Jijo-2: An Office Robot That Communicates and Learns

Hideki Asoh, Yoichi Motomura, Futoshi Asano, Isao Hara, Satoru Hayamizu, Katsunobu Itou, Takio Kurita, and Toshihiro Matsui, *National Institute of Advanced Industrial Science and Technology (AIST)*

Nikos Vlassis, Roland Bunschoten, and Ben Kröse, *RWCP Autonomous Learning Functions SNN, University of Amsterdam*

The authors combine speech recognition, dialogue management, and statistical learning procedures to develop Jijo-2, an office robot that can communicate with humans and learn about its environment.

Nowadays, robots are moving off of factory production lines and into our everyday lives.[1] Unlike stationary and pre-engineered factory buildings, an everyday environment, such as an office, museum, hospital, or home, is an open and dynamic place where robots and humans can coexist and cooperate. Hence, instead of the capabilities of precise motion, dexterous manipulation, and so forth—capabilities a factory robot requires—the increasingly popular interactive robot must be able to learn from and adapt to its dynamic environment and communicate with people.

We have built an office robot, Jijo-2, as a testbed for autonomous intelligent systems that interact and learn in the real world (see Figure 1). Jijo-2's most notable properties are its communication and learning skills: it can communicate with humans through a sophisticated Japanese spoken-dialogue system, and it navigates by using models that it learns by itself or through human supervision. It achieves the former through a combination of a microphone array, speech recognition module, and dialogue management module.[2] It achieves the latter through statistical learning procedures by which the robot learns landmarks or features of its environment that let it construct useful models (or maps) for navigation.

Unlike a factory robot, Jijo-2's tasks are not clearly defined. Our intention was not to design special-purpose hardware and software tuned for a particular task but to build a mobile agent that, being physically embodied in the human world, would exhibit some generic aspects of intelligence. In particular, we have emphasized in Jijo-2 the role of (semi) autonomous learning: By operating in an office environment, the robot is expected to learn how to perform services such as guiding visitors, delivering messages, managing office members' schedules, arranging meetings, and other similar tasks. Currently, Jijo-2 can demonstrate most of these enthralling tasks.

A robot–human dialogue

Figure 2 illustrates an example of a dialogue between human users (U1, U2) and Jijo-2 (R) involving several different behaviors. The robot's behaviors included in this example are

- turning to the sound source (U1),
- detecting and recognizing the user's face,
- referencing the database of the office member's current location,
- calling a member by sending email,
- guiding the user to a member's office, and
- registering new location information in the location database.

This example also illustrates how the omitted salient information in the users' utterances is reconstructed. The words in square brackets in the English translation of the Japanese utterance are omitted in Japanese. For example, in U1's fifth utterance, only the name of a person (Hara-san) is mentioned. Because the utterance is also a question, the system creates a database query frame and fills a slot using

1094-7167/01/$10.00 © 2001 IEEE

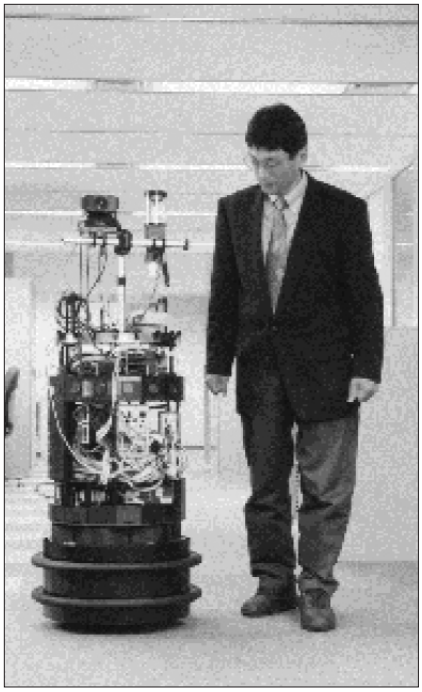


Figure 1. Our speaking robot, Jijo-2.

U1: Kon-nichiwa. (Hello.)
The robot turns to the speaker, detects his face, and recognizes him.
R: Kon-nichiwa, Matsui-san. (Hello, Dr. Matsui.)
U1: Asoh-san wa doko? (Where is Mr. Asoh?)
R: Honkan ni imasu. ([He is] in the main building.)
U1: Yonde kudasai. (Please call [him].)
R: Asoh-san ni me-ru wo okurimashouka? (Shall I send an email to Mr. Asoh?)
U1: Hai. (Yes, please.)
The robot sends an email.
U1: Hara-san wa? ([Where is] Dr. Hara?)
R: Office ni imasu. ([He is] in [his] office.)
U1: Tsureteitte kudasai. (Please take me [to his office].)
The robot plans a path to the office and executes the navigation plan.
During the navigation, it becomes uncertain about its location.
R: Koko wa doko desuka? (Where am I?)
U1: Asano-san no tokoro desu. ([You are] in front of Dr. Asano's office.)
R: Arigatou. (Thanks.)
The robot confirms the current position, replans the path to its destination, and resumes navigation.
R: Hara-san no heya ni tsukimashita. ([We are] at Dr. Hara's office.)
U2: Shokuji ni yukimasu. ([I'm] going out for lunch.)
R: Schedule wo touroku shimasu. ([I'll] register [the data] to the schedule database.)
The robot sends SQL commands to the schedule database.
U1: Matsui mo. (Matsui, too.)
R: Schedule wo touroku shimasu. ([I'll] register [the data] to the schedule database.)
The robot sends SQL commands to the schedule database.
U1: Arigatou. (Thanks.)
R: Douitashimashite. (You are welcome.)

Figure 2. A typical dialogue between Jijo-2 and a human. (U1 and U2 are the human users, Toshihiro Matsui and Isao Hara, respectively; R is the robot.)

the information "doko" (where) in the attentional state. Such an omission of salient information is often encountered in natural languages, especially in Japanese; therefore, the capability of estimating the omitted information is essential for building a realistic dialogue system.

In the middle of the dialogue, during its navigation to Hara's office, the robot becomes uncertain about its position. This can occur because of noise in sensor signals, occlusion, and so forth. The navigation module recognizes the problem and sends the request "ask location" to the dialogue manager. The robot actively starts a new conversation to confirm its location by asking, "Koko wa doko desuka?" (Where am I?)

The Jijo-2 architecture

Jijo-2 is based on Nomadic's Nomad 200 mobile-robot platform. It is equipped with various sensors such as ultrasonic range sensors, infrared proximity sensors, tactile sensors, and an odometric sensor. Figure 3 shows a schematic diagram of its architecture. The onboard computer is a PC running Linux and is connected to a LAN through radio Ethernet. We added a microphone array, a CCD (charge-coupled device) color camera, an omnidirectional camera, digital signal processors (DSPs) for sound signal processing, and a Japanese speech synthesizer.

