Proceedings of the 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems Raleigh, NC July 7-10, 1992

Behavior-Based Intelligent Robot in Dynamic Indoor Environments

In So Kweon, Yoshinori Kuno
Research & Development Center
Toshiba Corporation
1, Komukai, Toshiba-Cho, Saiwai-Ku
Kawasaki, 210, Japan

Mutsumi Watanabe, Kazunori Onoguchi
Kansai Research Lab.
Toshiba Corporation
8-6, Motoyama-Minami-Cho, Higashinada-Ku
Kobe, 658, Japan

Abstract

We present a navigation system using multiple sensors for unknown and dynamic indoor environments. To achieve the robustness and flexibility of the mobile robot, we propose a new behavior-based architecture with three groups of clustered (reflexive, purposive, and adaptive) agents that realizes both efficiency in attaining the mission of the robot and robustness against the various kinds or failures that may occur in a dynamic environment.

Basic behaviors required for navigation, such as, avoiding obstacles, moving towards free space, and following targets, are redundantly developed as agents and combined in the behavior-based system architecture.

We demonstrate the capabilities of our system in unstructured real office environments, using an indoor mobile robot developed by Toshiba.

1 Introduction

Autonomous mobile robots have been extensively studied since the 1960's. Two control architectures have been proposed to date: functional decomposition, and behavior-based decomposition.

Robots with functional decomposition normally use a SMPA (Sense-Model-Plan-Act) framework. The SMPA approach consists of modules that each sense the world, build two or three dimensional models, and plan actions for the robot using the model in a sequential manner. All the modules must be complete and working before any action take place. Although several successful demonstrations have been made, (for example, [4, 11, 12]), a serious problem in this approach is reliability. If one module fails, either due to software or hardware bugs, the entire system will break down.

On the other hand, the behavior-based approach decomposes the system into individual modules, each of which is responsible for one behavior to be performed by the entire system [2, 3]. Each behavior contains a complete path, from sensing to action, and is executed in a completely parallel manner. The behavior-based approach is more reliable than the SMPA approach. Even if one module fails, other behaviors can still produce meaningful actions for the robot.

Although this decomposition greatly improves robustness in the face of sensing and processing failures, it tends to be inefficient from the viewpoint of achieving the mission of the robot. A behavior-based system with fixed priorities tends to adopt a lower-level behavior in a dynamic environment, which is inefficient from the viewpoint of achieving a mission.

Recently, some advanced behavior-based systems have been proposed to give a purposive capability to the robust reflexive/reactive configuration. Arkin [1] proposed the motor schema-based architecture. "Schema" is a methodology used to describe the interaction between perception and action. Each individual motor schema corresponds to a reflexive behavior. A dynamic network is utilized instead of a layered configuration. Noreils [10] presented a three level (functional, control, planning) configuration for indoor mobile robots. This system is more flexible than conventional subsumption systems because the functional level is programmed by the control level depending on the mission. Currently, the planning part comprises two rule-based modules: the general-planner and the path-planner. The planner is used to recover from failures occurred in navigation by applying prepared rules. Hartley [6] developed a prototype of a behavior-based airplane controller. To solve the "lack of modularity" problem, control (Inhibit, Suppress) connections between behavior modes are utilized. Other possible methods for the arbitration of behavior were also proposed in this paper, including priority lists and hysteresis.

In spite of several improvements in behavior-based architecture, these systems still seem to have a weakness when purposive behaviors to attain a mission are combined with reactive/reflexive behavior, especially from the viewpoint of failure recovery. Planning-only failure recovery or behavior arbitration by restricted priority list may negate the essential merits of behavior-based architecture, that is, robustness and flexibility. Our motivation is to fill the gap between reflexive/reactive behavior and purposive behavior in order to take advantage of behavior-based architecture.

