THE UNIVERSITY OF MICHIGAN

-- Summary of Progress for 1991 --

DOE Robotics Project

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

MASTER

DOE/NE/37969--3

BIMONTHLY PROGRESS REPORT Covering Dec/Jan 1991
by
The University of Michigan

1. Significant Accomplishments

During the past two months, experiments using stereo vision for hallway navigation have been performed. The goal has been to use stereo vision to allow a mobile platform to judge distances to doorframes in a hallway. This information is then combined with sonar readings to allow the mobile platform to successfully navigate through the doorway. The integration of vision data is valuable since sonar information alone gives a very limited view of the world. In particular sonar information has limited range and angular resolution. In contrast, stereo vision gives a robot long-range sensing capabilities (from three to ten meters).

In these experiments, we chose to use our TRC Labmate robot (shown in Figure 1). The Labmate has sixteen sonar sensors in a partial ring. Mounted on top of the Labmate is a CCD camera on a pan/tilt/translate head. The pan/tilt/translate head is used to translate the camera in order to get the stereo pair of images. The images are digitized and analyzed on a Silicon Graphics workstation. Motor and sonar sensor control is provided by the same workstation.

The robot was placed in an ordinary hallway located in our laboratory. The robot took two images of the hallway, separated by a displacement of 180mm (the top two images in Figure 2). Vertical edges were extracted from the images (the bottom two images in Figure 2), which correspond to possible doorframes. The vertical edges in the two images were matched and a displacement was calculated. By knowing the focal length of the camera (f), the baseline (b) and the displacement (d), the distance (z) to each doorframe is calculated using the equation:

$$z = b * (f/d).$$

The results of the stereo vision algorithm are quite promising. For the pair of images shown in Figure 2 the following depths are calculated (scanning the images from left to right):

image feature	calculated depth	actual depth
front of drinking fountain	434cm	396cm
back of first door on left	627cm	610cm
last door on left	1254cm	1403cm
door at end of hall	1410cm	1586cm
third door on right	1128cm	1159cm
second door on right	868cm	930cm
front of first door on right (open door)	342cm	305cm

As might be expected, the depth values are more accurate for closer doors and less accurate for distant doors (Figure 3A shows a floorplan of the hallway and depth readings).

The next step in the experiment was to have the robot actually go through a door. We instructed the robot to enter the first door on the right of the hall (the open one in the images). The robot

used the depth information returned from its stereo algorithm to quickly proceed to the approximate location of the first door on its right. The robot then used its sonar to find the actual location of the door. After finding the front and back edges of the door (using sonar), the robot positioned itself in the center, turned ninety degrees and proceeded through the doorway (Figure 3B shows the robot's actions).

Much work remains to be done on the robustness of the door entering routines. The next step in this project is to create a library of robot actions (e.g., go-down-hall, enter-right-doorway, etc.) that are tested and robust. After this is done, more complex tasks, such as moving from one part of the facility to another should be possible. A long-term goal of this project will be for the robot to learn a map of its environment based on visual and sonar information, plus the actions that are to be taken at each point in the environment in order to achieve the robot's goals.

As was seen in the above experiments, autonomous navigation of mobile systems requires the instantaneous absolute position of the controlled vehicle. This can be achieved by fusing dead-reckoning with measurements of landmarks. the landmarks provide frequent absolute position measurements, and the dead-reckoning (encoders) provides the incremental position between successive absolute position measurements. Each of these measuring methods has errors. The accuracy of dead-reckoning is affected by the terrain (slippage, uneven surface, etc.), and by the vehicle (difference in wheel diameter, wheel misalignment, and unbalanced load on the vehicle). The accuracy of absolute position measurement with landmarks is limited by the resolution of the measurement system, and is also a function of the relative position of the vehicle and the landmark.

In order to improve the precision and reliability of navigation, we have developed four algorithms of absolute-incremental data fusion, and have evaluated them both theoretically and experimentally. The algorithms are described in the enclosed report. the best results, in terms of precision in achieving the target position, are provided by the method of minimizing the variance of the new estimate. The improvements on the position estimation depend on the particular location of the point at which the measurement is made. In the enclosed result, the improvements at two particular locations are 32% and 38% in simulation, and 21% and 33% in experiment, as shown in Figure below.

Comparisons of error from the real position for different position estimation algorithms: (A) computer simulation; (B) experiment. At each point, from left to right, the algorithms are based on (a) absolute measurement; (b) weighted average based on sensitivity; (c) least-squares method; and (d) minimizing the variance of the new estimate.

In our evaluation, we assume the vehicle has a balanced load and moves on a smooth floor in one direction (i.e., no slippage). While this gives us perfect measurements of the incremental system, this is not the case in practice. To remedy this deficiency, we are now examining a method to identify, in real time, the vehicle parameters that affect dead-reckoning. The envisioned identification scheme would use the steering and velocity commands, the encoder readings, and the landmark measurements as inputs, and would provide a compensation gain for each encoder in real time. Furthermore, by executing a sequence of designed tests, the system would also be able to identify floor conditions (in terms of slippage) and take them into account in the data fusion. The combined identification-fusion method could provide more accurate and reliable position information for autonomous navigation.

While these efforts are intended to yield more immediate results, we are also continuing to develop more advanced techniques useful for the coordination of multiple robots. Indeed, to perform tasks in complex environments, such as nuclear power plants, a cooperative efforts of many artificial agents may be necessary. A crucial hindrance to effective coordination, however, is that intelligent agents might not know enough about each other's intentions, abilities and perspectives to anticipate future interactions. Unless they are somehow initially designed with complete models of what the others know, the intelligent agents must rely on communication to share the knowledge that is vital for coordination. It is clear that the agents attempting to share all of the knowledge contained in their data bases with all of the others would result on an overwhelming burdens on resources needed for communication (time, communication bandwidth, etc.) and impede timely decisionmaking.

We have recently developed algorithms that can be used by agents to estimate the impact - or a utility - of any message on both the sender and receiver. Our method uses a Recursive Modelling Method (RMM) to explicitly compute the expected utility of a message. The utility of a message is defined in terms of the utilities of the physical actions available to the agents. A transmitted message might, for instance, inform other agents what the sender of the message intends to do. The message results in the changes in the models the sender has of the others and removes some of the uncertainty about what they will do, which can lead to the increased efficiency of interaction. Alternatively, in some situations, instead of broadcasting their intentions, agents may prefer sharing some important information about the environment that the others are unlikely to have due to their different viewing positions and sensing capabilities.

In the area of radiation imaging, we have tested the new spectrographic capabilities of the radiation camera. Images of multienergetic gamma ray fields were successfully performed, as shown in the attached paper submitted for the summer ANS national meeting. The integration of spectroscopy into the camera represents a major accomplishment in terms of camera capabilities. Our next task is to examine ways to display these images in meaningful ways.

2. Technical Reports and Major Correspondence

"Multichannel Spectrum Analysis for a Gamma Ray Imaging Camera", by D. Dolin, T. A. DeVol, and D. K. Wehe, submitted for the ANS Summer 1991 Meeting, Orlando, FL, June 1991.

3. Meetings

The University of Michigan has maintained a nearly continuous presence at ORNL for the past reporting period. Engineers have installed the navigation software needed for DEMO'90. Numerous problems were encountered, requiring strenuous efforts to overcome. As of this writing, the navigation software has been successfully installed, although some additional tuning of algorithm parameters remains. Major parts of the IGA algorithm (depth through motion) have been successfully installed, but one part remains uncompleted. The next trip to ORNL is scheduled for February 6-7, 1991 to complete this work.

The TRC Labmate mobile platform

Figure 1

Top: Stereo pair of images. Baseline = 180mm
Bottom: Vertical edges corresponding to door frames are extracted.

Figure 2

A. Depth readings returned by stereo.

B. Robot is instructed to enter first door on its right.

Figure 3

BIMONTHLY PROGRESS REPORT Covering Feb/Mar 1991
The University of Michigan

SIGNIFICANT ACCOMPLISHMENTS

Infrared Sensing

We have completed the assembly of all 24 infrared (IR) range sensors. As reported earlier, these sensors can distinguish 16 discrete ranges and are currently set up for a maximum range of 2 m. The IR sensors are based on triangulation, which makes them almost completely insensitive to the reflectivity and orientation of the reflecting surface. We have also completed the construction of the interface electronics. All IR sensors are now installed on CARMEL, located such that each IR sensor faces in the same direction as one of the ultrasonic (US) sensors.

The objective of our work in adding infrared sensors is to improve the accuracy of sensor-generated maps. More accurate maps improve the obstacle avoidance performance and can be used by higher-level planning algorithms for navigation. To accomplish this, we have developed an algorithm which combines data from ultrasonic (US) and infrared (IR) sensors, which we call pre-process verification (PPV). Unlike other sensor fusion algorithms, PPV evaluates the readings from the two sensing systems before they are used to update the histogram grid used for obstacle avoidance. This is possible for our architecture since corresponding sensor pairs are fired at the same time.

Theoretically, the readings of the two complementary sensors should be identical, as both "see" the same object at the same time. However, US and IR sensors have vastly different characteristics which introduce a different behavior in mapping the environment. Based on our experimental experience with both types of sensors, we have formulated a comprehensive set of heuristic rules that are applied to each pair of sensor readings. In practice, each pair of readings is categorized by the difference between the readings. Then, according to the category, a certainty value (CV) is assigned to a cell in the histogram grid.

At this time we have preliminary results from three different versions of PPV. These versions differ in the level of complexity of the comparison criteria. To compare and evaluate the correctness of the heuristic rules, we employ our previously developed Index of Performance (IP). The IP is physically significant; it is the weighed average of the distance between each filled cell in the histogram grid and the nearest actual obstacle boundary (hence, the physical dimension "cm"). The evaluation is performed by comparing the IP for a map obtained from both sensor systems (fused map) with the IP for the map obtained from using US sensor information only (sonar map), during the same run. A typical result yielded an IP of 9 cm from the fused map, as compared to an IP of 14 cm from the sonar map. This quantitative difference provides a measure of the improvement provided by adding the infrared sensors. However, even without the IP, one can observe that the fused map is smoother and of better definition than the sonar map (see Fig. 1). As expected, the PPV algorithm improves the mapping of objects that are difficult to detect by the US sensors, such as tilted or smooth surfaces. Although we have not yet fully implemented all the heuristic rules for PPV, we are encouraged by the early results from the algorithm. We expect to further improve its performance during the coming months.

One way in which the results from the infrared sensors can be improved is through improved spatial resolution. At present, we are limited to 16 discrete ranges from each IR sensor. We have developed a method which recognizes when the discrete range changes to the next step as the robot moves. The range is known quite accurately at this transition, and becomes a fiducial point. Range values between the transitions are determined by the Kalman Filter applied to the known velocity vector of the robot, in combination with any other range sensor data. The details of this method and the results of system tests are contained in a technical report also prepared during this reporting period.

