Highway Scene Analysis in Hard Real-Time *

Margrit Betke†
Boston College
Chestnut Hill, MA 02167

Esin Haritaoglu and Larry S. Davis
University of Maryland
College Park, MD 20742

ABSTRACT

A hard real-time vision system has been developed that analyses color videos taken from a car driving on a highway. The system uses a combination of color, edge, and motion information to recognize and track the road boundaries, lane markings and other vehicles on the road. Cars are recognized by matching templates that are cropped from the input data online, by detecting image features, and by evaluating how these features relate to each other. Cars are also recognized by temporal differencing and by tracking motion parameters that are typical for cars. The system recognizes and tracks road boundaries and lane markings using a recursive least squares filter. Experimental results demonstrate robust, real-time car recognition and tracking over thousands of image frames.

1 INTRODUCTION

The general goal of the our research is to develop an intelligent, camera-assisted car that is able to interpret its surroundings automatically, robustly, and in real-time. Even in the specific case of a highway's well-structured environment, this is a difficult problem. Traffic volume, driver behavior, lighting and road conditions, and so forth are unpredictable. Our initial vision system, introduced in Ref. [1], failed in heavy traffic or on highways with cluttered roadsides. Our new system overcomes some of these problems by analyzing the whole highway scene, in particular, by segmenting the road using color information, and then recognizing and tracking lane markings, road boundaries and multiple cars on the road. Our vision system does not need any initialization by a human operator, but recognizes the cars it tracks automatically. The video data is processed in real time without any specialized hardware. All we need is an ordinary video camera and a low-cost PC with an image capture board.

Due to safety concerns, camera-assisted or vision-guided vehicles must react to dangerous situations immediately. Not only must the supporting vision system do its processing extremely fast, i.e., in *soft real time*, but it also must guarantee to react within a fixed time frame under all circumstances, i.e., in *hard real time*. Hansson et al. [7] have developed a non-vision based, distributed real-time architecture for vehicle applications that incorporates both hard and soft real-time processing. However, computer vision research for intelligent vehicles has so far only aimed at soft real-time performance, i.e., fast processing without timing guarantees. In contrast, we use a *hard* real-time system that can predict in advance how long its computations take. We utilize the advantages of the hard real-time operating system "Maruti," whose scheduling guarantees – prior to any execution – that the required deadlines are met and the vision system will react in a timely manner [14].

Approaches for recognizing and/or tracking cars from a moving camera are, for example, given in Refs. [5, 6, 10, 11, 12, 16, 17, 18] and for lane detection, e.g., in Refs. [4, 11, 12, 13]. Related problems are autonomous convoy driving, e.g.,

---

*The work was supported by ARPA, ONR, and the Philips Labs, grants N00014-95-1-0521, N6600195C8619, DASG-60-92-C-0055.

†The author was at the University of Maryland. Email: betke@cs.bc.edu

0-7803-4269-0/97/$10.00 © 1998 IEEE

Ref. [15] and traffic monitoring using a stationary camera, e.g., Ref. [3, 9].

2 VISION SYSTEM OVERVIEW

Given an input of a video sequence taken from a moving car, the vision system outputs an online description of road parameters and locations and sizes of other vehicles in the images. This description could be used to estimate the positions of the vehicles in the environment and their distances from the camera-assisted car. The vision system contains four main components: the car detector, the road detector, the tracker, and the process coordinator (see Figure 1). Ref. [1] provides the details of an early version of the system.