We are developing a behavior-based mobile robot with three clustered behavior modes. Basic navigation behavior is developed as an independent agent, and directly connected to the Motion-Executor which determines the motion commands for robot control. The agents that provide the behavior are clustered into three groups: reflexive-level, purposive-level and adaptive-level. The adaptive-level group is updated to recover from failure of the purposive-level or deadlock situations. In a sense, the adaptive-level is ready to fill the gap between purposive behaviors and reflexive/reactive behaviors. An independent arbitration mechanism is prepared for each group. The reflexive-level group, which consists of multiple sensory agents, adopt the "subsumption-like" arbitration because its role is to maintain minimal safety of the robot. For the purposive-level group," mission-based arbitration" is adopted.

Multi-layered behaviors include the reflexive-level ***obstacle-avoider***, the adaptive-level ***free-space-explorer***, ***wall-follower*** and ***open-space-explorer***, and the purposive-level ***target-tracker*** and ***target-searcher***. Vision-based target tracking behaviors are complemented with sonar-based behaviors. For example, intersection tracking behavior is coupled with the sonar-based wall following behavior. Doorway tracking behavior is also associated with the sonar-based open-space-explorer. Such redundant behaviors using two different sensors can be very useful for detecting and recovering from failure of each behavior.

2 System Hardware

We have developed a compact cart-type robot system, BIRDIE (Behavior-based Intelligent Robot in Dynamic Indoor Environment), as shown in Figure 1. Eighteen ultrasonic ranging sensors are installed around the robot at intervals of 20 degrees, which corresponds to the view angle of each sensor. Passive infrared sensors are installed on the front of the robot to detect thermal obstacles, especially humans. Flexible belt-type touch sensors using electrically conducting rubber are attached to the lower part of the robot. A movable stereo camera platform is mounted on top of the robot. A flux-gate compass is located in the center of the robot to detect the self direction of the robot.

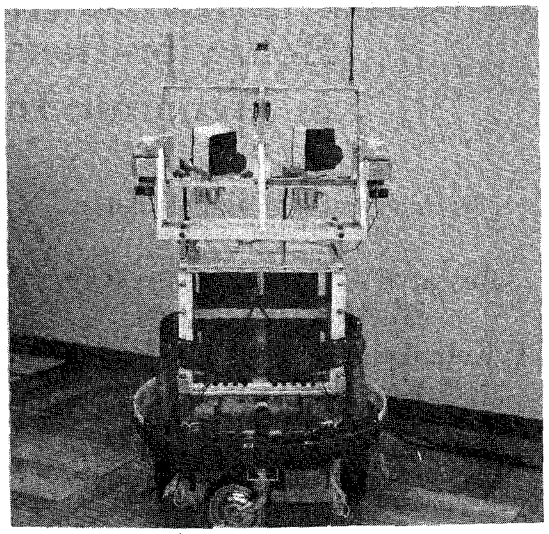
Figure 1: Mobile robot.

Currently, the implementation of the system is distributed in a computer network external to the robot and in an onboard CPU within the robot. The computer network consists of engineering workstations and the vision system. Sensor data(sonars, infrared, contact, compass) and motion command are communicated between the onboard CPU and the network through a wireless serial link. The onboard CPU controls the sensor driver circuits and motor drive circuits for the camera platform and the wheels.

3 System Architecture

3.1 System Design

We adopt a behavior-based architecture with three groups of clustered agents. Figure 2 shows the configuration of the system. The system is roughly separated into four parts: a group of reflexive-level agents, a group of purposive-level agents, a group of adaptive-level agents, and the Motion-Executor. The roles of the agents which belong to the reflexive-level group are to maintain minimal safety of the robot, that is, obstacle detection and collision avoidance. We adopt a multi-sensory configuration to avoid the danger caused by a failure in obstacle detection.

The roles of the agents belonging to the purposive-level group are to assist in navigating the robot to the final goal efficiently. In the current system, the navigation sequence to reach the goal is given as the behavior of specified agent pairs consisting of target-tracking and target-searching. The targets correspond to the objects which characterize subgoals and the final goal to be reached. Multiple Target-Tracker and Target-Searcher pairs are prepared according to possible subgoals and goals. The Target-Tracker provides a *move toward* behavior. The minimal necessary information for fixation to the target (that is, steering angle) is calculated by extracting the projected target region from image in camera-centered coordinates. The Target-Searcher provides a *rotate itself around the current position* behavior to search for a target.