.BIMONTHLY PROGRESS REPORT Covering Feb/Mar 1991 The University of Michigan

In addition to these tests, we have also been testing the infrared sensors in a very high gamma radiation field. We have placed a sensor close to the 10^4 Ci ^{60}Co source and observed no unusual behavior. We have currently irradiated this sensor to approximately 10^5 R without failure in fields of $>10^2$ R/hr, but are continuing irradiations until failure or fluences (an ANL-West suggestion) of 10^{6-7} R.

In addition to the ultrasonic and infrared range sensors now working on CARMEL, the Odetics Laser Range Scanner is also now working at the University of Michigan using an IBM PC AT and Matrox interface board. This is truly a significant accomplishment.

Navigation

Beyond the work at the reflexive obstacle avoidance level described above, we continue to develop increasingly complex methods of navigation which combine sensing and prior information. In this report, we describe progress in two areas of this research: system architecture and spatial mapping.

- With respect to the system architecture, we have further refined the structures of the architecture associated with the world model and the environment model. Also, we have started work on effective and efficient interconnections between these two models. We have implemented mechanisms for managing and accessing spatial knowledge: moving information from/to the Environment Model to/from the World Model, in the form of a context-based caching.
- With respect to spatial mapping we are developing a theory of the relations between past actions and spatial mapping that can be used as a bases for learning. We are in the process of designing representations of sequences of actions (a memory of past explorations of the environment) and of spatial information (a "learned" map of the environment) that we can use in experiments in adaptive navigation this summer.

Details of this work are described below. The attached figures are useful for following the discussion.

World/Environment Model

One of the key guidelines of the architecture is the support of both reactive and deliberate activities at all levels. This guideline is maintained by entertaining, at each layer of the system, external and internal input sources: external sources coming from the previous layer (lower layer in the case of the perception line) which contain a combination of the output of that layer and an evaluation of that information (also coming from the same layer), and internal sources coming from the expectations which are based on internal models. This guideline leads to a system which has a similar structure at different levels, and thus the emphasis of the design shifts to the types of 'flow lines' in the system and the types of junctions that should be supported. We now have completed an initial fine grain definition of the structure of these lines and connections.

We have presented earlier the ideas of Environment Models (EM) and World Models (WM) hierarchies. To review, the EM serves as a 'local' memory, where locality is temporal, spatial, and contextual. The WM contains ALL the information the system has, whereas the EM contains only that information which is necessary (or more precisely which is expected to be necessary). This model hierarchy supports 'active' storage elements that avail the processing elements with the necessary information. During this reporting period, we have implemented a mechanism for 'activating' information and for moving information, or contexts, into the local memory. This could be regarded as a context-based cache where there is a finite size faster memory module which

BIMONTHLY PROGRESS REPORT Covering Feb/Mar 1991
The University of Michigan

is what the processing elements of the system have access to. Unlike regular caches, the information here contains internal indexing to other contexts. A small number of contexts are active at a particular level at any given time, and a set of 'activation tests' determines whether the active context should be modified. On 'activation' the contexts to which the new active context is pointing to are loaded into the 'local memory' (EM), with their activation tests.

We have implemented this mechanism within the interface and simulation program previously described in an earlier progress report. We have demonstrated these ideas in the context of a robot navigating in a structured indoor environment as might be expected in an advanced nuclear reactor. Here, contexts are 'places', and the activation tests demonstrated are spatially and resolution based. This idea of context based caching and context activation could be used in addition to guaranteeing performance measures for some module (due to a restricted data size), also for performing self analysis of action and deductive learning by using the list of activations (activation history) as a memory of events. These activation 'points' are event landmarks and represent all the system needs to store in order to 'remember' a sequence of events (and internal states) through time. (Additional figures and videotape are available for more details).

Spatial Theory

Much of the research in robotics to date (such as the work with CARMEL) has dealt with getting a robot to move about in the environment without running into anything. While this work is crucial if we are to have truly autonomous robots, it does not go far enough. The next step is to develop a system that remembers where it has been (this results in a network-like map of those places in its environment) and uses its map to reason about its actions.

In our work, we are developing a theory of spatial mapping and navigation that will allow a robot to learn routes between places in its environment. Each route is represented as a set of triples, with each triple consisting of a goal location, a perception and an action. When the robot has a certain goal location (a target for action) and it sees a certain set of features then it takes a certain action that will lead it towards its goal location. A sequence of such triples (of goal, features, and actions) is one route through the environment. The robot learns routes by either exploring or by being guided along the route. Once the robot has been guided along a route, it should be able to follow that route by itself the next time. As the robot becomes more experienced in an environment, it develops a whole network of routes that it can use to navigate from place to place in that environment.

In addition to developing a network of routes, the robot should also be able to reason about its environment. That is, if a known route is blocked, the robot should be able to intelligently choose an alternative action that will lead it to its goal location. It should also, eventually, be able to jump from route to route, thus creating new routes. Obviously, when the robot has chosen an alternative action it can save that action and thus "teach itself" new routes.

As we develop this theory, and the representations and methods that it are implied by it, we will provide our robot platform (CARMEL) with new types of spatial reasoning. We will be able to move mobile robotics up the evolutionary ladder from "obstacle-avoidance" platforms to robots that learn about the spatial environment.

Radiation Imaging

During this past reporting period, the work in radiation imaging has focused on two areas: testing the spectral capability of the existing camera, and developing more sensitive detector elements. Additional images have been produced of Cs-137 sources placed between radioactively hot steel plates. Without using motion to improve the spatial resolution, we have found the camera

BIMONTHLY PROGRESS REPORT Covering Feb/Mar 1991
The University of Michigan

to be effective at resolving both spectral and spatial differences from distances of approximately 2 meters. We also accelerated our detector testing in preparation for fabrication of the second camera.

In order to be able to effectively form images in high radiation fields, the camera must be capable of operating effectively at both high and low count rates. The existing camera functions in moderate radiation fields with established detector materials. However, we have been actively pursuing BaF₂ as a new detector material to replace BGO and NaI₂. BaF₂ has the advantage of reasonably good spectral sensitivity, and has the potential to be the fastest detector for this type of work. Unfortunately, it is accompanied by a dominant (~80%) slow component which destroys its ability to function at high count rates. We are using the difference in wavelengths between the fast and dominant slow component to try to emphasize the fast component response and de-emphasize the slow component. During this reporting period, we have tested wavelengths shifters, including p-terphenyl, BM-terphenyl, and several others, in different solvents, including p-dioxane, coumarin and toluene. Using a spectrophotometer, we determined those combinations which enhance the fast component escaping the detector. Our best results during the past two months show equal fast and slow components, thus representing a factor of 4-5 improvement over the unshifted case. However, we are hopeful that even further improvements are possible using other combinations of fluors and solvents, and will continue work in this area.

In addition to the work for high count rates, we are also looking at improved spectral response. Our work in CsI has progressed to the point where we are routinely achieving 7% energy resolution, but believe it should be possible to reach as low as 4%. We have studied the reasons why CsI, when coupled with a PIN photodiode, is not yielding close to its best resolution. We believe the problem is due to either a situation-dependent ampliphonic sensitivity or the detector material does not follow the expected energy dependence for resolution based on Poisson statistics. While the latter would reveal some interesting new physics about this material, the former problem is correctable rather than an inherent limitation. If this coupling could be made to yield the expected excellent resolution, then we would have a detector element for the camera which is small, requires low voltage and current, and is extremely portable.

Thus, our work has continued on two fronts: we have continued testing of the prototype camera using its new inherent spectral sensitivity, and we continue our investigation of detector materials which be used in the second generation camera awaiting fabrication.

REPORTS AND MAJOR CORRESPONDENCE DURING REPORTING PERIOD

Borenstein, J. and Koren, Y., "The Vector Field Histogram Method for Fast Obstacle-Avoidance for Mobile Robots," was accepted for publication in the *IEEE Journal of Robotics and Automation*.

Borenstein, J. and Koren, Y., "Histogramic In-motion Mapping for Mobile Robot Obstacle Avoidance" was accepted for publication in the *IEEE Journal of Robotics and Automation*.

Fujii, Y., "Sub-Step Range Estimation using an Infrared Sensor with a Built-in 16-step A/D Converter in a Mobile Robot Application", UM Technical Report, March 1991.

The paper "A Decision Theoretic Approach to Coordinating Interactions" was accepted for presentation at the International Joint Conference on Artificial Intelligence, in Sidney, Australia in August 1991. Another paper on decision theory applied to exploration was accepted for presentation at AAAI-91.

***BIMONTHLY PROGRESS REPORT Covering Feb/Mar 1991
The University of Michigan***

Two papers on the radiation imaging work were recently published in Nuclear Instruments and Measurements. Another paper, highlighting the spectral sensitivity of the radiation camera, was accepted for presentation at the ANS Summer Meeting in Orlando, June 1991.

MEETINGS AND CORRESPONDENCE

The University has maintained a seemingly constant presence at ORNL in preparation for DEMO'90. In addition, University personnel attended the Albuquerque, NM meetings, and Board of Governors Meeting at ORNL during this reporting period. We have also communicated independently with ANL-West personnel who have expressed an interest in the radiation imaging work and radiation sensitivity tests for ultrasonic and infrared sensors.

Fusion Map

Sonar Map

Fig. 1

A view of the Processing module -- this is just a small portion of the architecture.

The environment model is a combination and an extension of the ideas of cache memories and of short term memories. Whenever new local information is brought in, not all of the presently local information is removed, thus supporting temporal connectivity.

A screen view of the Interface and Simulation Program

1. Initial environment

C (start) → D → E → G → H → J (goal)
 J (start) → H → G → E → F (goal)
 A (start) → B → C → D → E → F (goal)
 F (start) → E → G → H → J (goal)
 J (start) → H → C → D → E → F (goal)

2. Robot's experience in the environment reflected in routes between starting locations and goal locations

3. Robot constructs a network of routes

4. Robot deduces a route from H → A and generalizes the network of routes

Figures showing the construction of a route-based spatial map of an environment. The initial environment is shown at the top. The robot is guided through a series of routes from which it constructs a network. The robot then deduces a new route and generalizes its network.

1. SIGNIFICANT ACCOMPLISHMENTS

During this reporting period, Michigan has made significant progress in defining and implementing a method for correlating the environmental model (from sensor data) with the world model (from the a priori database). Since this represents a potentially major thrust, we present these concepts, in what we have defined as "Context-based Caching" in detail in this report.

A. CONTEXT-BASED CACHING (CbC)

The models a complex system should maintain include information about a work space, objects, properties of objects, relationships among them, events that can occur, and any other relevant information. The Context-based Caching (CbC) paradigm is strongly tied with the model hierarchy which dynamically secludes the local information from the rest of the information.

