LOLA

Probabilistic Navigation for Topological Maps

Ricardo Gutierrez-Osuna and Ren C. Luo

■ LOLA's entry in the Office Delivery event of the 1995 Robot Competition and Exhibition, held in conjunction with the Fourteenth International Joint Conference on Artificial Intelligence, was the culmination of a three-month design and implementation period for an indoor navigation system for topological maps. This article describes the major components of the robot's navigation architecture. It also summarizes the experiences and lessons learned from the competition.

The Office Delivery event of the 1995 Robot Competition and Exhibition, held in conjunction with the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95), was similar in concept to previous years: It required the robots to navigate in an officelike partitioned environment that consisted of hallways and rooms. However, a few new twists were added to the 1995 competition. First, a complete topological map of the environment was not available. Instead, a set of instructions, for example, "turn third left" and "go past foyer," would guide the robot through the hallways toward a goal room. Second, it would be possible that the instructions contained an error, such as directing the robot toward a nonexistent hallway or room. Third, the information provided in the instructions only specified a number of "openings" that had to be detected before turning into another hallway or entering the goal room. Only the nature of the last opening of every instruction could be inferred (a hallway in the case of a turn instruction or a doorway in the case of an enter instruction), but the intermediate openings could be of any type.

The lack of a more qualitative description of the environment limited the capabilities of the probabilistic navigation algorithm on the robot, which could only be used as a sophisticated feature counter (figure 1).

Navigation Architecture

The navigation architecture implemented on LOLA consists of four major modules: (1) state-set progression, (2) feature detection, (3) low-level motion routines, and (4) registration. The interaction between these four modules and the sensory-motor devices on the robot is depicted in figure 2.

State-set progression (Nourbakhsh, Powers, and Birchfield 1995) provides a probabilistic inference for topological navigation that copes with the uncertainties in sensing. With a complete topological map of the environment, this algorithm is able to compensate for misclassified or undetected features, allowing the robot to localize robustly. The *feature-detection module* supplies the abstract features that drive the probabilistic inference. These features are extracted by fusing vision and sonar information. The *low-level motion routines* provide a set of standard primitives, such as goal-seek, direction-follow, and wall-follow, that perform obstacle avoidance and velocity control of the vehicle. The *registration module* is responsible for computing the robot's orientation by fitting line segments to the sonar boundaries using least mean squares estimation.

In the following subsections, these modules are described in more detail.

State-Set Progression

Building a robust navigation system requires modeling the environment with a level of abstraction consistent with the sensing capa-

Figure 1. LOLA.

bilities on the robot. Highly detailed descriptions of the world (that is, geometric) require accurate sensors or, more commonly, sophisticated algorithms to extract useful information from the raw sensor data. They also call for elaborate path-planning strategies. However, such detailed knowledge about the state of the robot is not always necessary. In most cases, the state of the robot can be represented in a more qualitative manner, such as the way humans do it.

Topological maps are a good example of these ideas (Kortenkamp and Weymouth 1994; Meng and Kak 1993; Kuipers and Byun 1991). In a topological map, *nodes* can be thought of as locations in the real world suitable for sensing and reasoning, and *edges* represent paths between nodes that can be accomplished with a collection of competent low-level motion routines. Sensing between nodes is irrelevant to the reasoning modules (Becker et al. 1995) and is considered useful in helping the low-level motion routines traverse paths. Within this framework, the state of the robot can be represented by a node in the topological graph.

However, what if the sensing capabilities on the robot are so limited that feature recognition, and even detection, cannot always be guaranteed? In this case, the state of the robot must be represented as a set of nodes in the graph (state set) that are believed to be possible, each one having a certainty value (Nourbakhsh, Powers, and Birchfield 1995; Russel and Norvig 1995; Kortenkamp and Weymouth 1994; Kortenkamp et al. 1994).[1] With the arrival of new percepts from the sensors, the state set is updated. *Good percepts* reduce uncertainty by sharpening the state set to a few nodes with high certainty values. *Bad percepts* blur the state set but still provide useful information.

