

Towards Vision-Based Intelligent Navigator: Its Concept and Prototype

Jun Miura, Motokuni Itoh, and Yoshiaki Shirai,

Abstract—This paper proposes a novel concept of *intelligent navigator* that can give the driver timely advice on safe and efficient driving. From both the current traffic condition obtained from visual data and the driver's goal and preference in driving, it autonomously generates advice and give it to the driver. Not only *operational level* advice, such as emergency braking due to an abrupt deceleration of the front vehicle, but also *tactical level* advice, such as lane changing due to the congested situation ahead, can be generated. Two main components of the intelligent navigator, the advice generation system and the road scene recognition system, are explained. A Three-level reasoning architecture is proposed for generating advice in dynamic and uncertain traffic environments. On-line experiments using the prototype system show the potential feasibility of the proposed concept.

Keywords—ITS, intelligent navigator, visual recognition of traffic scene, hierarchical reasoning architecture.

I. INTRODUCTION

In recent years, there have been growing interests in ITS (intelligent transportation systems). One ultimate goal of ITS research is to realize a fully autonomous vehicle [1], [2]. It is, however, still difficult to achieve this goal because a very high reliability and safety will be required for deployment. Thus, as a practical step towards the goal, we propose the *intelligent navigator*, that can, in place of a human navigator sitting on the next seat, give the driver appropriate advice on safe and efficient driving.

To drive to a destination, we first select a route from the current location to the destination, and then drive a vehicle to follow the route. Concerning the assistance to route selection, navigation systems have already been deployed which use various information such as GPS data, map data, and traffic condition information provided by local traffic control centers [3]. On the other hand, the assistance to driving itself is still an active research area. This is mainly due to lack of reliable sensory systems and intelligent assistance planner.

Tasks in driving can usually be divided into two levels: at the higher level, maneuvers such as lane changing and overtaking are determined to meet the objective of driving (e.g., a target arrival time) under the constraints imposed by the actual traffic condition; at the lower level, the selected maneuver is translated into actual operations of steering, accelerating, and braking. These levels are called as the *tactical level* and the *operational level*, respectively [4], [5].

Operational level driving can be assisted relatively straightforwardly using various sensing capabilities such as vision for lane detection [6] or for detecting other vehicles [7], [8]; if some dangerous situation arises, the driver can be warned. Sev-

J. Miura and Y. Shirai are with Department of Computer-Controlled Mechanical Systems, Graduate School of Engineering, Osaka University, Suita, Osaka 565-0871, Japan (e-mail: jun.miura@ieee.org and shirai@mech.eng.osaka-u.ac.jp).

M. Itoh is with Graduate School of Science and Technology, Kobe University, Kobe, Hyogo 657-8501, Japan.

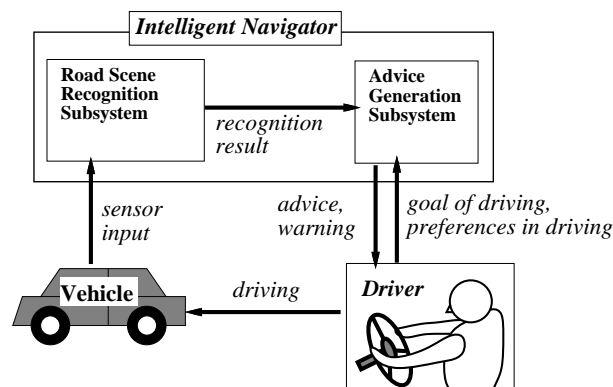


Fig. 1. Overview of intelligent navigator system.

eral commercial products for local level assistance have recently been reported [9]. However assistance systems for the tactical level have little been developed so far.

Sukthankar et al. [5] pointed out the importance of tactical level driving in realizing safe and efficient autonomous driving. This is also true for driver assistance systems. Since the quality of maneuver selection may have considerable effects on safety and efficiency, it is important to generate advice on appropriate maneuvers (i.e., *tactical level advice*) in a timely fashion.

In this paper, we propose an architecture of *intelligent navigator* which can generate tactical level advice as well as local level advice. Fig. 1 schematically depicts the proposed architecture of the intelligent navigator system. The driver gives the system the goal of driving (e.g., the target arrival time) and his/her preference to specific driving styles (e.g., the driver may want to avoid lane changing as much as possible). The road scene recognition subsystem recognizes the current traffic situation using vision. The advice generation subsystem generates appropriate advice and give it to the driver. The driver may control the vehicle according to the given advice. Since there have been many works on visual recognition of road scenes, we can adopt their results to implement our road scene recognition system. Concerning generation of tactical level advice, however, there have been little previous works; this paper, therefore, focuses on how to construct such an advice generation system.

In designing an advice generation system, we have to consider the following two issues. One is the *uncertainty* (or *ambiguity*) in road scene recognition results based on which advice is generated. The other is the *dynamics* of road scene; i.e., the traffic situation evolves as time advances. Therefore, the tactical level advice should be generated based on the prediction of the future traffic condition with consideration of uncertainty. As a mechanism for such advice generation, we propose a three-level

reasoning architecture with probabilistic traffic modeling.

We have developed an intelligent navigator prototype by combining the advice generation system, which is based on the layered reasoning architecture, with a vision system to detect lanes and vehicles. The prototype is for driving on a highway without branches and is implemented as a stand-alone system such that it is easily installed in a normal, unmodified vehicle. To the authors' knowledge, the prototype is the first *on-line* system for tactical level advice generation.

It is very costly to always use a real prototype for verifying the advice generation algorithm and the traffic models. Therefore we developed a highway traffic simulator; before on-line experiments, we ran simulated experiments many times to improve the algorithms and the models. The simulator can generate various traffic conditions by setting the road configuration, the number of vehicles, and the driving strategy of each vehicle.

The rest of the paper is organized as follows. Section II describes the proposed three-level reasoning architecture with comparison with previous works. Section III explains how to generate tactical level advice; that is, how the driver gives the system the goal of driving and his/her preferences in driving, how the uncertain and dynamic traffic situation is modeled probabilistically, and how tactical level advice is generated from such information. The Meta-tactical level reasoning is also explained in this section. Section IV describes our simple but robust traffic scene recognition subsystem. Section V describes preliminary experimental results. Section VI summarizes the paper and discusses future works.

II. THREE-LEVEL REASONING ARCHITECTURE

This section describes our three-level reasoning architecture suitable for advice generation in uncertain and dynamic environments.

