CAIR-2

Intelligent Mobile Robot for Guidance and Delivery

Hyun S. Yang, Jiyoon Chung, Byeong S. Ryu, and Juho Lee

■ CAIR-2 from the Korea Advanced Institute of Science and Technology (KAIST) placed first in the Office Delivery event at the 1995 Robot Competition and Exhibition, held in conjunction with the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95). CAIR-2 is a totally self-contained and autonomous mobile robot, and its control architecture incorporates both behavior-based and planner-based approaches. In this article, we present a short description of CAIR-2's hardware, system and control architecture, real-time vision, and speech recognizer.

Now it is our turn in the final round! The judges are ready. A lot of people are watching. Can CAIR-2 (figure 1) accomplish its office-delivery task as well as it did in the two preliminary rounds?

The environment is similar in size, shape, and furnishings to the first floor of an office building, containing a foyer as well as offices. CAIR-2 starts in a specific room with only one exit and is given instructions verbally by speech recognition on how to get to the goal room. Some instructions might be faulty, and it must be able to overcome. No topological map of the office environment is given.

CAIR-2 starts to wander around the room looking for an exit marked with a landmark pattern. It recognizes the target pattern in the landmark using vision and keeps tracking the target in real time while approaching the exit to adjust its position and orientation. After confirming the position and orientation several times, CAIR-2 starts to exit the room. It exits the room safely; it says "exit room" to confirm that it has completed its first subtask.

Now it moves toward its goal room. It knows it must make a right turn at the first junction to take the hallway. It finds two doors using sonar and confirms by saying "door found" until it reaches the next junction for another right turn. It knows it is on the correct path to the goal room.

Measuring the distance from its body to the left- and right-side walls using ultrasound, it keeps its path near the center of the hallway. It finds a foyer and again confirms it verbally.

CAIR-2 now tries to find the third right open where it is supposed to make a right turn, as given in the initial instructions. However, it cannot find a third open! There is a little confusion.

CAIR-2 activates a behavior responsible for finding any mistake. It fails to find a mistake and an alternative path to the goal and concludes that the given instruction was incorrect. It announces it verbally, ending the first phase of the task.

The second phase of the task is rather straightforward and similar to the first, except there is no incorrect instruction. CAIR-2 finally approaches the goal room and finds the entrance. It enters the room carefully, keeping its body at the center of the entrance. There was not a single mistake. CAIR-2 completed its mission better than we expected. The crowd cheers, and we expect a high score.

In summer 1990, we started a project, sponsored by the Center for Artificial Intelligence Research (CAIR) at the Korea Advanced Institute of Science and Technology, to develop an intelligent mobile robot. Because most mobile robot platforms available at that time were costly and still did not satisfy our requirement, we had to design and build an entire mobile robot system all by ourselves. In winter 1991, we developed a prototype intelligent mobile robot named CAIR-1 that had almost all the functions that CAIR-2 has now. In summer