The roles of the agents belonging to the adaptive-level group are to fill the gap between the purposive-level and the reflexive-level: that is, to recover from failure of the purposive-level agent and deadlock situations. A high-performance wandering capability can be realized by activating these agents. Four agents using ultrasonic ranging sensors are now in place: an Open-Space-Explorer, a Wall-Follower, an Obstacle-Boundary-Follower and a Free-Space-Explorer. The Open-Space-Explorer provides a *move toward open space* behavior. "Open space" means a place with a broad view: for example, an entrance. The Wall-Follower provides a *walk along the boundary of an obstacle* behavior. The Free-Space-Explorer provides a *go in the safe direction* behavior. The "safe direction" means the direction with the greatest distance between the robot and the observed boundary of an object.

The role of the Motion-Executor is to create a motion command which controls the motor drive circuit of the robot by selecting an appropriate behavior according to the global mission and the situation. An additional agent is also ready to detect the self-direction of the robot.

3.2 System Implementation

To be a useful system in a complex indoor environment, the robot must be able to carry out a global mission by selecting an appropriate behavior in sequence among many available behaviors. The integration of multiple behaviors into such a coherent system raises three important issues: the representation of a global mission, behavior arbitration, and failure recovery.

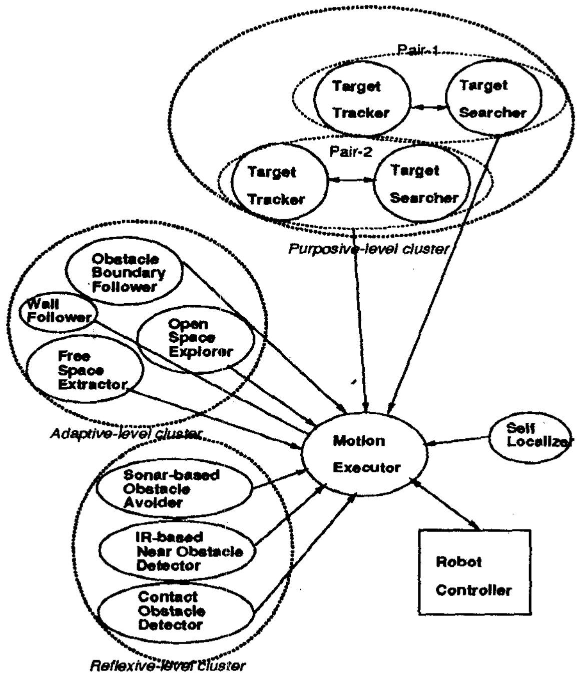

Figure 2: System configuration.

3.2.1 Mission representation

Any typical indoor environment, e.g., an office building can be represented by a combination of rooms, corridors, and intersections [5]. Based on this fact, we observe that the robot can demonstrate a goal-directed navigation capability only with three target tracking behaviors: doorway tracking, wall tracking, and intersection tracking behaviors. Therefore, we can represent a global navigation mission as a sequence of these three tracking behaviors. Currently, this sequence is generated by a human operator. Using if-then rules, a list of adaptive-level candidate behaviors is also generated for the recovery from the failure of each target tracking behavior. For example, when the robot is in the room and a doorway tracking behavior fails, the open-space-explorer is the best alternative behavior to get out of the room. If the open-space-explorer is not activated or fails, the wall-follower provides the behavior to get out of the room by following the walls of the room.

3.2.2 Behavior arbitration

The motion command is determined inside the Motion-Executor by selecting an appropriate behavior. Three arbitrators prepared for the reflexive-level, purposive-level, and adaptive-level groups work in parallel to determine the motion command. Although conventional systems adopt only one means of behavior arbitration, such as subsumption or map-based arbitration, our system uses two kinds of behavior arbitration: arbitration between different-level groups and that inside each group [13].