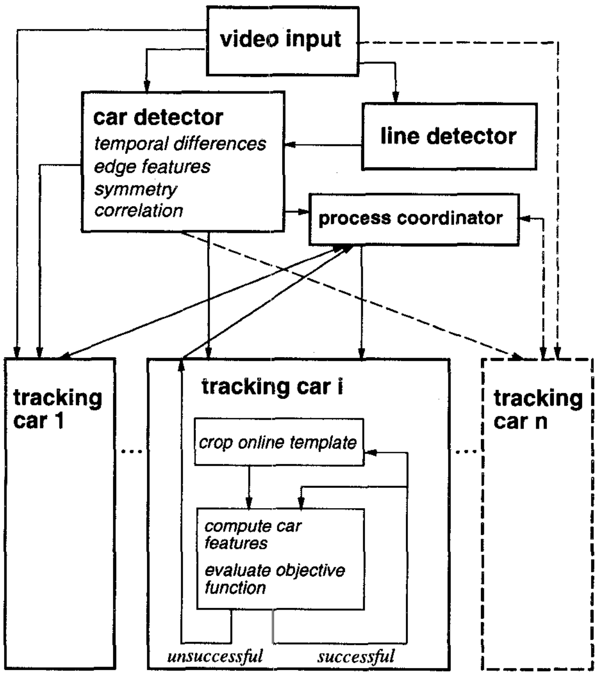

Figure 1: The real-time vision system.

3 COLOR IMAGE ANALYSIS

Color provides information that makes feature detection more robust. For example, in Figure 2, the hue and saturation images highlight the traffic signs, some car features, and the lane markings. The algorithms described below use a composite image as an input that is comprised of the sum of the horizontal and vertical brightness edge maps, hue and saturation images. Figure 2 illustrates the results of modelling the road color using the composite image.

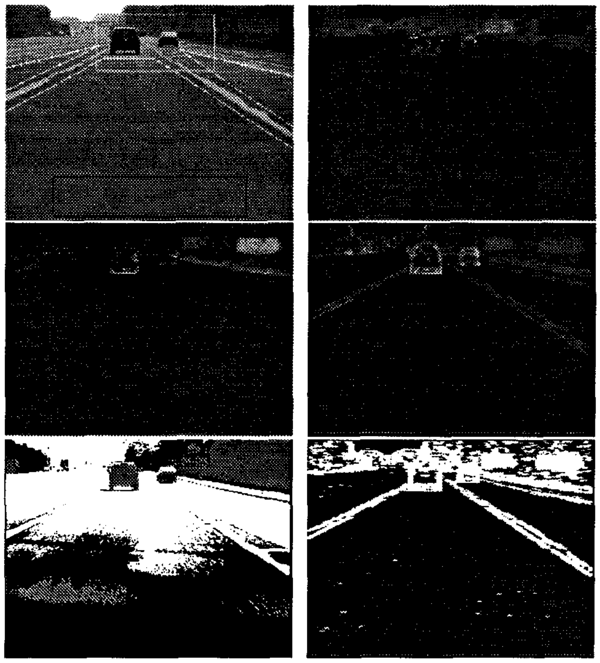

Figure 2: The black-bordered road region in the top left image is used to train road parameter models. The top right and middle images illustrate hue, saturation, and the composite image of the scene. Using the grayscale model, the pixels classified to belong to the road within the white-bordered region in the top left image are shown in black in the bottom left image (only 60% correct classification). Using the composite of hue, saturation and edge information, however, 96% of the pixels are classified correctly as shown in the bottom right image.

4 VEHICLE DETECTION AND TRACKING

The input data of the vision system consists of image sequences taken from a camera mounted inside our car, just behind the windshield. The images show the environment in front of the car – the road, other cars, bridges, and trees next to the road. The primary task of the system is to distinguish the cars from other stationary and moving objects in the images and recognize them as cars. This is a challenging task, because the continuously changing landscape along the road and the various lighting conditions that depend on the time of day and weather are not known in advance. Recognition of vehicles that suddenly enter the scene is difficult. Cars and trucks come into view with very different speeds, sizes, and appearances. In Ref. [1], we describe how passing vehicles are recognized by an analysis of the motion information provided by multiple consecutive image frames. We also describe how vehicles in the far distance, which usually show very little relative motion between themselves and the camera-assisted car, can be recognized by an adaptive feature-based method. Immediate recognition from one or two images, however, is very difficult and only works robustly under cooperative conditions (e.g., enough brightness contrast between vehicles and background). Therefore, if an object cannot be recognized immediately, our system evaluates several image frames and employs its tracking capabilities to recognize vehicles.