State-set progression, presented by Nourbakhsh, Powers, and Birchfield in the 1994 Robot Competition and Exhibition, is an effective implementation of this approach. Sensing is modeled as a collection of distinctive features in the environment and a collection of percepts that the sensors can return. Ideally, each percept maps to a unique feature in the collection. In practice, each percept can be generated from any feature, and a conditional certainty $C(feature|percept)$[2] is used to represent the likelihood of each feature-percept pair. At time k, a percept arrives from the sensors, and the new certainty of each state $C(S_i)^k$ is calculated as the sum of progression certainties from all previous states $S_j|_{j<i}$. The progression certainty from state S_j to state S_i is the certainty of state S_j at time k-1, $C(S_j)^{k-1}$ times the likelihood of having traversed the path between these states given the evidence of the percept. Shown in figure 3, the likelihood of such a path is the same as missing all intermediate features in the path times the likelihood of a match between *percept* and the feature of S_i.

$$C(S_i)^k = \sum_{j<i} C\left(S_j \xrightarrow{path} S_i \,\middle|\, percept\right) C(S_j)^{k-1} \qquad (1)$$

$$C\left(S_j \xrightarrow{path} S_i \,\middle|\, percept\right) = C\left(feature_{S_i} \,\middle|\, percept\right) \prod_{j<k<i} C\left(feature_{S_k} \,\middle|\, no\_percept\right) \qquad (2)$$

On the right-hand side of equation 2, the first term, $C(feature\ s_i|\ percept)$, represents the certainty of being at state S_i given the *percept* and no information about the state set at time k – 1. When merged in the summation of equation 1, it measures the likelihood of single-edge progressions. The second term,

$$\prod_{j<k<i} C\left(feature_{S_k} \,\middle|\, no\_percept\right),$$

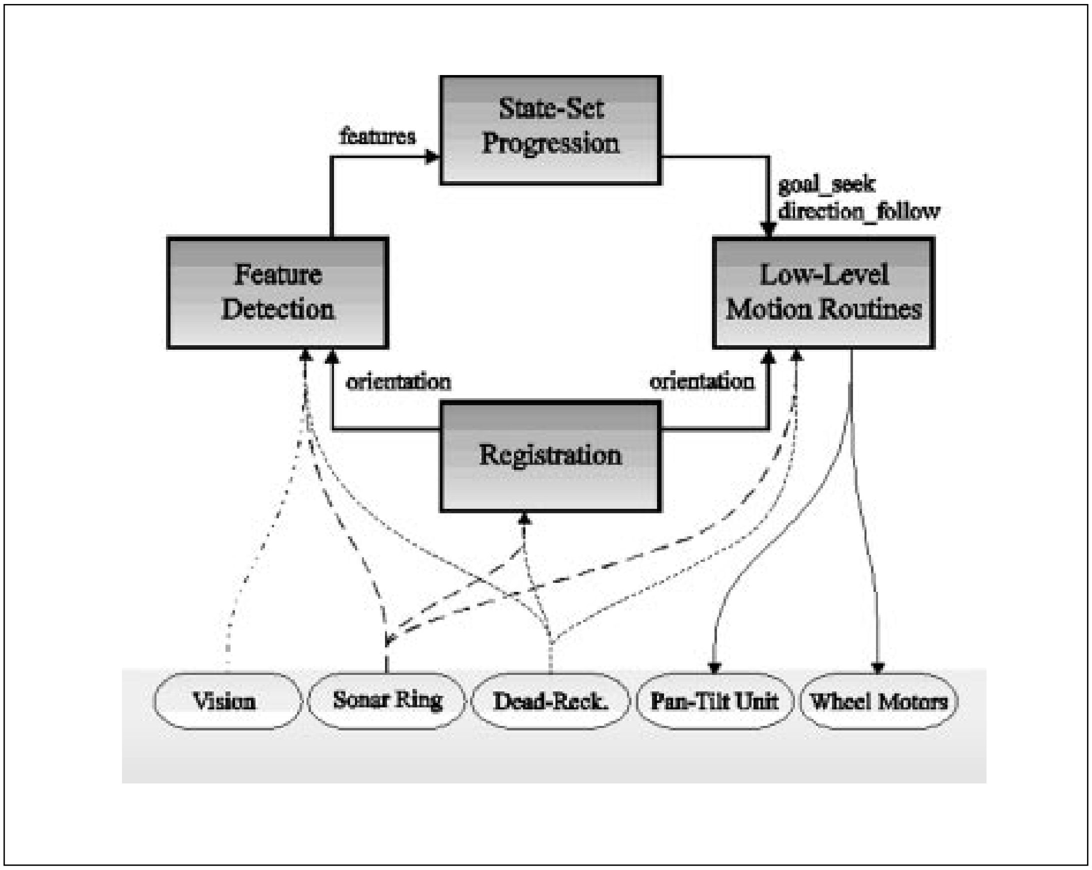

Figure 2. Navigation Architecture.

accounts for the possibility of missing features, allowing a state to progress to states separated by more than one edge in the graph. Because only the relative magnitude of the state-set certainty values is important, normalization constants can be ignored.