We use the fixed priority order between the three groups. First of all, a behavior from the reflexive-level group is preferred to other levels except for a deadlock situation because safety is the most important function for the survival of the robot and obstacles(especially, humans). Secondly, a behavior from purposive-level group is preferred while tracking or searching is successfully executed, to achieve efficient navigation as long as minimal safety is guaranteed. If a failure of the purposive-level behavior or a deadlock situation occurs, a behavior from the adaptive-level group is selected though the adaptive-level agents are working all the time to efficiently recover from failure.

Different behavior arbitration mechanisms for the reflexive-level, purposive-level, and adaptive-level work in parallel to determine the motion command. For the reflexive-level group, priority-based arbitration is adapted according to the reliability of each sensor. If the behavior from an agent with higher priority emerges, others from agents with lower priority are ignored.

For the purposive-level group, the priority of each paired target tracker and target searcher is determined according to the mission. An agents-pair list is produced from the behavior sequence. Currently, this stage is done interactively. Even if all the subgoals are not specified, our system still works well by using adaptive-level agents.

The Mission-Arbitrator picks up a behavior from multiple pairs following the agents list. When the robot has reached the current subgoals, the next agent pair is selected. When the current purposive-level pair fails, the Mission-Arbitrator sends a message to an Adaptive-Arbitrator to recover from failure by invoking a suitable adaptive behavior. The adaptive-level behavior can be invoked either when a target tracking behavior failed or a deadlock situation occurred. For the failure of the tracker, an appropriate behavior is selected from the list of adaptive behaviors of the current tracking behavior. Detecting the failure of the target tracking behavior is done by comparing the motion command from the target tracker with that from the best adaptive-level behavior. For the adaptive-level group, the Adaptive-Arbitrator determines an appropriate behavior following if-then rules.

Each agent provides the behavior as a vector(distance and direction) that the robot should move in the next period. If a selected adaptive agent cannot recover from the failure or its value becomes lower than the threshold, the next agent written in the list is selected. For example, while the robot is passing through a corridor, the Wall-Follower provides the proper(tangential) behavior. When the robot approaches an intersection, the behavior from the Wall-Follower becomes less reliable, and the behavior from the Open-Space-Explorer becomes more reliable. When the value of the Wall-Follower becomes lower than the threshold value, the Adaptive-Arbitrator switches the dominant agent to the Open-Space-Explorer that guides the robot to the center of the intersection. That is to say, the change in relationship among the behavior modes reflects the environmental situation. The Adaptive-Arbitrator utilizes this information to select the proper agent according to the situation.

3.2.3 Failure recovery

If no failure occurs in robot navigation, only a few behavior modes are needed to guide a robot to its goal. However, several failures frequently occur in the case of navigation in a dynamic environment. We treat two kinds of failures in the current system: failures in achieving the role of an agent (local failure) and failures in efficient navigation (global failure).

Local failures of this type occur during sensing or execution of processing. Conventional systems with functional decomposition lose robustness when this kind of failure happens. They are divided into two categories: detectable failures and undetectable failures. In the current system, only the failures in the purposive-level group are detectable ones.

Local failures in the purposive-level agents are generally detected inside each agent by comparing the predicted results with the real results. For example, a failure in target tracking in the Target-Tracker is noticed in the Target-Tracker itself by checking the detected size and position with previous results. For agents in the reflexive-level and the adaptive-level groups, there is no way to detect failures because they are caused by unexpected environmental changes. For the reflexive-level group, a redundant multi-sensory configuration maintains safety in the presence of failures.

Global failures occur when the dominant behavior does not coincide with the achievement of the global mission: that is, contradiction between individual behavior modes of agents and the global mission of the robot. Conventional subsumption architecture has no way to actively avoid this failure because no sophisticated arbitration mechanism is available. This failure is usually observed as a deadlock situation. Several complex deadlock situations occur in actual navigation. Our system handles two typical cases: deadlock stopping and deadlock motion sequencing such as moving forwards and backwards between two obstacles. Details are given in [13].