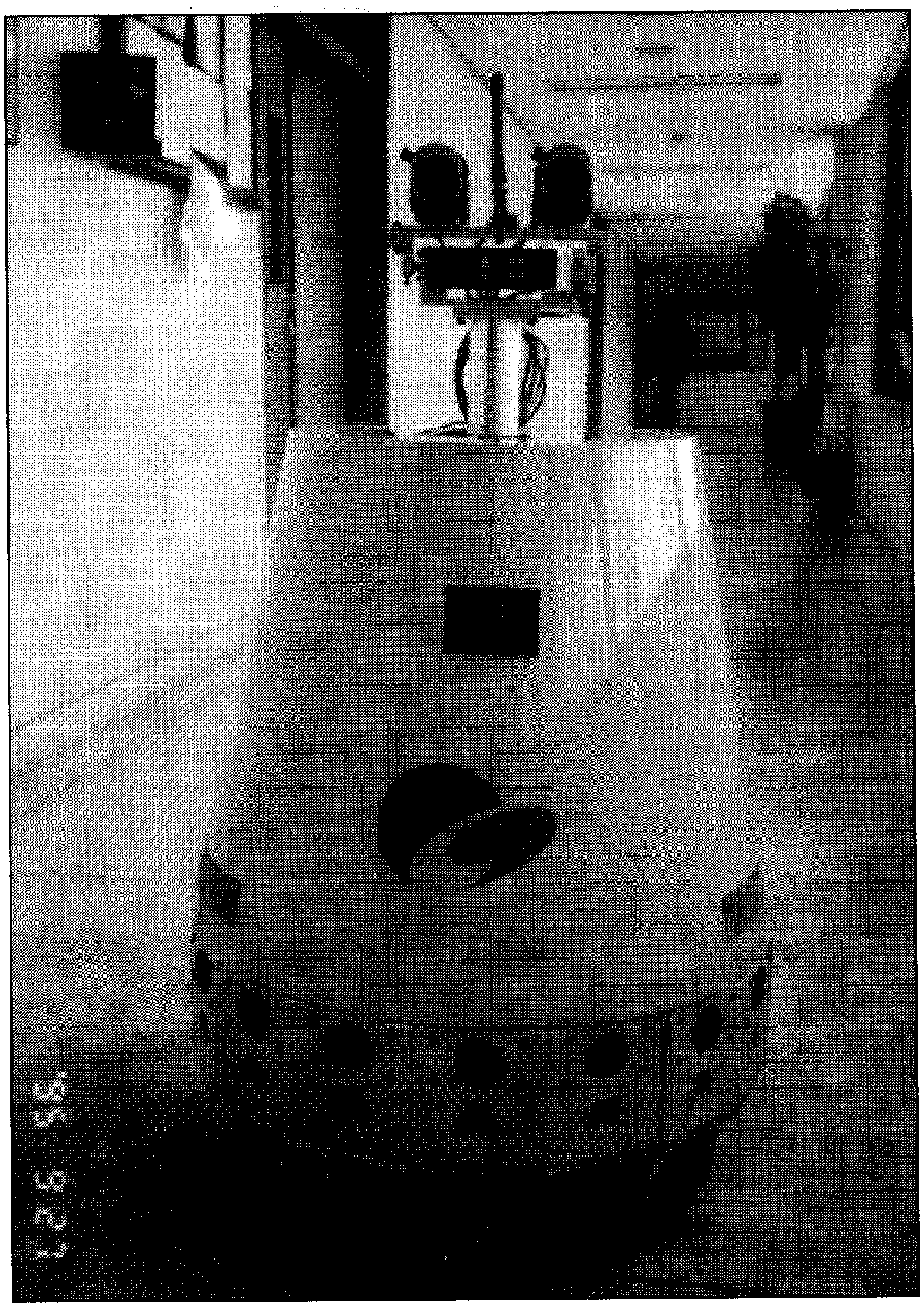


Figure 1. CAIR-2.

Only the gods knew what would happen to us carrying a 65-kilogram robot across the Pacific!

1992, CAIR-2 was completed; we redesigned CAIR-1 mechanically to provide more flexibility during navigation and improved such functions as ultrasound ranging, real-time visual tracking, infrared proximity sensing, and control architecture for navigation.

During the 1993 World Exposition held in Taejon, Korea, we were invited by the expo organizing committee to exhibit CAIR-2 for three months outdoors, with real-time demonstrations for the public. To accomplish such a challenging exhibition successfully, we had to make CAIR-2 robust in both hardware and software. We added some hardware for ventilation, safety, and wireless communication and built two mobile robots whose shapes resembled the 1993 expo's mascots, Dream Boy and Dream Girl. We developed software to make these two robots autonomously interact with each other and play with the public in real time. We could run these robots successfully for three months outdoors in the middle of a large crowd. With this event, we could have a lot of real-world experience and make the robot robust and practically useful for both indoor and outdoor tasks.

We were then ready to introduce CAIR-2 to the world. The 1995 Robot Competition and Exhibition, held in conjunction with the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95), seemed the best opportunity, although we realized we had to make that extra effort to carry the robot from Korea to Montréal, Québec, Canada. Only the gods knew what would happen to us carrying a 65-kilogram robot across the Pacific! Thanks to the gods, CAIR-2 made it to the competition site without any damage and won the competition.

In what follows, we provide a short description of CAIR-2's hardware, system architecture, control architecture, real-time kernel, real-time vision, and speech recognizer.

Hardware and System

CAIR-2 is approximately 100 centimeters tall and 60 centimeters (diameter) wide. It weighs about 65 kilograms. Most of the mechanical parts are made with aluminum alloy to reduce the weight. Pneumatic tires and two kinds of ring spring are used to absorb shocks from the uneven floor or road.

The brains of CAIR-2 are two on-board MC 68040 single-board computers that control other VME boards, such as motor controllers, frame grabber, ultrasound range finder, infrared sensor controller, voice synthesizer, and random-access memory disk, through the VME bus (A32/D32) (figure 2). CAIR-2 is equipped with 16 sonar sensors for measuring the range between 30 and 300 centimeters and 8 infrared sensors for proximity sensing within 30 centimeters. Two video cameras with a pan-and-tilt mechanism attached on the head are exploited for object recognition, scene understanding, target tracking, and stereo vision.

Control Architecture

For CAIR-2 to behave adaptively to the dynamically varying environment, we developed a new control strategy, as shown in figure 3, that combines the merits of both behavior-based and planner-based approaches (Simmons 1994; Lyons 1993; Saffiotti 1993; Mataric 1992; Payton, Rosenblatt, and Keirsey 1990; Kaelbling 1986). Sensory data queues, behaviors, and the blender are primarily exploited to control the robot reactively, but the other units are used to plan and accomplish complex physical actions and tasks. In this article,