On the robot and the remote host, we implemented several software modules for navigation and dialogue. Figure 4 depicts the system's overall structure, including the reactor and the integrator modules. We implemented integrator modules in EusLisp, an object-oriented Lisp for robot control, and the reactor modules in C for the sake of real-time control. The modules are managed in an event-driven architecture and realize both reactive and deliberative behaviors.[3] Communication between modules occurs over TCP/IP connections. The major advantages of this event-driven multiagent architecture are the implementation of concurrent behaviors and the plug-and-play aspects of the software modules.

The dialogue system

Jijo-2's spoken dialogue system is composed of several parts. The first is a sound source localization and signal separation module. It can reduce the noise level and improve speech recognition—even in an office environment. The next part is a speaker-independent continuous Japanese speech recognition module, which decodes an input speech signal to its semantic contents. The third part interprets utterances using a task frame and any system knowledge. The final part manages the dialogue process and the robot's behavior using a set of predefined reply templates with a context-based slot-filling method.

How Jijo-2 listens to the human voice

For Jijo-2 to carry out a smooth dialogue in real environments, dynamic noise suppression is needed to maintain good speech recognition performance. We applied a microphone array composed of eight omnidirectional microphones around the robot's top tray. The sound from a speaker arrives at each microphone with a different delay. The sound signal at each microphone is digitized and fed to the first DSP (TI-C44). Based on the delay-and-sum beam-forming method,[2] the direction to the sound source is computed and then used to form a beam to pick up the speech and reduce ambient noise. We observed a noise reduction of approximately 10 decibels in the frequency region over 1,000 Hz, which is crucial in recognizing consonants. As Figure 5 shows, our multimicrophone system usually performs better than a single-microphone system, even in noisy environments.

The noise-free digital sound data is sent to the second DSP through the direct–memory–access-driven communication port. The second DSP does the frequency analysis and emits the vector quantization code (VQ code) for each phonetic element every 10 ms. The VQ codes are sent to the onboard PC through a serial link.

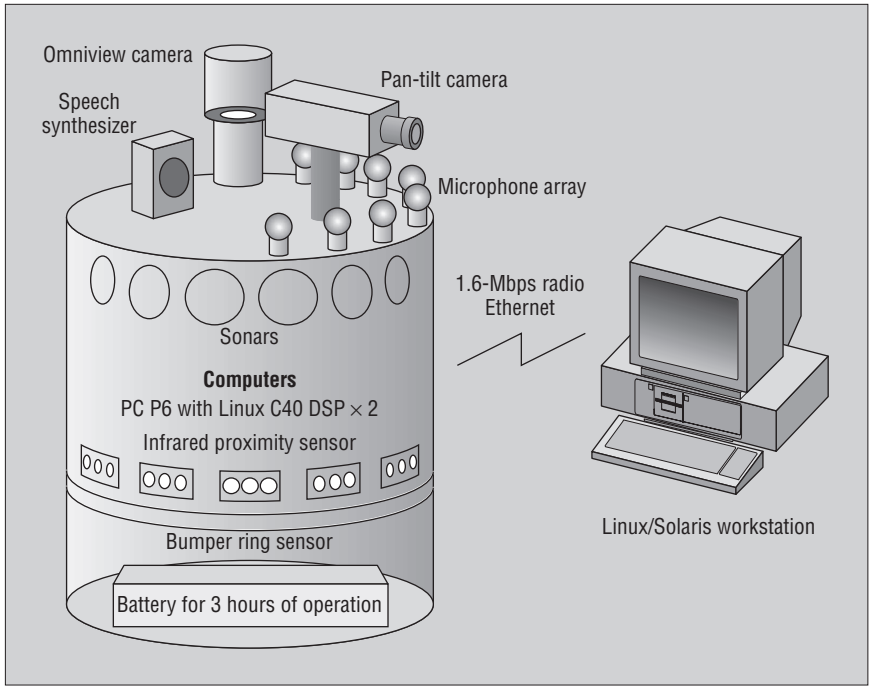


Figure 3. A diagram of Jijo-2's architecture.

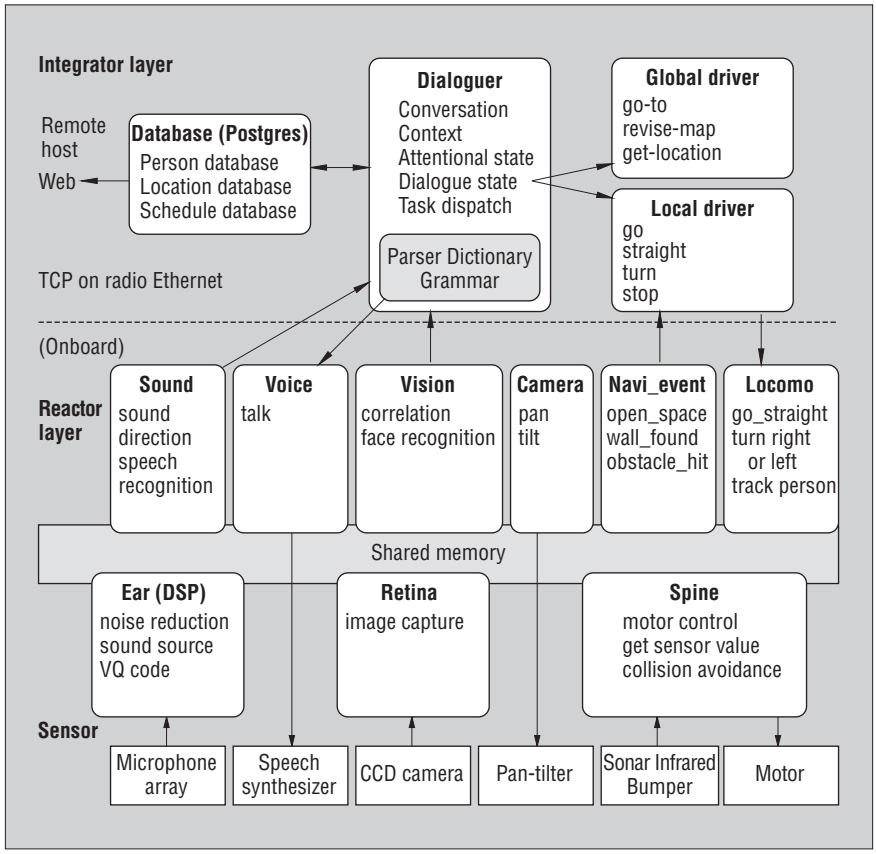


Figure 4. A diagram of Jijo-2's system modules.

How Jijo-2 recognizes speech

The speaker-independent continuous Japanese speech recognizer, Ninja, is a hidden Markov model-based system developed in the Electrotechnical Laboratory, which evolved into AIST. Using phoneme models, a word dictionary, and a grammar created beforehand, Ninja searches for a series of word symbols that is consistent with the grammar. Thus, the speech recognition module produces a list of symbols such as *hello*, *right to turn*, *straight to go*, and *here is Matsui's office* together with a recognition confidence value and a direction angle to the sound source.