There have been several works on decision-making considering the uncertainty and the dynamics of environments. Niehaus and Stengel [10] modeled the movement of a nearby vehicle using a probabilistic distribution, which is continuously updated using the Kalman filtering, and generated a safe plan considering the probable worst-case scenarios. Only a local and short-time prediction is performed in planning. Forbes et al. [11] proposed to model all levels of planning for an automated vehicle using a fixed probabilistic network. Although they proposed an efficient computation method for the network, extending the approach to more complicated scenarios may still be difficult because of increasing computational cost. Sukthankar et al. [5] proposed a distributed reasoning scheme for the tactical level planning. Independently operating planning modules with different algorithms vote for the desirable action, and the high-scored action is selected and executed. The parameters and the relative weight of each planning module are tuned through an evolutionary learning method. The proposed scheme seems fitted to the tactical level planning that requires a relatively short-term prediction.

To make a plan with a long look-ahead tends to be computationally expensive if all alternatives are considered in every situation. Moreover, it may be inefficient to always carry out such a planning. Therefore, we propose to introduce a knowledge-based meta-level planning (called the *meta-tactical level*) to

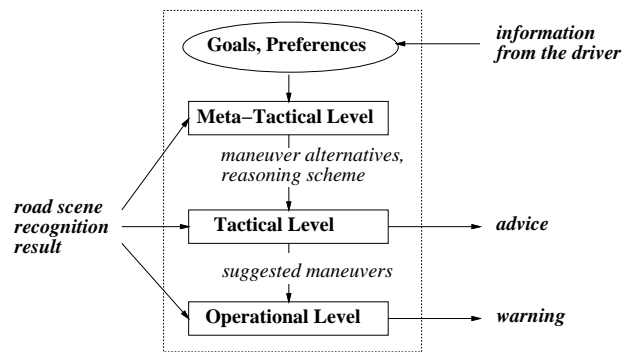


Fig. 2. Three-level architecture of advice generation subsystem.

control the tactical level, i.e., to adaptively limit the search space of the tactical level and to activate the tactical level only when it is necessary, according to both the history of maneuvers and the current traffic condition. The resultant control architecture is composed of three levels: *meta-tactical level*, *tactical level*, and *operational level*.

Fig. 2 illustrates the internal architecture of the advice generation subsystem, which is based on the above three-level reasoning architecture. The meta-tactical level continuously watches designated events on traffic and, on occurrence of an event, activates an appropriate tactical level maneuver selection procedure. Then the tactical level determines the best maneuver to suggest and give it to the driver. The operational level mainly checks immediate dangers, such as an abrupt deceleration of the front vehicle, by watching near surrounding areas of the vehicle.

In robotics, several layered architectures have been proposed. Gat [12] proposed a three-level control architecture for autonomous robots. In his architecture, called ATLANTIS, the controller is responsible for controlling primitive activities, which are usually reactive sensorimotor processes; the deliberator controls time-consuming computational activities such as planning and world model maintenance; the sequencer coordinates such various activities by initiating and terminating them according to the current goal and situation. Pell et al. [13] proposed a similar architecture for autonomous spacecraft. Such works mainly discuss how to integrate deliberative and reactive activities, which correspond to the tactical- and operational levels of our architecture, respectively. Therefore our approach of putting a knowledge-based meta-level is different from those of the previous works; the meta-level can also be applied to their architectures.

III. GENERATING TACTICAL LEVEL ADVICE

A. Information from the Driver

The intelligent navigator receives the driver's goal and preference in driving which are to be used for advice generation. We represent such information in the forms of *loss function* and *cost assignment*, because a statistical decision theory is used for advice generation.

Loss function $L(t)$ is used to represent drivers' requirements on the time of arrival at the destination (the target exit). $L(t)$ is defined using the *target* arrival time t_{target} and the *estimated*

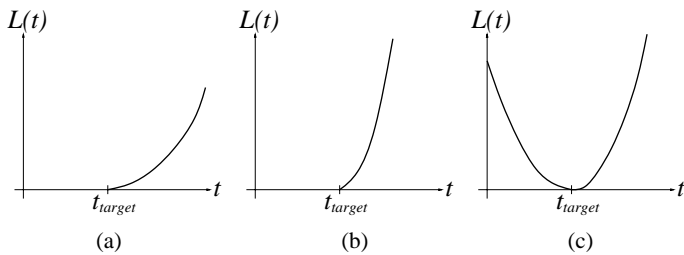


Fig. 3. Examples of $L(t)$:

- (a) loose requirement on the arrival time;
 (b) tight requirement on the arrival time;
 (c) both late and early arrivals are undesirable.

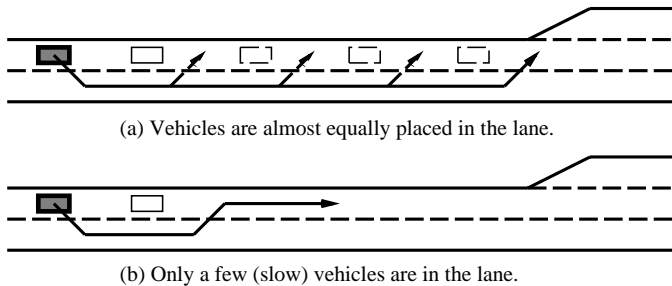


Fig. 4. Estimated traffic situations and overtaking scenario.

arrival time t . Fig. 3 shows some examples of $L(t)$. A tighter requirement on t_{target} is represented by a larger gradient of $L(t)$. The loss function could be changed during driving according to the change of the goal.

Cost C is used to represent the driver's degree of preference to each maneuver. For example, C_{change} is the cost related to the lane changing maneuver; if the driver puts the largest importance on safety and very much wants to avoid possible risks related to lane changing, a high C_{change} is given to the system so that the lane changing maneuver is least likely selected except in a truly emergency situation.

B. Tactical Level Reasoning Considering Uncertainty and Ambiguity of Road Scene Recognition

The tactical level reasoning basically runs as follows. First, the expectation of the arrival time at the destination when each maneuver is adopted is estimated from the probabilistic model of the current traffic situation. Using the estimated arrival time, we calculate the loss of each candidate maneuver using the loss function and the cost attached to the maneuver. Finally the best maneuver is selected which minimizes the loss.

We also consider the *ambiguity* of a recognition result. When the road is moderately occupied by vehicles, the visual recognition subsystem attached to the vehicle cannot obtain information of the whole surroundings due to occlusion and, thus, may not be able to determine the traffic situation uniquely. In such a case, the best maneuver in each possible situation is first selected, and then, if a maneuver is selected in *every* situation, that one is selected for advice. Otherwise, any advice is not generated. Using information from the driver, however, a useful advice could be generated. That is, since the recognition ability of the driver is usually better than that of the recognition system, the driver may

have more information (e.g., the driver can see the vehicles far ahead through the windows of the front vehicle). Therefore, the system provides *conditional* advice which complementarily uses the driver's recognition ability, when it is useful. An example is shown below.