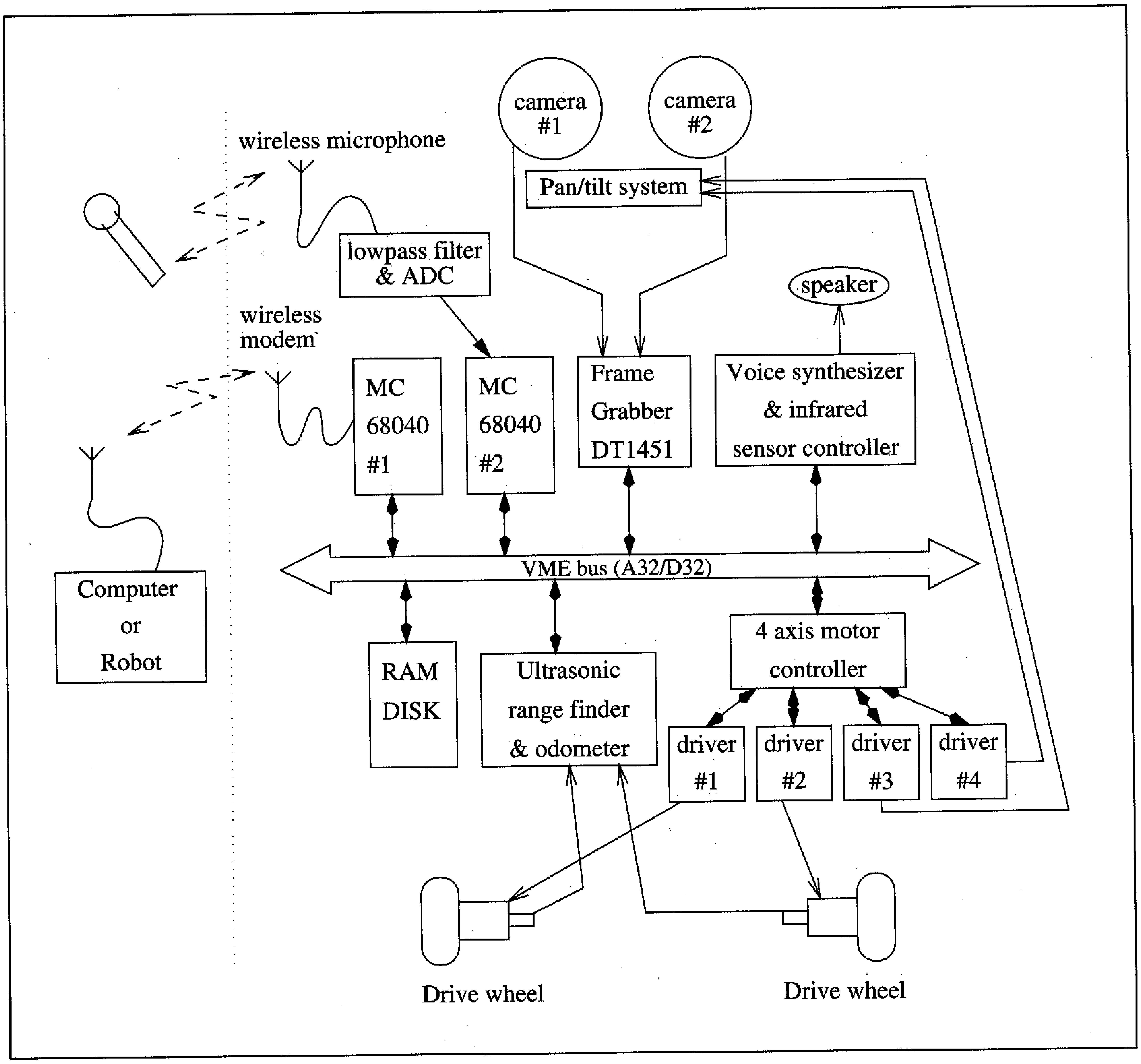


Figure 2. System Architecture of CAIR-2.

we briefly describe the functions of units exploited for the competition.

Behaviors

Each behavior is basically independent and has four links to its related units (figure 4). The *coordinator* activates one or more behaviors required to complete a given task by sending wake-up signals. Each behavior then collects and processes sensory data and outputs the direction and the velocity of the robot. For the competition, we used 17 behaviors, for example:

BH_AVOID_COLL: Avoid the collision by keeping the robot a safe distance from the obstacle using sonar and infrared sensors.

BH_OPEN_SPACE_EXPLORER: Move the robot to the open space, and let it wander around.

BH_GO_FORWARD: Measure the distance from the robot body to the left- and right-side walls using sonar, and try to keep its path along the center of the hallway.

BH_PASS_FOYER: Find the foyer, and pass safely.

BH_FIND_OPEN: Find doors by sonar sensors.

BH_HALLWAY_END: Find the end of the hallway.

BH_TRAP: Check the validity of the current instruction, and see if there is any mistake when the robot is in the trap. If any mistake is detected, try to solve. Otherwise, deactivate all currently active behaviors, and announce the situation verbally.

Coordinator

The coordinator incorporates a set of behaviors, whenever necessary, to accomplish complex tasks that cannot be completed by a single behavior, according to the guidelines set by the planner and the fuzzy state estimator.