Our implementation of state-set progression for the competition is as follows: For each instruction given to the robot, an array of states such as the one shown in figure 3 is created, with the exception of Go Hallway End, which is handled without state-set progression. Five possible features and percepts are considered: (1) Hallway, (2) Open_Door, (3) Closed_Door, (4) Foyer, and (5) Wall. Figures 4 and 5 show how the information is stored in the state array.

For a (Turn|Enter) (First|Tenth) (Right|Left) instruction, only the location of the features (left or right) and the nature of the last opening are known.[3] Because no other information is available for the intermediate openings, they can have any of these three features: (1) Hallway, (2) Open_Door, or (3) Closed_Door. For a Go Past Foyer instruction, the location of the foyer is unknown, and both sides need to be monitored. Also, all other features are possible before and after the foyer node. A slight modification of the state-set progression algorithm is necessary because several possible features are stored in the intermediate nodes. Equation 2 is then rewritten as

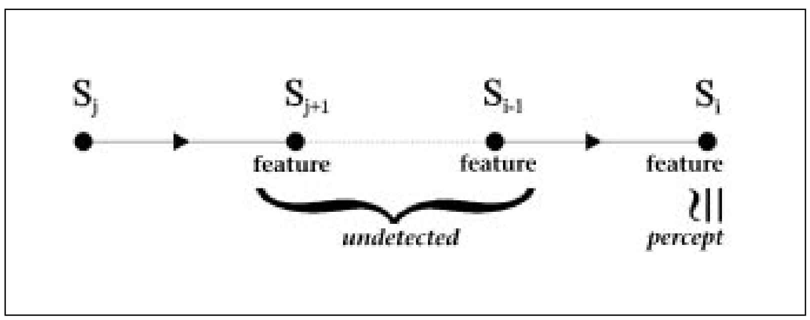

Figure 3. A Path between Two States.

$$C\left(S_j \xrightarrow{path} S_i \middle| percept\right) = \left[\sum_{m=1}^{\#\,features\,in\,S_i} C\left(feature_{m,S_i} \middle| percept\right)\right] \cdot \prod_{j<k<i} \left[\sum_{m=1}^{\#\,features\,in\,S_k} C\left(feature_{m,S_k} \middle| no\_percept\right)\right] \quad (3)$$

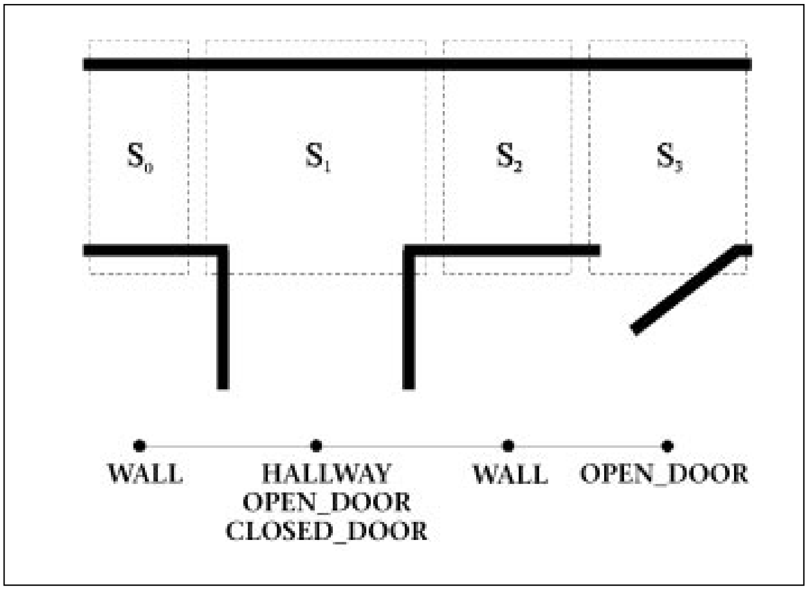

Figure 4. Enter Second Right.