We have preliminarily evaluated the speech recognizer's recognition rate using a prepared data set, which is a recording of six male speakers reading 95 typical input sentences of the dialogue with Jijo-2. When we use a standard microphone, which is set near the speaker's mouth, the average recognition rate for a whole sentence is 86 percent. However, in an office setting, the recognition rate decreases because of ambient noise and echo caused by the distance between the speaker and the microphones. When an omnidirectional microphone—set about 50 cm from the speaker—records the data, the recognition rate drops to 47 percent.

To cope with recognition performance degradation, we introduced multiple grammars for recognition. When the robot starts up, three speech recognition processes start running. Each recognizer handles one grammar—the reply grammar, location grammar, or full grammar. The dialogue manager in the integration layer chooses one grammar at a time. When a yes-or-no reply is expected, the dialogue manager activates the reply grammar. When location or person names are expected, it invokes the location grammar. Otherwise, it uses the full grammar. Because evaluating the performance quantitatively online is difficult, we have not yet evaluated the recognition rate's increase. From experience with more than 100 demonstrations to visitors, however, we conclude that introducing multiple grammars significantly improves speech recognition.

Because the pipelined multiprocessors (two DSPs and a MPU) form the audio beam, generate VQ code, and perform speech recognition, the total recognition finishes almost in real time. The longest delay is introduced by the 0.4-second pause to identify the end of an utterance. This is also important for realizing fluent communication between the robot and humans.

To understand a speaker's intention and respond correctly, an utterance's semantic content is extracted from the result of speech recognition. We use a simple task-dependent dictionary and grammar from the speech recognizer to parse the recognition result. In this word dictionary, we embed task-dependent semantic equivalences.

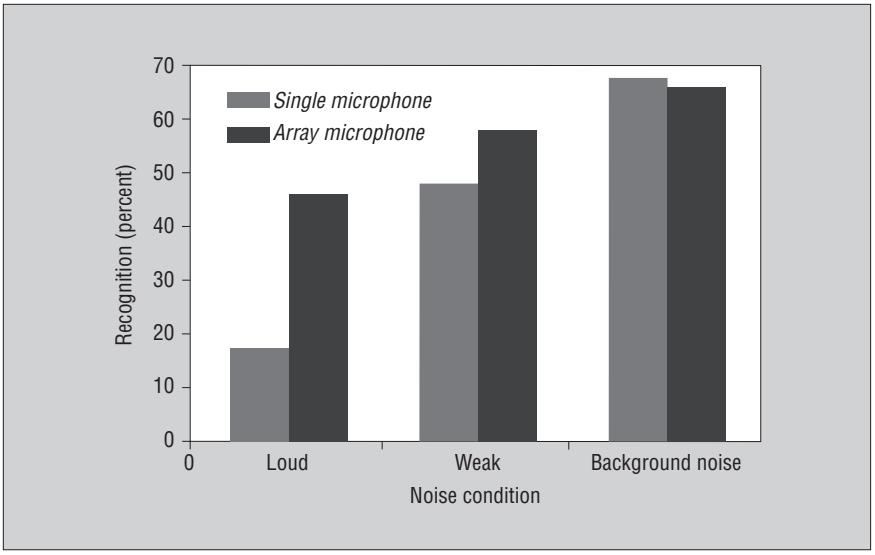


Figure 5. The speech recognition performance of Jijo-2's multimicrophone array in various noise conditions.

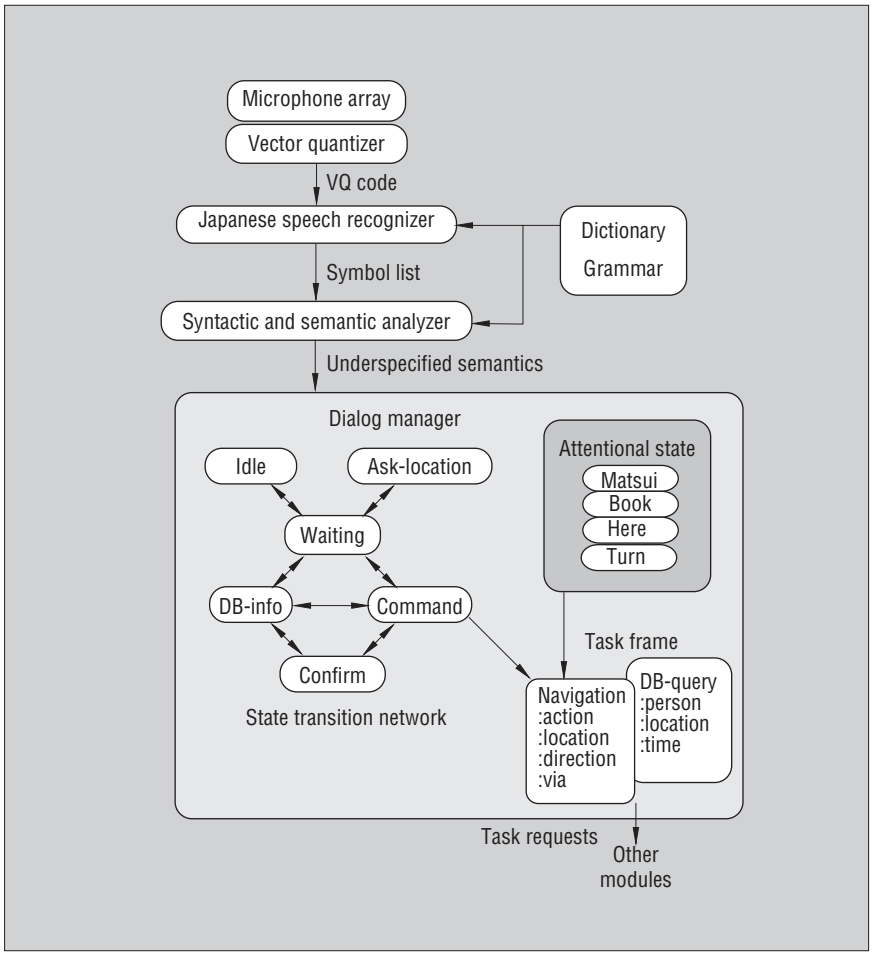


Figure 6. The state transition network that controls the dialogue process.

How Jijo-2 carries out a dialogue

The dialogue manager maintains the state of the dialogue and outputs appropriate responses from the robot. A state transition network controls the dialogue process, where the current state is represented as a state in the finite-state automaton network (see Figure 6).