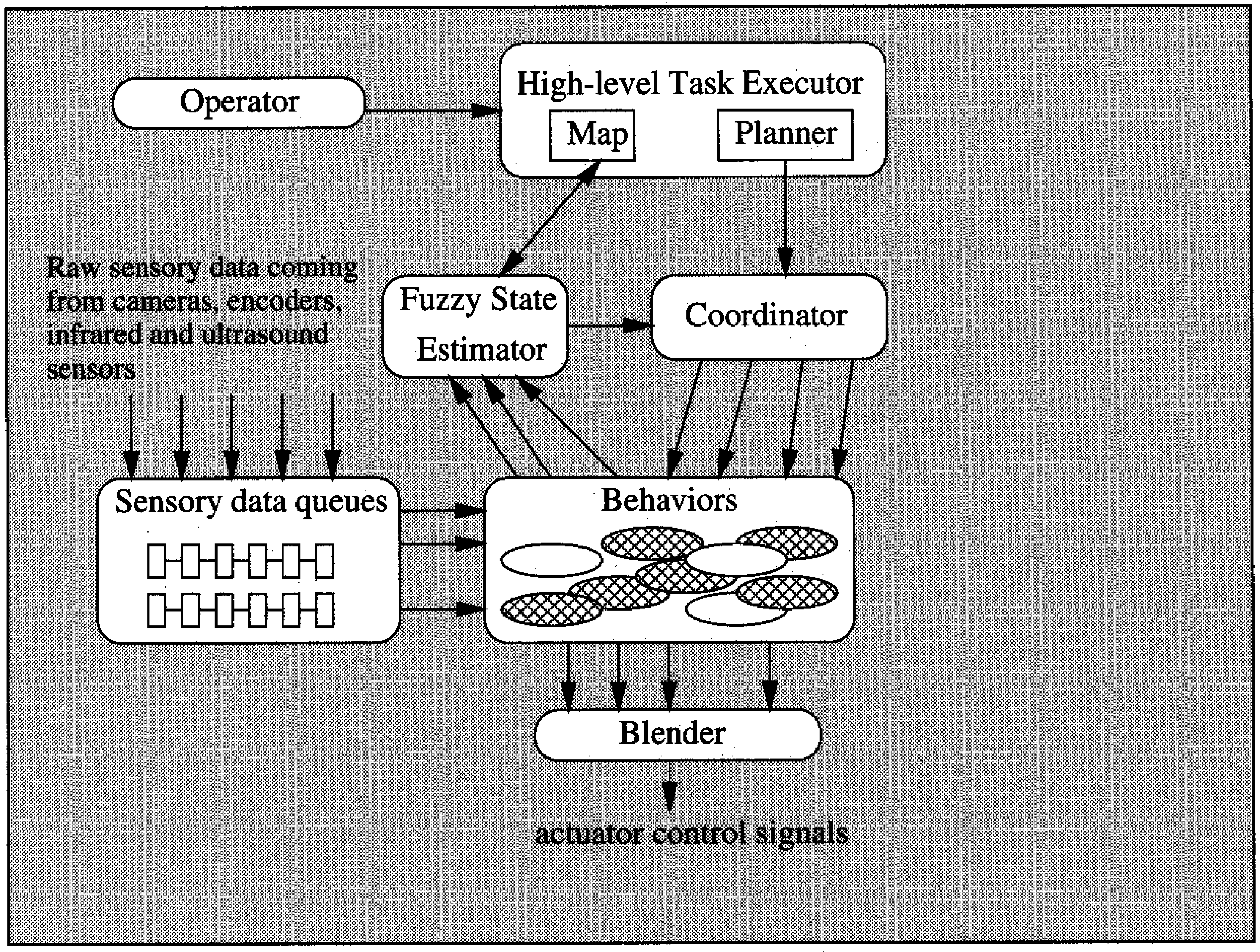


Figure 3. Control Architecture of CAIR*-2.*

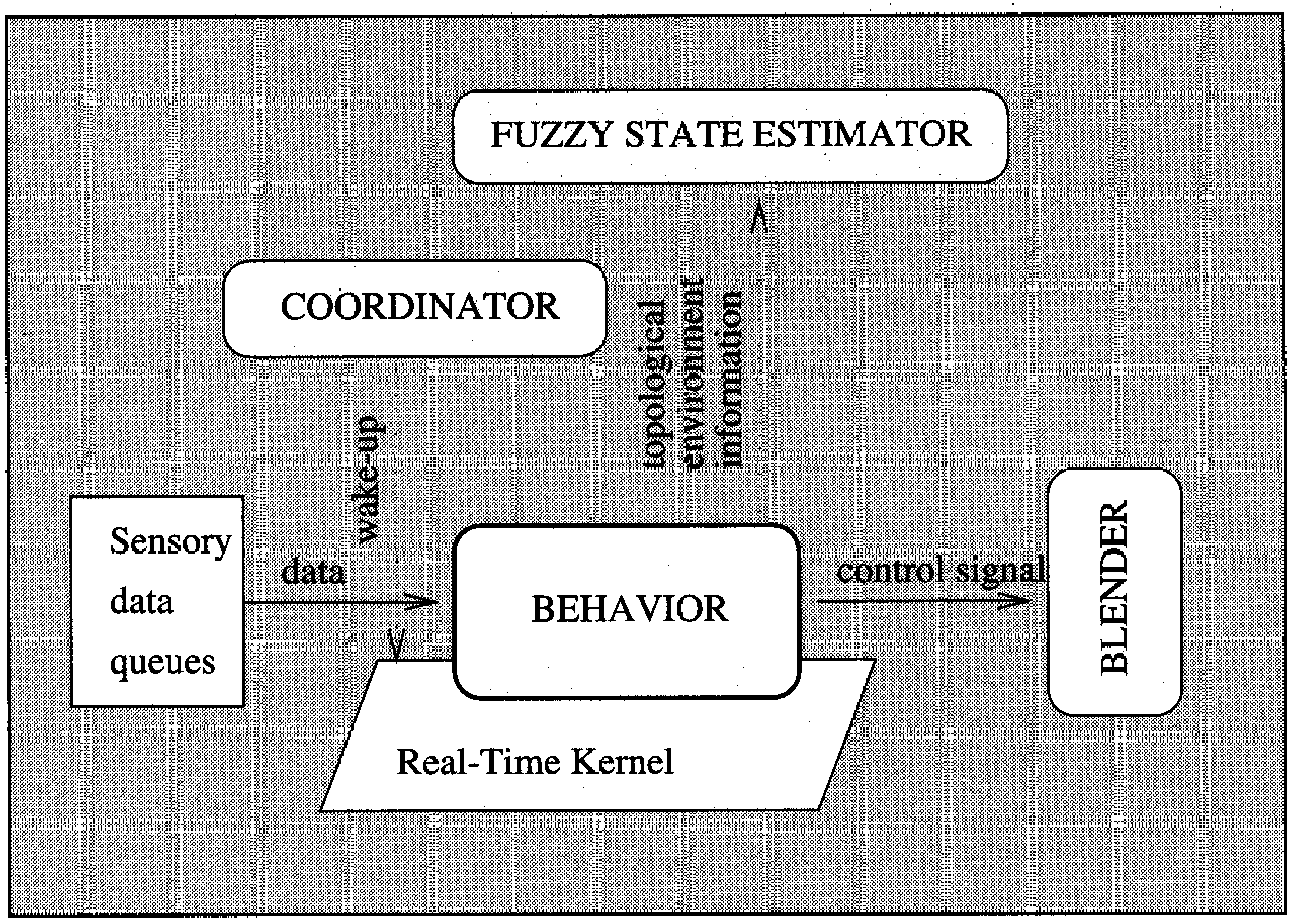


Figure 4. Diagram Showing How Behavior Interacts with Other Units.

Blender

The *blender* resolves conflict made by multiple control signals coming from activated behaviors. We currently use two kinds of conflict-resolution method: One is based on the vector-summation method, and the other is based on the winner-take-all strategy. In most cases, the first method is used to merge the multiple output into one representing vector. However, in case the output from the behavior that must not be merged with others because of its inherent characteristic and must have priority over the others, the winner-take-all strategy is exploited. For example, the output from the behavior used for collision avoidance must have priority over the others so a robot does not move to the prohibited area decided by the behavior.

An Example: Exit Room

To help understand how behaviors actually incorporate to complete a specific task for the competition, we describe Exit Room. As shown in figure 5, five behaviors incorporate to achieve the common goal, Exit Room. The solid lines denote the activation sequence of the behaviors.