Consider the scenario shown in Fig. 4¹. The vehicle with the intelligent navigator (called *MyVehicle*, drawn as a painted rectangle in the figure) on the left lane² is approaching the exit to take. Since the speed in the current lane is becoming a little bit slow, the intelligent navigator starts thinking of advising the driver to overtake vehicles ahead. The overtaking maneuver is generally faster, but there may be risks of lane changing itself and of missing the exit. We evaluate two maneuvers, *keep-lane* and *overtake*, as follows.

Suppose only vehicles just before and behind *MyVehicle* are visible. There are two possible cases for the occluded area:

- *congested*: vehicles are almost equally placed in the lane (Fig. 4(a))
- *not congested*: just a few (slow) vehicles are blocking our lane (Fig. 4(b)).

The system tries to identify the situation using the method described in Section III-C. If the situation is identified, we evaluate candidate maneuvers for the identified situation. Otherwise, evaluation is carried out for both situations as follows.

Let t_A be the estimated arrival time of *overtake* maneuver in the congested situation, t_B be that of the same maneuver in the other situation, and t_S be that of *keep-lane* maneuver (the result of this maneuver is supposed to be equal in both situation). In addition, the cost $C = C_{change}$ is considered for *overtake* maneuver.

The result of comparison and given advice are as follows:

- If $L(t_A) + C > L(t_B) + C > L(t_S)$, *keep-lane* maneuver is always better than the other; the given advice is "Keep Lane" (see Fig. 5(a)).
- If $L(t_S) > L(t_A) + C > L(t_B) + C$, *overtake* maneuver is always better; the advice is "Change Lane" (see Fig. 5(b)).
- Otherwise (i.e., $L(t_A) + C > L(t_S) > L(t_B) + C$), *overtake* maneuver is better only in *Not Congested* situation; thus, the conditional advice "Change Lane If Not Congested" is generated (see Fig. 5(c)).

C. Traffic Situation Estimation Using Velocity Map

In some cases, it is possible to estimate the situation of occluded areas by the velocity difference of the two lanes. For example, normally the velocity difference can be supposed to be small at the position far from any entrance or exit. If the velocity of our lane is slower than the other, it is likely to be that just a few slow vehicles are blocking our lane and it is not congested.

We verified the validity of such a state estimation using the highway traffic simulator. We performed simulation run many times with setting various parameters to vehicles; in each situation occurred, we recorded the average velocity of both lanes and, at the same time, manually classified the situation into *congested* and *not congested*. Fig. 6 shows the compiled result for

¹This was originally presented by Sukthankar et al. [5].

²Note that the slower lane is the left one in Japan.

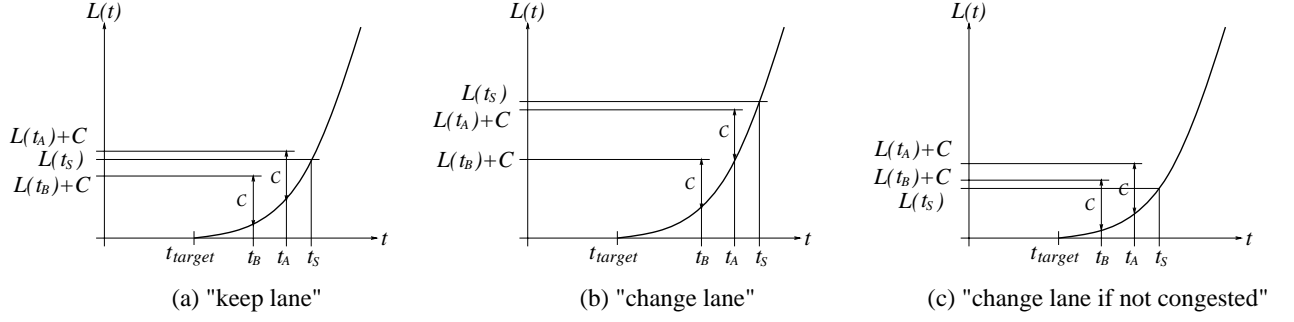


Fig. 5. Maneuver evaluation and given advice.

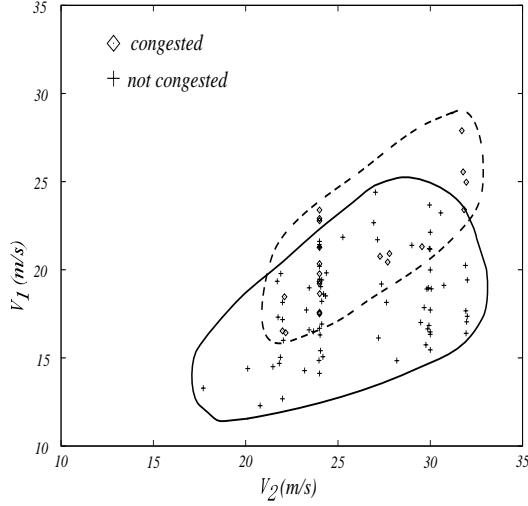


Fig. 6. The result of simulation experiment for the velocity map.

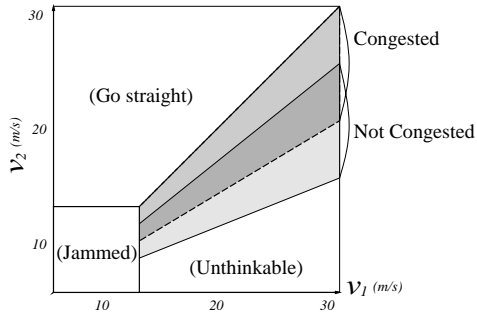


Fig. 7. An example of the velocity map.

the case that the distance to the next exit is medium. In the figure, v_1 and v_2 represent the average velocity in the slower and the fast lanes, respectively. The result shows that the *congested* situation occurs when the velocity difference is smaller (the area in the dotted line) and the *not congested* situation occurs when the velocity difference is large (the area in the solid line). Using the result, we constructed the *velocity map* like the one shown in Fig. 7 that supports the estimation of occluded areas.

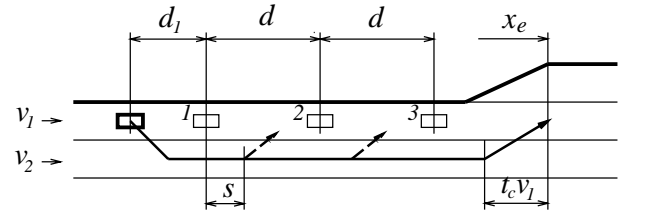


Fig. 8. Overtaking with consideration of approaching exit.

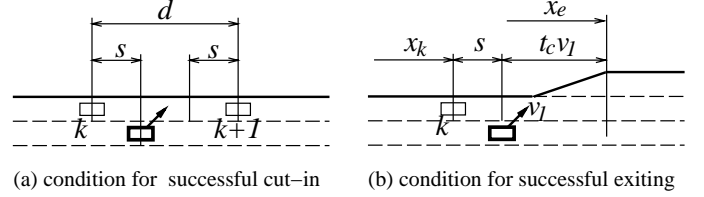


Fig. 9. Conditions for overtaking.