The main idea behind CbC is to restrict the accessible information to a fixed memory size in order to increase speed performance (for example if a search through the knowledge base is required), while guaranteeing that the necessary information is contained in this restricted local memory. As an example, consider listening to one of your favorite records -- you may not know the exact order of the musical pieces but at the end of one piece you can usually start humming the next piece. Evidently something at the end of the first piece caused the information about the following one to surface --this is what CbC will do. However, unlike the one dimensionality of a sequence of musical pieces through time, numerous connections can be supported. The goal of providing the necessary information is shared with the regular cache approach (see Fig. 1). However, in the traditional cache approach, information is judged on the single available dimension of memory location (as demonstrated in Fig. 2), whereas in this approach the relational dimensions are limitless. There is, of course, a trade off in that the information in CbC has to be properly constructed (see Fig. 3-4). This is the difference between knowledge and data, and CbC is a mechanism for managing knowledge.

Figure 1: Standard caching (secondary memory) scheme

Figure 2: Standard address based caching scheme

Figure 3: The Context-based Caching scheme

Figure 4: Context structures in the CbC approach

CbC: FORMAL DEFINITION

The Environment Model (EM) consists of a restricted size memory (either based on the number of contexts or the total memory size) containing all the local contexts, and their associated status:

$$EM = \{ C_i, S_i \}_{i=1..n}$$

The EM also maintains a smaller subset of contexts, called the *active contexts* AC , which represent the contexts which are currently being used. The AC could be regarded as the present goal, position, or complex state of the system.

The *context status* S_i represents the status of the particular context C_i within the local model (the EM). For example, it can indicate whether a context is active, has been active (similar to being used), the lapse of time since it has been used, etc. This information is necessary for the replacement scheme (i.e., when local contexts need to be switched).

A context C is a set:

$$C = \{ I, T, R, A \}$$

where:

- I = *information content*
- T = *activation test*
- R = *activation rule*
- A = *associated contexts*

The information content is a set of objects $\{O_i\}$. Objects can be divided into *declarative objects* and *procedural objects*. Declarative objects contain information about entities they could be defined as objects that could answer "what?" questions. Procedural objects contain information about operations they could be defined as objects answering "how?" questions. Each object contains a set of identifiers which indicate the 'type' of the object (which might itself be an object) and possibly the particular applicable procedural objects. The intention is to support any representation scheme both for procedural and declarative objects, thus, procedural knowledge as well maintains indicators to how it is to be interpreted. For example, if an operation is defined as a C code text, it needs to be compiled in order to be applied, etc.

The *activation test* T is a binary function: $T = f(s, AC) \Rightarrow \{T, F\}$ mapping the *state* of the system s and the active contexts AC to a binary space (*True* or *False*). The *state* s is a set of variables which are constantly modified by the systems operation. They signify the 'direction' in which the system is moving. A successful activation test (*True*) result, will initiate a *context switch*. In a *context switch*, the context whose *activation test* yielded a positive result will become active.

The *activation rule* R determines what method to use in the context switch. In particular, whether to replace one particular active context,

replace all active contexts, or just add an active context to the AC list. R could be constant for simple cases or could be a function as well: $R = f(s, AC)$.

The *associated contexts* A is a set of contexts to which a particular context C has connections to: $A = \{C_i\}$. Whenever a context switch occurs, the contexts associated with the newly active context are brought into the local memory (EM), replacing existing local contexts. The replacement scheme is a meta level operation, i.e., it is not determined by the knowledge or the state of the processing at this level, but is determined by an external strategy scheme independent of the present level. It could be a static operation or could be modified according to a set of criteria at this strategic meta level. LRU (least recently used) is an example of such a replacement scheme.

Associated with this scheme is a specific short term memory object called the *activation list*. The *activation list* maintains a list of 'temporal operational landmarks' via the temporally ordered list of active contexts (possibly with the *state* at the activation instant). This list provides a concentrated account of the processing history. It could be used for repeating operations, for reporting, or most importantly for retrospection and learning. Learning here should be seen in the context of the suggested data structures, i.e., forming new connections, modifying connections, adjusting activation tests and rules etc.

The World Model (WM) is a memory containing all the knowledge (in the form of contexts) the system has. Regular caching could be regarded as a (contextless) subset of CbC in which contexts are pages in the memory, information content (for the knowledge managing mechanism is disparate and word oriented, activation test is based on a single parameter in the *state* address request, and associated contexts are the next page (for a program) and all pages which might be referred to by the page. (The information on the pages might have a symbolic meaning which could be interpreted by some program, but the caching mechanism does not recognize any structures but words and pages.)

The advantage of CbC is in the fact that the structures being dealt with have a real knowledge content and are dealt with as such. Such structures support a level of operation where 'knowledge' objects are the medium of operation. This is in contrast with the regular caching scheme in which memory addresses and single computer words are the objects. Evidently these two caching schemes serve the same goal of making available potentially necessary information, however, these operations work on different levels. Whereas regular caching works on the computer hardware level and byte operations, CbC works in a knowledge level. It needs, nevertheless, to have implementation support from the levels below it (which is partially supplied by the sections above).

The tradeoff of the CbC mechanism is the same tradeoff off of adding any abstraction level to a system: objects have to be structured in a particular way at the level below. For example, introducing the symbol level (computer high level languages) on top of the assembly language level requires and forces a special structure of assembler. This requirement is entertained by interpreters and compilers. The gain is in being able to define operations in a higher level medium. The CbC is an example of such, it allows us to perform

expectation based analysis at the knowledge level. Evidently the complexity of these expectations is not easily represented directly at the lower levels.

CONTEXT ACTIVATION

In order to support the CbC approach, the whole database is constructed of 'contexts' as its atomic elements (more on which will be said next). Note that there is no restriction on the data structures for the different contexts. At every instant the system maintains a small set of 'active contexts'. These contexts are usually contexts which relate to the immediate spatial location or state of the system. The system also maintains a larger set of local contexts (the EM) which are either directly related to the active contexts or are contexts which are related to previously active contexts but have not yet been replaced. The size of the local context set which is fixed will determine the amount of these temporally resident contexts (which provide the temporal continuity to the past).

Each context in the database has associated with it a set of contexts to which it relates in some way (see figure 5). For example, for spatial contexts, such an associative relation can be based on spatial proximity, overlap, containment, connection, etc. A connection from a context *A* to context *B* identifies context *B* as being potentially relevant when context *A* is active. The

Figure 5: Context-Based Caching main data structures

connections themselves are not sufficient as they only provide a static relation. The dynamic characteristics of the system models is introduced by the switching of the active contexts. Therefore, associated with each context to context connection is an activation test T . If context A is active and an activation test T associated with the connection from context A to context B yields a positive result, then we have a *context switch*. The simple form of a context switch will be replacement: when B replaces A as the active context (in the example above). More complex activation schemes can be devised such as replacing all the active contexts (e.g., a new context take over), adding an active context (e.g., subtasking), or performing some initialization (e.g., an unexpected situation out of the scope of the existing internal indexing). These additional activation schemes can be specified within the activation tests.

When a context switch occurs, the contexts connected to the newly active context are brought into the local context set. Evidently these contexts have to replace existing contexts in the set. Standard methods from cache theory can be used to select the context to be replaced such as least recently used (LRU) etc. In our current implementation, we preferred a circular replacement scheme in which a pointer is constantly advanced through the local context set addresses. This scheme is close to the LRU approach but in fact implements a 'least recently made local' (LRML) approach regardless of whether a context has been used (activated) or not. The advantage of this approach over the LRU approach is that it does not require any additional attributes associated with the local elements nor does it generate any computational overhead. Nevertheless, the replacement scheme is an important issue which we have not fully addressed at the present. The information management system which checks the activation tests and performs their activation scheme upon a successful test can be entirely decoupled from the regular processing elements of the system or from the human user. Thus, the whole operation of context switching and of the changing of the local (available) contexts is potentially transparent to the users. It is important to notice that when a context switch occurs the newly active information is already available, thus no delay is introduced here. The information which is brought into the local set consists of contexts related to the newly activated context which are not expected to be immediately 'requested'. A situation similar to thrashing in standard cache memory will, however, occur if there are rapid context switches in a 'direction' out of the local set. Another type of 'miss' which needs to be supported is the case of unexpected situations, i.e., a situation for which the existing internal indexing have not been sufficient in anticipating the necessary information. In this case, an apparent delay will probably occur as the system will have to resort to the existing (non-context based) techniques.

Since the information management operation is working 'behind the curtains' the system has the illusion of temporal continuity despite the discrete context switching events. This is like eating the cake and still keeping it: on the one hand context switches are indeed discrete events which can be operated upon, stored for analysis or learning etc, yet on the other hand, from the perspective of the active elements (provided the associations have been properly defined) the necessary information is continuously available. The CbC approach could be regarded as providing a buffer between fast changing data and slower behaving components. Its role is analogous to a heavy flywheel which has to provide enough inertia to a physical system.

OBJECT ORIENTED MODELING

Our definitions of contexts follow the object oriented paradigm. Each element (context) has associated with it (in addition to the connections discussed earlier) an object identification slot. The object identification determines what types of operations can be performed on this object type, such as creating the data structure, maintaining it, graphically displaying it (if it is a physical object) and so forth. This allows us to support any type of data structure as a context. The trade off in not forcing a uniform representation is that for the diverse object types, multiple operations have to be defined. This is an acceptable price to pay for the ability to support any representation scheme. Moreover, using this approach, meta level knowledge such as object types or procedural knowledge could be placed in the same database. The object oriented paradigm has an additional advantage for the visual simulation in that symbolic attachments are available for the objects and their elements. Thus, specific portions could be highlighted on request, and more importantly in the case of sensor simulations and sensory data registration it lends itself for obtaining symbolic attachments for the sensory data itself.

SUMMARY. THE ADVANTAGES OF CbC

To summarize, the advantages of the Context based Caching are the following:

- Increased performance (speed).
- Possible guaranteed performance limits.
- Accommodating a potentially infinite data set. As only a limited size of information is resident in the local context sets at a given time, the amount of information which can be managed is practically limitless.
- Transparency from the information users.
- Accommodating any data type (modularity). The contexts and their sub parts are defined in the object oriented paradigm, and different operations can be attached to the object types depending on the required tasks. This allows us to borrow any representation scheme for new objects, as long as we attach the proper operations to the object types.
- Providing a mechanism for focusing processing attention. The CbC approach does not preclude procedural knowledge. It can thus be used both for focusing on the most prominent data, and in case there are procedural options (specified as contexts as well), it helps focus on the prominent procedures.
- Providing contextual and temporal continuity. One of the main problems in abstracting sensory information for the use by reasoning modules is that the rate of information changes is much higher for the sensory elements than for the reasoning elements (which need some relatively temporarily stable state to reason about). The discrete context changes provide exactly that: changes which are happening at a lower rate.
- Creating a discrete event history to be analyzed. A sequence of operations performed by the system could now be memorized in a compact form by storing only the *landmark events* which are the context switches.