Depending on the state, the robot's responses to input utterances change. The rules for response generation and state transition are encoded as logical statements using a Prolog-like language interpreted by EusLisp. The robot knows about a particular task or task frame, which is a frame structure with several slots to represent necessary information for task execution. The dialogue manager tries to fill the slots using the dialogue's information content. After filling all slots in a specific task frame, the robot can execute the task. Currently, we've prepared five kinds of task frames: database query, database update, identify person, navigation, and call person. The task frame for navigation has four slots, such as destination, direction, action, and modifier (fast, slow, and so forth). In this case, depending on the action slot's value, destination or direction must always be filled before executing a task. The other slots are filled with default values if the user does not specify them.

When the system starts up, the dialogue module is in the *Idle* state and must wait for input from the user (such as "hello"). When a user calls the robot, the state changes to *Waiting*, and the system waits for the user's request. If the request relates to database access, the state moves to *DB info*, and the system generates a task frame for "database query" or "database update," depending on the input sentence's mode (interrogative or declarative). If the request relates to navigation, the state moves to the *Command* state, and the system generates a task frame for the specific command. When the robot asks the user a yes or no question, the state moves to *Confirm*, and the system expects a yes-or-no response from the user. In the confirm state, the yes–no grammar is used for speech recognition. The state transition network effectively eliminates wrong responses to spurious utterances generated by noise and prevents the system from catastrophic faults. In case some utterance is misidentified, the system reconfirms the utterance before executing the task.

Knowing the current context is also helpful for understanding utterances. We use an attentional state that lists salient entities referred to in the preceding utterances. In natural Japanese conversations, salient information, which is easily surmised from the context, is not repeated and is often omitted from the utterance. Hence, to understand underspecified sentences, the system must keep track of salient information in a conversation. For this purpose, we tentatively introduced a simple short list of salient discourse entities

(the people, objects, and events being discussed), and we use the information stored in the list to fill the task frame slots.

From the viewpoint of taking turns while talking, the user controls the dialogue. Using the task frame, the system can accept fragmental information distributed along several input utterances in any order. When the user does not specify necessary slot values within some determined time, the system prompts the user for input.

How Jijo-2 executes tasks

Jijo-2 is an integrated robot that carries out dialogues to provide services combined with its navigation and sensing capabilities. When the user's utterances fill all required slots of a task frame and confirm the contents of the slots, the dialogue module tries to execute a task using these capabilities.

If the current task is a database query or update, the dialogue module dispatches the task to the database module. Normally, a response is immediate, and the speech synthesizer can pronounce the result for the query or update task. If it takes too long, the user might want to start another conversation, which the dialogue module handles, because the modules all run concurrently in an event-driven manner. The database containing people's schedules and locations is implemented on a database server (Postgres server), which also provides Web-based access. Therefore, once a dialogue about a user's schedule lets the system update the database, the information is available for public access.

The dialogue module dispatches the navigation task to the driver module, which also maintains a map of the environment. The dialogue module can always command the driver to go to somebody's office, but the driver might not know how to go there. In such a case, the driver module requests the dialogue module to ask for navigation instructions from a person nearby. Additionally, during navigation, the driver might encounter an unexpected landmark, which could also lead the dialogue module to conduct a conversation to confirm the new landmark's location.

As Figure 4 shows, all modules related to dialogue and task execution run concurrently and are controlled in an event-driven manner. When some event such as speech input, finding a landmark, or detecting an obstacle occur, the respective modules for processing the event communicate with each other and collaboratively process the event. Hence, Jijo-2 can maintain a dialogue while it navigates a corridor. If the dialogue module can handle the dialogue within the module or the database—such as a query about the current date and time—then it is locally processed without interfering with the navigation. On the other hand, if it contains commands to stop navigation or change destination, the dialogue module retracts the current command to the driver and restarts another behavior. Jijo-2 knows its current status of task execution, and software developers can program handling methods for interruptive input such as "stop" or metacommands such as "cancel," depending on the task. This means, however, that we should implement the handling methods for all exceptional cases, which becomes difficult when the task variation and complexity increases. Solving the binding of interruptive or metacommands to the specific task is also difficult. The current system implements a simple binding rule that chooses the task on top of the stack of required tasks.

An omnidirectional camera's main advantage is its large field of view. This view, for a mobile robot application, lets many landmarks be simultaneously present in the scene, leading to more accurate localization.

How Jijo-2 finds and recognizes people

When Jijo-2 hears "hello" while in its idle state, it turns to the user, using information about the sound source direction detected by the microphone array unit. Simultaneously, it invokes a skin color detection function in the vision module to find the user's face.[4] Using this, the camera module controls the CCD camera's pan tilt and tries to locate the human face (the largest region of skin color) at the center of the image. Then, the face recognition module runs to recognize the extracted face region.

The current face recognizer is based on combining a log-polar transform of the image and higher-order local autocorrelation features (HLAC).[5] Because the log-polar transform maps the image's rotation and scaling to a translation of the log-polar image—and because HLAC is translation invariant—this combination makes the recognition result robust to the input image's rotation and scaling.

A person's face is memorized as a template feature vector of 105 dimensions created from several shots of training images of the face by linear-discriminant analysis. When a face image is input, the system extracts the feature vector from the image, compares it with the memorized template vectors, and chooses the closest one. The label accompanied with the face is output from the recognition module and inserted as the speaker's name in the dialogue manager's attentional state stack.

The omnidirectional vision system

Among the several sensor devices used in robotics, vision provides the richest source of information—but it is traditionally restricted to standard CCD cameras. However, omnidirectional vision systems are becoming increasingly popular in mobile robotics for tasks such as environment modeling and navigation.[6] Jijo-2 is equipped with an omnidirectional imaging device, mounted on top of the robot and consisting of a vertically mounted standard camera aimed upward and looking into a spherical mirror (see Figure 7). Each omnidirectional image has a resolution of 320 × 240 pixels.

An omnidirectional camera's main advantage—compared to a traditional camera—is its large field of view. This view, for a mobile robot application, lets many landmarks be simultaneously present in the scene, leading to more accurate localization. This, in turn, obviates the need for expensive active vision mechanisms for landmark detection, making the robot localization task easier.

How Jijo-2 extracts features

Prior to building environment models to be used for vision-based navigation, the robot must be able to extract appropriate features from images. These features can be natural or artificial,[7] and their purpose is to facilitate localization performance. In statistical terms, the need for feature extraction arises from the fact that, normally, the dimensionality of the robot's sensor data is high, making any statistical inference in the original space unrealistic.