D. An Example of Probabilistic Traffic Modeling: Overtaking with Consideration of Approaching Exit

This subsection describes an example of probabilistic traffic modeling. The situation for modeling is the one shown in Fig. 4(a). Using the model, we calculate the expectation of the arrival time at the target exit when taking *overtake* maneuver (i.e., overtaking front vehicles and returning to the left lane before the exit).

Let us consider Fig. 8. Let v_1 and v_2 be the average speed on the left and the right lanes, respectively, and d be the distance between vehicles in occluded areas. We model the uncertainty of d by a normal distribution. Let μ and σ^2 be the mean and the variance of the normal distribution, respectively. These parameters are related to the lane speed, and have been empirically obtained [14]. Given μ and σ^2 , the distribution of the current position $x_k(0)$ of the k th car ($k \geq 2$) ahead at time $t = 0$ is specified by the following mean u_k and variance σ_k^2 :

$$\mu_k = d_1 + (k-1)\mu, \quad (1)$$

$$\sigma_k^2 = \sigma_1^2 + (k-1)\sigma^2, \quad (2)$$

where d_1 and σ_1^2 are the mean and the variance of the distance to the front vehicle, respectively, obtained by vision.

Let us calculate the probability that *MyVehicle* can safely

takes the exit after overtaking k cars ahead. First, we consider the condition that *MyVehicle* can cut in the space between the k th and $k+1$ th vehicles (see Fig. 9(a)). The probability P_{i_k} that this condition holds is given by

$$P_{i_k} = P(2s \leq d), \quad (3)$$

where s is the safety margin for entering. Next, consider the condition that *MyVehicle* does not miss the exit. This condition is restated as the condition that the position of *MyVehicle* when it finishes overtaking the k th vehicle is sufficiently before the exit (see Fig. 9(b)). The probability that this condition holds is calculated as:

$$P_{e_k} = P(x_k(t_k) + s \leq x_e - t_c v_1), \quad (4)$$

where t_c is the time for lane changing, x_e is the position of the exit; t_k is the time for overtaking k vehicles. Since t_k satisfies the following equation:

$$x_k(t_k) = x_k(0) + v_1 t_k = v_2 t_k - s, \quad (5)$$

t_k is given by

$$t_k = \frac{x_k(0) + s}{v_2 - v_1}. \quad (6)$$

From eqs. (4)(5)(6), we obtain

$$P_{e_k} = P\left(x_k(0) + s \leq \frac{(x_e - t_c v_1)(v_2 - v_1)}{v_2}\right). \quad (7)$$

Assuming that the above two conditions are mutually independent, the probability P_k of overtaking k vehicles and then successfully taking the exit is $P_k = P_{i_k} P_{e_k}$. In addition, the elapsed time t_{e_k} until *MyVehicle* reaches the exit after overtaking k vehicles is given by

$$t_{e_k} = (x_e - \mu_k - s)/v_1. \quad (8)$$

Using the above equations, the expectation of the arrival time t_A (see Sec. III-B) is given by

$$\begin{aligned} t_e &= T_n, \\ T_k &= P_k t_{e_k} + (1 - P_k) T_{e_{k-1}}, \quad (k = 1, \dots, n) \\ T_0 &= t_f, \end{aligned} \quad (9)$$

where n is the index of the farthest vehicle that *MyVehicle* possibly overtakes (i.e., $P_{i_n} > 0$ and $P_{i_{n+1}} = 0$); t_f is the expectation of the arrival time in case that *MyVehicle* cannot overtake any vehicles ahead, and can be calculated similarly to the case of t_A .

On the other hand, the estimated time t_S of the *keep-lane* maneuver (see Sec. III-B) is given by x_e/v_1 . To verify the feasibility of the above model, by using the simulator, we calculated and compared t_A and t_S for several combinations of μ 's (the mean of distance between vehicles) and x_e 's (the distance to the target exit) in the case where $v_1 = 80$ [km/h] and $v_2 = 100$ [km/h]. The result is summarized in Fig. 10. From the figure, we see that the larger μ is, or the larger x_e is, the more *overtake* maneuver has an advantage. This result seems reasonable compared with our intuition.

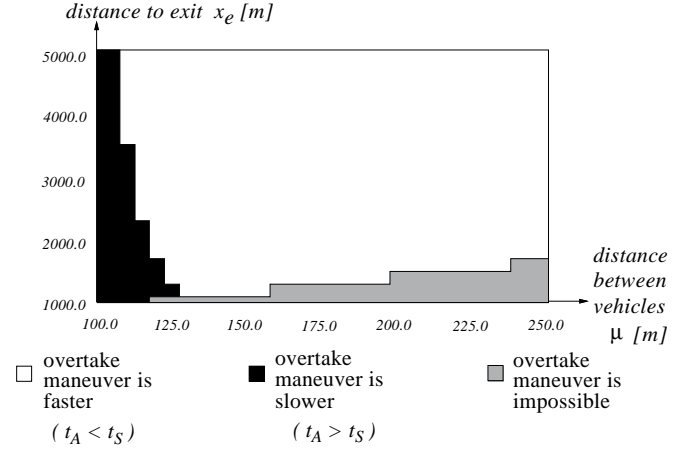


Fig. 10. Comparison of expected arrival times for *overtake* and *keep-lane* maneuvers. Gray area indicates the case where *overtake* maneuver is impossible because the distance to the target exit is too small compared with that to the front vehicle.

E. Other Probabilistic Models

We classified possible situations which may occur in highway driving and, for each situation, we constructed similar probabilistic models to calculate the expected arrival time of each maneuver [15]. In addition to the situation described above, models are constructed for the following situations: overtaking near the target exit without congestion (see Fig. 4(b)), overtaking near an entrance or an exit which *MyVehicle* does not take, overtaking without consideration of any entrance or exit, and lane changing for exiting. An appropriate model for the current situation is selected on-line using the meta-level reasoning described below.

F. Meta-Tactical Level Reasoning

The meta-tactical level (see Fig. 2) continuously watches important events on traffic. Examples of possible events are: the average speed of the current lane slows down; the exit is approaching. It also periodically updates the estimate of the arrival time.

