Multiple Vehicle Detection and Tracking

Margrit Betke Esin Haritaoglu Larry S. Davis

University of Maryland
Center for Automation Research and
Institute for Advanced Computer Studies
College Park, MD 20742-3275
Phone +1-301-405-6716, Fax +1-301-314-9658, E-mail:betke@umiacs.umd.edu
http://www.umiacs.umd.edu:80/users/betke/index.html

ABSTRACT

A hard real time vision system has been developed that recognizes and tracks multiple cars from video sequences taken from a car driving on highways and country roads. Recognition is accomplished by combining the analysis of single image frames with the analysis of the motion information provided by multiple consecutive image frames. In single image frames, cars are recognized by matching deformable gray-scale templates, by detecting image features, such as corners, and by evaluating how these features relate to each other. Cars are also recognized by tracking motion parameters that are typical for cars.

The vision system utilizes the hard real-time operating system Maruti which guarantees that the timing constraints on the various vision processes are satisfied. The dynamic creation and termination of tracking processes optimizes the amount of computational resources spent and allows fast detection and tracking of multiple cars. Experimental results demonstrate robust, real-time recognition and tracking over thousands of image frames.

Keywords: object recognition, real-time image processing, motion vision, car detection, vehicle tracking, deformable template matching, intelligent vehicles, hard real-time system Maruti.

1 Introduction

We have developed a hard real time vision system that recognizes and tracks cars from video sequences taken from a car driving on highways and country roads. Recognizing and tracking objects in images taken by a *moving* camera (or a fixed camera carried by a moving car) is much more challenging than real-time tracking with a *stationary* camera. Not only is there motion of the objects in the images, but also relative motion between the camera, the objects, and the environment. Our method uses the relative motion between the camera-assisted car and its environment to detect potential cars. To handle cases where there is little relative motion, our method searches for features that are typical for cars. Recognition of a car is verified if an objective function yields a high value. The objective function defines how likely it is that an object with certain parameters is a car. The objective function combines evaluation of the history of tracking a potential car with correlation of the potential car with a deformable template of a car created on-line using the method by Betke and Makris.[3] A detailed description of our hard real-time vision system can be found in Ref. 2.

[0]The support of DARPA, the Office of Naval Research, the Army Strategic Defense Command, and Philips Laboratories under Contracts N00014-95-1-0521 and DASG-60-92-C-0055 is gratefully acknowledged.

SPIE Vol. 2962 • 0277-786X/97/$10.

Various approaches for recognizing and/or tracking cars from a moving camera have been suggested in the literature – for example, detecting symmetry,[17,13,4] approximating optical flow,[16,7] exploiting binocular or trinocular stereopsis,[8,4,10] matching templates,[10] and training a neural net.[11] Related problems are autonomous convoy driving,[15] road detection and following,[1,5,6,8,12] and image stabilization.[9] Unlike some methods described in the literature, our vision system can track more than one car at a time. In addition, it does not need any initialization by a human operator, but recognizes the cars it tracks automatically. Our method also does not rely on having to estimate road parameters. Unlike other methods, our vision system processes the video data in real time without any specialized hardware. All we need is an ordinary video camera and a low-cost PC. Simplicity is the key to the real-time performance of our method. We developed a system that is simple enough to be fast, but sophisticated enough to work robustly.

2 The hard real-time vision system

Given an input of a video sequence taken from a moving car, the vision system outputs an on-line description of the locations and sizes of other cars in the images. This description could be used to estimate the positions of the cars in the environment and their distances from the camera-assisted car. The vision system contains three main components: the car detector, the process coordinator, and the tracker. Once the car detector recognizes a potential car in an image, the process coordinator creates a tracking process for each potential car and provides the tracker with information about the size and location of the potential car. For each tracking process, the tracker analyzes the history of the tracked areas in the previous image frames and determines how likely it is that the area in the current image contains a car. If it contains a car with high probability, the tracker outputs the location and size of the hypothesized car in the image. If the tracked image area contains a car with very low probability, the process terminates. This dynamic creation and termination of tracking processes optimizes the amount of computational resources spent.