COSIM

We have implemented our demonstration of the CbC paradigm within our Control and Simulation Interface program for Mobile Platforms (COSIM). This program allows us to simulate disparate platforms and different environments both indoors and outdoors (see figure 6). The main feature of COSIM are the sensor simulations and the provision for capturing real sensory data and registering the simulated data with it (see figure 7).

Figure 6: The Control and Interface Simulation program

Figure 7: Registering Sensory information with Expected Information

COSIM is implemented on the Silicon Graphics computer using primitive *gl* library functions. The user interface in COSIM is using the NASA Panel library 9.7 which supports a variety of buttons sliders and actuators. Mouse and keyboard based user interaction governs various window operations such as opening and closing the different possible windows, rotations, zooms, pans, as well as a large set of simulation control operations and robot control, and many more. COSIM is a powerful simulation tool which can be used for algorithm development and analysis. In COSIM various aspects can be investigated which cannot be tested with the real robots, or which might be too dangerous to experiment with in the real world. We can modify the sensor accuracy and look for the breaking point of the algorithms, change the physical parameters of the platform such as maximum speed, processing capabilities, and check their effect on the various algorithms etc. As a demonstration of its power, we have used it to analyze the VFH algorithm [1] for obstacle avoidance using sonar data in real time.

In addition to being a tool for experimenting with existing algorithms, the realistic sensors simulation capabilities lend themselves as a tool for first stage development of vision and perception algorithms, for which COSIM has already been used. As is the case with CbC, COSIM is a modular tool which can 'grow' upon request and need.

CbC IN THE SPATIAL MODELING DOMAIN

We used the CbC approach within COSIM to simulate and visualize graphically a structured work environment for a mobile platform. The program displays several views of the environment: a map view and a side view (see Fig. 8). The viewing positions can be either environment-centered (fixed position) or robot-centered (fixed relative to the robot). There is also a

Figure 8: COSIM overview

simulated camera view which displays a view which would be captured by a camera mounted on the robot. An actual Hero 2000 robot, similar to the one simulated, is in fact placed in the same position in the physical building. It can be controlled from COSIM and moved along with the simulated platform, and its sensory input (ultrasonic sensors and vision) can be compared and registered with the simulated sensors.

In the implementation of CbC in COSIM, we use places as the contexts. The particular places partially model the Advanced Technology Building which houses the AI lab at the University of Michigan. 'Building places' are objects which are comprised of a set of more primitive objects such as walls, doors, ceilings, floors, etc., each being constructed out of its own particular set of parameters as is specified by the object type. For example, 'door objects' have additional parameters for specifying the kind of door, the rotation axis position, and the opening angle. The robot can be moved through the building, and as its position changes, different portions of the environment are displayed in COSIM's windows. To demonstrate the ideas of 'active' and 'local' information, the display mode can be switched from showing just the active 'context' to showing the 'local' contexts, or even all the contexts (this is possible here because the information management process is not really decoupled from the rest of the system, nor is the primary memory -World Model- hidden from the system).

The definition of the actual context places and their connections is strongly tied with the task at hand. In the demonstrated implementation, the context places on the high resolution levels represent small portions of corridors or rooms which roughly correspond (order of magnitude) to sonar sensory range (see figure 9). On this level, intralevel connections between the contexts are based on spatial neighborhood, and the activation tests are based on spatial position thresholds and the directed orientation. At a lower resolution level, whole corridors are places, and yet at the lowest resolution, the whole building is a place. Intralevel connections remain spatially based,

Figure 9: An Active place (highlighted) and local places

however, interlevel connections (e.g. corridors to building) are based on immediate containment, i.e., the building place is connected to all the corridors whereas corridors are connected to the subcorridor places. The activation tests for these connections are based on spatial position of interest (mostly where the platform is), and on the size of the region of interest in the 'map window'. The larger the region of interest, the lower the resolution of the active place should be.

This implementation provides both a demonstration and explanation of the CbC paradigm, and at the same time it demonstrates the power of this approach for visualizing spatial structured environments in particular. Since we have a 'switch' in our implementation which allows us to turn on and off the availability of the local models versus all the information, we can examine the drastic change in speed performance when we switch from one mode to the other. In this implementation, the information management (activation tests etc.) could not be decoupled from the regular processing tasks and has been incorporated into the graphics loop. Despite that, the power of allowing access to only a fraction of the database at a given time was evident.

SENSORY INFORMATION REGISTRATION

In COSIM, we attempted to produce as realistic as possible sensor simulations. The types of sensors currently implemented are ultrasonic (sonar) sensors, dense range cameras, and grey scale or color cameras. The sensor models accommodate different parameters which help us approach a realistic sensory simulation. Our intention with COSIM is to provide a simulation environment in which the boundaries between real sensory data and simulated data disappear. By integrating the platform control loop within the simulation, we want to use sensory expectations (simulated sensors), with real data in order to help with the sensory analysis task. The object oriented paradigm will allow us to concentrate on parts of objects with their symbolic attachments in the simulated data, and match these elements to some real data features. The problem of bridging the gap between signal information (from the sensors) to symbolic information (high level abstractions) has been a central one for automatic perception and artificial intelligence. The scheme presented here introduces expectation based symbolic attachments into the sensory data which is a great leap forward in the processing of sensory information. Our approach in COSIM is motivated by the direction outlined in [2]. Simpler ideas of using model based expectations for sensory analysis with ultrasonic sensors are described in [3].

As these approaches are at their infancy, development of better algorithms will require visualization capabilities for the developers. COSIM provides the capability of visualizing sensor information, simulated sensor information, and any other desirable internal parameters (as was discussed earlier).

ONGOING RESEARCH EFFORTS AND OPEN QUESTIONS

There are numerous interesting research issues arising from this research effort. We are currently working on the sensory to expectation registration and hope to have positive results in the near future. Other interesting issues with respect to CbC are:

1. How to automate the database specification? The goal here would be to generate a context-based model (connections, activations, tests, etc.) from a given CAD model of a structured environment and a set of constraints (which will define the required granularity for the context places, the necessary associations, etc.).

2. How to dynamically modify the context based connections based on learned experience? For example, if in a particular situation the system finds out that two contexts tend to appear together, how does it provide both the mechanisms for detecting new associations and for recording them? A potentially difficult problem here would be how to automatically generate the activation tests once a connection has been established.

3. Another issue which was mentioned earlier is: How to properly replace local contexts? A good replacement scheme may add to the temporal stability of a system.

4. How does the complexity of the activation tests affect the performance of the whole system? At present, we have used very simple activation tests, however, there is no inherent restriction on the complexity of these tests. It is conceivable that more complex tests will be desirable. Therefore, it is important to assess when the complexity of the tests might over burden the system, and render the approach useless.

5. How should the system detect and deal with unexpected situations? At present, the default context is always the presently active one. However, it is desirable to introduce a 'null' test which will identify a necessary change of context but to an unknown context (not presently local). One could similarly think of a test to reaffirm the active contexts if no switch has occurred. This issue will become more important for non-structured environments or for highly dynamic ones where it cannot be expected to be able to anticipate all situations, and thus 'surprises' will occur.

Other interesting issues may involve computation and complexity issues with respect to the CbC paradigm.

B. ULTRASONIC SENSOR TESTING IN SEVERE RADIATION ENVIRONMENTS

Ultrasonic range sensors (URSSs) can be widely used in nuclear power plants and ER&WM applications, to provide real-time obstacle avoidance data both for mobile robots and for robotic arms.

In this section we present the results from experiments designed to study the behavior of URSSs inside a radiation environment. The purpose of

these experiments was to determine the level of gamma radiation at which an URS becomes inoperable.

The most significant result was that a commercially available ultrasonic sensor system would remain functional even after accumulating 10 Mrads (about 10 times the amount of radiation expected in an actual robotic application). This result was obtained after separating radiation-sensitive digital components from the other (analog) components. The digital components were placed outside of the radiation environment, connected through a 100 ft cable with the analog components and the actual ultrasonic transducer.

DETAILED DESCRIPTION

The URS system in our experiment comprised of a Polaroid electrostatic transducer and a Texas Instruments SN28827 Sonar Ranging Module. The Polaroid transducer is the one used by most robotics researchers for obstacle avoidance (including HERMIES-II and HERMIES III). The transducer consists of an ultrathin gold plated membrane about 1.5" in diameter that can be used to either transmit or receive ultrasonic sound waves. The SN28827 is an electronic circuit board with two custom designed analog and digital microchips as well as coils and other analog components.

Digital logic circuits are known to be sensitive to radiation. The sensitivity may vary from an erroneous operation (e.g., a logic '1' becomes logic '0') to total, unrecoverable failure of the circuit. Although the sensitivity to high energy neutrons may be greater, our application in nuclear facilities will primarily center in high gamma ray fields. Applications at facilities such as HFEF at ANL-W, a major part of the IFR program, also have fields which are dominated by intense fields of gamma rays.

To verify the susceptibility of the digital components to radiation, we operated the URS with all (analog and digital) components inside an intense gamma radiation environment and found that the sensor failed at a radiation dose of 90,000-95,000 rads, which was only 10% of the target radiation level. The failure was not permanent and the sensor recovered after two days.

To protect the digital components from radiation, we separated the electronic circuit into an analog and a digital module. The digital module was to be kept outside the radiation environment. The analog module board held the rest of the parts, which, as believed, were not affected by radiation. The two boards were connected by a 4-line, 100 ft unshielded cable. This configuration is shown in Fig. 10.

The new configuration worked well but we observed some signal loss along the long cable resulting in a wave propagation cone with a slightly narrower opening angle. This is only a minor decrement in performance, insignificant for typical applications.

Laboratory Setup

The experiments were conducted in a 3.5m x 3.5m x 3.5m room which accommodates the Cobalt-60 irradiator at the Phoenix Laboratory facilities of the University of Michigan, Michigan Memorial Phoenix Project. Dose rates were measured with Reuter-Stokes chamber model RS-C4-1606-207, serial number 2-8943, which is calibrated annually by the manufacturer against a National Bureau of Standards Source. The URS was placed at a short distance from the radiation source.

A styrofoam board was placed as an "object" in front of the URS and the distance between URS and object was measured manually for reference. A 100 ft long cable connected the on-site analog module with the remote digital module, located outside of the radiation zone. The digital module was interfaced to a computer which triggered the sensor and accepted a binary signal when the URS received an echo. The time between triggering and receiving an echo represents the time-of-flight of the ultrasonic sound waves, and can be converted into distance. The laboratory setup is shown in Figures 10-14.