Recently, there has been a growing interest in automatic procedures that learn such features from a data set. This automatic learning of features is a natural objective, because, in principle, it can make the process inde-

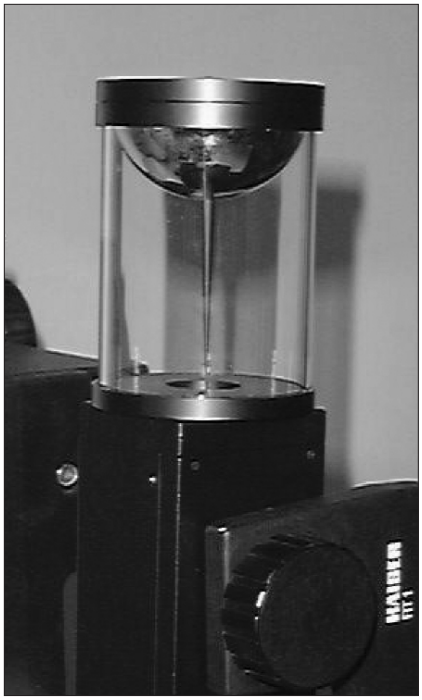


Figure 7. The omnidirectional camera.

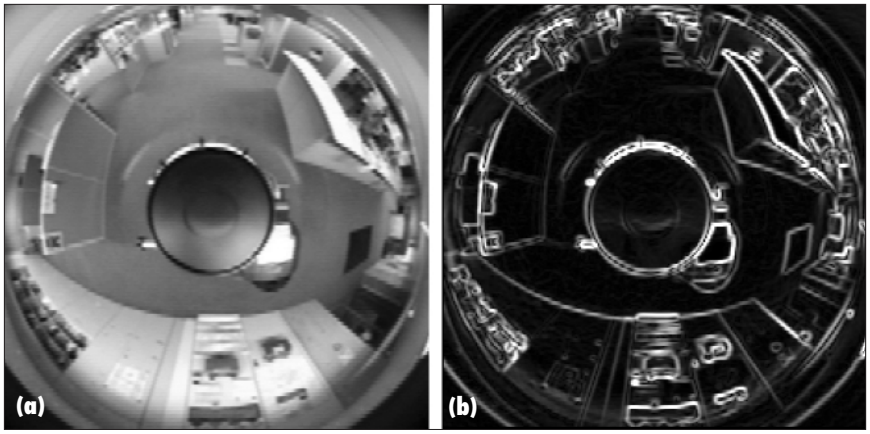


Figure 8. (a) An omnidirectional image; (b) the same image after edge detection.

pendent of environment, giving rise to a so-called *appearance-modeling approach*.[8] Learning is most often carried out with statistical methods, and the easiest and most commonly used is principal component analysis.[9] In this method, a set of sensor measurements is linearly projected to a low-dimensional subspace, which is easily computed by solving a matrix eigenvalue problem. The nice thing about using PCA is that it combines many optimality properties and is simple to implement. There have been several recent reports in the robotics literature that apply PCA on omnidirectional images.[10,11]

Jijo-2 carries out feature extraction through edge detection, which solves the problem of illumination changes in the robot's environment. The original color image is converted to grayscale and then linearly normalized so that the range of gray values is within [0, 255]. Then a Sobel operator performs edge detection. Figure 8 shows a snapshot from the omnidirectional camera and an image after edge detection.

Next, we threshold the image and retain only the edge pixels with the highest intensity. To capture general characteristics of one scene and avoid searching for particular structure from these edge pixels, we use the spatial density of these pixels.[6] The idea is that in those parts of the image where many edge pixels appear, there is a potentially good landmark on which to focus (such as the desk outline in the upper right corner of Figure 8). Moreover, edge pixels that appear as outliers should be ignored as unnecessary details of the scene—the density estimation algorithm should detect this.

To compute the spatial density, we use the Parzen method. This amounts to using one Gaussian kernel per edge pixel and then adding the contributions from all pixels to compute the local density. Unfortunately, this operation has quadratic complexity and is inconvenient for real-time processing. However, there is a fast algorithm for density estimation based on the fast Fourier transform whose logarithmic complexity makes the total cost drop significantly. The density function is computed on the panoramic transformation of the omnidirectional snapshot with a resolution of 10×30 (lower than the original 320×240 image), which constitutes a 300-dimensional feature vector. To further reduce the dimensionality of the feature vector, we apply PCA. (You can find more details elsewhere.[6])

How Jijo-2 localizes itself

Having extracted a feature vector from an omnidirectional image while the robot moves, the goal is to predict in real time the robot's position from this feature vector. There are many possible ways to establish associations between a set of robot positions and feature vectors. To predict the robot's position from a feature vector, we use a simple nearest-neighbor method: When the robot observes a new image, it compares the corresponding feature vector—after Parzen density estimation and PCA—to all feature vectors stored in the database in terms of the Euclidean distance. The position that corresponds to the feature vector with the smallest distance to the current feature vector provides a maximum a posteriori estimate of the robot's position.

This nearest-neighbor matching procedure is efficient, because after PCA, the training set of feature vectors is low-dimensional. We can achieve further speedup in the nearest neighbor search if we use an appropriate structure to store the feature vectors—such as a k-dimensional tree. Alternatively, a regression method can replace the nearest-neighbor search.

An example of edge-based feature extraction

We carried out an experiment involving 400 omnidirectional images while the robot was moving along a corridor in our office environment. We applied the edge-based feature extraction method and PCA to further reduce the dimension. Figure 9 shows cumulative variance as a function of the number of principal components in PCA. We note that the first nine principal components provide more than 90 percent of the original data set's total variation. So, with little loss of information, we can discard the other dimensions. The complete database information that we need to store and use during real-time robot localization is the 400×9 training-feature set for nearest-neighbor matching and a single projection matrix, which maps the edge feature image to the 9-dimensional PCA feature vector.

Using this data set and a nearest-neighbor method, we observed in most cases that the robot had good localization performance. Figure 10 plots the estimated offset (in meters) of the robot along the corridor as a function of the true offset. Except for the three outlier points, our algorithm can predict with good accuracy the robot's true position. The three wrong predictions are due to occlusion of parts of the scene by objects that did not appear in the original PCA database construction.

Supervised feature extraction

When the robot collects its observations in a supervised manner—that is, when they are annotated in the sample with the robot's