Search Exit: When Exit Room is initiated or whenever the robot fails to find an exit, this behavior stops the robot at its current position and causes it to scan 360 degrees around its main axis searching for the exit. While the robot turns its head or body to scan, a vision module takes and analyzes a sequence of images every 1/30 second until it finds a landmark (figure 6), which is used to detect an exit. Once the landmark is found, Track Exit is triggered. Otherwise, Open-Space Explorer is triggered, allowing the robot to move out to the open area for a better view.

Track Exit: Once the robot recognizes the landmark, it starts to track the landmark continuously. This behavior helps the robot to get a more precise and reliable estimation of the position and the orientation of the landmark. CAIR-2 can track its landmark in real time using what we call the *two-stage visual tracking method* (TSVTM). While the robot tracks the landmark, it continues to estimate the position and the orientation of the exit using the geometry between the landmark and the exit. After the robot arrives at the front of the exit, control passes to Confirm Exit.

Open-Space Explorer: Whenever the robot fails to find the landmark, this behavior moves the robot toward the open space for a better view. After the robot travels a certain distance from the current position, this behavior is turned off.

Confirm Exit: This behavior confirms (several times) the position and the orientation of the exit with respect to the robot when the robot reaches the front of the exit to get more precision for passing the exit. If the position and the orientation are consistent over several measurements, this behavior assumes that the current information is reliable, and the robot is ready to exit the room.

Exit Room: This behavior actually conducts the action Exit Room and verbally

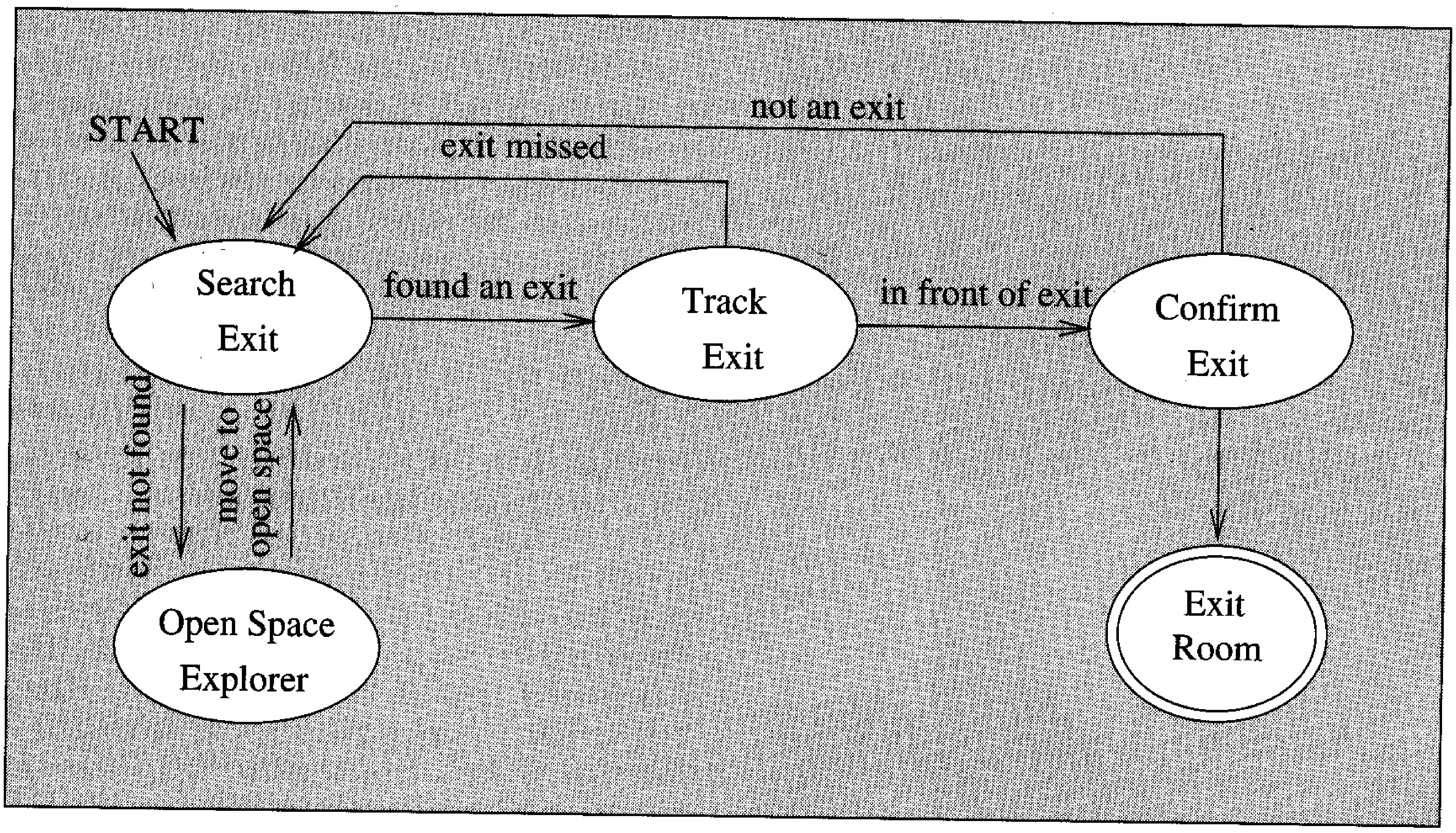


Figure 5. Diagram Showing How Behaviors Incorporate to Achieve Exit Room.

announces its mission is completed after the robot successfully exits the room.

Real-Time Kernel

A *real-time kernel* based on Earliest Deadline First was developed to manage behaviors efficiently. Whenever a processor is ready to be used, the kernel assigns the processor to execute the task with the highest priority and the closest deadline. To accomplish this task, the kernel divides the tasks into three priority classes: (1) hard, (2) soft, and (3) NRT (nonreal time). A *hard task* has the highest priority and is either a periodic or a sporadic process with a critical deadline. A *soft task* has fewer strict time constraints and is either periodic or sporadic. An NRT task has the lowest priority and no time constraint at all.