The ultimate goal of our vision system is to provide a car control system with a sufficient analysis of its changing environment, so that it can react to a dangerous situation immediately. Whenever a physical system like a car control system depends on complicated computations, such as are carried out by our vision system, the timing constraints on these computations become important. A "hard real-time system" guarantees – prior to any execution – that the system will react in a timely manner. In order to provide such a guarantee, a hard real-time system analyzes the timing and resource requirements of each computation task. The temporal correctness of the system is ensured if a feasible schedule can be constructed. The scheduler has explicit control over when a task is dispatched. Our vision system utilizes the hard real-time system Maruti.[14] Maruti is a "dynamic hard real-time system" that can handle on-line requests. It either schedules and executes them if the resources needed are available, or rejects them. We implemented the processing of each image frame as a periodic task consisting of several subtasks (e.g., distant car detection, car tracking). Since Maruti is also a "reactive system," it allows switching tasks on-line. For example, our system could switch from the usual cyclic execution to a different operational mode. This supports our ultimate goal of improving traffic safety: The car control system could react within a guaranteed time frame to an on-line warning by the vision system which may have recognized a dangerous situation.

3 Car recognition

The input data of the vision system consists of image sequences taken from a moving car. The camera is mounted inside the car just behind the windshield. It takes images of the environment in front of the car, for example, the road, other cars, and trees next to the road. The 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 objects that suddenly enter the scene is difficult. Cars and trucks come into view with very different speeds, sizes, and appearances.

To facilitate robust and fast recognition of cars, we distinguish between recognizing cars that appear in the field of view after having passed the camera from behind, and cars that appear in the far distance in front of the camera. Once a car is recognized, it is tracked in subsequent image frames until it moves out of sight.

3.1 Recognizing passing cars

When other cars pass the camera-assisted car, they are usually nearby and therefore cover large portions of the image frames. They cause large brightness changes in such image portions over small numbers of frames. Large brightness changes over small numbers of frames are detected by differencing the current image j frame from an earlier frame k and checking if the sum of the absolute brightness differences exceeds a threshold in an appropriate region of the image. Region R of image sequence $I(x, y)$ in frame j is hypothesized to contain a passing car if $\sum_{x,y \in R} |I_j(x, y) - I_{j-k}(x, y)| > \theta$ where θ is a fixed threshold. If large brightness changes are detected in consecutive images, a gray-scale template of a size corresponding to the hypothesized size of the passing car is created from a model image using the method by Betke and Makris.[3] It is correlated with the image region R that is hypothesized to contain a passing car. The normalized sample correlation coefficient is used as a measure of how well region and template image correlate or match.

3.2 Recognizing distant cars

Cars that are being approached by the camera-assisted car usually appear in the far distance as rectangular objects. Generally, there is very little relative motion between such cars and the camera-assisted car. Therefore, any method based only on differencing image frames will fail to detect these cars. Therefore, we use a feature-based method to detect distant cars. We look for rectangular objects by evaluating horizontal and vertical edges in the images. The horizontal edge map H and the vertical edge map V are defined by a finite difference approximation of the brightness gradient.

Due to our real-time constraints, our recognition algorithm consists of two processes, a coarse and a refined search. The refined search is employed only for small regions of the edge map, while the coarse search is used over the whole image frame. The coarse search determines if the refined search is necessary. It searches the thresholded edge maps for long, uninterrupted edges. Whenever such edges are found in some image region, the refined search process is started in that region. Since the coarse search takes a substantial amount of time (because it processes the whole image frame), it is only called every 10th frame. In the refined search, the vertical and horizontal projection vectors $\mathbf{v}$ and $\mathbf{w}$ of the horizontal and vertical edges H and V in the region are computed. Figure 1 illustrates the horizontal and vertical edge maps H and V and their projection vectors $\mathbf{v}$ and $\mathbf{w}$. A large component of $\mathbf{v}$ indicates pronounced horizontal edges along the corresponding row in the horizontal edge map H. A large component of $\mathbf{w}$ indicates pronounced vertical edges along the corresponding column of V. The threshold for selecting large projection values is half of the largest projection value in each direction. The projection vector $\mathbf{v}$ of the vertical edges is searched starting from the left and also from the right until a vector component is found that lies above the threshold in both cases. The positions of these components determine the positions of the left and right sides of the potential car. Projection vector $\mathbf{w}$ is searched similarly to find the top and bottom of the potential car.