Figure 10.

Figure 11.

Figure 12.

Figure 13.

Figure 14.

Design of the Experiments

The experiments were divided into three categories.

- 1) Experiments outside the radiation environment, BEFORE the URS was irradiated.
- 2) Experiments inside the radiation environment WHILE the sensor was being irradiated.
- 3) Experiments AFTER the sensor had been irradiated up to a certain amount of radiation.

The first category of experiments would provide a reference for the sensor behavior. The second category would provide the behavior of the URS within the expected environment. The third category would show whether the behavior of the sensor is affected by the accumulated radiation or by the radiation itself, i.e., by temporarily altering the operation of the electronics. We hypothesized that if the accumulated radiation affects the sensor, then the altered behavior would not be a function of the environment, as long as we had reached a certain level of radiation that would serve as an upper bound for affecting the sensor. If, on the other hand the behavior from cases 1 and 3 was the same and different from case 2, then this would mean that the radiation itself (the byproduct electrons of the gamma radiation) affect the sensor. The experiments consisted of repeated range readings to an obstacle which was placed at a known distance and orientation. The readings were taken every time the radiation would reach a specific level. The total accumulated radiation was on the order of 10 Mrads.

For a radiation level of 10.5 Mrads there was no significant range error. We stopped the experiments after a level of 10.5 Mrads was accumulated, and we do not know if the sensor would have failed at a higher level. However, a level of 10 Mrads is 10 times higher than the anticipated level of a "real-world" application of 1 Mrad. Above this dose level, we would expect other critical components to have failed. After accumulating 10.5 Mrads, we performed experiments outside the radiation environment and there was no change in the behavior of the sensor. This means that exposing the sensor to gamma radiation does not cause any permanent damage.

C. RADIATION SENSING AND IMAGING

This is a report of the progress that has been made in the development of a scintillation spectrometer for use in high radiation environments. Barium Fluoride (BaF₂) is the fastest known inorganic scintillator and has demonstrated radiation hardness up to 10^7 rad. One of the problems with BaF₂ is that only about 20% of the light output is fast (decay time of 0.6 ns). To make a suitable spectrometer for the application listed above one has to separate the fast light from the slow (decay time of 620 ns) light and do spectroscopy only with the fast light. At the University of Michigan, we have decided to separate the fast and slow emissions with organic compounds known as wavelength shifters. Most other methods that are currently developed to measure the emission spectrum of BaF₂ make no attempt to separate the fast and slow components. To alleviate the problem of pulse pile-up in high radiation fields, one would like to measure only the fast component of the BaF₂ light.

Over the past several months, we have worked on acquiring the equipment and expertise to test and evaluate the different wavelength shifters. There are three different experiments that are being run to evaluate the various wavelength shifters: molar absorptivity, emission wavelength, and ratio of fast to slow-fluorescent light from the wavelength shifter. The ratio of the amount of fast light to slow light for our crystal is 0.14. This was measured with a quartz end-window photomultiplier tube. Making the same measurement with a conventional PMT (wavelength cut-off ~300nm) this ratio would be 0.0 due to the fast emission peaking at 220 nm. By measuring the scintillation decay spectrum of the BaF2/wavelength shifter system with a conventional PMT we can increase this ratio from 0.0 to 0.16. Theoretically we have found an organic dye that is capable of a fast to slow ratio of 1.0, something which had never been achieved experimentally. Ultimately our goal is to achieve a fast to slow ratio of ~10.

By measuring only the fast component of the BaF2 light we expect to have an energy resolution of 25% at 662 keV. Although this is considered poor by most scientists, it still allows for crude energy spectroscopy in very high radiation fields, a capability which is currently unavailable. Achieving the fast to slow ratio proposed above would allow the BaF2/wavelength shifter system to be used in a field that is about 50 times more intense than possible with the BaF2 system.

Over the next progress reporting period we will be investigating how we can increase our currently obtainable fast to slow ratio of 0.16. To do this we will study Stern-Volmer and Forster kinetics of organic molecules and also consider if there is a geometric method (i.e., crystal and PMT geometry) that can be used to increase this ratio.

REPORTS AND MAJOR CORRESPONDENCE ISSUES

1. Koren, Y. and Borenstein, J., 1991, "Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation," was published in the Proceedings of the IEEE Conference on Robotics and Automation Sacramento, California, April 7-12, 1991.
2. Johann Borenstein and Yoram Koren were awarded US patent #5006988, entitled "Obstacle-Avoiding Navigation System" (with Simon Levine). This patent includes histogrammic in motion mapping (HIMM), the virtual force field (VFF), the vector field histogram (VFH), tele-autonomous operation, and 32 other claims.
3. Michigan contacted P. Bacca of ANL-W to followup on potential IFR-DOE/Robotics collaborations. Mr. Bacca noted two areas of interest: radiation imaging and IGRIP modelling. He noted that ANL-W would be interested in providing the testbeds for the radiation imaging cameras, although preferably after their upcoming DOE inspection.

REFERENCES

- [1] J. Borenstein and Y. Koren, "The vector field histogram - fast obstacle avoidance for mobile robots," *IEEE Journal of Robotics and Automation*, Revised 1990.
- [2] Ramesh Jain and Yuval Roth, "Towards integrated autonomous systems," In *International Symposium on Optical Engineering and Photonics in Aerospace Sensing, Proc., Applications of Artificial Intelligence IX, SPIE Vol. 1468*, Orlando, FL, pages 188-201, April 1991.
- [3] Y. Roth and T. Weymouth, "Environment model for mobile robots indoor navigation," Technical Report CSE-TR-53-90, The University of Michigan, Computer Science and Engineering Division, Artificial Intelligence laboratory, Ann Arbor, MI 48109-2110, April 1990.

SIGNIFICANT ACCOMPLISHMENTS

Summary

During the past two months, the Michigan group has focused its efforts on solving navigation problems. In this report, we describe a new method for improving the capabilities of ultrasonic sensors. The key idea is to modulate the timing of sensor firings to eliminate spurious noise and crosstalk. This should improve the reliability of these sensors and permit faster obstacle avoidance. At a higher level of intelligent control, progress has been achieved in building maps, without a priori knowledge of the environment, by registering depth cues obtained from visual maps. By manually driving the robot through the environment, the robot acquires these cues which enable to perform the task autonomously the next time. Sparse depth maps have also been acquired using a new algorithm which integrates vision, infrared, and wheel encoder data. The technique is totally automated and requires no apriori environmental knowledge. For many situations, an environmental map will be available. Progress has also been made in deriving the position correction factors to account for the disparity between the robot's actual view of the world and the view of the world from the robot's predicted position. The progress report concludes with a brief summary of other work in progress.

Error Eliminating Rapid Ultrasonic Firing for Mobile Robot Obstacle Avoidance

1. THE NOISE AND CROSSTALK PROBLEM OF ULTRASONIC SENSORS

We have developed error eliminating rapid ultrasonic firing (EERUF), a method that allows multiple ultrasonic sensors (URSSs) to be fired at five to ten times faster rates than those that are customary in typical mobile robot applications. Fast sampling improves the reliability and robustness of sensor based obstacle avoidance. At the same time, EERUF reduces by one to two orders of magnitude the number of erroneous readings due to environmental ultrasonic noise or due to the interference caused by other mobile robots (with ultrasonic sensors) that operate in the same environment. Most importantly, EERUF dramatically reduces mutual interference among a robot's own ultrasonic sensors, a phenomena known as crosstalk. With these unique features, EERUF allows multiple mobile robots to collaborate in the same environment, even if they use ultrasonic sensors operating at the same frequencies.

Two of the foremost (and well-known) limitations of ultrasonic sensors are their susceptibility to noise and their susceptibility to specular reflections. We can distinguish different sources of noise, namely environmental noise, noise from sensors on other

mobile robots, and crosstalk from onboard ultrasonic sensors. Specular reflections occur when soundwaves approach a surface not frontally ($\alpha = 0^\circ$) but at a larger angle (e.g., $\alpha > 30^\circ$), as shown in Fig. 1. If the surface is smooth, it acts like a mirror and reflects the incident soundwaves away from the sensor. Thus, the sensor does not receive an echo and consequently does not "see" the object.

Figure 1: A specular reflection from a smooth wall renders the object "invisible".

Many mobile robots have URSs installed on their periphery at 15° intervals, to guarantee complete coverage of the area around the robot in all directions. Another important reason for using densely spaced, multiple URSs (arranged in a circular or semi-circular formation around the robot) lies in the ability of this arrangement to partially overcome the problem of specular reflections. This is shown in Fig. 2a where a mobile robot approaches a smooth wall. When the wall first comes into the range of one of the sensors (here: sensor #3), the angle α is too large and the incident beam is reflected away from the sensor. Consequently, the robot does not detect the object and continues to travel in the original direction, toward the obstacle. However, moments later the object appears in range of sensor #4 and is detected, since the relative angle between sensor #4 and the wall is more favorable. Our experiments show that a sensor ring with 15° spacing between sensors will detect practically any object at any orientation, although the robot may have approached the object very closely before this happens. For this reason it is crucial to fire all sensors rapidly, so that an object is detected as soon as it comes in range of a suitable sensor.

Another reason for firing URSs rapidly is the detection of obstacles right in front of the robot. Clearly, there is a close relation between the firing rate of the sensors and the speed at which a mobile robot can travel safely. But since this relation depends strongly on the characteristics of objects in the environment, it is difficult to formulate this relation mathematically.

Figure 2: Multiple ultrasonic sensors, densely spaced around the robot, help overcome the problem of specular reflections.

The fastest possible way of firing multiple ultrasonic sensors is to immediately re-fire each sensor after it receives an echo or after a preset time-out. Theoretically, firing rates (i.e., the rate at which each sensor is fired at least once) are 30 - 60 ms. Existing applications don't employ this firing scheme because rapid firing generates significant amounts of noise and crosstalk. However, in combination with our error rejection method described below, this fast firing scheme has now become feasible.

In the design of EERUF, we investigated and classified the different types of ultrasonic noise that might interfere with the operation of ultrasonic sensors. Based on these results, we were able to design EERUF to reject noise from the environment, other ultrasonic sensors, and internal noise.

Environmental ultrasonic noise may occur near certain machine tools or possibly in the vicinity of discharging high pressure air or other fluid-flow installations. Possible sources are not well-investigated mostly because this kind of noise occurs only in rare instances. Theoretically, powerful, persistent ultrasonic noise can completely disable ultrasonic sensors (jamming), although we consider this a highly unlikely condition. More likely is a situation where a machine sweeps through the spectrum of ultrasonic frequencies during power-up or power-down, briefly covering the particular frequency range to which the URS is sensitive.