4 Range-Based Behaviors

In a behavior-based mobile robot, each behavior computes a motion command directly from sensor data without building any internal representation. Therefore, behaviors become more robust when multiple sensor data are used to compute motion commands. We have presented a motion decision method, based on active contour models, to compute the position and direction of the robot by fusing multiple sensor date [8]. This method computes the position and orientation of the robot by maintaining the equilibrium of all forces from multiple sensors. Forces acting on the robot include internal and external forces:

- As internal forces, we consider a smooth force, $\mathbf{f}_s$, from the smoothness constraints on the robot's trajectory and a damping force, $\mathbf{f}_d$, from the robot dynamics [1],
- The range forces, $\mathbf{f}_i$, from multiple range measurements, pushes the robot to the location where equilibrium is achieved between the range forces (the safest position), and
- Other external forces, $\mathbf{f}_t$ (e.g., the target forces) can be added to attract the robot.

Many mobile robot systems use acoustic sensors, because they provide an inexpensive means to obtain range information around the robot by computing the echo travel time. The ultrasonic ranging sensor has been also proven to be very effective for indoor navigation.

Using the energy-based motion decision algorithm and ultrasonic sensors, we have developed four range-based behaviors: free-space exploring, obstacle avoiding, obstacle-boundary following, and open-space exploring [9].

[1]We can easily add another external force due to acceleration of the robot

The free-space-explorer pushes the robot to the largest open space, which corresponds to the safest area for the robot. In other words, the greater sonar measurements provide the larger attractive forces on the robot. As a result, the robot moves to the location where equilibrium between range forces is achieved.

The obstacle-avoider only uses range measurements reflected from nearby obstacles, because any range measurements that are greater than a specified distance guarantees the robot safety. For the obstacle-avoider, the range forces from nearby obstacle act as repulsive forces on the robot. The range force is inversely proportional to the range measurement. In other words, nearby obstacles provide the greater repulsive forces.

The obstacle-boundary-follower uses an algorithm similar to the obstacle-avoider. In the case of the free-space-explorer and the obstacle-avoider, the range forces from nearby obstacles were either attractive or repulsive. For the obstacle-boundary-follower, the range measurements are available without any systematic error, such as specular reflection for sonar, the robot will never collide with obstacles, because it will always move along the tangential directions of nearby obstacles.

In indoor navigation, a robot often needs the open-space-explorer which can find an exit (e.g., a doorway) in a room by just using robust reflexive behaviors, without using any sophisticated vision systems or algorithms. The open-space-explorer is based on the free-space-explorer. The basic idea is to hypothesize an oblique wall, immediately behind the robot and to update the actual range measurements with the synthetic range readings reflected from this hypothesized wall. Then, with these updated range readings, a new robot location is computed by the algorithm of the free-space-explorer.

To determine whether the robot goes through a doorway or not, we use high-level heuristics about the environment as well as low-level sensor information. First, it computes the change of range readings from the left and right side sonars. When the robot passes through a doorway, large fluctuations in both the left and right sonar readings must be observed. When this large fluctuation is detected, it checks the existence of an intersection or a wall to confirm the completion of the behavior.

5 Vision-Based Behaviors

In most examples of indoor mobile robot navigation, one of the most important capabilities is to recognize some features, such as intersections and walls, and then track them to guide the robot - target tracking.

The target-tracker extracts two kinds of information from an input color image: a steering angle directing the robot towards a target and verification of the tracking accuracy [9]. The robot normally controls its steering, merely by using the steering angle computed from the tracker. However, this steering angle includes some errors, because of inaccurate motion of the robot. The tracking trajectory is used to verify whether the robot exactly moves towards the target or not. If the robot moves exactly toward the center between targets, the trajectories of these two targets on the image will diverge from the image center to the image boundary. Using these lines, the tracker checks whether the robot accurately follows the commands or not. When both targets disappear from an image frame at the same time, the tracker assumes that its mission has been completed. If only one target disappears, the tracker commands the robot to turn in the direction of this target, until the target is again visible. Then it repeats the same procedure until it accomplishes the tracking.