4 Car recognition based on tracking

Immediate recognition from one or two images 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. The process coordinator creates a separate tracking process for each potential car. 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 edge maps and templates in subsequent image frames. In each frame a refined search within the tracking window provides new estimates of the outline of the potential car and uses them to determine the new window boundaries. In every 10th frame a car template is created on-line and correlated with the object in the tracking window. In each frame, the objective function evaluates how likely it is that the object tracked is a car. It checks the credit and penalty values associated with each process. The process coordinator ensures that two tracking processes do not track objects that are too close to each other in the image. This may happen if a car passes another car and eventually occludes it. In this case, the process coordinator terminates one of these processes. (For example, in Figure 2 process 1 is too close to process 2 in frame 41 and is terminated.)

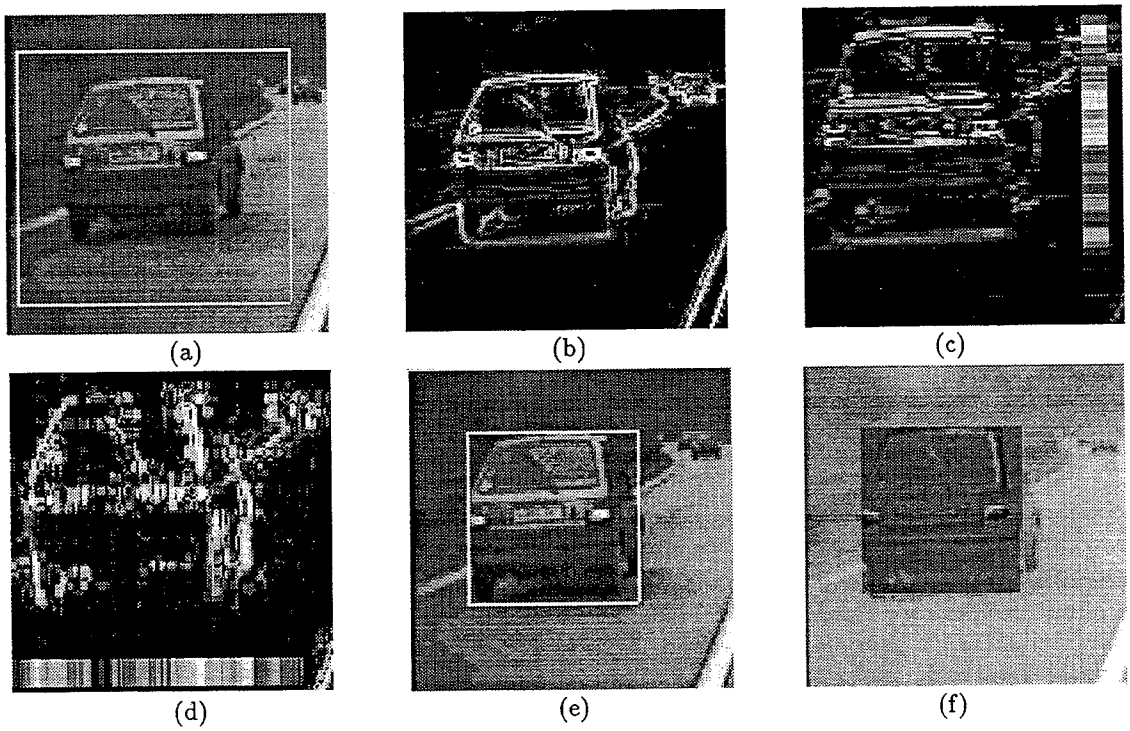

Figure 1: (a) An image with its marked search region. (b) The edge map of the image. (c) The horizontal edge map H of the image (enlarged). The column on the right displays the vertical projection values $\mathbf{v}$ of the horizontal edges computed for the search region marked in (a). (d) The vertical edge map V (enlarged). The row on the bottom displays the horizontal projection vector $\mathbf{w}$. (e) Thresholding the projection values yields the outline of the potential car. (f) A car template of the hypothesized size is overlaid on the image region shown in (e); the correlation coefficient is 0.74.