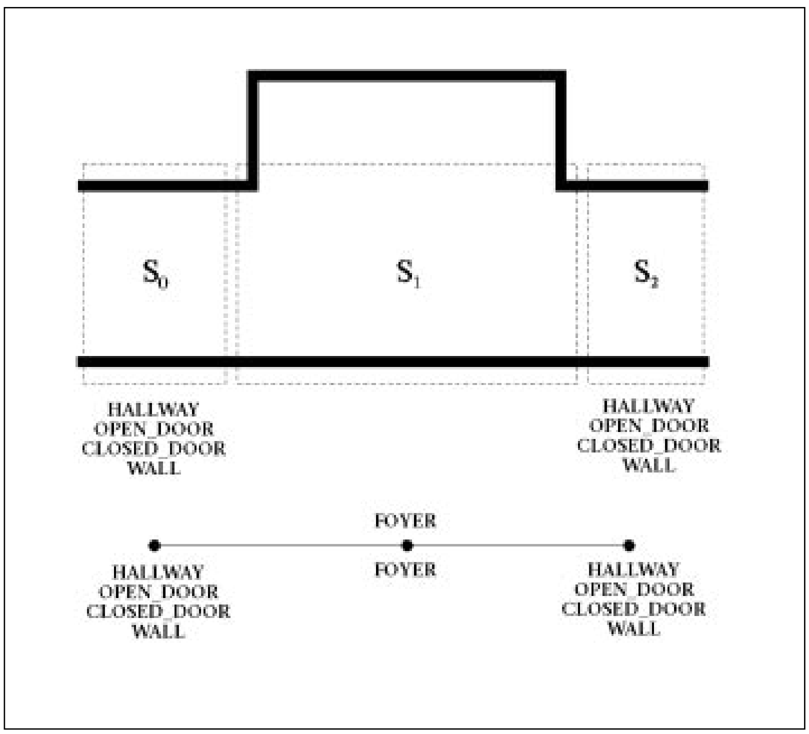

Figure 5. Go Past Foyer.

The uncertainty in the intermediate nodes blurs the certainty distribution in the state set, limiting the localization capabilities of the probabilistic inference. There is simply not enough discriminative information in the nodes to extract useful conclusions from the perceptions. As an example, if the robot reaches the end of a hallway before having detected the desired number of openings, there is no way to tell from the information stored in the state set if this circumstance is the result of a faulty instruction or an error in feature detection (an undetected feature). Under these circumstances, the state-set progression algorithm can only be used as a sophisticated feature counter.

Feature Detection

The *feature-detection module* provides the abstract percepts to be fed into the state-set progression algorithm. Our initial implementation of this module was based exclusively on sonar, which has been shown to suffice if a topological map of the environment is available (Nourbakhsh, Powers, and Birchfield 1995). To improve robustness, vision was added in the last week before the competition.

Sonar

The sonar feature detector utilizes just two transducers of the sonar ring (one on each side) to build a contour line on each side of the robot. To determine which pair of transducers has to be used (ideally those most aligned with the side walls), the robot must keep a decent orientation estimate with respect to the hallway. With reasonably well-behaved surfaces, a half-cone detection of 15 degrees can be assumed for each transducer. This value determines the maximum error in orientation estimate before the feature detector starts looking at the wrong pair of transducers. Figure 6 shows an experimental contour line extracted with this simple strategy.

Because of the errors in orientation estimation and dead reckoning, the scan lines tend to drift over time (in other words, they are not absolutely parallel to the *x* axis in figure 6). This circumstance is taken into account and corrected by periodically updating the orientation estimate and the side references (distance to the side walls).