In this section, we describe two tracking behaviors: wall tracking and intersection tracking.

Part of an indoor environment can usually be described by sets of parallel 3-D lines (e.g., walls and corridors). In the image, they form a vanishing point with associated lines [7].

We can extract vanishing points by using the fact that a set of parallel lines in the world intersect at a vanishing point in the image. However, the image lines will never meet at one point because of image noise and the errors in localizing lines in the image. We represent the uncertainty in edge location using a 1-D Gaussian distribution along the direction perpendicular to each edge line. Therefore, the search for vanishing points can be easily accomplished by finding a small neighborhood in the image plane intersected by a sufficient number of straight lines. Since this vanishing point specifies the 3-D orientation of the parallel lines, a mobile robot can be aligned with the parallel lines using this vanishing point. The relation between the camera and world coordinate systems is specified by the image coordinates of this vanishing point [8]:

$$\mathbf{pv} = (x_v, y_v) = (f\frac{\tan\alpha}{\cos\theta}, f\tan\theta) \qquad (1)$$

where f is the focal length of the camera, and θ and α indicate the pan and tilt angle of the camera with respect to the world coordinate system. Therefore, the angle needed to align the robot with the wall - steering

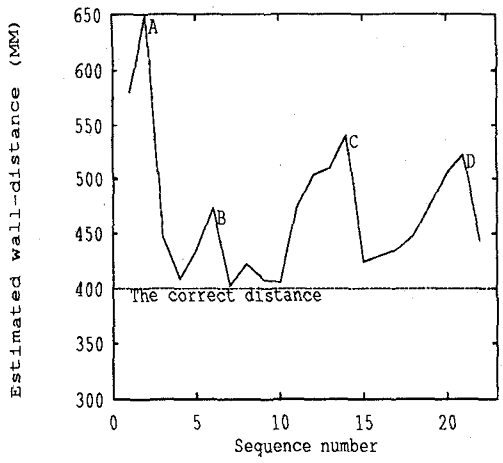

Figure 3: The computed distances from the wall by wall-following behavior.

angle - becomes:

$$\theta = \arctan\frac{y_v}{f}. \qquad (2)$$

We have done a series of navigation experiments in a real office environment. Using the extracted vanishing point, the wall-following behavior commanded the robot to move along the corridor. Figure 3 shows the correct distance (i.e., half of the corridor width) and the computed wall distances by wall-following behavior. When the computed distance shows a large discrepancy with respect to the correct distance (e.g., at the points A, B, C, and D in the figure), the behavior controls the robot to the center of the corridor. The robot navigated a distance of 10 meters along a narrow corridor without any failure in repeated trials.

In an indoor environment, when a behavior-based robot fails to accomplish its mission, such as finding a specific target, a better strategy is to find and move to a nearest intersection rather than wandering around.

In the present work, we use a simple heuristic - intersections exist at the end of walls - to extract an intersection. Vertical edges on the walls are candidates of intersections. The algorithm works in the following steps:

1. First compute a vanishing point;

2. Find two corresponding straight lines on the floor, which intersect at the vanishing point;

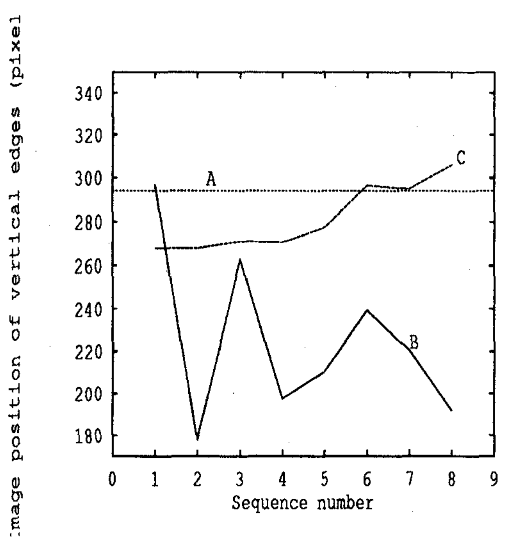

Figure 4: The average column value of two vertical edges, tracked by intersection tracking behavior.