5 Experimental results

Our data consists of more than 15,200 images taken by a video camera from a moving car on an American and a German highway. The images are evaluated in both hard and virtual real time in the laboratory. Figure 2 shows how three cars are recognized and tracked. Table 1 provides some of the results of our vision system for data that was analyzed in hard real time. The system recognized and tracked a black van for 47 seconds. The duration of tracking is shorter for the other sequences, because the tracked cars quickly disappear in the distance or are occluded by other cars. Processing each image frame takes 98 ms on average in hard real time; thus, we

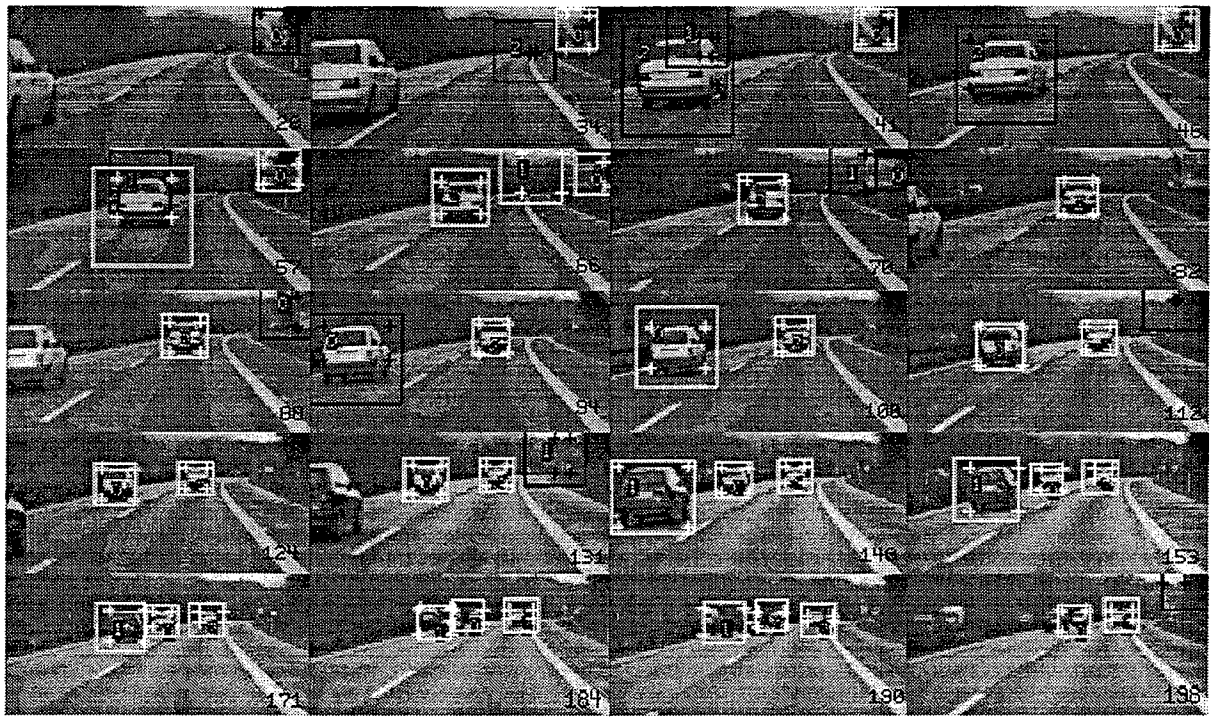

Figure 2: Example of an image sequence in which cars are recognized and tracked. On the top, images are shown with their frame numbers in their lower right corners. The black rectangles show regions within which moving objects are detected. The corners of these objects are shown as crosses. The rectangles and crosses turn white when the system recognizes these objects to be cars. A traffic sign causes a false alarm and is tracked by process 0 in frames 34–66. The graph on the bottom shows the x-coordinates of the positions of the three recognized cars and the false alarm.

Table 1: Hard Real-Time Recognition

Vehicle	Duration of Recognition and Tracking
black van	47 s
white mid-size car 1	8 s
white compact car	13 s
station wagon	13 s
white mid-size car 2	17 s
white car	13 s

achieve a frame rate of approximately 10.2 frames per second. Note that the numbers are averages, because some images are processed quickly (if only a small car is being tracked), while others take longer (e.g., if the complete image is searched). The average amount of processing time is summarized in Table 2.

