deceleration generally cause tilting of the vehicle and as a consequence the estimation of the horizon line may be somewhat imprecise at times. For this reason, we generate hypotheses for the presence of other vehicles within $\pm\delta$ vertical pixels of the imaged horizon line.

Given these observations, we generate hypotheses along a band of pixels comprised within the set of image lines $\{L\} : f(x) = mx + b \pm \delta$. For the smallest image area considered for hypothesis generation, we posit that there could be a vehicle in any of the image regions with all the possible x-axis coordinates acting as upper-left corners within the image band. For taller vehicles, we allow every generated hypothesis to grow above $l \in L$ by as much as $\frac{2}{3}$ of its own side height h. A total of four classifiers are defined and trained for the detection of

1) front views
2) back views
3) front-side and back-side views
4) side views

with some of them requiring different aspect ratios[2]. For rear and front views, the smallest hypothesis rectangle is of size 15×15 pixels, 15×36 pixels for front-side and rear-side views,

[2] The aspect ratio is 1 for the front and rear classifiers, 2.4 for vehicle front side and back side classifiers and 3.0 for the side classifier.

and 15×45 pixels for side views. Each generated smallest-area hypothesis is then used to generate other, larger hypotheses to account for the detection of vehicles that are at closer range, and thus appear larger on the image plane. For this purpose, we scale each hypothesis by a factor of 1.2 repeatedly, until the region outgrows the image plane.

V. Hypothesis Verification

AdaBoost (short for Adaptive Boosting) introduced by Freund and Schapire [22] is a method for choosing and combining a set of weak classifiers to build a strong classifier. Combining the concept of the integral image as an efficient way of computing Haar-like features and cascaded AdaBoost, Viola and Jones introduced a powerful method for object recognition [21]. We adopted this approach for the hypothesis verification stage. We used four cascaded AdaBoost classifiers to discriminate positive from false-positive hypotheses. The Haar-training module from OpenCV is used to train our classifiers. We used in excess of two hundred vehicle images (positive samples) for each classifier. The images used for different vehicular views are borrowed from the dataset used by Cornells and Leibe [23], [24]. For the negative examples we used a set of more than five hundred images randomly downloaded from the Internet.

In order to increase the robustness of the hypothesis verification stage, we considered the fact that there should be a peak in the disparity histogram where obstacles above the ground plane are imaged. Figure 2 shows areas with or without obstacles, and their corresponding disparity histograms. Although disparity histogram peaks have been used for obstacle detection before [14], we are not aware of any work using our approach for testing several rectangular areas from the image. The main reason is probably the high computational cost of repetitive histogram generation for overlapping areas. To efficiently compute the disparity histogram, an integral disparity histogram is extracted from the disparity map. This integral disparity histogram defined by us is inspired from the integral image concept introduced by Viola and Jones [21]. To reduce the required processing time, the original disparity map is down-sampled to half of its size, and then the integral value for each bin of the histogram is computed. A total of twenty bins are used for the histogram. Computing the disparity histogram for any rectangular area amounts to performing two additions and one subtraction for each bin. The processing time required to compute the integral histogram is 5ms on conventional hardware.

Given a positive vehicle detection from our classifiers, we can further affirm that, for a given image area, if the disparity histogram peak is greater than a certain threshold, there is little probability that this hypothesis constitutes a false positive. Conversely, and again for a verified hypothesis, the absence of a clear peak is considered to signify a false-positive. Furthermore, if the peak disparity in the histogram (as a clue to obstacle depth) is inconsistent with the image area for which there is a detection, the hypothesis can be rejected outright.

VI. Hypothesis Fusing and Tracking

In every frame, image regions standing as verified hypotheses may overlap. It thus may be necessary to fuse these hypotheses into one, more consistent detection event. Hypotheses sharing more than 40% of their area are then fused into the smallest rectangular image region containing the overlapping hypotheses. This list of rectangles is used for maintaining and confirming current detections.