Recognition by tracking. The process coordinator creates a separate tracking process for each potential car (see Fig. 1). It uses the initial parameters for the position and size of the potential car that are determined by the car detector and ensures that no other process is tracking the same image area. The tracker creates a "tracking window" that contains the potential car and is used to evaluate composite information, edge maps, and templates in subsequent image frames. The position and size of the tracking window in subsequent frames is determined by a simple recursive filter: The tracking window in the current frame is the window that contains the potential car found in the previous frame plus a boundary that surrounds the car. The size of this boundary is determined adaptively. The outline of the potential car within the current tracking window is computed by a feature search described in detail in Ref. [1]. As a next step, a template of a size corresponding to the hypothesized size of the vehicle is created from a stored model image using the method developed in Ref. [2]. The template is correlated with the image region that is hypothesized to contain the vehicle. If the normalized correlation of the image region and the template is high and typical vehicle motion and feature parameters are found, it is inferred that a vehicle is detected. Note that the correlation coefficient is invariant to constant scale factors in brightness and can therefore adapt to the lighting conditions of a particular image frame. The model images shown in Ref. [1] are only used to create car templates online as long as the tracked object is not recognized to be a vehicle yet; otherwise, the model is created online by cropping the currently tracked vehicle.

Online model creation. The outline of the vehicle found defines the boundary of the image region that is cropped from the scene. This cropped region is then used as the model vehicle image from which templates are created that are matched with subsequent images. The templates created from such a model usually correlate extremely well (e.g., 90%), even if their sizes substantially differ from the cropped model, and even after some image frames have passed since the model was created. As long as these templates yield high correlation coefficients, the vehicle is tracked correctly with high probability. As soon as a template yields a low correlation coefficient, it can be deduced automatically that the outline of the vehicle is not found correctly. Then the evaluation of subsequent frames either recovers the correct vehicle boundaries or terminates the tracking process.

Symmetry check. In addition to correlating the tracked image portion with a previously stored or cropped template, the system also checks for the portion's left-right symmetry by correlating its left and right image halves. Highly symmetric image portions with typical vehicle features indicate that a vehicle is tracked correctly.

Adaptive window adjustments. An adaptive window adjustment is necessary if – after a vehicle has been recognized and tracked for a while – its correct current outline is not found. This may happen, for example, if the rear window of the car is mistaken to be the whole rear

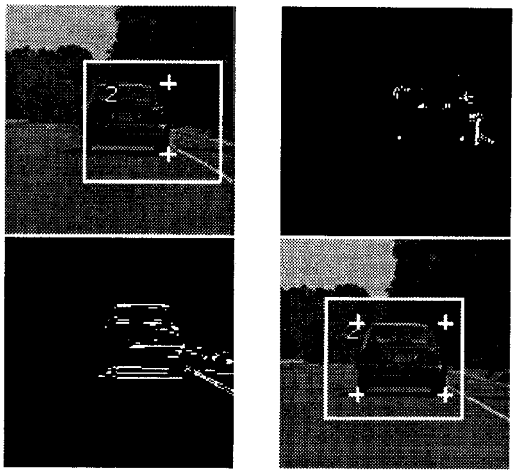

Figure 3: The left car corners are identified incorrectly (top left). The vertical edges (top right) of the car's left side can be found by searching the horizontal edge map (bottom left) and shifting the window to the left (bottom right). The car is fully captured in subsequent image frames.

of the car, because the car bottom is not contained in the current tracking window (the camera may have moved up abruptly). This can be determined easily by searching along the extended left and right side of the car for significant vertical edges. In particular, the pixel values in the vertical edge map that lie between the left bottom corner of the car and the left bottom border of the window, and the right bottom corner of the car and the right bottom border of the window, are summed and compared with the corresponding sum on the top. If the sum on the bottom is significantly larger than the sum on the top, the window is shifted towards the bottom (it still includes the top side of the car). Similarly, if the aspect ratio is too small, the correct positions of the car sides are found by searching along the extended top and bottom of the car for significant horizontal edges, as illustrated in Figure 3.

The window adjustments are useful for capturing the outline of a vehicle, even if the feature search encounters thresholding problems due to low contrast between the vehicle and the environment. The method supports recognition of passing cars that are not fully contained within the tracking window, and it compensates for the up and down motion of the camera due to uneven pavement. Finally, it ensures that the tracker does not lose a car even if the road curves.