It is inefficient to always check all events. It is also annoying for the driver to be given the same advice such as “Keep Lane” again and again. Therefore, we need to make the system monitor only selected events which are considered to be important in the current state. To realize such an adaptive focus of attention, we construct a state transition graph shown in Fig. 11. For each state, possible events and their corresponding tactical level reasoning procedures are retrieved from the graph. For example, at state [Exit: Medium, Lane: Left] (which means that the distance to the exit is medium and the vehicle is on the left lane), possible events are: (1) the speed becomes slower (Speed: Slower); (2) the estimate of the arrival time is updated (Estimate arrival time); (3) the exit becomes near (Exit: Near); and (4) the driver changed the lane without advice (Change lane by the driver). For the first two events, the tactical level planning overtaking with approaching exit is executed. For the last two events, only the state update is performed. The current transition graph is based on the following assumptions: there are

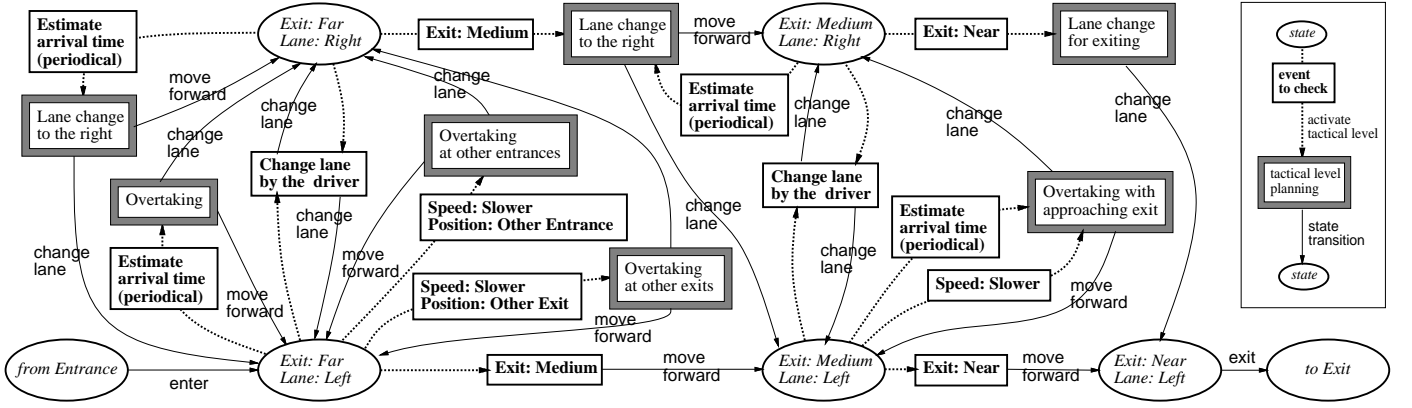


Fig. 11. State-transition graph for meta-tactical level.

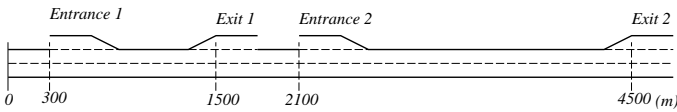


Fig. 12. A highway configuration for simulation.

only two lanes and no branches; the right lane is always faster than the left. It is, however, not difficult to extend the graph to remove such assumptions.

G. Simulation Results

We tested the tactical level reasoning using the highway traffic simulator. Fig. 12 shows the highway configuration used for simulation. *MyVehicle* enters at *Entrance 1* and exit from *Exit 2*. The loss function used is:

$$L(t) = \begin{cases} 0 & t \leq t_{target} \\ (t - t_{target})^2 & t > t_{target} \end{cases} \quad (10)$$

Other vehicles are set to have their own target speed and distance to the front vehicle, and to overtake whenever a slower vehicle is ahead.

We examined the variation of driving of *MyVehicle*, which is assumed to always follow the generated tactical level advice, for various target arrival time t_{target} and the cost of lane changing C_{change} . For each parameter set, we ran the simulation 15 times with setting the parameters for other vehicles randomly, and calculated the average number of lane changes and the average arrival time. Table I summarizes the result. The earlier the target arrival time is, or the smaller the cost of lane changing is, the more *MyVehicle* performs lane changing. Since this result coincide with our intuition, we think our tactical level reasoning is effective.

IV. ROAD SCENE RECOGNITION

This section describes the road scene recognition subsystem (see Fig. 1) which detects lanes and vehicles. Although many works uses stereo vision (e.g., [16], [17]), that requires more cameras and more computing power than using a single camera. We, therefore, use a single camera for each direction (front and

TABLE I
Number of lane changes (left) and actual arrival time in seconds (right).

t_{target}	C_{change}							
	100		400		900		1600	
160	2.7	178.9	2.1	181.3	1.7	182.5	1.5	188.5
170	2.4	180.3	1.9	181.9	1.8	186.7	1.5	188.5
180	1.9	180.5	1.3	183.5	0.9	187.1	1.0	192.9
190	1.7	184.5	0.9	187.1	1.0	192.9	0.7	197.3
200	0.9	187.1	1.1	193.0	0.7	197.3	0.5	198.6

rear). The rear-directed camera is used for recognizing the vehicles passing from the rear in order to check the safety of lane changing maneuver. The effective use of the rear camera will be shown in Sec. V-B. We use shadows under vehicles as the cue for vehicle detection [18].

For the purpose of driver assistance, the recognition system has to process images almost in real-time. On the other hand, we have to consider the uncertainty in visual information for reliable recognition. Therefore, we have decided to adopt a simpler image processing for each frame with an explicit uncertainty modeling, and to statistically integrate information from an image sequence to reduce the uncertainty.

The recognition process is composed of the following steps:

1. Detect lane boundaries and estimate the position of *MyVehicle*.
2. Detect other vehicles and estimates their relative position, relative velocity, and their uncertainties.
3. Make correspondence between frames and integrate data using Kalman filter.
4. Track vehicles based on template matching.

A. Lane Boundary Detection and Vehicle Position Estimation

First, the system extracts white regions corresponding to the two white boundaries of the current lane by thresholding the image and labeling. Then, a line is fitted to each set of white regions. The region between the two lines is considered as the current lane. Using the width of the lane, the image regions of other lanes can be extracted. Figs. 13(a)-(c) show an input image, the detected white regions, and the detected lane boundaries. From

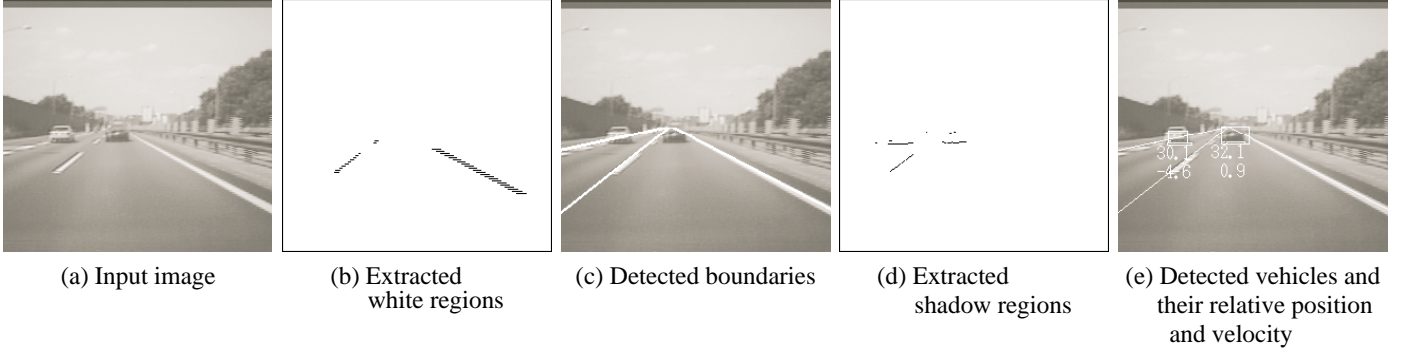


Fig. 13. Lane and vehicle detection.