Environmental noise from other ultrasonic sensors is very likely to occur when more than one vehicle with ultrasonic sensors operate in the same environment (see Fig. 3). Typically, multiple vehicles would use the same kind of URSs, which, unfortunately, would practically guarantee mutual interference. In fact, the direct path in Fig. 3a results in a powerful disturbance.

sonof12b.dwg 4/25/91
 sonof12b.dwg 4/25/91

Figure 3: Ultrasonic noise generated by another mobile robot. Severe interference is most likely as both robots use URSs operating at the same frequencies.

Internal noise from onboard ultrasonic sensors or crosstalk represent the third major noise source. Figure 4 shows a mobile robot equipped with multiple URSs in two typical indoor environments; both environments differ substantially in the way they promote crosstalk. For the following discussion, we define the term "critical path" as any path of ultrasound waves that are transmitted by one sensor and are received by another, thus creating crosstalk. The sensor that transmitted the ultrasound waves is labeled x, and the receiving sensors are labeled y.

Figure 4a shows a direct critical path, where the robot is near a single wall. Because of the symmetry in Fig. 4a, two sensors are labeled 'y', since they are both on a critical path with sensor x. If sensors y were fired shortly after sensor x, they would be awaiting the echo to their own signals by the time the echo from sensor x reaches them. Thus, the reading from sensors y would result in some arbitrary error, depending on the time difference T_{lag} between firing sensors x and y.

The situation is more complex for the indirect critical path in Fig. 4b. Here, at an instance t_0 , sensor x is fired and its sound waves are reflected off three walls. Assuming the walls are fairly smooth (e.g., drywall), the reflected wavefront will reach sensor y after traveling through the distance $L=l_1+l_2+l_3+l_4$. If, at this time, sensor y is awaiting an echo of its own, then it will receive the signal from sensor x and interpret it as its own echo. Note that critical paths may also exist when the robot is traveling off the center line between corridor walls.

As is evident from Fig. 4b, crosstalk is not a phenomena that occurs only under very extreme conditions. In fact, crosstalk is a much more severe source of error than the sporadic misreadings from ultrasonic noise in the environment.

Once a critical path exists, crosstalk is a particularly damaging condition because it will continuously cause false readings in sensor y, until the robot moves out of the critical path situation. The following numeric example shows how serious this problem is.

Fig. 4b shows the conical propagation profile of a wavefront generated by sensor x. In Fig. 4b, this profile is only 5° wide (while the actual width is usually 15° to 30°), to consider only that portion of the wavefront that holds most of the ultrasonic energy (i.e., close to the acoustic axis). We make the reasonable assumption that any receiving sensor placed in the area of the reflected cone (i.e., after completing the roundtrip of length $L=l_1+l_2+l_3+l_4$) is a candidate for crosstalk.

For the symmetric case depicted in Fig. 4b it is clear that similar symmetric conditions prevail when the robot is a little closer or further from wall 2, as long as wall 2 lies between points a and b (located on the axes of symmetry, S). Therefore, the critical path of Fig. 4b exists while the robot travels through the distance D_c . For the geometric conditions in Fig. 4b, $D_c = 0.60$ m (approximately). Assuming a travel speed of $V = 0.5$ m/sec and a sampling rate of $T_p = 60$ ms for each sensor, sensors y will sample

$$n_c = \frac{D_c}{V T_p} = \frac{0.60}{0.5 \times 0.06} = 20$$

readings. All of these readings will differ only slightly from each other but all will be totally false, since they result from crosstalk. Any algorithm that relies on multiple samples to gain confidence in the measured location of obstacles will be misled to place very high confidence in the accuracy of these $n_c = 20$ recurring readings.

Figure 4: How crosstalk from onboard sensors is generated:
a. Direct critical path.; b. Indirect critical path.

Our experimental observations indicate that in an indoor environment, up to 20% of all sensor readings can result from crosstalk when firing at a (slow) rate of 120 ms. At faster firing rates, up to 90% of all readings can be in error, because of crosstalk.

2. ELIMINATING NOISE AND CROSSTALK WITH THE EERUF METHOD

In this section we introduce two methods for noise rejection. The first method, comparison of consecutive readings, is straight forward. However, as we will show, this method cannot reject crosstalk. To reject crosstalk, we introduce a modification, called comparison with alternating delays.

2.1 Comparison of Consecutive Readings

One simple method for eliminating occasional random noise is to compare two consecutive readings from the same sensor. The difference between any two consecutive readings, T_δ , is small if the readings result from "good" measurements (i.e., not caused by noise). One cannot assume $T_\delta = 0$ because of the robot's motion and the discrete resolution of the sensors. In the following discussion we will call consecutive readings that differ only by less than a small amount T_δ near-identical readings. However, if a reading was caused by random noise, it is highly unlikely to be near-identical to the previous reading, whether the previous reading was "good" or caused by noise, too. Thus, comparison of consecutive readings can identify false readings due to environmental noise and subsequently reject such readings.

While comparison of consecutive readings is an efficient way to reject false readings caused by random noise, it is unsuitable for reducing crosstalk. This is because crosstalk does not occur at random, as can be seen in Fig. 4. If, for example, sensor y was always fired a fixed period of time (T_{lag}) after sensor x, then sensor y would repeatedly produce near-identical erroneous readings T_{err} , until the spatial conditions in the environment have changed enough (due to the robot's motion) that y does not receive the signal from x (i.e., the critical path is interrupted).

To support this claim, we introduce the timing diagram in Fig. 5a. The first firing of sensor x after a critical path is established is labeled t_0 . Sensor y is fired some arbitrary time T_{lag} later. For the configuration given in Fig. 4b, the sound waves travel through the critical path L in the time $T_{L,n}$ (for the nth firing, with $n = 0, 1, \dots$), resulting in an erroneous reading $T_{err,n}$ in sensor y. Sensor x receives its "legitimate" echo after a period $T_{echo,n}$. We will assume that $T_{echo,n}$ and $T_{L,n}$ change only insignificantly between subsequent readings. As can be readily seen from Fig. 5a, subsequent erroneous readings $T_{err,n}$ are near-identical and can therefore not be rejected.

2.2 Comparison With Alternating Delays

To overcome this problem, we introduce an alternating delay T_{wait} , before each sensor is fired (Fig. 5c). For each sensor i, T_{wait} alternates between two values, a_i and b_i . a_i and b_i can be very small, on the order of a few milliseconds. Now, any sensor i can be re-fired immediately after receiving an echo, and after waiting for the short period of time, $T_{i,wait}$. After each firing, $T_{i,wait}$ is toggled between $T_{i,wait,a}$ and $T_{i,wait,b}$. Fig. 5c shows the timing diagram for comparison with alternating delays.

Figure 5: Timing diagram for asynchronous, immediate refiring.
a. No wait-times; b. Constant wait-times; c. Alternating wait-times

The first crosstalk reading produces an arbitrary erroneous reading $T_{err,0}$. However, subsequent errors alternate between two values, $T_{err,n}$ (even) and $T_{err,n}$ (odd). Thus, we achieve protection from crosstalk if we reject every reading that differs by more than T_δ from the preceding one. One should note that "good" readings are unaffected by this scheme, since echo-readings from actual obstacles are independent of the alternating delays and will be near-identical, differing at most by T_δ .

3. CONCLUSIONS

We have introduced a new method for identifying and rejecting erroneous readings due to noise and crosstalk. This method is based on the principle of comparison of consecutive readings, but, in addition, employs alternating wait states before firing each sensor. This latter measure artificially injects differences in between consecutive crosstalk readings, while leaving "good" readings unaltered. Experimental results show successful rejection of both direct and indirect path crosstalk (from onboard sensors), and errors caused by external sources. In crosstalk-promoting test-environments, the EERUF method consistently produced one to two orders of magnitude fewer errors than a "conservative" firing scheme (i.e., without error rejection) running at the same firing rate.

In summary, our new EERUF method will permit multiple mobile robots to operate in the same environment, without interference among their ultrasonic sensors. The reliability and robustness of single robot obstacle avoidance will improve considerably. Objects moving (e.g., people walking in the robot's path) at a higher speed can be avoided. Mobile robots will be able to safely traverse obstacle-cluttered environments 2 to 4 times faster than with today's state-of-the art methods (e.g., 2 to 3 m/sec vs. 0.8 m/sec).

Experiments with a scene-based, multi-level representation for mobile robot spatial mapping and navigation

We are in the process of implementing a scene-based representation that will give a mobile robot three levels of functionality. Each new level builds on the functionality of the lower levels and increases the spatial abilities of the robot. Higher levels of functionality emerge after repeated experience in an environment. Only after the robot has sufficient mastery of one level can it begin to function at the next level. The first level of functionality (and the most basic) is place recognition. Place recognition is the robot's ability to determine its position relative to other places in the environment. The second level of functionality is the ability to follow routes. A route is a sequence of actions that lead the robot from a starting location to a goal location. The robot is initially guided through the environment and shown routes from place to place. After this training period the robot should be able to retrace the routes autonomously. The final level of functionality we desire for our robot is the ability to find detours and shortcuts, including deducing routes that the robot has never actually traversed. This will give the robot added flexibility when its original route is blocked or it is under time constraints that make it worthwhile to look for a shortcut.

Using the TRC Labmate mobile platform (see Figure 6) equipped with sonar sensors and a single, translating camera, we have begun implementing the first two levels of functionality. We began by guiding the robot through a route from its starting location to a goal location (see Figure 7). At the starting point, the goal point, and at each turn in the route, the robot took a pair of images. From these images it extracted a set of locational cues. Locational cues are stable features in the environment that will allow the robot to identify a place in the future. For this particular experiment we choose vertical edges as locational cues. The vertical edges were extracted using a Sobel edge detector. The camera was translated along its axis and a second image was taken. Locational cues from the two images were matched and depth information was retrieved by noting the displacement of a pair of features between the two images. Locational cues are stored in a locational grid that represents their relationships and stored for further use. (See Figure 8). Finally, some of the locational cues are associated with future actions that the robot takes. For example, locational cue #1 in Figure 8 is associated with the right turn into room 50 since it is approximately the same distance away as the turn and on the correct side of the corridor.

Figure 6: The TRC Labmate mobile platform with sonar sensors and single, translation camera.

Figure 7: The robot is guided along a route from the start to room 50. The route has two turns.

In the second phase of the experiment, the robot is instructed to autonomously traverse the same route from the same starting position into room 50. It again extracts locational cues from images and uses the cues to determine its position in the environment. After determining its position, it chooses an action that will lead it towards its goal location. The action it chooses is remembered from its previous, guided traversal of the route. If the robot extracts a locational cue that is associated with a turn (such as locational cue #1 in the Figure 8) it can proceed rapidly to that location before it needs to determine its position once again.