The representation of detected vehicles consists of a list of rectangles together with a detection counter. If this detection counter becomes higher than a certain threshold τ_2, the rectangle state changes to *active* (the probability of a vehicle in the image region is high), otherwise it remains in the list to be either activated based on future hypothesis confirmation, or removed after sufficient evidence accumulates that no vehicle is present in it. Before new hypotheses can be added to the current list of confirmed detections, it is necessary to maintain it by tracking the detected vehicles. To this end, an optical flow field is computed for each detection along with a Kalman filter to perform the tracking.

As a new image frame becomes available, hypotheses (rectangular areas) are added to the list with their detection counter set to a value τ_1, corresponding to their maximum lifetime (measured in number of frames) before they can be removed from the list, if no confirming evidence can be found for the presence of a vehicle. We use the same technique as above to perform the merging of overlapping areas and the setting of thresholds τ_1 and τ_2. A hypothesis is removed from the list when its detection counter reaches zero. As a result, a tracking continues if a vehicle is detected every τ_1 frames on average, and a tracking is terminated if it is not detected for τ_2 frames in a row[3].

VII. Experimental Results and Discussion

We used the RoadLAB instrumented vehicle for recording sets of sequences in the urban area of London Ontario, Canada (see Figure 5) [25]. The stereo system used for the experiments has a baseline of length $b = 357\,\mathrm{mm}$, a smallest detectable 2D disparity of $\frac{1}{16}$ of a pixel, a focal length of $f = 12.5\,\mathrm{mm}$, and a physical pixel square size of $4.40\,\mu\mathrm{m}$. The image resolution was set to 320 × 240 pixels. Typical vehicle speeds during the experiments ranged from 0 to $60\,\mathrm{km/h}$.

The algorithm has been tested on 7,814 frames containing 13,513 vehicles in different lanes, orientations, and directions. Table I shows the accuracy of the system for vehicles closer than 50m, 100m, and 150m for *leading, oncoming, other*, and *all* vehicles, where *other* includes parked vehicles and those in other views such as at intersections. Figure 4 shows some correctly and incorrectly detected vehicles.

In the sum of frames used for the experiments, the number of false positives amounts to 2,008 (about 0.26 per frame or less than 7×10^{-6} False Positive Per Window (FPPW)). Since false positives are mostly detected outside the area occupied by the roadway, better results could still be achieved by rejecting hypotheses related to detections that are not located on the

[3] Acceptable results are obtained with $\tau_1 = 3$ and $\tau_2 = 10$.