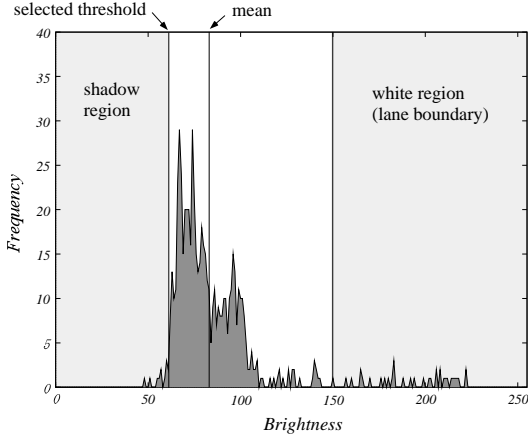


Fig. 14. An example of histogram of the lane region and threshold.

the detected boundaries, the lateral position of *MyVehicle* (i.e., the lane on which *MyVehicle* is) is determined. The longitudinal position of *MyVehicle* is estimated by observing the *dashed* lane boundaries between lanes. The size and the relative position of such lane boundaries are determined by traffic regulations. By counting how many such boundaries pass *MyVehicle*, the system can roughly estimate the longitudinal position from the starting point. From the map of the road and this longitudinal position, the system estimates the distance to an entrance or an exit, and if the distance becomes less than a certain threshold, *MyVehicle* is considered to reach the entrance or the exit. The empirical accuracy of the longitudinal position was a few hundred meters for a driving of about 12 km. This accuracy will easily be increased by using GPS systems.

B. Vehicle Detection

Once the lane regions are extracted, the system searches them for vehicles. Since there is a shadow area under a vehicle, we extract a dark region whose brightness is less than a threshold, which is adaptively determined by the mean and variance of the histogram derived from brightness on the lane region (see Fig. 14). Fig. 13(d) shows the extracted shadows for the image shown in Fig. 13(a), which are the candidates of the vehicle position.

Fig. 15 is the projection of shadows and the detected vehicles

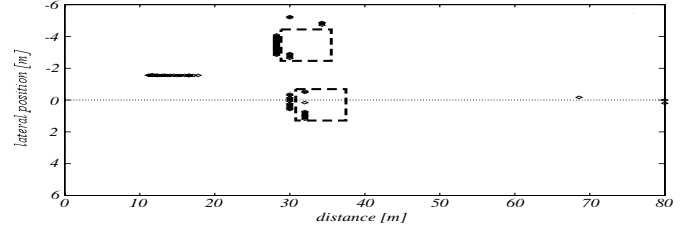


Fig. 15. Projected vehicle position.

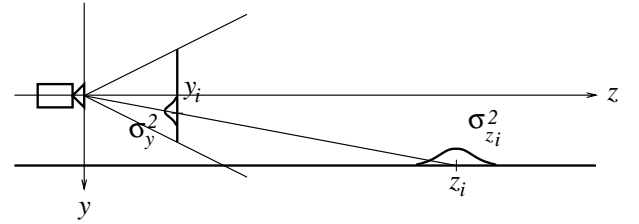


Fig. 16. Calculation of vehicle position and uncertainty.

(dotted rectangle) on the road surface. Since the approximate size of vehicles is known, the shadows of different sizes are determined not to be vehicles.

For each vehicles, we calculate the longitudinal position z_i and its uncertainty $\sigma_{z_i}^2$ (see Fig. 16) by:

$$z_i = \frac{fh}{y_i},$$

$$\sigma_{z_i}^2 = \left(\frac{fh}{y_i^2} \right)^2 \sigma_y^2 = \frac{z_i^4}{(fh)^2} \sigma_y^2, \quad (11)$$

where y_i is the averaged vertical position of a dark region and σ_y^2 is its variance; the uncertainty of the image position is mainly due to the quantization error.

C. Making Correspondence over Frames

We make correspondence of extracted vehicles over frames for reliable recognition. A newly obtained vehicles is matched with a previously detected vehicle if:

1. Both are on the same lane, *and*
2. The difference of positions is within a certain range computed from the previous uncertainty estimate.

TABLE II
Processing time.

process	time (msec)
lane boundary detection	4
vehicle detection	4
filtering	20
tracking	17
total	45

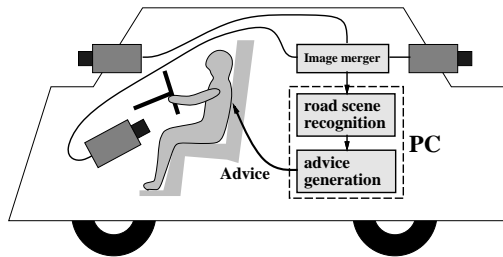


Fig. 17. Configuration of prototype system.

The data of a matched vehicles is integrated with the previous data using Kalman filter [19].

D. Tracking using Template Matching

When a vehicle is detected, the corresponding image region of a certain size is registered as a template. Then the vehicle is tracked by template matching based on the normalized correlation, and the result of the tracking is used for the check of lane changing motion of other vehicles. The rectangles in Fig. 13(e) show the results of template matching (each white rectangle indicate a template). The relative position and velocity of each vehicle are also indicated below the template.

E. Processing Time

Table II shows the computation time for each step of recognition. We used a PentiumII-400MHz PC with Hitachi IP-5000 image processor and measured the time for the case where three vehicles are being tracked.

V. EXPERIMENTAL RESULTS

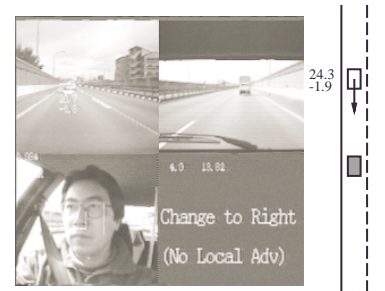
A. Prototype System and Experiments

Fig. 17 shows the overview of the prototype system. Two cameras are used to see the forward and the backward directions. Another camera will be used for observing the driver's behavior (e.g., response to the advice); at present, the system detects just the direction of the face, and the information is not used for the assistance. Two subsystems are working on one PC (Pentium-II 400MHz), and the processing ability reaches 10 frames/sec.