Table 2: Average Computation Time for the German Data

Step	Image Area	Time
computing edge maps	complete image	88 ms
	(every 10th frame)	
computing horizontal lines		35 ms
differencing image frames	window	1.2 ms
detecting passing cars		19 ms
matching templates	(every 10th frame)	105 ms
tracking cars		2-35 ms

We obtain two sets of results for the German highway data by visual inspection using the virtual runtime environment. In the first set every other frame is processed, and in the second set (given in parentheses) every eighth frame is processed. A total of 23 (20) out of 28 cars are detected and tracked successfully. The image size of the detected cars is between 10×10 and 80×80 pixels. On average, each detected car is tracked during 106 (31) frames. Then it usually is either occluded by other cars, or becomes too small. The car detector notices brightness changes immediately when a passing car appears in the image. The car is tracked by a window of the correct size after 14 (4.6) frames. It takes another 7 (3.7) frames on average until the car tracking has become stable and the detector verifies that it found a car. We encounter 3 (2) false alarms during a total of 58 (19) frames, i.e., scenarios where the system detects a car, but the image does not contain a car at this location. One such false alarm is created by a traffic sign; it is shown in Figure 2, We obtain similar results for the American data. A detailed description of our experimental results can be found in our technical report.[2]

6 Conclusions

We have developed and implemented a hard real-time vision system that recognizes and tracks multiple cars from sequences of gray-scale images taken from a moving car. Unlike several methods described in the literature, our vision system tracks more than one car at a time, recognizes the cars automatically, and relies only on simple low-cost hardware, such as an ordinary video camera and a PC. Our system is robust unless it encounters uncooperative conditions, e.g., too little brightness contrast between the cars and the background, very bumpy roads, sharp tree shadows cast onto the highway, or congested traffic. Extensions of our work will address driving situations under such conditions.

Acknowledgment

We thank the German Department of Transportation for providing the European data and Prof. Ashok Agrawala and his group at the University of Maryland for support with the Maruti operating system.

7 REFERENCES

[1] Reinhold Behringer. Detection of discontinuities of road curvature changes by GLR. In *Proc. Intelligent Vehicles Symposium*, pages 78–83, 1995.

[2] Margrit Betke, Esin Haritaoglu, and Larry Davis. Multiple vehicle detection and tracking in hard real time. Technical Report CAR-TR-834, CS-TR-3667, UMIACS-TR-96-52, University of Maryland, July 1996.

[3] Margrit Betke and Nicholas C. Makris. Fast object recognition in noisy images using simulated annealing. In *Proc. International Conference on Computer Vision*, pages 523–530, Cambridge, MA, June 1995. Also published as MIT, AI Memo-1510.

[4] Stefan Bohrer, Thomas Zielke, and Volker Freiburg. An integrated obstacle detection framework for intelligent cruise control on motorways. In *Proc. Intelligent Vehicles Symposium*, pages 276–281, 1995.

[5] Jill D. Crisman and Charles E. Thorpe. SCARF: A color vision system that tracks roads and intersections. *IEEE Trans. on Robotics and Automation*, 9(1):49–58, 1993.

[6] Ernst D. Dickmanns and Volker Graefe. Applications of dynamic monocular machine vision. *Machine Vision and Applications*, 1(4):241–261, 1988.

[7] W. Krüger, W. Enkelmann, and S. Rössle. Real-time estimation and tracking of optical flow vectors for obstacle detection. In *Proc. Intelligent Vehicles Symposium*, pages 304–309, 1995.

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

[9] Carlos Morimoto, Daniel DeMenthon, Larry S. Davis, Rama Chellappa, and Randal Nelson. Detection of independently moving objects in passive video. In *Proc. Intelligent Vehicles Symp.*, pages 270–275, 1995.

[10] Y. Ninomiya, S. Matsuda, M. Ohta, Y. Harata, and T. Suzuki. A real-time vision for intelligent vehicles. In *Proc. Intelligent Vehicles Symposium*, pages 101–106, 1995.

[11] D. Noll, M. Werner, and W. von Seelen. Real-time vehicle tracking and classification. In *Proc. Intelligent Vehicles Symposium*, pages 101–106, 1995.

[12] Dean Pomerleau. RALPH: Rapidly adapting lateral position handler. In *Proc. Intelligent Vehicles Symposium*, pages 506–511, 1995.

[13] U. Regensburger and V. Graefe. Visual recognition of obstacles on roads. In *IROS*, pages 982–987, 1994.

[14] Manas C. Saksena, James da Silva, and Ashok K. Agrawala. Design and implementation of Maruti-II. In Sang Son, editor, *Principles of Real-Time Systems*. Prentice-Hall, Englewood Cliffs, NJ, 1994. Also available as UMD CS-TR-3181, UMIACS TR-93-122.

[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] Thomas Zielke, Michael Brauckmann, and Werner von Seelen. Intensity and edge-based symmetry detection with an application to car-following. *CVGIP: Image Understanding*, 58(2):177–190, September 1993.