Each sonar reading, once projected on the contour line, is compared to a threshold to determine if it corresponds to an opening, an indentation (from a closed door), a wall, or an obstacle. Two monitor flags on each side

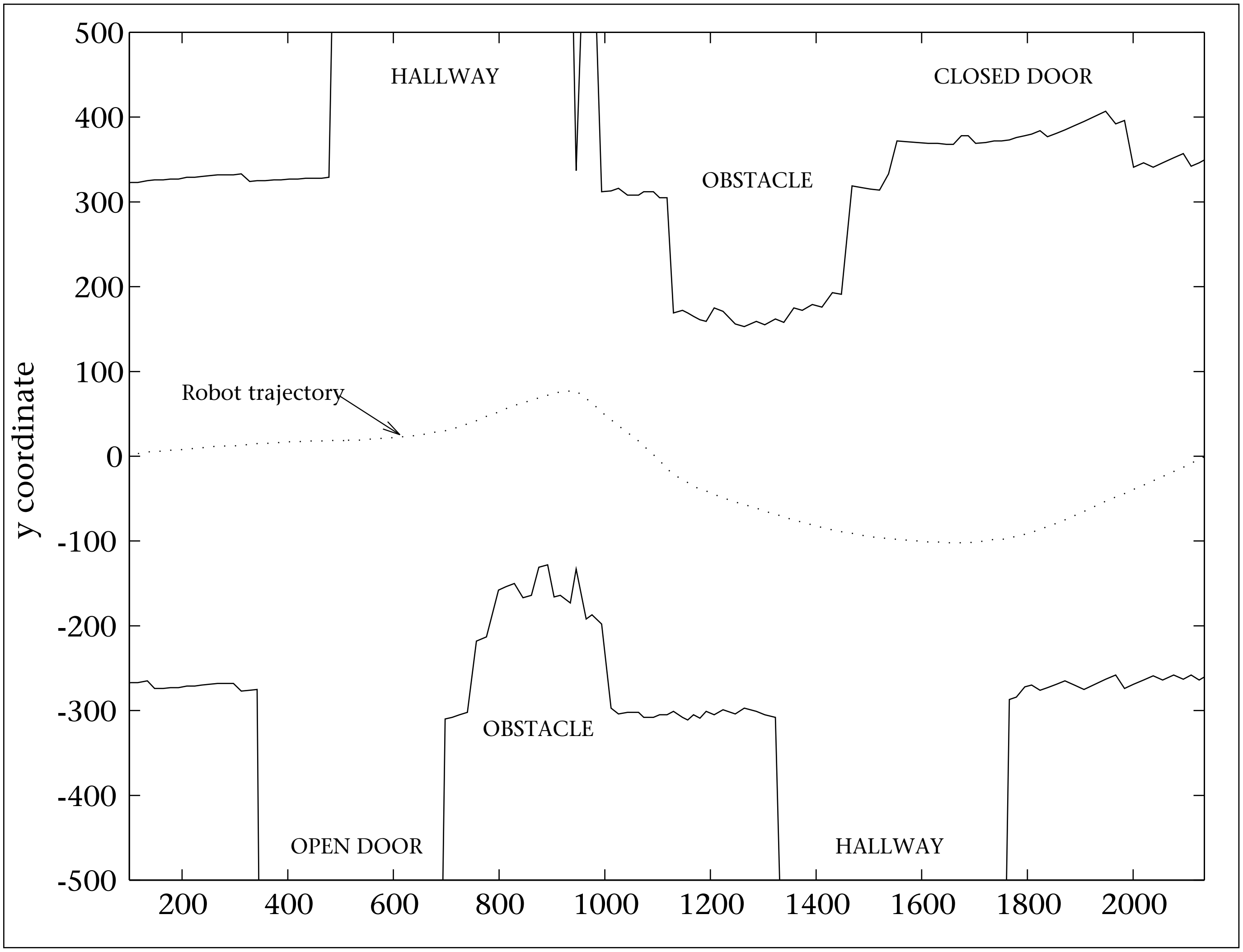

Figure 6. Experimental Sonar Contour Line.

track openings—one for foyers, hallways, and open doors and one for closed doors. When the start of an opening is detected, the corresponding monitor flag is activated. After the opening has been passed and the transducers return echoes from the wall, the monitor flag is disabled, and the width of the opening is compared to a threshold to determine if the percept was a false alarm, a doorway, a hallway, or a foyer.

This strategy was tested in our laboratory, in an artificial office environment made of cardboard partitions. With this type of surface, our feature-detection algorithm showed good results in general. However, we could not assume similar performance in the competition arena.[4] In particular, closed-door detection posed a potential source for problems. With state-set progression seriously handicapped because of the lack of a complete topological map, we were forced to guarantee 100-percent feature detection.