Real-Time Vision

The vision system of CAIR-2 is based on the TSVTM, which is an efficient tracking method capable of tracking targets in real time with general-purpose hardware (Chung 1995). The TSVTM consists of a real-time kernel, an image saver, a database, and a vision module.

The *vision module* is divided into two modules, according to the information available: (1) the first-stage vision module (FSVM) and (2) the second-stage vision module (SSVM). The FSVM has to process the whole image to recognize targets; so, it needs a lot of computation time. However, the SSVM can easily find and track targets by focusing on interesting parts of an image that have already been obtained from the previous execution of either the FSVM or the SSVM. Because the SSVM knows the approximate location and useful features of the targets, it can achieve a fast response time. Therefore, the overall computational requirement of the TSVTM becomes less strict. Because the FSVM needs more computation time than given, the incoming images during the execution will be lost. Thus, we designed the image saver to take responsibility for keeping all the incoming images until they can be processed. The database keeps both the estimated and the predicted location, velocity, intensity, and so on, of each region that makes up the target. Using the information in the database, the SSVM verifies its result by using the predefined constraints. Whenever a fault is discovered, it sends a signal to the real-time kernel to invoke the FSVM.

The tasks, including the TSVTM, are scheduled with their own time constraints. The image saver has a very strict time constraint because it has to put an incoming image from the camera into the queue every one-thirtieth of a second. The FSVM has no time constraint at all. However, it should be as fast as possible because it affects the response time of the TSVTM. The SSVM has a time constraint, but violation of the deadline does not make the overall system fail because the image saver will save the incoming images.

Figure 7 shows an overview of the TSVTM. There are three computing modules. Each of them has its own computational require-

The vision system of CAIR-2 is based on the TSVTM, which is an efficient tracking method capable of tracking targets in real time with general-purpose hardware.

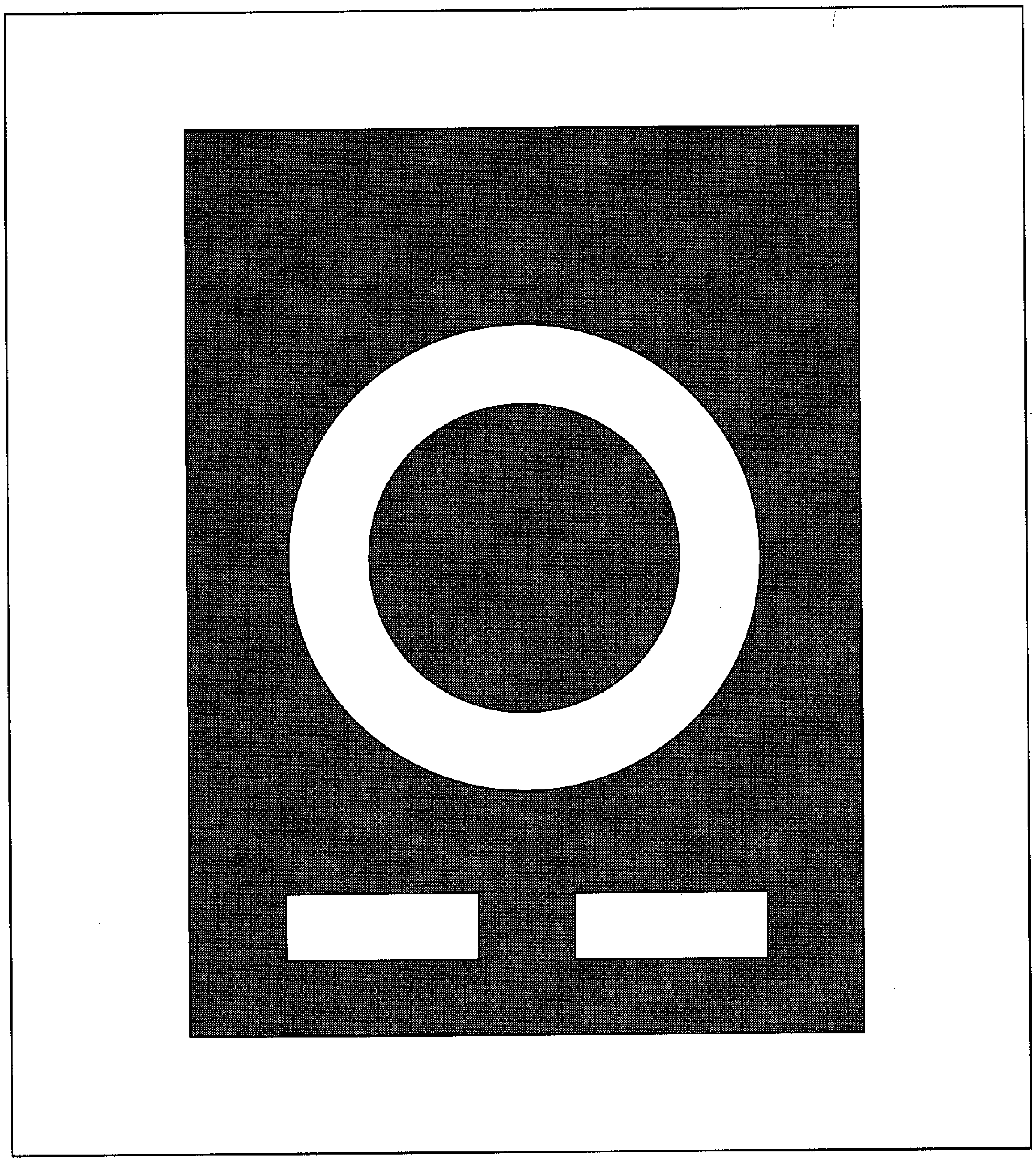


Figure 6. Landmark Used to Locate an Exit.