3. Extract vertical lines on each straight line, located at approximately the same distance from the robot;

4. Track the two vertical lines by a real-time tracker.

The steering angle of the robot is determined by

$$\tan\theta = \frac{C_{avg} - C_i}{f} = \frac{\Delta C}{f} \tag{3}$$

where f indicates the focal length, C_i is the column value of the image center, and C_{avg} is the average column value of two vertical edges.

Figure 4 shows the computed average column value of two vertical edges from the experiments. The straight line A represents the column value of the correct image center. The curve B indicates the average column position of two vertical edges before the robot moves. The discrepancies between A and B, corresponding to ΔC, are used in the intersection-tracking behavior to control the robot. The curve C shows the resulting average column positions, corrected by the intersection-tracking behavior. Under ideal conditions, we should not observe any errors between the correct image center (the line A) and the corrected average column position of the two vertical edges (the curve C). However, when vertical edges are distant (e.g., 10 meters from the robot) in the beginning of the experiment, large errors (30 pixels) are observed. As the robot approaches the vertical edges, the error is reduced to less than 5 pixels.

6 Experimental Results

We have carried out a series of real navigation experiments in an unmodified office environment, using BIRDIE.

Missions for the robot are specified as a sequence of behaviors to be accomplished: (1) go through a doorway, (2) turn to the left after passing a doorway, and (3) go to an intersection by following a long flat wall on the left-hand-side. The motion executor monitors all the behaviors to confirm whether or not this sequence of behaviors is accomplished. Work for a more general mission level interface is still underway.

Figure 5 shows a diagram of an experiment in a real office environment. The robot should accomplish the mission, given as a sequence of behaviors, without colliding with any obstacles. In this experiment, five behaviors (obstacle-avoider, open-space-explorer, obstacle-boundary-follower, intersection-tracker, intersection-searcher) were working in parallel.

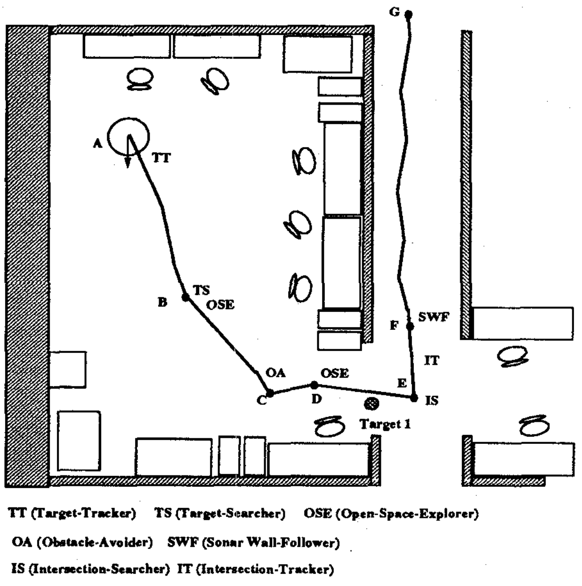
G
A
TT
B
TS
OSE
F
SWF
IT
OA
OSE
E
C
D
IS
Target 1
TT (Target-Tracker) TS (Target-Searcher) OSE (Open-Space-Explorer)
OA (Obstacle-Avoider) SWF (Sonar Wall-Follower)
IS (Intersection-Searcher) IT (Intersection-Tracker)

Figure 5: An experimental result from a real mobile robot navigation.