6 BOUNDARY AND LANE DETECTION

Road boundaries and lane markings are detected in each frame by a spatial recursive least squares filter (RLS) [8]. The image in Figure 4 illustrates the detected lane and boundary points as black "+" symbols and shows the lines fitted to these points. The graph shows the slope of lane 2, updated after each new lane point is detected.

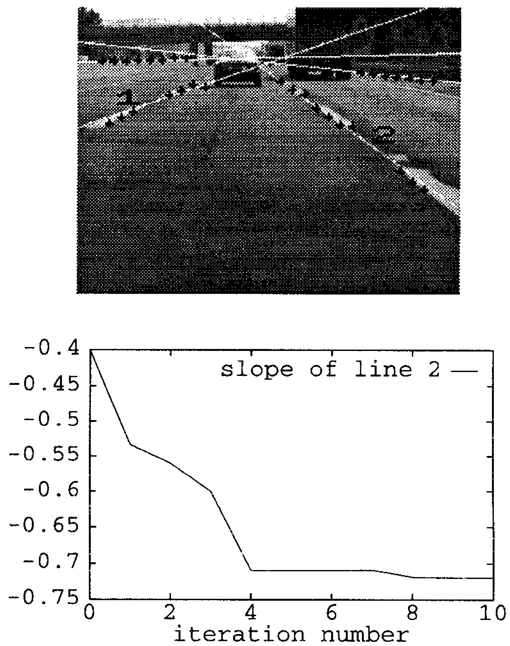

Figure 4: The RLS line and boundary detection.

7 EXPERIMENTAL RESULTS

The analyzed data consists of more than 1 hour of RGB and greyscale video taken on American and German highways. The images are evaluated in both hard and soft real time in the laboratory. A Sony CCD video camera is connected to a 166 MHz Pentium PC with a Matrox Meteor image capture board, and the recorded video data is played back and processed in real time. Table 1 provides some of the timing results of our vision algorithms. Some of the vehicles are tracked for several minutes, others disappear quickly in the distance or are occluded by other cars. Processing each image

Table 1: Duration of Vehicle Tracking

Tracking Time	No. of Vehicles	Average Time
< 1 min	41	20 s
1–2 min	5	90 s
2–3 min	6	143 s
3–7 min	5	279 s

frame takes 68 ms on average; thus, we achieve a frame rate of approximately 14.7 frames per second. The average amount of processing time per algorithm step is summarized in Table 2. To reduce computation costs, the steps are not computed for every frame.

Table 2: Average Processing Time

Step	Time	Average Time
Searching for potential cars	1–32 ms	14 ms
Feature search in window	2–22 ms	13 ms
Obtaining template	1–4 ms	2 ms
Template match	2–34 ms	7 ms
Lane detection	15–23 ms	18 ms

Table 3 reports a subset of the results for our German and American highway data using Maruti's virtual runtime environment. During this test, a total of 48 out of 56 cars are detected and tracked successfully. Since the driving speed on American highways is much slower than on German highways, less motion is detectable from one image frame to the next and not as many frames need to be processed.

The lane and road boundary detection algorithm was tested by visual inspection on 14 minutes of data taken on a two-lane highway under light traffic conditions. During about 75% of the time, all the road lanes or boundaries are detected and tracked (3 lines); during the remaining time, usually only one or two lines are detected.

Our system is robust unless it encounters uncooperative conditions, e.g., too little brightness contrast between the cars and the background and very congested traffic.

8 CONCLUSIONS

We have developed and implemented a hard real-time vision system that recognizes and tracks lanes, road boundaries, and multiple vehicles in videos taken from a car driving on German and American highways. Our system is able to run in real time with simple, low-cost hardware. All we rely on is an ordinary video camera and a PC with an image capture board.

The vision algorithms employ a combination of brightness, hue and saturation information to analyze the highway scene. Highway lanes and boundaries are detected and tracked using a recursive least squares filter. The highway scene is segmented into regions of interest, the "tracking windows," from which vehicle templates are created online and evaluated for symmetry in real-time. From the tracking and motion history of these windows, the detected features and the correlation and symmetry results, the system infers if a vehicle is detected and tracked. Experimental results demonstrate robust, real-time car recognition and tracking over thousands of image frames.