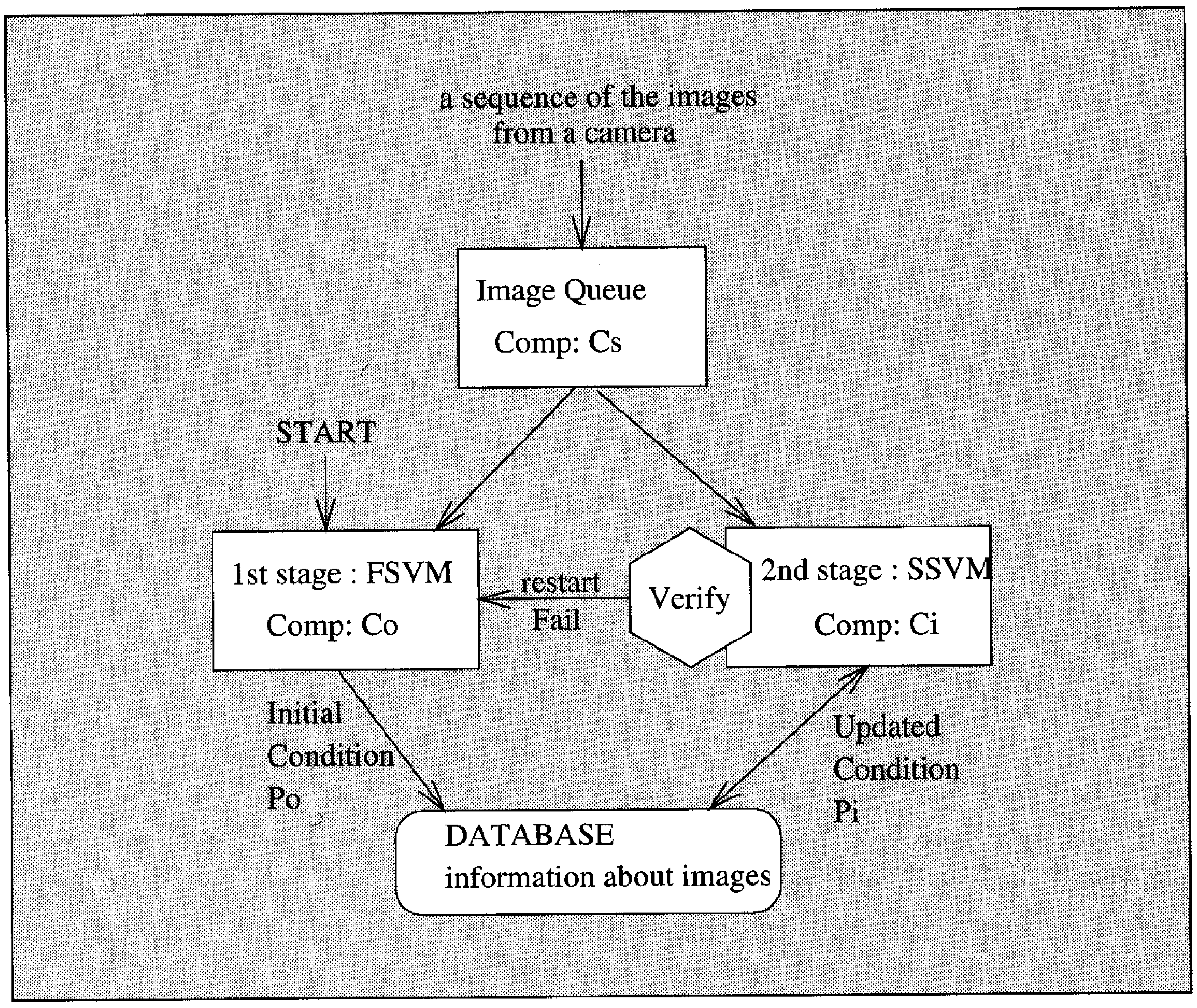


Figure 7. Two-Stage Visual Tracking Method.

ments: C_0, C_i, and C_s denote the computation time of the FSVM, the SSVM, and the image saver, respectively. C denotes the computation time equivalent to the time that it takes to grab an image, one-thirtieth of a second.

There are two constraints in the TSVTM:

$$C_i + C_s < C_f \tag{1}$$

$$\left[\frac{C_0 - (C_f - C_s)}{C_f - C_s} \right] < \frac{S_m}{I_s} \tag{2}$$

Equation 1 shows the constraint on the computation time of the SSVM and the image saver. Their sum must be less than C_f, or the length of the queue in the image saver will grow boundlessly. Equation 2 shows the constraint on memory that is necessary for the image saver to store the images in the queue. The denominator on the left side of equation 2 is the given computation time for the FSVM, and the numerator is the excess of the computation time over the given computation time. Thus, the image saver must have the same number of image buffers as the left side of the equation. We assume that the size of an image buffer is I_s, so the minimum memory space of the TSVTM should be larger than S_m.

$$\frac{C_0 - (C_f - C_s)}{C_f - (C_s + C_i)} \tag{3}$$

The TSVTM will have the fastest response time only after the SSVM has been invoked as many times as defined in equation 3 since the last invocation of the FSVM. Therefore, the TSVTM is well suited for applications that need to track targets in real time and endure a slightly longer initial response time.

Speech Recognizer

As a tool for human-robot interaction, we implemented a speech recognizer and a synthesizer. Using these, CAIR-2 understands the verbal instructions given by the operator and announces its current status whenever each part of the instructions is completed, or it detects a trouble. For the competition, we developed a simple but robust speech recognizer that can recognize 50 words spoken by microphone in the middle of a crowd with over 99-percent accuracy. (Note that all the instructions used for the competition could be made using a combination of these 50 words.)