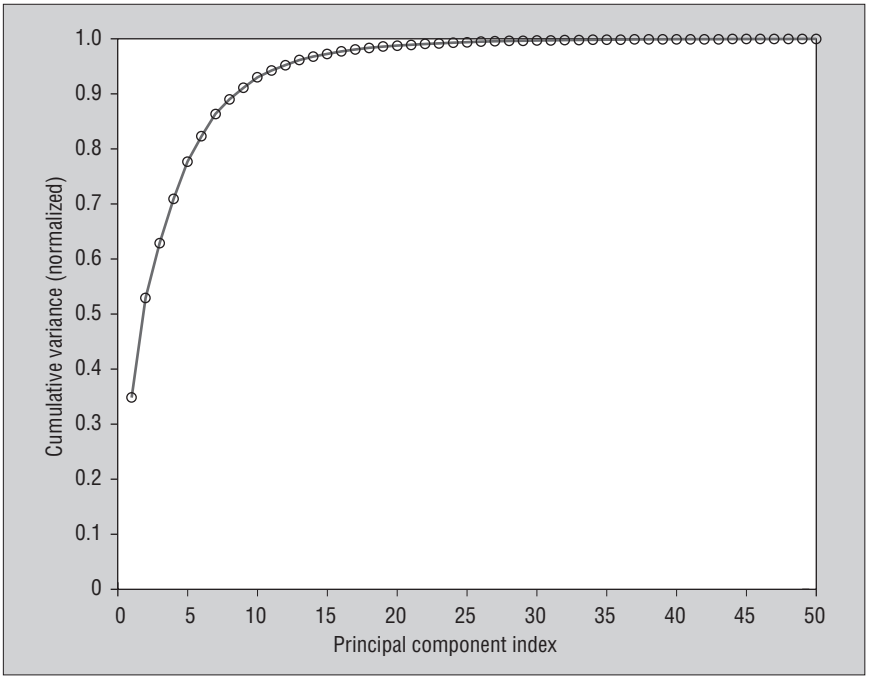


Figure 9. The cumulative variance per principal component.

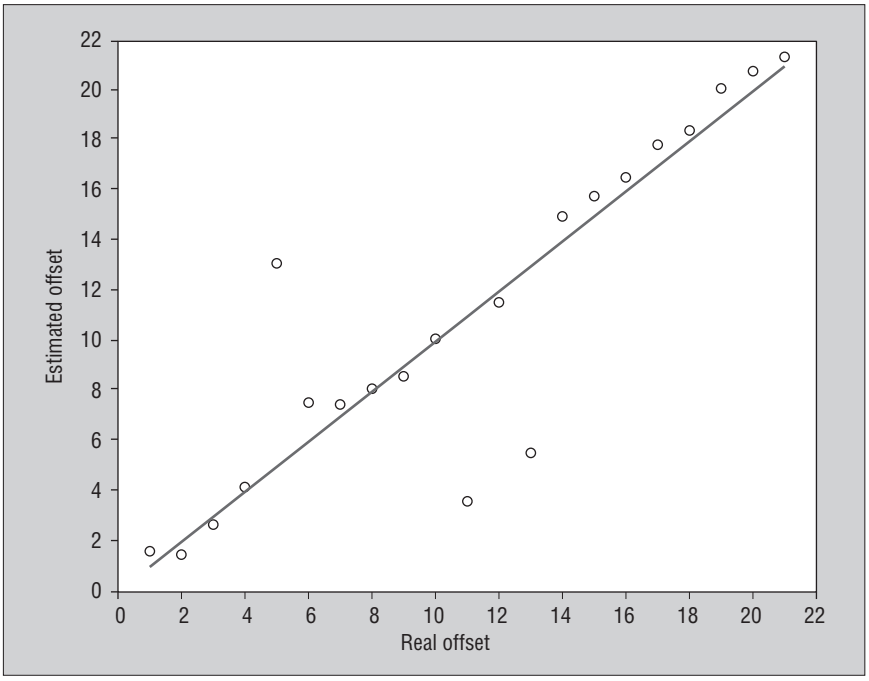


Figure 10. Estimated robot position versus the real position. The three outlier points are due to occlusion.

position—then PCA can be suboptimal. The reason is that PCA is an unsupervised feature-extraction method that uses only the observed sensor vectors to compute the projection directions, and thus the extracted features can have little discriminatory power between robot positions. If feature extraction is to be used for tasks such as robot localization and navigation, then a supervised projection method could substitute for a PCA.[12]

The principal idea is that an optimal projection method should account for the topological structure of the observed data, which, under certain assumptions about the robot's environment, can be assumed to form a low-dimensional manifold embedded in the observation space (plus noise). An optimal linear projection is one that preserves this resulting manifold's topological structure—for example, it reduces the number of self-intersections.

Figure 11 shows results from using the supervised feature extraction method on 100 images collected by the robot along a trajectory. We clearly see the advantage of the supervised projection method over PCA. From the projected manifold's shape, we see that considering the pose information during projection can significantly improve the resulting features. There are fewer self-intersections of the projected manifold in our method than in PCA, which, in turn, means better robot position estimation on average.

Socially embedded learning

In the design of traditional machine-learning systems, the learning systems are fed with clean training data prepared by the users, and they learn simple functional relationships hidden in the data. The learning systems are isolated from the information-rich environment around them. On the other hand, teaching by direct human supervision, as is the case for young children in close interaction with their parents, can be the most powerful teaching strategy. This kind of closely coupled interaction with the environment can significantly support and accelerate an agent's learning capabilities. We call this kind of learning process *socially embedded learning*.[13]

In the current Jijo-2 system, we explored the first step in this direction in the map-learning scheme by using a robot–human dialogue. Map learning is important for mobile robots, and although a variety of techniques have been proposed,[1,14] it remains a difficult problem. In particular, map learning with uncertain sensors is difficult because of accumulation of uncertainty about the robot's location. Here we have introduced the robot–human dialogue as a potential solution to that problem. Jijo-2 starts with no map information and acquires a probabilistic map under human supervision, as Figure 12 shows. This procedure is similar to the one we execute when we have a newcomer in our office and shows how a simple dialogue can drastically improve the map-learning process. This is a simple example of the socially embedded learning principle—of course, more research is needed, including analysis of human-to-human interaction in various learning situations.

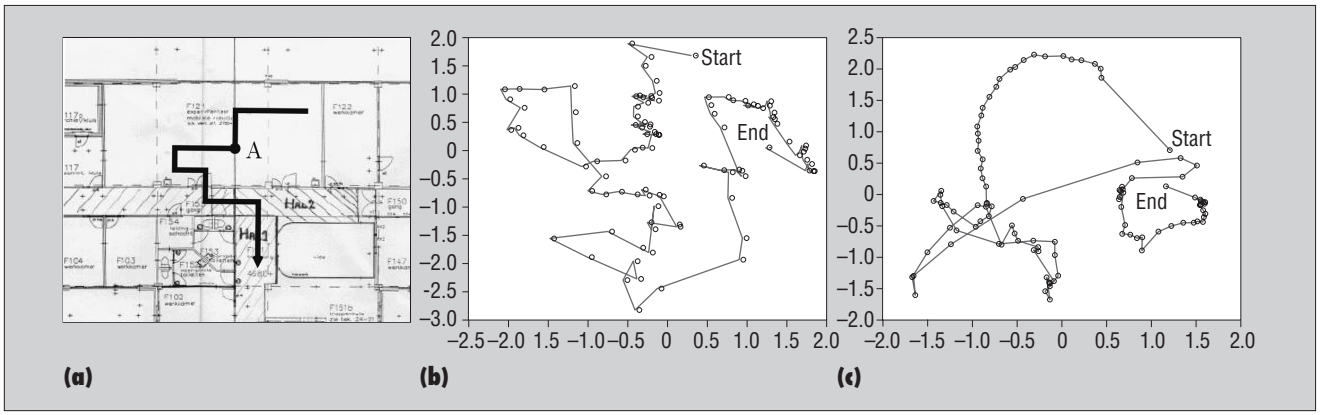


Figure 11. (a) The robot trajectory; projection of the panoramic image data to 2D, (b) using the proposed method and (c) projection on the first two principal components. The start and end points are the projections of the panoramic images captured by the robot at the beginning and end, respectively, of its trajectory.