Vision

To improve the robustness in feature detection, we decided to incorporate vision in the last week before the competition. The color back-projection algorithm our teammate Rich LeGrand had implemented for the Office Cleanup event had been giving good results on LOLA for a while. Porting it to the Office Delivery event was trivial,[5] and it increased performance tremendously. The solution we adopted was to place a fluorescent marker on the floor, next to every doorway, as shown in figure 7.

To speed the color back-projection algorithm, only one-third of the image is pro-

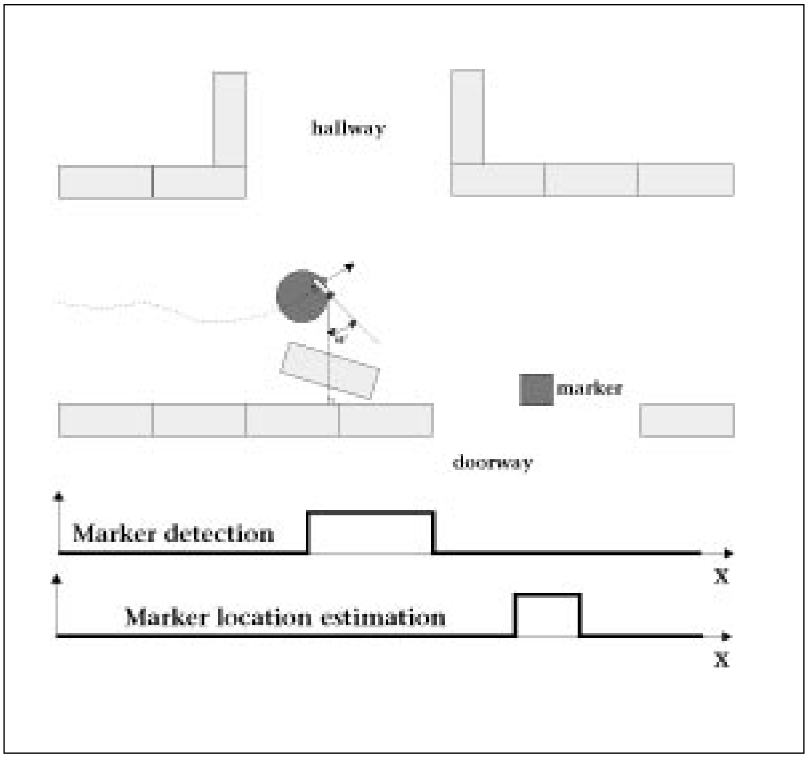

Figure 7. Visual Marker Setup.

	sonar	
	opening	**indentation**
marker detected	*open door*	*closed door*
no marker	*hallway/foyer*	*closed door*[2]

1. Discrimination between these two percepts is done by comparing their width to a threshold.
2. For the competition runs, this option was disabled, and closed-door detection was based solely on vision.

Table 1. Merging Sonar and Vision.

cessed. The tilt axis on LOLA's pan-tilt unit is fixed to the position used in event 2 to be able to use the same calibration lookup table. A simple proportional derivative compensator on the pan axis keeps the camera at the right angle when the robot steers away from obstacles.

Because the vision system detects the markers before the robot has reached the doorway, each marker detection is stored in an array and merged with the sonar features when the robot is next to the marker. Table 1 shows the possible combinations of sonar and vision percepts.

The performance of the feature-detector module with this addition was excellent, with 100-percent success in detection and recognition in a variety of test runs and throughout the competition.

Low-Level–Motion Routines

Within the framework of topological navigation, these routines are control strategies that specify how the robot should follow the edges connecting two nodes (Kuipers and Byun 1991). The geometric characteristics of the environment determine which routine should be used at each time. The three basic routines are (1) direction-follow, (2) wall-follow, and (3) goal-seek. *Direction-follow* is usually applied to move along corridors, *wall-follow* is the safest strategy when trying to traverse wide-open areas, and *goal-seek* is best suited for approaching a certain remote landmark.

Shipped with every Nomad mobile robot is a motion library that includes these three routines. The advantage of these routines is that they have built-in obstacle avoidance, isolating the programmer from the low-level control of the vehicle. Interaction with high-level processes, which is a common problem with most obstacle-avoidance algorithms (with the exception of navigational templates) (Kortenkamp 1995), is implicit in this approach because (1) the low-level motion routines have specific behaviors and (2) it is the responsibility of the high-level modules to activate the appropriate routine.