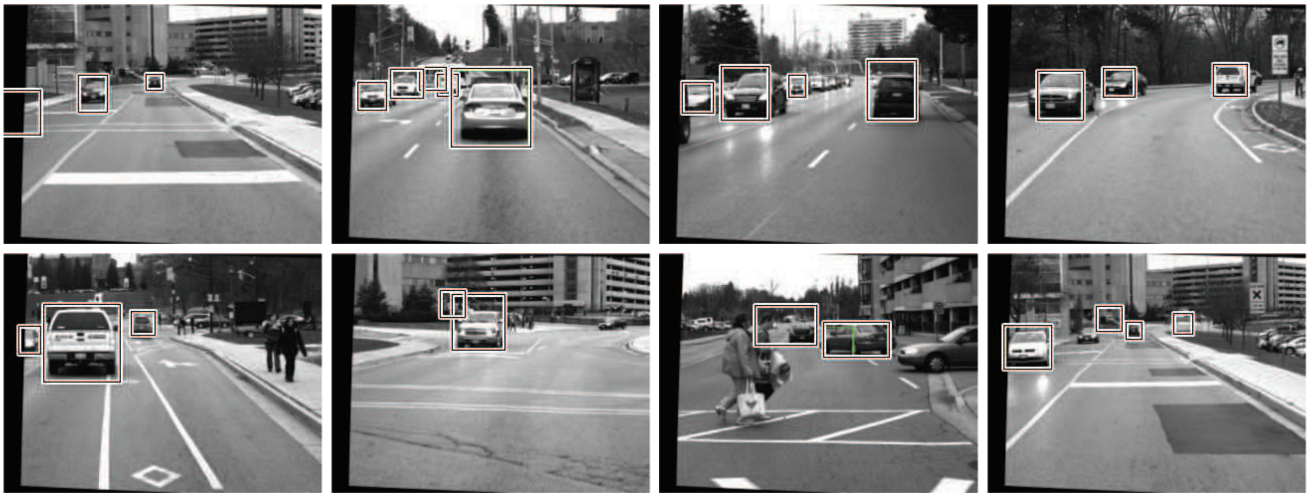

Fig. 4. *Examples of detection results for various frames, including errors*

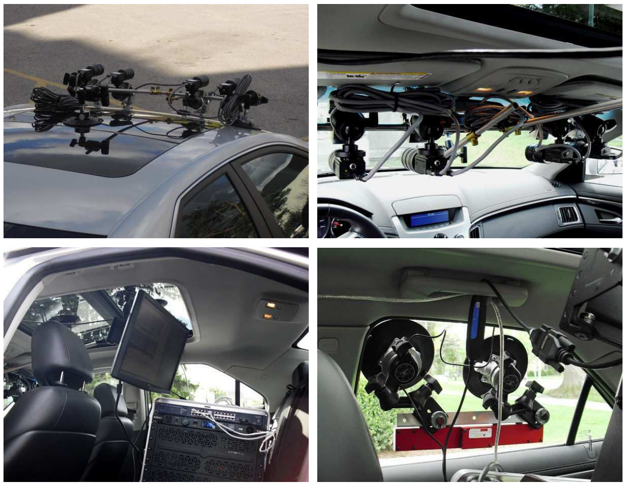

Fig. 5. *The RoadLAB in-vehicle laboratory*

TABLE I
DETECTION RATES FOR DIFFERENT DISTANCES AND VEHICLES

Hit Rate	Leading	Oncoming	Other	All
Dis. < 50	0.9853	0.9886	0.8004	0.9858
Dis. < 100	0.8669	0.8135	0.8028	0.8548
Dis. < 150	0.8171	0.7393	0.8145	0.7990

roadway. The execution time of the algorithm is 25fps on generic hardware.

Table II shows the number of false positives and algorithm speed, using AdaBoost with and without the horizon line constraint and the depth histogram. Removing the disparity histogram constraint results in twice as many false positives, accompanied by a modest increase of the frame rate.

Figure 6 illustrates the difference between the detection rates for on-coming, leading, and other vehicles. Detection rates for on-coming vehicles are slightly worse than those of leading vehicles. A possible explanation may be that headlights are on in our sequences while off in the training images. The detection rates are worse for parked vehicles and those at intersections, possibly because of partial occlusion.

TABLE II
THE EFFECT OF THE IMAGE HORIZON LINE AND DISPARITY HISTOGRAM CONSTRAINTS ON FALSE POSITIVES AND FRAME RATES

	FP	Frame Rate
Complete Technique	2008	25
Without Disparity Histogram	5676	27
Without Disparity Histogram and Horizon	7874	18

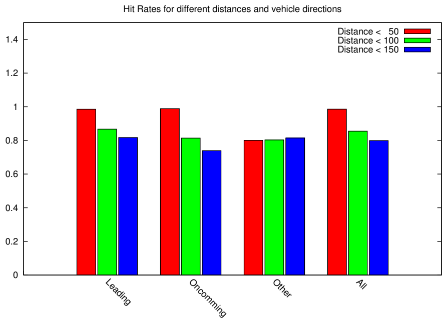

Fig. 6. *Detection hit rates for different vehicle positions and distances*

Figure 7 depicts the relation between hit-rates (provably correct detections) and vehicle distance. The performance of our technique decreases dramatically for vehicles located farther than 80m. A possible cause of this performance degradation may be that the smallest windows for which our classifiers are trained are of modest size (15×15 pixels) and provide crude resolution. We have compared the frame rates, hit rates, and the farthest detectable vehicles with other contributions cited in Table III. Among these other techniques, ours has the best frame rate, with comparable hit rates for distances under 50m.