Although Jijo-2 can communicate with humans using a spoken-dialogue system and learn good features and a probabilistic map of its environment, there are still many open issues to make the system really adaptive and robust.

Our dialogue system's biggest problem is that the design of the state transition network and task frames is ad hoc and strongly task-dependent. The scheme of representing semantic information of the utterance is also informal. These problems make it difficult to extend the system to cover wider tasks. When a new task is introduced, the system designer must redesign large parts of the structures. To avoid this problem and make such incremental extensions easier, designing a more systematic semantic representation is necessary. For the design, we need a deeper understanding of the robot's tasks and the task domain's ontology. Attention control of the dialogue system is currently also rather simple. Introducing theories of discourse structure,[15,16] such as in the Collagen system,[17] is an important issue.

A basic element we are continuously investigating is the seamless integration of the robot's communication and learning. In particular, the robot can use its communication skills effectively for collecting information necessary to learn the environment. On the other hand, the embodied knowledge about an environment can prove essential for executing natural conversation. Such coevolution of the communication and learning skills is currently the most demanding part of the Jijo-2 architecture and constitutes an ongoing research issue in our groups.

Ongoing research also involves extending the current simple topological map to a hybrid topological–metric map. The benefit of using such a map is that a rough description of the environment is incorporated in the topological map, allowing precise metric information to be used only in those areas of the environment requiring accurate navigation. We can smoothly integrate such a framework within the state transition network that controls the dialogue process (see Figure 6), allowing the definition of a general state transition network in a joint verbal–world space. Beside the scientific challenges and innovations that such a joint-space approach might entail, we intuitively expect that it can provide elegant solutions to practical problems, such as optimal backtracking in task execution when retracting commands—such as "stop"—are issued.

R: Where am I?
U: You are in front of Dr. Asano's office.
R: Where shall I go?
U: Please go to Dr. Hara's office.
R: Sorry, I don't know how to go to Dr. Hara's office.
U: OK. Please go straight.
R: OK.

The robot moves straight ahead until the next salient landmark is perceived.
R: Where am I?
U: You are in front of Dr. Matsui's office.

Figure 12. Teaching Jijo-2.

In parallel, introducing the user to the robot's control loop, in the socially embedded manner we have described, immediately suggests the use of reinforcement-learning techniques for map learning and navigation.[18] We could then regard user feedback as a reinforcement signal that rewards or penalizes the robot's decisions, while we could allow stochastic actions between the (discrete) states of the verbal–world space, and augment the joint-transition network with a state–action reward function. Then, learning of optimal decisions within a particular environment could be achieved using dynamic programming or alternative reinforcement learning techniques, leading to an efficient learning-by-experience paradigm.

Finally, it would be interesting to see how Jijo-2 could adapt itself within a community of socially embedded robots and to what extent the interaction of multiple semisentient robots can facilitate their learning and task execution capabilities. ■

Acknowledgments

We thank John Fry (Stanford University) for his contribution and the reviewers for their motivating comments. The Real World Computing Program supported this work.

References

1. W. Burgard et al., "Experiences with an Interactive Museum Tour-Guide Robot," *Artificial Intelligence*, vol. 114, nos. 1–2, Oct. 1999, pp. 3–55.

2. T. Matsui et al., "Integrated Natural Spoken Dialog System of Jijo-2 Mobile Robot for Office Services," *Proc. 16th Nat'l Conf. Artificial Intelligence* (AAAI 99), AAAI Press, Menlo Park, Calif., 1999, pp. 621–627.
3. T. Matsui, H. Asoh, and I. Hara, "An Event-Driven Architecture for Controlling Behaviors of the Office Conversant Mobile Robot Jijo-2," *Proc. IEEE Int'l Conf. Robotics and Automation*, IEEE Press, Piscataway, N.J., 1997, pp. 3367–3371.
4. I. Hara et al., "Communicative Functions to Support Human Robot Cooperation," *Proc. 1999 IEEE/RSJ Conf. Intelligent Robots and Systems*, IEEE Press, Piscataway, N.J., 1999, pp. 683–688.
5. K. Hotta, T. Kurita, and T. Mishima, "Scale Invariant Face Detection Method Using Higher-Order Local Autocorrelation Features of Log-Polar Image," *Proc. Int'l Conf. Automatic Face and Gesture Recognition* (FG '98), IEEE CS Press, Los Alamitos, Calif., 1998, pp. 70–75.
6. N. Vlassis et al., "Edge-Based Features from Omnidirectional Images for Robot Localization," *Proc. IEEE Int'l Conf. Robotics and Automation*, IEEE Press, Piscataway, N.J., 2001, pp. 1579–1584.
7. J. Borenstein, B. Everett, and L. Feng, *Navigating Mobile Robots: Systems and Techniques,* A.K. Peters, Wellesley, Mass., 1996.
8. S.K. Nayar, S.A. Nene, and H. Murase, "Subspace Methods for Robot Vision," *IEEE Trans. Robotics and Automation*, vol. 12, no. 5, Oct. 1996, pp. 750–758.
9. I.T. Jolliffe, *Principal Component Analysis*, Springer-Verlag, New York, 1986.
10. M. Jogan and A. Leonardis, "Robust Localization Using Panoramic View-Based Recognition ," *Proc. 15th Int. Conf. Pattern Recognition*, IEEE CS Press, Los Alamitos, Calif., 2000, pp.136–139.
11. B.J.A. Kröse et al., "A Probabilistic Model for Appearance-Based Robot Localization," *Image and Vision Computing*, vol. 19, no. 6, Apr. 2001, pp. 381–391.
12. N. Vlassis, Y. Motomura, and Ben Kröse, "Supervised Dimension Reduction of Intrinsically Low-Dimensional Data," to be published in *Neural Computation*, vol. 14, no.1, Jan. 2002.
13. H. Asoh et al., "Socially Embedded Learning of the Office-Conversant Robot Jijo-2," *Proc. 15th Int'l Joint Conf. Artificial Intelligence*, Morgan Kaufmann, San Francisco, 1997, pp. 1552–1557.
14. S. Thrun, "Probabilistic Algorithms in Robotics," *AI Magazine*, vol. 21, no. 4, Winter 2000, pp. 93–109.
15. B.J. Grosz and C.L. Sindner, "Attention, Intentions, and the Structure of Discourse," *Computational Linguistics*, vol. 12, no. 3, July–Sept. 1986, pp. 175–204.
16. K.E. Lochbaum, "A Collaborative Planning Model of Intentional Structure," *Computational Linguistics*, vol. 24, no. 4, 1998, pp. 525–572.
17. C. Rich, C.L. Sindner, and N. Lesh, *Collagen: Applying Collaborative Discourse Theory to Human–Computer Interaction*, tech. report TR-2000-38, Mitsubishi Electric Research Laboratories, 2000.
18. R.S. Sutton and A.G. Barto, *Reinforcement Learning: An Introduction,* MIT Press, Cambridge, Mass., 1998.