Speech recognition was made as follows: The analog speech signal coming from the receiver of a wireless microphone is amplified and low-pass filtered. The on-board analog-to-digital converter then converts it to 12-bit 8-kilohertz digital data. With the forward-averaging technique, a threshold between the speech and the silence regions is automatically determined. A speech waveform is then converted to a parametric representation. We adopted a short-time spectral analysis with a bank-of-filters model, one of the most widely used parametric representation methods. The similarity between the input pattern and each of the reference patterns is then computed using the likelihood distortion measure for recognition.

Conclusions

In this article, we presented a brief description of CAIR-2, with an emphasis on its control architecture and real-time visual tracking. The competition was a good opportunity for us to compare our technology with others and to test the robustness and the flexibility of our robot.

Acknowledgments

We would like to thank Jung Wan Cho, chief director of the Center for Artificial Intelligence Research at the Korea Advanced Institute of Science and Technology, for his support of our mobile robot project since 1990. Thanks are also owed to the organizing committee of the 1995 IJCAI-AAAI Mobile Robot Competition, especially David P. Miller, David Kortenkamp, and David Hinkle for providing a grant for part of our travel expense to Montreal and their helpful comments and suggestions.

References

Kaelbling, K. 1986. An Architecture for Intelligent Reactive Systems, SRI Technical Note 400, SRI International, Menlo Park, California.

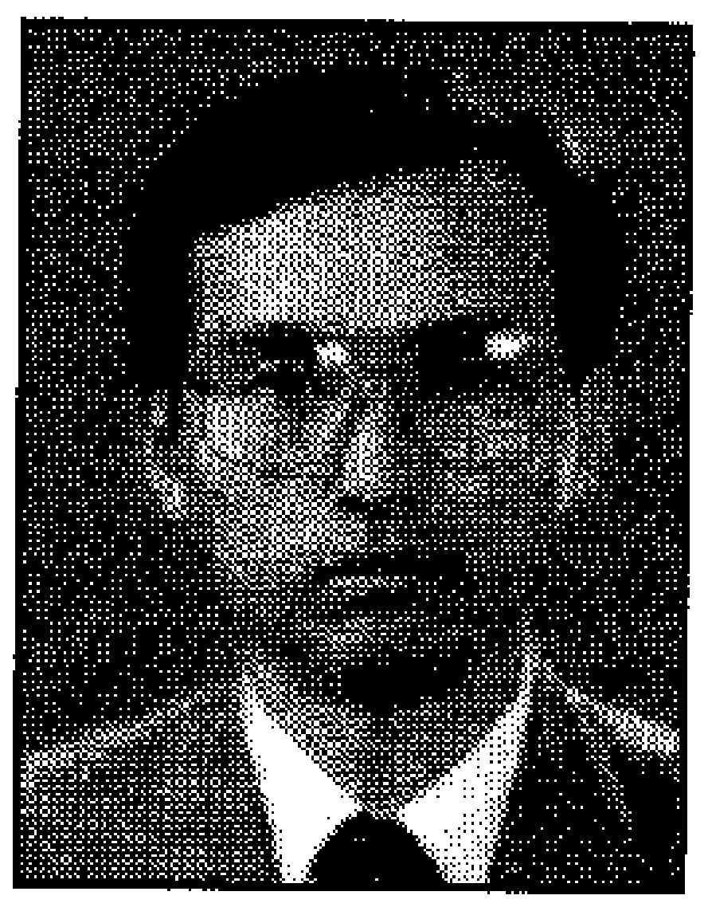
Lyons, D. M. 1993. Representing and Analyzing Action Plans as Networks of Concurrent Processes. *IEEE Transactions on Robotics and Automation* 9(3): 241–256.

Mataric, M. J. 1992. Integration of Representation into Goal-Driven Behavior-Based Robots. *IEEE Transactions on Robotics and Automation* 8(3): 304–312.

Payton, D. W.; Rosenblatt, J. K.; and Keirsey, D. M. 1990. Plan-Guided Reaction. *IEEE Transactions on Systems, Man, and Cybernetics* 20(6): 1370–1382.

Saffiotti, A. 1993. Some Notes on the Integration of Planning and Reactivity in Autonomous Mobile Robots. In *Foundations of Automatic Planning: The Classical Approach and Beyond, Papers from the 1993 Spring Symposium,* 122–126. Menlo Park, Calif.: AAAI Press.

Simmons, R. G. 1994. Structured Control for Autonomous Robots. *IEEE Transactions on Robotics and Automation* 10(1): 34–43.



Hyun S. Yang received a B.S.E.E. from the Department of Electronics Engineering, Seoul National University, Korea, in 1976, and an M.S.E.E and a Ph.D from the School of Electrical Engineering at Purdue University in 1983 and 1986, respectively. He was with the Department of Electrical and Computer Engineering at the University of Iowa from 1986 to 1988 as an assistant professor. He is currently with the Department of Computer Science at the Korea Advanced Institute of Science and Technology (KAIST) as an associate professor and a director of the Computer Vision and Intelligent Robotics Lab in the Center for Artificial Intelligence Research. His current research interests include computer vision, robotics, AI, and multimedia.

Jiyoon Chung received a bachelor's degree from the Department of Computer Science at Chungang University in Seoul, Korea. in 1990 and a master's degree from the Department of Computer Science at the Korea Advanced Institute of Science and Technology (KAIST) in 1992. He is currently a Ph.D. student in the Department of Computer Science at KAIST. His research interests include robot vision, AI, and mobile robotics.

Byeong-Soon Ryu received a B.S. and an M.S. from the Department of Computer Science at the Korea Advanced Institute of Science and Technology (KAIST) in 1991 and 1993, respectively. He is currently a Ph.D. student in the Department of Computer Science at KAIST. His research interests include AI, computer vision, and mobile robotics.

Ju Ho Lee received a B.S. from the Department of Computer Science at Chungang University in Seoul, Korea, in 1993 and an M.S. from the Department of Computer Science at the Korea Advanced Institute of Science and Technology (KAIST) in 1995. He is currently a Ph.D student in the Department of Computer Science at KAIST. His research interests include robot vision, AI, and mobile robotics.