VIII. CONCLUSION

We have developed a real-time, multi-vehicle detection and tracking system using stereo information, optical flow, Kalman filters, and an AdaBoost classifier. The technique executes at 25fps on generic hardware and has been tested on the RoadLAB instrumented vehicle [25]. Research is currently being conducted to determine whether vehicular odometry

TABLE III
COMPARISON ON FRAME RATES, DISTANCE (OR SIZE), HIT RATES, FALSE POSITIVES, AND VEHICLE VIEWS

Authors	Distance of Farthest Detection	FPS	HR	FP	Notes
Chang and Cho [26]	32×32 image region	5	99%	12%	Rear detection
Southall *et al.* [27]	40m	16	99%	1.7%	Single lane rear detection
Bergmiller *et al.* [28]			83.12%	16.7%	Rear detection
Sun *et al.* [29]	32×32 image region	10	98.5%	2%	Rear detection
Alonso *et al.* [30]			92.63%	3.63%	Rear and front detection
Cheng *et al.* [11]		20	90%	10%	Rear and front detection
Our Results	120m	25	98.6%	13%	Multi-view

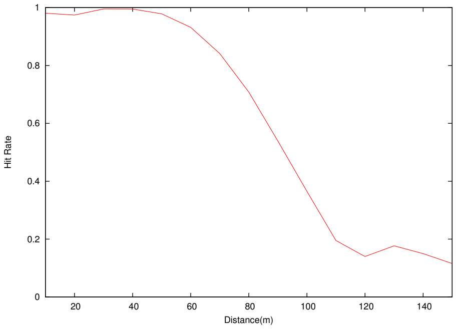

Fig. 7. *Overall hit rate with respect to distance*

may provide additional constraints to improve the quality of our results. We plan to use a similar framework, possibly with different constraints, to detect pedestrians in real-time.

REFERENCES

[1] A. Amditis, K. Pagle, S. Joshi, and E. Bekiaris, "Driver-vehicle-environment monitoring for on-board driver support systems: Lessons learned from design and implementation," *Applied Ergonomics*, vol. 41, pp. 225–235, 2010.

[2] M. Bertozzi, A. Broggi, M. Cellario, A. Fascioli, P. Lombardi, and M. Porta, "Artificial vision in road vehicles," *Proceedings of the IEEE*, vol. 90, no. 7, pp. 1258–1271, 2002.

[3] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection: A review," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, pp. 694–711, 2006.

[4] B. Leibe, A. Leonardis, and B. Schiele, "Robust object detection with interleaved categorization and segmentation," *International Journal of Computer Vision*, vol. 77, no. 1, pp. 259–289, 2008.

[5] B. Ulmer, "Vita-an autonomous road vehicle (arv) for collision avoidance in traffic," in *Intelligent Vehicles Symposium, Proceedings of the*, 1992, pp. 36–41.

[6] M. Bertozzi and A. Broggi, "GOLD: A parallel real-time stereo vision system for generic obstacle and lane detection," *Image Processing, IEEE Transactions on*, vol. 7, no. 1, pp. 62–81, 2002.

[7] C. Thorpe, M. Hebert, T. Kanade, and S. Shafer, "Vision and navigation for the Carnegie-Mellon Navlab," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 10, no. 3, pp. 362–373, 2002.

[8] Z. Sun, R. Miller, G. Bebis, and D. DiMeo, "A real-time precrash vehicle detection system," in *Applications of Computer Vision, 6th IEEE Workshop on*, 2003, pp. 171–176.

[9] A. Bensrhair, M. Bertozzi, A. Broggi, P. Miche, S. Mousset, and G. Toulminet, "A cooperative approach to vision-based vehicle detection," in *Intelligent Transportation Systems*, 2002, pp. 207–212.

[10] R. Mandelbaum, L. McDowell, L. Bogoni, B. Reich, and M. Hansen, "Real-time stereo processing, obstacle detection, and terrain estimation from vehicle-mounted stereo cameras," in *Applications of Computer Vision, 4th IEEE Workshop on*, 1998, pp. 288–289.