For the competition, we used direction-follow to traverse hallways and goal-seek to position the robot before making turns into hallways and the goal room. Wall-follow, which had been widely used to exit the start room in previous contests, was not used on LOLA.

The *exit-room routine* can be viewed as another low-level motion routine because it determines a strategy to traverse the link from the initial node in the topological map (defined as the robot being inside the start room) to the next node (defined as the robot being in the first hallway, just outside the room).

LOLA uses a combination of sonar and vision to exit the room. The position of the doorway can easily be obtained from the fluorescent marker placed directly outside the room. Before searching for the marker, the robot moves to an open area in the room using sonar. This step is important to obtain a good field of view for the camera and ensure a decent angle of attack when pursuing the doorway. Once the robot has found a location far enough from obstacles and walls, it performs a visual scan until the marker is found. Then, the goal-seek routine is called and positions the robot on the marker. Finally, the robot makes a sonar scan to obtain a rough orientation estimate from the door frame and makes the final move into the hallway.

Registration

The last module in our navigation architecture is *registration,* which determines the direction of the hallways and the orientation of the robot (Kortenkamp et al. 1994). This information is critical for two reasons: First, the direction-follow routine requires information concerning the axis of the hallways. Second, and more important, the sonar feature detector must know which transducers are facing the side walls, and the pan-tilt unit needs to track one of the sides of the hallway at a specific angle.

We perform registration using sonar. While it navigates along a hallway, the robot continuously maps the sonar readings into a local reference frame and fits line segments to the resulting points using least mean squares estimation. The number of sonar frames and, therefore, the distance that must be traveled along a hallway to get a good orientation estimate (let's say within $\pm 10^{o}$) depend on the surface of the walls. As an example, with the use of cardboard partitions, the robot needs to move no more than a few inches to obtain the estimate. For smoother surfaces, this distance can increase as much as a few meters.

The first registration run is performed after exiting the start room. At this point, the robot is located in the hallway and has turned left or right according to the first instruction. The rough orientation estimate extracted from the door frame allows the robot to follow the hallway and collect points for registration. During this period, feature detection is disabled because this orientation estimate is not accurate enough. Once the line-fitting algorithm returns with the first orientation estimate, if the robot has traveled more than a specified distance (half the width of a hallway), it must return to the initial position in the hallway in case it passed any features. Once this preliminary registration is performed, the feature-detection module is activated, and the robot starts executing the instructions that will take it to the goal room. Subsequent orientation updates improve the accuracy of previous estimates and correct the errors in dead reckoning.

The Real Thing

After three intense months of design, programming, and testing, we finally had a working system. If we had made the correct assumptions about both arena and rules, we would only have to tune up some parameters to optimize the robot's performance in the competition arena. If we had not, who knows? However, the big moment had come, the real thing, the competition.

After a long 18-hour drive from Raleigh, North Carolina, we arrived in Montreal. Our first contact with the arena did not offer many surprises. The environment layout for the competition was the same as the sample one the organization had offered to the teams in the previous months. Also, the problems experienced in previous years with the partition's sonar specularity had been eliminated by choosing cardboard and textured plastic surfaces. Finally, the convention center offered uniform lighting conditions and a solid gray floor, excellent for the color back-projection algorithm. We realized then that the assumptions we had made were mostly correct. We had to spend some time tuning up the exit-room routine because obstacle avoidance would not let the robot cross the narrow doorway leading to the hallways. For a couple of days, we even used the same histogram models for the doorway markers we had used in our laboratory under different visual conditions, but we finally made new ones for the preliminary rounds. Otherwise, LOLA seemed to have fun navigating through the hallways, avoiding whatever obstacles it found on the way.

During the preliminary rounds, we had a chance to compare our system with those of the other competitors. LOLA was the only robot that used vision and artificial markers to navigate through the hallways. The performance of the other robots was similar, but LOLA was at a disadvantage (because of marker penalties) except in cases where the complexity of the environment makes sonar feature detection unreliable. The preliminary rounds demonstrated that we had built a robust system that could easily handle cluttered environments with static obstacles and humans. Unfortunately, they also proved unnecessary the use of visual markers in the absence of obstacles in the hallways. Rather than disabling visual feature detection for the final round—thus eliminating the penalties—we decided to increase the maximum speed of the robot by 50 percent and have some fun with it. We could afford playing with LOLA on its way through the arena, why not do it then? After all, we were there to share a good time with the other teams and learn from the experience. All the robots showed their best in the final round, and we reached a second place that tasted like victory.