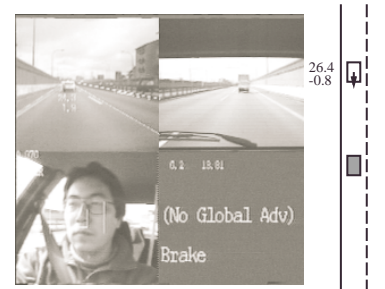
We show the result of an on-road experiment. The vehicle ran from Toyonaka entrance to Ibaraki exit of the Meishin expressway; the travel distance is about 12 km. We gave the system the loss function indicated by eq. (10), target arrival time $t_{target} =$

TABLE III
Measured values and estimated times and losses.

triggered time t		126.5 (sec.)
distance from the entrance		2783 (m)
average speed	left	21.9 (m/sec.)
	right	25.3 (m/sec.)
estimated arrival time	keep lane	421.0 (sec.)
	change lane	364.3 (sec.)
estimated loss	keep lane	7638.8
	change lane	3448.6



(a) $t = 48.9$



(b) $t = 50.2$

Fig. 19. The driver is advised (warned) to brake.

460(sec), and the cost of lane changing $C_{change} = 2500$. This cost means that 50 seconds delay is allowed in exchange for lane changing. The actual arrival time was 527(sec.); during the travel, the system generated the tactical level and the local level advice 5 times and 10 times, respectively.

B. Tactical Level Advice

Fig. 18 shows the situation where tactical level advice "Change to Right Later" was generated due to a reduction of the speed of the current lane. The upper-left and the upper-right part of each image are respectively the forward and the backward view. The lower-left part is the observation of the driver, and the advice is displayed on the lower-right part. The corresponding 2D map of vehicle placement (i.e., relative position and velocity) is also shown at the side of each image. In the map, dotted rectangles indicate vehicles which are invisible but are being tracked by the Kalman filter.

In this case, by referring to the velocity map, only the situation *Congested* was selected. Table III summarizes the measured values of the road scene and the estimated times of arrival

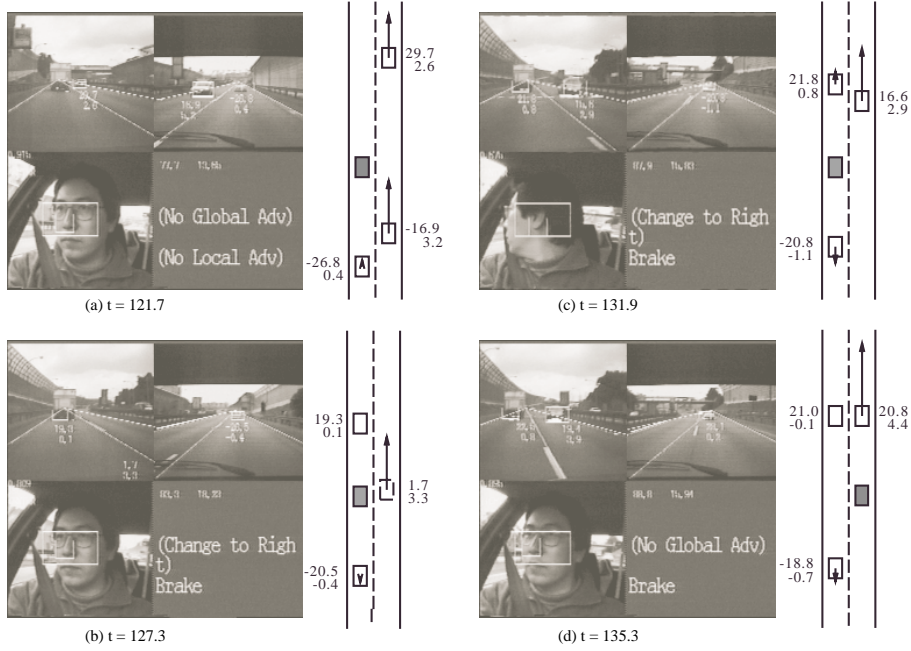


Fig. 18. The driver is advised to change lane later.

at the goal and their losses. The maneuver “Change Lane” was selected at the tactical reasoning module, but on the faster lane there was a passing vehicle. The vehicle, which had been detected by the backward camera (see Fig. 18 (a)), was tracked by filtering although it was not detected by two cameras at the moment (Fig. 18 (b)). Thus, the system advised the driver to “Change to Right Later” (Fig. 18 (b)) (Note that the advice in the parentheses means that the action should be executed later). Following the advice, the driver checked the faster lane (Fig. 18 (c)), and changed to the lane after the vehicle had passed (Fig. 18 (d)).

C. Operational Level Advice

Fig. 19 shows the situation where a local level advice “Brake” was generated due to the deceleration of the front vehicle. At present, this advice is generated if the following condition is satisfied:

$$d_{curr} + v_{curr}T_f < d_{thresh}, \quad (12)$$

where d_{curr} (v_{curr}) is the current relative position (velocity) of the front vehicle with respect to *MyVehicle*; T_f is a constant time duration (currently 3 (sec.)); d_{thresh} is a threshold (currently 20 (m)). In the case of Fig. 19, d_{curr} and v_{curr} were estimated as 24.5 (m) and -2.0 (m/sec.); thus, the relative position after T_f (sec.) was estimated as 18.5 (m), which is less than d_{thresh} .

VI. CONCLUSIONS AND DISCUSSION

This paper has proposed the concept of intelligent navigator that can give the driver timely advice on driving in a dynamic and uncertain traffic environment. The intelligent navigator system is composed of the advice generation and the road scene recognition subsystems. We have proposed a three-level reasoning architecture for tactical level advance generation. We also

constructed a prototype system and conducted experiments on the actual highway. The experimental results show that the intelligent navigator can provide reasonable advice on-line.

Experimental evaluation of the system like the intelligent navigator is, however, difficult because traffic situations that we would like for evaluation do not necessarily occur at the time of experiments. So we tested and improved our advice generation strategy by performing simulation repeatedly for various simulated traffic situations, and then, implemented it on the prototype system. We conducted several on-road experiments; during the experiments, one of the authors actually drove the vehicle, and followed the advice when it was generated. His subjective impression is that most of advice were generated in appropriate timings.

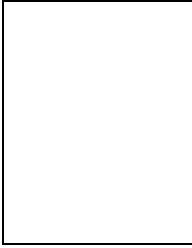
It is not an easy work for the driver to give the loss function and the cost assignment, which are used in the tactical level reasoning. We are planning to employ a learning-based approach in which the parameters are continuously refined through the interaction with the driver.