References

[1] M. Betke, E. Haritaoglu, and L. Davis. Multiple vehicle detection and tracking in hard real time. In *Intelligent Vehicles*, pages 351–356, 1996. Also UMD CAR-TR-858, http://www.cfar.umd.edu:80/ftp/TRs/CVL-Reports-1996/TR3667-Betke.ps.gz.

[2] M. Betke and N. C. Makris. Fast object recognition in noisy images using simulated annealing. In *Proc. ICCV*, pages 523–

Table 3: Detection and Tracking Results on Grayscale Video

Data origin	German		American
total number of frames	ca. 5572		9670
No. of frames processed (in parentheses: results normalized for every frame)	every 2nd	every 6th	every 10th
detected and tracked cars	23	20	25
cars not tracked	5	8	3
size of detected cars (in pixels)	10×10–80×80	10×10–80×80	20×20–100×80
avg. no. of frames during tracking	105.6 (211.2)	30.6 (183.6)	30.1 (301)
avg. no. of frames until car tracked	14.4 (28.8)	4.6 (27.6)	4.5 (45)
avg. no. of frames until stable detection	7.3 (14.6)	3.7 (22.2)	2.1 (21)
false alarms	3	2	3

530, 1995. Also MIT AI Memo 1510 and http://www.cs.bc.edu/~betke/.

[3] D. Beymer and J. Malik. Tracking vehicles in congested traffic. In *Intelligent Vehicles*, pages 130–135, 1996.

[4] J. D. Crisman and Ch. E. Thorpe. SCARF: A color vision system that tracks roads and intersections. *IEEE Trans. on Robotics and Automation*, 9:49–58, 1993.

[5] U. Franke and I. Kutzbach. Fast stereo based object detection for stop & go traffic. In *Intelligent Vehicles*, pages 339–344, 1996.

[6] V. Graefe and W. Efenberger. A novel approach for the detection of vehicles on freeways by real-time vision. In *Intelligent Vehicles*, pages 363–368, 1996.

[7] H. A. Hansson, H. W. Lawson, M. Strömberg, and S. Larsson. BASEMENT: A distributed real-time architecture for vehicle applications. *Real-Time Systems*, 11:223–244, 1996.

[8] S. Haykin. *Adaptive Filter Theory.* Prentice Hall, 1991.

[9] T. Ikeda, S. Ohnaka, and M. Mizoguchi. Traffic measurement with a roadside vision system – individual tracking of overlapped vehicles. In *ICPR*, pages 859–864, 1996.

[10] L. Kaminski, J. Allen, I. Masaki, and G. Lemus. A sub-pixel stereo vision system for cost-effective intelligent vehicle applications. In *Intelligent Vehicles*, pages 7–12, 1995.

[11] Q. Luong, J. Weber, D. Koller, and J. Malik. An integrated stereo-based approach to automatic vehicle guidance. In *ICCV*, pages 52–57, 1995.

[12] M. Maurer, R. Behringer, S. Fürst, F. Thomarek, and E. D. Dickmanns. A compact vision system for road vehicle guidance. In *ICPR*, volume 3, pages 313–317, 1996.

[13] D. Pomerleau. RALPH: Rapidly adapting lateral position handler. In *Intelligent Vehicles*, pages 506–511, 1995.

[14] M. C. Saksena, J. da Silva, and A. K. Agrawala. Design and implementation of Maruti-II. In *Principles of Real-Time Systems.* Prentice-Hall, 1994.

[15] H. Schneiderman, M. Nashman, A. J. Wavering, and R. Lumia. Vision-based robotic convoy driving. *Machine Vision and Applications*, 8(6):359–364, 1995.

[16] S. M. Smith. ASSET-2: Real-time motion segmentation and shape tracking. In *ICCV*, pages 237–244, Cambridge, MA, 1995.

[17] F. Thomanek, E. D. Dickmanns, and D. Dickmanns. Multiple object recognition and scene interpretation for autonomous road vehicle guidance. In *Intelligent Vehicles*, pages 231–236, 1994.

[18] T. Zielke, M. Brauckmann, and W. von Seelen. Intensity and edge-based symmetry detection with an application to car-following. *CVGIP: Image Understanding*, 58:177–190, 1993.