Acknowledgments

We would like to thank teammate Rich LeGrand, who worked with LOLA on the Office Cleanup event, for his many good suggestions and his help with the vision system as well as for being a real buddy during the preparation for the competition. The American Association for Artificial Intelligence is also greatly acknowledged for the generous financial support that helped the North Carolina State team travel to Montreal. We thank Jim and John Slater, Philippe Pignon, and Keith Mason, from Nomadic Technologies, for their help and support during the inevitable last-minute debugging and tuning up in Montreal. Finally, we would like to express my sincere appreciation to David Hinkle, David Kortenkamp, David Miller, and many others involved with IJCAI-95 who helped organize and coordinate this competition.

Notes

1. It is possible for these certainty values to be greater than 1.0. For this reason, the term *probability* should not be used in this discussion.

2. In the Bayesian sense, this is the a posteriori conditional certainty, which can be expressed as $C(f|p) = C(p|f)\,C(f)/C(p)$. $C(p|f)$ is estimated through experimental data. $C(f)$ is a uniform distribution when the map is unknown. $C(p)$ is also uniform for an unbiased feature detector. Therefore, $C(f|p) = \alpha\, C(p|f)$, where α is a constant that can be dropped because we are only interested in the relative magnitude of the state-set certainty values.

3. The door to the goal room would be opened, according to the rules.

4. We couldn't make this assumption especially after the experiences reported in previous years.

5. In fact, vision was added in one night.

References

Becker, C.; Salas, J.; Tokusei, K.; and Latombe, J. 1995. Reliable Navigation Using Landmarks. In Proceedings of the 1995 IEEE International Conference on Robotics and Automation, Volume 1, 401–406. Washington, D.C.: IEEE Computer Society.

Kortenkamp, D. 1995. Perception for Mobile Robot Navigation: A Survey of the State of the Art. In Proceedings of the NASA Dual-Use Space Technology Transfer Conference. Washington, D.C.: National Aeronautics and Space Administration.

Kortenkamp, D., and Weymouth, T. 1994. Topological Mapping for Mobile Robots Using a Combination of Sonar and Vision Sensing. In Proceedings of the Twelfth National Conference on Artificial Intelligence, 979–984. Menlo Park, Calif.: American Association for Artificial Intelligence.

Kortenkamp, D.; Huber, M.; Koss, F.; Lee, J.; Wu, A.; Belding, W.; and Rogers, S. 1994. Mobile Robot Exploration and Navigation of Indoor Spaces Using Sonar and Vision. In Proceedings of the AIAA/NASA Conference on Intelligent Robots in Field, Factory, Service and Space (CIRFFSS 1994). Washington, D.C.: National Aeronautics and Space Administration.

Kuipers, B., and Byun, Y. 1991. A Robot Exploration and Mapping Strategy Based on a Semantic Hierarchy of Spatial Representations. *Robotics and Autonomous Systems* 8:47–63.

Meng, M., and Kak, A. C. 1993. Mobile Robot Navigation Using Neural Networks and Nonmetrical Environment Models. *IEEE Control Systems* 13(5): 30–39.

Nourbakhsh, I.; Powers, R.; and Birchfield, S. 1995. DERVISH: An Office-Navigating Robot. *AI Magazine* 16(2): 53–60.

Russell, S., and Norvig, P. 1995. *Artificial Intelligence: A Modern Approach.* New York: Prentice Hall.

Ricardo Gutierrez-Osuna is a graduate student in the Department of Electrical and Computer Engineering at North Carolina State University. He received his B.S. in industrial engineering from the Polytechnic University of Madrid, Spain, in 1992 and his M.S. in computer engineering from North Carolina State University in 1995. His research interests include uncertain reasoning, mobile robot navigation, and computer vision.

Ren C. Luo is currently a professor in the Department of Electrical and Computer Engineering and the director of the Center for Robotics and Intelligent Machines at North Carolina State University in Raleigh. From 1992 to 1993, he was Toshiba Endowed Chair Professor in the Institute of Industrial Science at the University of Tokyo. He received his Diplom-Ing. and Ph.D. from Technische Universitaet Berlin in 1979 and 1982, respectively. Luo is also a fellow of the Institute of Electrical and Electronics Engineers.