The Authors

Hideki Asoh is a group leader at the Information Technology Research Institute, National Institute of Advanced Industrial Science and Technology, Japan. His research focuses on constructing learning intelligent systems. He received a BS in mathematical engineering and an MS in information engineering from the University of Tokyo. He is a member of the Japanese Society of Artificial Intelligence; the Inst. of Electronics, Information, and Communication Engineers; and the Japanese Neural Network Society. Contact him at the Information Technology Research Inst., National Inst. of Advanced Industrial Science and Technology (AIST), Tsukuba-Central-2, 1-1-1 Umezono, Tsukuba, Ibaraki 305-8568, Japan; h.asoh@aist.go.jp.

Nikos Vlassis is an assistant professor at the Computer Science Institute of the University of Amsterdam. His research interests include statistical feature extraction, robot environment modeling, and multiagent systems. He received a BS in electrical and computer engineering and a PhD in artificial intelligence, both from the National Technical University of Athens. He holds the Dimitris N. Chorafas Foundation prize (Lucerne, Switzerland) for young researchers in Engineering and Technology. Contact him at the Computer Science Inst., Univ. of Amsterdam, Kruislaan 403, 1098 SJ Amsterdam, Netherlands; vlassis@science.uva.nl.

Yoichi Motomura is a senior research scientist at the National Institute of Advanced Industrial Science and Technology, Japan. He also works for CREST, the Japan Science and Technology Corporation. His research interests include probabilistic models and statistical learning methods for real-world applications in robotics, intelligent systems, and information retrieval domains. He received his BS and MS in computer science from the University of Electro-Communications, Tokyo. He has received the research promotive award (1999) and the best presentation award (1998) from the Japanese Society of Artificial Intelligence. Contact him at the Information Technology Research Inst., National Inst. of Advanced Industrial Science and Technology (AIST), Tsukuba-Central-2, 1-1-1 Umezono, Tsukuba, Ibaraki 305-8568, Japan; y.motomura@aist.go.jp.

Futoshi Asano is a senior research scientist at the National Institute of Advanced Industrial Science and Technology, Japan. His research interests include array signal processing, adaptive signal processing, statistical signal processing, and blind source separation. He received a BS in electrical engineering and an MS and a PhD in electrical and communication engineering from Tohoku University, Sendai, Japan. Contact him at the Information Technology Research Inst., National Inst. of Advanced Industrial Science and Technology (AIST), Tsukuba-Central-2, 1-1-1 Umezono, Tsukuba, Ibaraki 305-8568, Japan; f.asano@aist.go.jp.

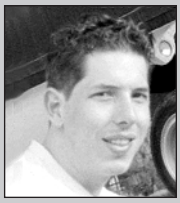
Isao Hara is a senior research scientist at the National Institute of Advanced Industrial Science and Technology, Japan. His interests include mobile robot navigation, multiagent system, robot-control systems, and networked robotics. He received his BS, MS, and PhD from Kyushu University, Fukuoka, Japan. Contact him at the Information Technology Research Inst., National Inst. of Advanced Industrial Science and Technology (AIST), Tsukuba-Central-2, 1-1-1 Umezono, Tsukuba, Ibaraki 305-8568, Japan; isao-hara@aist.go.jp.

Satoru Hayamizu is a senior research scientist at the National Institute of Advanced Industrial Science and Technology, Japan. He works on speech recognition, spoken dialogue, and communication with artificial systems. He received his BE, ME, and DrE from the University of Tokyo. Contact him at the Information Technology Research Inst., National Inst. of Advanced Industrial Science and Technology (AIST), Tsukuba-Central-2, 1-1-1 Umezono, Tsukuba, Ibaraki 305-8568, Japan; s.hayamizu@aist.go.jp.

Katsunobu Itou is a senior research scientist at the National Institute of Advanced Industrial Science and Technology, Japan. His research interest is spoken-language processing. He received his BS, ME, and PhD in computer science from the Tokyo Institute of Technology. Contact him at the Information Technology Research Inst., National Inst. of Advanced Industrial Science and Technology (AIST), Tsukuba-Central-2, 1-1-1 Umezono, Tsukuba, Ibaraki 305-8568, Japan; itou@ni.aist.go.jp.

Takio Kurita is a deputy director at the Neuroscience Research Institute, National Institute of Advanced Industrial Science and Technology (AIST), Japan. His research interests include statistical pattern recognition and neural networks. He received his BEng from Nagoya Institute of Technology and DrEng from the University of Tsukuba. Contact him at the Neuroscience Research Inst., National Inst. of Advanced Industrial Science and Technology (AIST), Tsukuba-Central-2, 1-1-1 Umezono, Tsukuba, Ibaraki 305-8568, Japan; takio-kurita@aist.go.jp.

Toshihiro Matsui works at the National Institute of Advanced Industrial Science and Technology, Japan. He led the Jijo-2 mobile office robot project team in the RWC partnership subsidized by METI, Japan. He received his BE, MD, and PhD in measurement engineering and information engineering from the University of Tokyo. Contact him at the Planning Headquarters, AIST, Tsukuba-Central-1, 1-1-1 Higashi, Tsukuba, Ibaraki 305-8561, Japan; t.matsui@aist.go.jp.



Roland Bunschoten is a PhD candidate in the Intelligent Autonomous Systems group at the University of Amsterdam. The Real World Computing Program funds his research. He is working on his dissertation on active map building and sensory representations, and his research interests include Markov robot localization, appearance-based environment modeling, omnidirectional vision, and multibaseline panoramic stereo vision. He received his MSc in artificial intelligence from the University of Amsterdam. Contact him at the Intelligent Autonomous Systems Group, Univ. of Amsterdam, Faculty of Science, Kruislaan 403, 1098 SJ Amsterdam, Netherlands; bunschot@science.uva.nl.

Ben Kröse is an associate professor at the University of Amsterdam, where he leads a group in computational intelligence and robotics. His interests include mobile-robot navigation and learning methods for world modeling. His special interest is the field of personal robots. He received his MSc and PhD from Delft University of Technology. Contact him at the Dept. of Computer Science, Univ. of Amsterdam, Kruislaan 403, 1098 SJ Amsterdam, Netherlands; krose@science.uva.nl.