At starting position, A, the robot found and tracked a red target, *Target*1, located at a doorway to simulate the doorway tracking behavior. The sonar-based open-space-explorer confirmed the resulting motion command from the target-tracker. While the robot was approaching B, *Target*1 was removed and the target-tracker failed to track the target. The motion-executor immediately detected this failure from the discrepancy between the two redundant behaviors, the simulated doorway-tracker and the sonar-based open-space-explorer. Since the target could not be found by the target-searcher, the motion-executor used an actuator command computed from the most appropriate adaptive-level behavior, the open-space-explorer, to reach C. At C, however, the obstacle-avoider detected a static obstacle and commanded the robot to D. The motion-executor again used a command computed from the open-space-explorer, because the first goal, going through open space, was not yet accomplished. When the robot passed through the doorway, from D to E, the successful accomplishment of the first goal was confirmed. At E, the intersection-searcher used the vanishing point to extract a candidate of intersection on the left hand side of the robot. From E, two redundant behaviors, such as the intersection-tracker and the sonar-based obstacle-boundary-follower, were working in parallel. From E to F, however, the intersection-tracker commanded the robot since there was no wall for the sonar-based obstacle-boundary-follower to follow. From F, the two redundant behaviors commanded the robot to move along a wall until it reached an intersection G. The accomplishment of the second goal was confirmed when both targets disappear from an image frame at the same time.

7 Conclusions

We have demonstrated that a navigation system, consisting of a few basic behaviors, along with a proper behavior selection mechanism can provide sufficient navigation capability for a mobile robot working in indoor environments. The three-clustered behavior-based architecture with the redundant adaptive-level behaviors have improved the robustness and flexibility of the behavior-based system by providing an efficient mechanism to recover the failure of the purposive-level behaviors.

We have also demonstrated that range-based behaviors, using an efficient energy-based motion decision algorithm, are very robust and flexible for mobile robot navigation in unmodified office environment, through real world experiments. From our preliminary experimental results, we have also found that vision-based behaviors using vanishing points are robust against errors of preprocessing (e.g., edge detection) and image noise.

Work is currently underway to improve our mobile robot, BIRDIE, by using other sensors, such as infrared sensors, touch sensors, and a digital compass.

References

[1] R. C. Arkin. Motor schema-based mobile robot navigation. *IEEE Journal of Robotics and Automation*, 8, April 1988.

[2] R. A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2, April 1986.

[3] R. A. Brooks. Intelligence without reason. Technical report, Artificial Intelligence Lab., MIT, 1991.

[4] V. Graefe. Dynamic vision systems for autonomous mobile robots. In *IEEE/RSJ IROS'89*, 1989.

[5] M. Habib and S. Yuta. Efficient navigation system consistution for indoor autonomous mobile robot. In *International Symposium on Advanced Robot Technology*, 1991.

[6] R. Hartley and F. Pipitone. Experiments with the subsumption architecture. In *IEEE Robotics and Automation*, 1991.

[7] Kanatani. Reconstruction of consistent shape from inconsistent data. *Int. J. Computer Vision*, 3, March 1989.

[8] I. Kweon, Y. Kuno, M. Watanabe, and K. Onoguchi. Behavior-based mobile robot using active sensor fusion. In *IEEE Robotics and Automation*, 1992.

[9] I. Kweon, M. Watanabe, K. Onoguchi, and Y. Kuno. Behavior-based mobile robot using multiple sensors. In *First Korea-Japan Joint Conference on Computer Vision*, 1991.

[10] F. Noreils. From planning to execution monitoring control for indoor mobile robots. In *IEEE Robotics and Automation*, 1991.

[11] C. Thorpe, M. Hebert, T. Kanade, and S. Shafer. Vision and navigation for the Carnegie-Mellon Navlab. *IEEE PAMI*, 10, May 1988.

[12] M. Turk, D. Morgenthaler, K. Gremban, and M. Marra. Vits - a vision system for autonomous land vehicle navigation. *IEEE PAMI*, 10, May 1988.

[13] M. Watanabe, K. Onoguchi, I. Kweon, and Y. Kuno. Architecture of behavior-based mobile robot in dynamic environment. In *IEEE Robotics and Automation*, 1992.