To deploy the intelligent navigator, the following two issues should be solved. One is the reliability of road scene recognition. The current vision algorithm works well under good lighting conditions but is not robust against drastic weather changes. We are working on developing more reliable image processing algorithms. Using other kinds of sensors such as millimeter-wave radar [20] may be a practical way. The other issue is the use of (more) global information for increasing the reliability of advice. From the on-vehicle sensors, only the information in the local area can be obtained and more global information is estimated from such local information. For example, we currently estimate the situation of occluded areas *indirectly* using the speed difference between lanes. It is useful to use roadside cameras [21], if available. In addition, more global information such as the congestion status on a far away location, which will

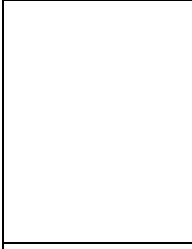
come from various traffic information systems [3], would also enhance the reliability of advice.

REFERENCES

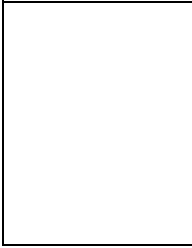
- [1] C. Thorpe, T. Jochem, and D. Pomerleau. The 1997 Automated Highway Free Agent Demonstration. In *Proceedings of 1997 IEEE Conf. on Intelligent Transportation Systems*, pp. 496–501, 1997.
- [2] Özgüner et al. The OSU Demo '97 Vehicle. In *Proceedings of 1997 IEEE Conf. on Intelligent Transportation Systems*, pp. 502–507, 1997.
- [3] Yilin Zhao. *Vehicle Location and Navigation Systems*. Artech House Inc., Boston, 1997.
- [4] J.A. Michon. A Critical View of Driver Behavior Models: What Do We Know, What Should We Know? In *Human Behaviors and Traffic Safety*. Plenum Press, 1985.
- [5] R. Sukthankar, S. Baluja, J. Hancock, D. Pomerleau, and C. Thorpe. Adaptive Intelligent Vehicle Modules for Tactical Driving. In *1996 AAAI Workshop on Intelligent Adaptive Agents*, pp. 13–22, 1996.
- [6] E. Dickmanns and A. Zapp. A Curvature-Based Scheme for Improving Road Vehicle Guidance by Computer Vision. In *Proceedings of the SPIE Conf. on Mobile Robots*, 1986.
- [7] U. Solder and V. Graefe. Visual Detection of Distant Objects. In *Proceedings of the 1993 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 1042–1049, 1993.
- [8] C. Stiller, W. Pöchmüller, and B. Hürtgen. Stereo Vision in Driver Assistance Systems. In *Proceedings of 1997 IEEE Conf. on Intelligent Transportation Systems*, pp. 888–893, 1997.
- [9] R. Bishop. A Survey of Intelligent Vehicle Applications Worldwide. In *Proceedings of 2000 IEEE Intelligent Vehicles Symp.*, pp. 25–30, 2000.
- [10] A. Niehaus and R.F. Stengel. Probability-Based Decision Making for Automated Highway Driving. *IEEE Trans. on Vehicular Technology*, Vol. 43, No. 3, pp. 626–634, 1994.
- [11] J. Forbes, T. Huang, K. Kanazawa, and S. Russell. BATmobile: Towards a Bayesian Automated Taxi. In *Proceedings of the 14th Int. Joint Conf. on Artificial Intelligence*, pp. 1878–1885, 1995.
- [12] E. Gat. Integrating Planning and Reacting in a Heterogeneous Asynchronous Architecture for Controlling Real-World Mobile Robots. In *Proceedings of AAAI-92*, pp. 809–815, 1992.
- [13] B. Pell, D.E. Bernard, S.A. Chien, E. Gat, N. Muscettola, P.P. Nayak, M.D. Wagner, and B.C. Williams. An Autonomous Spacecraft Agent Prototype. In *Proceedings of Agents-97*, 1997.
- [14] E. Yonetani. *Transportation Engineering*. Kokumin-kagaku-sha, 1977. (in Japanese).
- [15] M. Itoh. A Driver Assistance System Based on Visual recognition of Road Scene. Master's thesis, Department of Computer-Controlled Mechanical Systems, Osaka University, March 1999. (in Japanese).
- [16] Q.-T. Luong, J. Weber, D. Koller, and S. Russell. An Integrated Stereo-Based Approach to Automatic Vehicle Guidance. In *Proceedings of the 5th Int. Conf. on Computer Vision*, pp. 52–57, 1995.
- [17] C. Knoeppel, A. Schanz, and B. Michaelis. Robust Vehicle Detection at Large Distance Using Low Resolution Cameras. In *Proceedings of 2000 IEEE Intelligent Vehicles Symp.*, pp. 267–272, 2000.
- [18] S. Hirata, Y. Shirai, and M. Asada. Scene Interpretation Using 3-D Information Extracted from Monocular Color Images. In *Proceedings of the 1992 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 1603–1610, 1992.
- [19] N. Ayache and O.D. Faugeras. Maintaining Representations of the Environment of a Mobile Robot. *IEEE Trans. on Robotics and Automat.*, Vol. RA-5, No. 6, pp. 804–819, 1989.
- [20] H. Kuroda, M. Nakamura, K. Takano, and H. Kondoh. Fully-MMIC 76GHz Radar for ACC. In *Proceedings of the 2000 IEEE Conf. on Intelligent Transportation Systems*, pp. 299–304, 2000.
- [21] E. Ichihara, H. Takao, and Y. Ohta. Visual Assistance for Drivers Using Roadside Cameras. In *Proceedings of the 1999 IEEE Int. Conf. on Intelligent Transportation Systems*, pp. 170–175, 1999.



Jun Miura received the B.Eng. degree in mechanical engineering in 1984, the M.Eng. and the Dr.Eng. degrees in information engineering in 1986 and 1989 respectively, all from the University of Tokyo, Tokyo, Japan. In 1989, he joined Department of Computer-Controlled Mechanical Systems, Osaka University, Suita, Japan, where he is currently an Associate Professor. From March 1994 to February 1995, he was a Visiting Scientist at Computer Science Department, Carnegie Mellon University, Pittsburgh, PA. He received the Best Paper Award from the Robotics Society of Japan in 1997. His research interests include robotics, artificial intelligence, and computer vision.



Motokuni Itoh received the B.Eng. and the M.Eng. degrees in Computer-Controlled Mechanical Systems in 1997 and 1999, respectively, from Osaka University, Suita, Japan. He is currently a PhD student of the Graduate School of Science and Technology, Kobe University, Kobe, Japan.



Yoshiaki Shirai received the B.E. degree from Nagoya University in 1964, and received the M.E. and Dr.Eng. degrees from the University of Tokyo in 1966 and 1969, respectively. In 1969, he joined the Electrotechnical Laboratory. He spent 1971-72 at the MIT AI Lab as a Visiting Scholar. Since 1988, he has been a Professor of Department of Computer-Controlled Mechanical Systems, Osaka University. His research interest include computer vision, robotics, and artificial intelligence.