[11] H. Cheng, N. Zheng, C. Sun, and H. van de Wetering, "Vanishing point and gabor feature-based multi-resolution on-road vehicle detection," *ISNN 2006*, pp. 46–51, 2006.

[12] A. Sappa, F. Dornaika, D. Ponsa, D. Gerónimo, and A. López, "An efficient approach to onboard stereo vision system pose estimation," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 9, no. 3, pp. 476–490, 2008.

[13] C. Keller, D. Llorca, and D. Gavrila, "Dense stereo-based roi generation for pedestrian detection," *Pattern Recognition*, pp. 81–90, 2009.

[14] H. Jung, Y. Lee, B. Kim, P. Yoon, and J. Kim, "Stereo vision-based forward obstacle detection," *International Journal of Automotive Technology*, vol. 8, no. 4, pp. 493–504, 2007.

[15] R. Mandelbaum, L. McDowell, L. Bogoni, B. Reich, and M. Hansen, "Real-time stereo processing, obstacle detection, and terrain estimation from vehicle-mounted stereo cameras," in *Proceedings of the IEEE*, 1998, pp. 288–289.

[16] P. Parodi and G. Piccioli, "A feature-based recognition scheme for traffic scenes," in *Intelligent Vehicles Symposium, Proceedings of the*, 2002, pp. 229–234.

[17] D. Lowe, "Object recognition from local scale-invariant features," in *ICCV*. Published by the IEEE Computer Society, 1999, p. 1150.

[18] J. Wu and X. Zhang, "A PCA classifier and its application in vehicle detection," in *Neural Networks, International Joint Conference on*, vol. 1, 2002, pp. 600–604.

[19] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection using Gabor filters and support vector machines," in *Digital Signal Processing, 14th International Conference on*, vol. 2, 2002, pp. 1019–1022.

[20] N. Matthews, P. An, D. Charnley, and C. Harris, "Vehicle detection and recognition in greyscale imagery," *Control Engineering Practice*, vol. 4, no. 4, pp. 473–479, 1996.

[21] P. Viola and M. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[22] Y. Freund and R. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *Computational learning theory*. Springer, 1995, pp. 23–37.

[23] N. Cornelis, B. Leibe, K. Cornelis, and L. Van Gool, "3d city modeling using cognitive loops," in *3D Data Processing, Visualization, and Transmission, 3rd International Symposium on*, 2007, pp. 9–16.

[24] B. Leibe, N. Cornelis, K. Cornelis, and L. Van Gool, "Dynamic 3d scene analysis from a moving vehicle," in *Computer Vision and Pattern Recognition, IEEE Conference on*, 2007, pp. 1–8.

[25] S. Beauchemin, M. Bauer, D. Laurendeau, T. Kowsari, J. Cho, M. Hunter, and O. McCarthy, "RoadLab: An In-Vehicle Laboratory for Developing Cognitive Cars," in *International Conference on Computer Applications in Industry and Engineering, IEEE Conference on*, 2010, pp. 7–12.

[26] W. Chang and C. Cho, "Online boosting for vehicle detection," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 40, no. 3, pp. 892–902, 2010.

[27] B. Southall, M. Bansal, and J. Eledath, "Real-time vehicle detection for highway driving," in *Computer Vision and Pattern Recognition, IEEE Conference on*, 2009, pp. 541–548.

[28] P. Bergmiller, M. Botsch, J. Speth, and U. Hofmann, "Vehicle rear detection in images with Generalized Radial-Basis-Function classifiers," in *Intelligent Vehicles Symposium*, 2008, pp. 226–233.

[29] Z. Sun, G. Bebis, and R. Miller, "Monocular precrash vehicle detection: features and classifiers," *Image Processing, IEEE Transactions on*, vol. 15, no. 7, pp. 2019–2034, 2006.

[30] D. Alonso, L. Salgado, and M. Nieto, "Robust vehicle detection through multidimensional classification for on board video based systems," in *Image Processing, IEEE International Conference on*, vol. 4, 2007, pp. 321–324.