The combination of manually guiding the robot through an environment and having it autonomously navigate after training provides for a flexible system of robot mapping. The robot will only store information about routes that it knows it will have to traverse and it does not need any prior knowledge about the environment. As the robot gains more experience in the environment it will be able to traverse routes more efficiently. We plan on continuing to develop our scene-based, multi-level representation in the coming months.

Figure 8: Top left: Original image that the robot takes as it looks down the hallway after turn 1. Top right: The robot extracts vertical edges that serve as locational cues. Bottom: The robot places the locational cues into a grid that retains their relationships. Some locational cues (such as #1) are associated with future actions (such as turn 2).

Position Calibration Using an Environmental Map

Under normal operating conditions, one would expect the robot to function in a well-defined environment. Unlike the work described above, this means that a detailed environmental map should already be available to it. With this map, the robot has the ability to keep track of its position quite accurately by comparing its view of the environment to the map's features. This also allows the robot to compensate for the inevitable errors in its expected position due to the accumulating effects of friction, uneven ground, etc. Ignoring these errors will only increase the magnitude of the positional error and complexity of the problem.

We shall assume the robot working in a typical nuclear environment contains a model of the working environment internally and periodically compares a simulated image with a camera image. Any deviation between the two images indicates a position error. The robot then adjusts its actual position from its expected position and the above error.

This project can be broken down into two phases:

1. Matching Phase: Find matching points/lines between the camera image and the environment model. (Initially, we plan to work with points.)
2. Correction Phase: Calculate a correction vector (rotation & translation) for the set of matched points and modify the robot's simulated position in the environment model to match its actual position.

We decided to implement the correction phase first. Consequently, routines were constructed to allow the user to manually choose and match up points from the camera image to preselected points in the model. This bimonthly report describes the steps taken during the past two months to implement the correction phase. We first review the important terminology and assumptions that are needed, and then describe the implementation process.

2. Terminology and Assumptions

First, the term model refers to the robot's collective knowledge of its working environment. The robot constantly maintains information on its current position and orientation in the model; we refer to this information as the expected position. Unfortunately, the robot's expected position may not necessarily be the same as its actual position in the environment. The model simulated image is what the robot expects to see from its current expected position based on known camera geometry. The camera image is what the robot actually "sees" from its current actual position. When dealing with points, image points refer to points from the camera image, whereas model points refer to points from the model or model simulated image.

There are three coordinate systems that we deal with: global, camera, and image (see figure 9). All three of these are right-handed coordinate systems. The global coordinate system (GCS) is the base coordinate system for the entire model. The position and orientation of the GCS are arbitrary. The camera coordinate system (CCS) references points from the focal point of the camera. The CCS is oriented so that the z-axis points directly backwards into the camera, and the y-axis points vertically up from the camera. We may assume that the global position and orientation of the camera is known. Both the global and camera coordinate systems are three dimensional (3-D). The image coordinate system (ICS) is two dimensional (2-D) and refers to points on the camera image plane.

The ICS is positioned at the bottom left corner of the image with the x-axis horizontal and the y-axis vertical with respect to the image. Unless otherwise specified, we shall assume that points are given in global coordinates.

Figure 9. The three coordinate systems. (x_g, y_g, z_g) indicates the global coordinate system. (x_c, y_c, z_c) indicates the camera coordinate system. (X, Y) indicates the image coordinate system.

Since we are implementing the correction phase before the matching phase, we will assume that the input to the correction phase (received from the matching phase) will be a set of matched pairs of points. One member of each matched pair is a model point (in 3-D) and the other member is an image point (in 2-D).

3. Implementation

The correction phase was implemented in two parts. First, we must convert each chosen image point from 2-D image coordinates to 3-D global coordinates. Then, given the image and model points in global coordinates, we shall calculate a correction vector specifying the correction needed to convert the expected position into the actual position.

The input to the correction phase consists of two sets of n matched points, $Xmodel$ and $Pdata$. Each $xmodel_i$ in $Xmodel$, $i=1,...,n$ denotes a model point in global coordinates. Each $pdata_i$ in $Pdata$, $i=1,...,n$ denotes a corresponding image point in image coordinates.

3.1 Conversion From Image to Global Coordinates

The conversion from 2-D image coordinates to 3-D global coordinates requires us to make an important assumption regarding the distance from the camera to any given point. We begin with a set of n image points $pdata_i = (X_i, Y_i)$, $i=1,...,n$ and we wish to calculate (xg_i, yg_i, zg_i) in global coordinates for each (X_i, Y_i) . Figure 10 shows that for any point on the image plane, there could be an infinite number of corresponding points in 3-D space. For each point z_n on the z -axis, there is a different point (x_n, y_n, z_n) in 3-D space that corresponds to the image point (X, Y) . This one-to-many relationship is due to the fact we are gaining information in our conversion; we are gaining an extra dimension along the z -axis. Since we are not told which z_n value to use, we approximate by making the assumption that the actual distance between a point and the camera is identical or extremely close to the expected distance. In other words, the robot position error is expected to be very small.

$$d_i = \text{dist}(\text{camera to } pdata_i) = \text{dist}(\text{camera to } xmodel_i)$$

Knowing d_i and the focal length of the camera, F , we can calculate the point, (xc_i, yc_i, zc_i) , in camera coordinates for each $pdata_i$ with the following equations:

$$\begin{aligned} xc_i &= \frac{X_i z}{F} \\ yc_i &= \frac{Y_i z}{F} \\ zc_i &= \frac{d_i F}{\sqrt{X_i^2 + Y_i^2 + F^2}} \end{aligned}$$

All image points are processed in this way to yield a set of 3-D data points derived from the image points. Since we know the position and orientation of the camera, we can easily transform the set of camera points, $\{(xc_i, yc_i, zc_i), i=1,...,n\}$, to the set of global points $\{(xg_i, yg_i, zg_i), i=1,...,n\}$.

3.2 Correction Vector Calculation

Once the image points have been converted to global coordinates, our next task is to find the correction vector q . q is a 7-vector that describes the rotation and translation needed to transform the expected position of the robot to the actual position. To do this we use an method described by Horn[2] and referenced by Besl and McKay[1]. The algorithm described below closely follows that used by Besl and McKay[1].

Figure 10. Conversion from image coordinates to global coordinates. Note that for each (X, Y) 2-D (image) space, there are an infinite number of corresponding points in 3-D (global) space.

Figure 11. Point matching and conversion to the GCS. Selected points from the model are matched with corresponding points from the camera image. Converting the points from the ICS to the GCS results in two global matched sets. The correction is the difference between these two sets.

The vector q is actually made up of two vectors:

$$\vec{q} = \begin{bmatrix} \vec{q}_R & | & \vec{q}_T \end{bmatrix}^t$$

where $q_R = [q_0, q_1, q_2, q_3]^t$, $q_0 \geq 0$, $q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$ is the unit rotation quaternion, and $q_T = [q_4, q_5, q_6]^t$ is the translation vector. It is important to note that this algorithm calculates q_R first; then calculates q_T from q_R .

First, we must find the center of mass of each set of points (data and model). Let μ_p be the center of mass for the set $Pdata = \{p_i\}$ and μ_x be the center of mass for the set $Xmodel = \{x_i\}$:

$$\vec{\mu}_p = \frac{1}{n} \sum_{i=1}^n \vec{p}_i \quad \vec{\mu}_x = \frac{1}{n} \sum_{i=1}^n \vec{x}_i$$

Knowing μ_p and μ_x , we can calculate the cross-covariance matrix Σ_{px} of $Pdata$ and $Xmodel$ with the following equation:

$$\Sigma_{px} = \frac{1}{n} \sum_{i=1}^n [\vec{p}_i \vec{x}_i^t] - \vec{\mu}_p \vec{\mu}_x^t$$

Next, we create the matrix $A_{ij} = (\Sigma_{px} - \Sigma_{px}^t)_{ij}$ and the vector $\Delta = [A_{23} \ A_{31} \ A_{12}]^t$. From these we can form the symmetric 4x4 matrix $Q(\Sigma_{px})$:

$$Q(\Sigma_{px}) = \begin{bmatrix} tr(\Sigma_{px}) & \Delta^t \\ \Delta & \Sigma_{px} + \Sigma_{px}^t - tr(\Sigma_{px}) I_3 \end{bmatrix}$$

where I_3 is the 3x3 identity matrix and $tr(A)$ is the trace of a matrix, A .

The eigenvector that corresponds to the maximum eigenvalue of the matrix $Q(\Sigma_{px})$ is the desired vector $q_R = [q_0, q_1, q_2, q_3]^t$. Given q_R , we can calculate q_T as follows:

$$\vec{q}_T = \vec{\mu}_x - R(\vec{q}_R) \vec{\mu}_p$$

where

$$R(\vec{q}_R) = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_0 q_3) & 2(q_1 q_3 + q_0 q_2) \\ 2(q_1 q_2 + q_0 q_3) & q_0^2 + q_2^2 - q_1^2 - q_3^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_2 q_3 + q_0 q_1) & q_0^2 + q_3^2 - q_1^2 - q_2^2 \end{bmatrix}$$

The vector $q_R = [q_0, q_1, q_2, q_3]^t$ describes rotation in terms of an angle of rotation θ and an axis of rotation defined by the unit vector $\omega = [\omega_x, \omega_y, \omega_z]$. The elements of q_R are defined by the following equations:

$$\begin{aligned} q_0 &= \cos \frac{\theta}{2} & , & & q_1 &= \omega_x \sin \frac{\theta}{2} \\ q_2 &= \omega_y \sin \frac{\theta}{2} & , & & q_3 &= \omega_z \sin \frac{\theta}{2} \end{aligned}$$

Each element of $q_T = [q_4, q_5, q_6]$ represents the needed displacement on the x-, y-, and z-axes, respectively, to correct the current position.

4. Conclusion

The correction phase of this project has been relatively straightforward to implement. Basically, it involved translating Horn's algorithm into program code. The code was tested independently first; then later merged and successfully tested with the robot simulation program.

The next step of this project will be to implement the matching phase. This will involve using both low-level image processing and heuristic artificial intelligence techniques to find points in a camera image that correspond to selected points in the known model. The output to this phase, along with the set of selected model points, will be the input to the algorithm described above.

Sparse Depth Maps Using Vision and Infrared Sensors

The work described above used vision to register depth cues as environmental landmarks for later navigation, and vision as a means of correcting the position error of the robot using an environmental map. In the work described here, we assume that no environmental map is available, but we would like to use vision and range data for navigation and sparse map building. We would also like the procedure to be totally automated.

The infrared (IR) range sensors currently mounted on a mobile platform in the Mobile Robot Laboratory at the University of Michigan have a limited range resolution due to their 4 bit analog digital converter (ADC). However, it is possible to obtain better resolution without changing the current hardware configuration by using knowledge of the robot's motion. In this project, we developed the algorithm for IR range estimation with sub-step accuracy. As the robot moves, a transition monitor detects the instance that a trimmed mean of the IR sensor outputs changes from one ADC step to another and evaluates the importance of data depending on the type of transition. The information from the transition monitor is integrated recursively in time with the displacement information from an encoder to give the sub-step accuracy in the range estimates. Our algorithm also provides an improved error bound for each range estimate which is useful in sensor fusion and other applications. Details of the transition monitor and experimental results which demonstrated a successful implementation of the algorithm are described in a paper recently submitted for presentation at IROS'92. The idea behind the algorithm can be applied to other sensor modalities having resolution primarily limited by a digitization device.

A Review of Other Work in Progress

A brief summary of progress on other project areas which have been reported in past progress reports:

- The adaptive wheel controller is currently being tested on one of the TRC Labmate mobile platforms. Previous work using Hough Transforms for landmark navigation has been completed and is being prepared for publication.
- The algorithm which compares the infrared and ultrasonic sensor readings is being rewritten to place more weight on the infrared sensor readings. The simple algorithm which has been used lacked the necessary sophistication.
- Work has continued in the area of radiation imaging. In the wavelength shifting work, the fast/slow ratio has now been improved to nearly 1. This represents a factor of two improvement in just the past two months. Some recent improvements in reflector design may provide another quantum jump. The CsI(Tl) work is continuing, although inexplicable diode noise problems have plagued this work during the past month. A new amplifier is expected to shed light on these problems. Major efforts have also been expended in the design and acquisition of a new Compton radiation imaging camera. We expect initial results from this exciting new development later this fall.
- A background effort has continued in the development of standard architectures and protocols for operating the diverse set of mobile robots employed for this project. While lacking the glamor of the work highlighted above, this effort provides a standard interface and provides network access to the different platforms. The NASREM architecture serves as the template for this work.
- The work on decision theory applications for rational agents has continued, although most of the recent effort has been centered on preparing for the AAAI and IJCAI presentations. A new demonstration was also prepared for the expected DOE site visit.

REFERENCES

1. Besl, P. J. and McKay, N. D. 1991. A method for registration of 3-D shapes. Accepted for publication in PAMI.
2. Horn, B. K. P. 1987. Closed-form solution of absolute orientation using unit quaternions. J. Opt. Sci. Amer. A 4 4 (April), 629-642.
3. Press, Flannery, Teukolsky, and Vetterling, 1988. Numerical Recipes in C. Cambridge Press, UK.

REPORTS AND MAJOR CORRESPONDENCE ISSUED

The following two papers were published:

1. Borenstein, J. and Koren, Y., 1991, "The Vector Field Histogram Fast Obstacle-Avoidance for Mobile Robots." Accepted for publication in the IEEE Journal of Robotics and Automation, Vol. 7, No. 3., June 1991, pp. 278-288.

2. Borenstein, J, and Koren, Y., 1991, "Histogramic In-motion Mapping for Mobile Robot Obstacle Avoidance," IEEE Journal of Robotics and Automation, Vol. 7, No 4, 1991.

SIGNIFICANT ACCOMPLISHMENTS

Error Eliminating Rapid Ultrasonic Firing for Mobile Robot Obstacle Avoidance

The University of Michigan has successfully tested and demonstrated our new error eliminating rapid ultrasonic firing (EERUF) method. EERUF has been implemented on the LabMate robot, which is commercially available from TRC. The LabMate is 75 cm long, 75 cm wide (including bumpers), and has a maximum speed of 1 m/sec. The algorithm demonstrated significantly improved results and high potential for realistic powerplant environments.

In our experimental system we used eight POLAROID sensors that were symmetrically spaced at 15° intervals (see Fig. 1). A '386-20 Mhz computer ran the EERUF algorithm as an interrupt-driven background task, which communicated range readings to the main task via a first-in first-out (FIFO) queue in shared memory. The foreground task was the vector field histogram (VFH) obstacle avoidance method combined with the histogrammic in motion mapping (HIMM) method (both developed earlier under the DOE project).

We tested the performance of EERUF with the following experiments:

1. Obstacle avoidance through a densely spaced obstacle course. We set up an obstacle course consisting of pencil-thin (8 mm diameter) vertical poles spaced approximately 1.6 m from each other (see Fig. 2). With the EERUF method, the robot was able to traverse this course at its maximum speed of 1 m/sec and an average speed of 0.8 m/sec (the maximum speed was reduced before and during tight turns, for dynamic reasons).

It is important to note that such high travel speeds are not necessary for all currently proposed applications. However, successful obstacle avoidance at a higher-than-necessary speed is a good measure of the robustness and reliability of a method, and consequently indicates a margin of safety for applications at lower speeds.

2. Obstacle avoidance in the presence of crosstalk. In Fig. 2 one can distinguish several highly reflective smooth walls and doors at the far corner of the lab. These features strongly promote crosstalk. We further added triangular-shaped reflective poster boards whose surfaces are smooth like Plexiglass (also shown in Fig. 2). Again, the robot was able to consistently avoid all obstacles while traveling at up to 1 m/sec.

3. Obstacle avoidance in the presence of ultrasonic noise from other robots. In this experiment we found that EERUF allowed equally fast obstacle avoidance even in the presence of intense ultrasonic noise from another mobile robot. This experiment is depicted in Fig. 2, where CARMEL was standing in the obstacle course while firing its 24 sensors at a rate of 1 firing every 3.3 ms.

4. Obstacle avoidance in the presence of intense and continuous environmental ultrasonic noise. In this experiment we introduced continuous noise by discharging an 80 psi air hose through a high velocity nozzle. This

type of noise can be expected when a pneumatic line leaks or breaks. Automatic Guided Vehicle (AGV) experts at Caterpillar have identified this form of noise as the foremost disturbance in industrial applications of AGVs. In the presence of this type of noise, we were able to demonstrate obstacle avoidance at lower speeds (up to 0.5 m/sec), when the source of noise (the discharge nozzle) was more than 3 m away from the robot.

5. Stationary environment. In addition to the in-motion experiments described above, we tested EERUF extensively in a reproducible, stationary test-environment. Typical results show successful rejection of over 97% of direct and indirect path crosstalk (from onboard sensors), or errors caused by external sources. The EERUF method consistently produced one to two orders of magnitude fewer errors than a "conventional" firing scheme (i.e., without error rejection) operating at the same firing rate.

Undergraduate Summer Research Results

This last summer we had three undergraduates who worked directly with our mobile robots and were supported by this grant and matching university money. Their projects are summarized here.

1. **Gateway detection using sonar** In many mapping situations it is important to detect when openings have appeared in the environment. These openings could correspond to doorways or hallways that the robot may wish to enter. We have designated these openings as gateways, for they allow the robot to pass from one space to another. We desired a method to efficiently detect openings as the robot travels down a hallway. The method we developed relies on detecting changes in the sonar readings of the side facing sonar elements (i.e., the sonars facing directly to the left and to the right of the robots direction of travel). These changes are found by detecting spikes in the derivative of the sonar readings (see Figure 4, which plots sonar values as a function of distance travelled). We have characterized an opening as having a beginning and an end. The beginning can be recognized as a negative spike in the derivative, i.e., the wall is either no longer there or suddenly much farther away. The end can be recognized as a positive spike in the derivative, i.e., the wall has either suddenly reappeared or become much closer. Various additional techniques are used to eliminate false openings. These algorithms were tested on a TRC Labmate mobile robot in our laboratory.

2. **GOALS.** The Potential Field Method (PFM) of obstacle avoidance is widely known. In this method, obstacles exert imaginary repulsive forces upon the robot while its destination exerts an imaginary attractive force. The vector sum of these forces points in a direction assumed safe for robot travel. However, PFM has several drawbacks, including oscillations when moving down a hallway, collisions on corners and trap situations where the robot is in a box. In an attempt to solve these problems the Global Obstacle Avoidance using Localized Sensing (GOALS) method was developed. Instead of reacting to various objects in the environment, the GOALS method actively searches for openings large enough for the robot. A long-range planner looks further ahead for distant openings. Then, the robot looks for a direction with openings both next to and ahead of the robot. If there is no long-range opening, the long-range sensing is ignored. Finally, the robot sets a local goal

through the opening which is closest to the robot's destination (see Figure 5). While the robot drives toward the local goal, it looks for other openings and sets a new local goal. The GOALS method effectively navigates around the situations where PFMs have more difficulty. GOALS was also tested on a TRC Labmate mobile robot in our laboratory.

Adaptive Motion Control for Mobile Robots

The following tasks were completed during the reporting period:

- Conceptual study of the adaptive motion control idea, and primitive computer simulations to test the ideas.
- Comparative study of different cross-coupling control structures by computer simulation, as well as the theoretical analysis of the proposed cross-coupling controller.
- Implemented the cross-coupled controller in software on a TRC Labmate robot.

The following work is in progress:

- We are building a new controller which will bypass the current on-board controller. This will allow an increased sampling rate which will improve the control and permit greater flexibility in controller design.
- Stability analysis and optimization of the adaptive control system.
- Hardware implementation of sensors and computer interface.

MEETINGS AND CONFERENCES

Johann Borenstein attended the 4th International SME conference in Pittsburgh, Pennsylvania, Sept. 16 - 19, 1991. He chaired a session on autonomous vehicles and presented a paper.

The University of Michigan hosted its annual DOE site review. Attenders included representatives from the Office of Naval Research and Consumers Power Company. The University also attended the DOE Robotics Project PI meeting in Washington, D.C.

REPORTS AND MAJOR CORRESPONDENCE ISSUED

Borenstein, J. and Raschke, U., 1991, "Real-time Obstacle Avoidance for Non-Point Mobile Robots." Proceedings of the Fourth World Conference on Robotics Research, Pittsburgh, Pennsylvania, Sept. 17-19, pp. 2.1 - 2.9.

Kortenkamp, D., Chown, E., Kaplan, S., and Weymouth, T., "A scene-based, multi-level representation for mobile robot spatial mapping and navigation,"

submitted to the 1992 IEEE International Conference on Robotics and Automation.

Fujii, Y., Weymouth, T., and Wehe, D., "Integration of Active Monocular Vision and Infrared Range Sensor", submitted for publication in the ANS Proceedings of the Remote Systems Technology Division.

Figure 1: One of the University of Michigan's 4 *Labmate* robots, equipped with an array of eight URS (spaced at 15° intervals).

Figure 2: With EERUF, the *Labmate* zaps through an obstacle course at 1 m/sec. Another mobile robot nearby generates discrete ultrasonic noise at a rate of 1 firing every 3.3 ms.

Figure 3: Experimental set-up in crosstalk-promoting environment.

Figure 4.

Figure 5. Simple example of GOALS method finding opening through three obstacles

END

**DATE
FILMED**

01/31/92