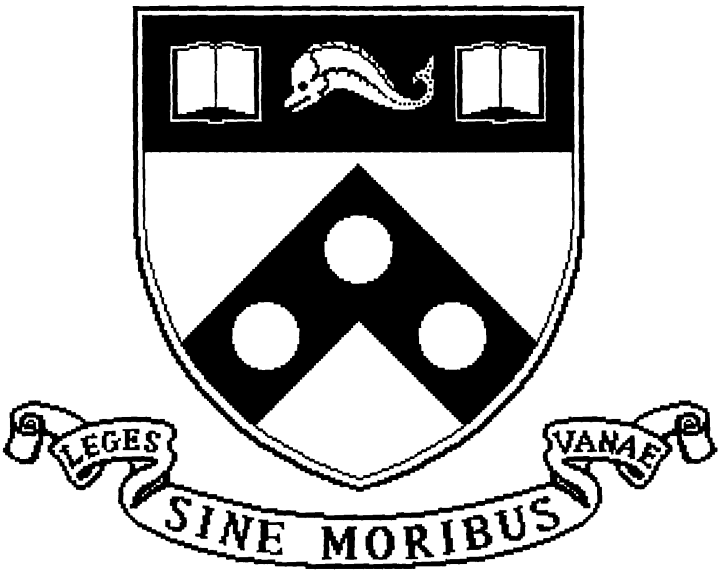
Cooperative Material Handling by Human and Robotic Agents: Module Development and System Synthesis

MS-CIS-95-01

J. A. Adams
R. Bajcsy
J. Kosecka
V. Kumar
R. Mandelbaum
M. Mintz
R. Paul
C. Wang
Y. Yamamoto
X. Yun



University of Pennsylvania
School of Engineering and Applied Science
Computer and Information Science Department
Philadelphia, PA 19104-6389

Cooperative Material Handling by Human and Robotic Agents: Module Development and System Synthesis *

J.A. Adams, R. Bajcsy, J. Kosecka, V. Kumar, R. Mandelbaum,
M. Mintz, R. Paul, C. Wang, Y. Yamamoto, X. Yun.

General Robotics and Active Sensory Perception (GRASP) Laboratory
Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104

December 15, 1994

1 Introduction

1.1 Overview

In this paper we present the results of a collaborative effort to design and implement a system for cooperative material handling by a small team of human and robotic agents in an unstructured indoor environment. Our approach makes fundamental use of human agents' expertise for aspects of task planning, task monitoring, and error recovery. Our system is neither fully autonomous nor fully teleoperated. It is designed to make effective use of human abilities within the present state of the art of autonomous systems. It is designed to allow for and promote cooperative interaction between distributed agents with various capabilities and resources. Our robotic agents refer to systems which are each equipped with at least one sensing modality and which possess some capability for self-orientation and/or mobility. Our robotic agents are *not* required to be homogeneous with respect to either capabilities or function.

Our research stresses both paradigms and testbed experimentation. Theory issues include the requisite coordination

*This research is funded in part by: ARPA Grants N00014-92-J-1647, DAAH04-93-G-0419; ARO Grants DAAL03-89-C-0031PRI, DAAL03-92-G0153; Gateway Grant 9109794; NASA Grants NGT-50729, NGT-70359; NIH Grant 3R01LM0521703S1; NSF Grants BCS92-16691, BCS92-21796, CISE/CDA-88-22719, CDA91-21973, CDA92-11136, CDA92-22732, GER93-55018, IRI92-10030, IRI92-09880, IRI93-03980, IRI93-07126, MSS91-57156-A 02; University Research Foundation Grant 370892; and The Whitaker Foundation

principles and techniques which are fundamental to the basic functioning of such a cooperative multiagent system. We have constructed a testbed facility for experimenting with distributed multiagent architectures. The required modular components of this testbed are currently operational and have been tested individually. Our current research focuses on the integration of agents in a scenario for cooperative material handling.

1.2 Related Work

There are several groups working on related problems, addressing the issues of cooperation at many different levels. Starting from the traditional symbolic planner-based control [9, 12], all the way to the lower levels where agents are involved in the direct physical interactions between each other or interactions mediated through the environment they reside in.

Some of the approaches looking at variations of the mobility problem are motivated by the ethological studies of animal societies (e.g. ant colonies, schools of fish, flocks of birds). By providing individual agents with sensory capabilities to recognize others of the kin and introducing new behaviors (e.g. staying close and/or away from another agent), simple cooperative behaviors such as flocking, dispersing and following emerged from the interactions of elementary reflexive behaviors. In these scenarios the societies of agents are homogeneous and the tasks such as exploration, wandering, and foraging food are usually achievable by a single agent [3, 23, 8, 35]. The multiplicity of agents increases the speed and efficiency while the communication between agents is not necessary to accomplish the task, when introduced the performance of the society improved.

The issue of communication plays different role when the cooperation between agents becomes more direct. For example cellular robots described in [38] interact with each other and reconfigure themselves, each cell robot sends a message about it's type and module to connect to. Different aspects of cooperation are emphasized in case of multi-arm manipulation also requiring closer coupling between agents [10]. As the complexity of tasks increases, requiring agents with different capabilities, the issues of cooperation must to be addressed at both high and low levels [31, 32]. This includes the task decomposition phase, where the task is subdivided and subtasks allocated to individual agents, followed by coordination phase where the robots coordinate their activities. The task decomposition problem brings another crucial issue to the control of a society of agents, that is the tradeoff between local and global control, i.e. to what extent should the members of the team be aware of the global intentions of the team lead by the leader versus just acting upon local information sensed through the environment [33]. The amount of global control needed is task dependent and generally the tasks which require optimization of some global resource (e.g. time, space, energy) require a global view [30].

1.3 Assumptions

Since the problem of agent-agent interaction and cooperation is immense, we must constrain it.

Our first constraint is that of the environment we consider. Our environment is *indoors* with somewhat controlled illumination, and a stable spatial layout (walls, doors and the basic furniture do not move around).

The second constraint concerns the design of our robotic agents. We currently have four mobile robotic agents, situated on a TRC mobile platform which has two degrees of freedom in movement: translation (back and forth) and rotation (clockwise or counter-clockwise). All platforms have position encoders on each wheel. Two agents are manipulatory agents. Agents C and D are equipped with six degree-of-freedom manipulators (one a PUMA 260 and the other a Zebra Zero). The remaining two agents are observer agents. Observation agent A has a suite of ultrasonic and infrared sensors, a light-striper and a stereo camera pair. Observation agent B is equipped with a stereo camera pair and a turntable on which is mounted an additional camera for tracking purposes. The human agent is supplied with a three dimensional graphical interface. A geometric model of the workspace is part of the a priori knowledge of this system. The raw sensory and processed data from the robotic agents are provided so the human may monitor the actions of the agents. The interface permits the human to act as a supervisor while also permitting interaction at any level of the system. Agent capabilities are described in more detail in section 2.

The third constraint involves the task. The task is to utilize the two mobile manipulatory agents to carry an object similar to a large pipe from one place to another, avoiding obstacles and passing through a narrow passage. The task of the observer agents is to scout the passage and *advise* (not commands) the mobile manipulatory agents as to the layout of the free space. The task of human agent is to monitor, advise and intervene when necessary.

The system is embedded in a Discrete Event Systems control theory framework where low level behaviors operate under supervisory control.

1.4 Scenario

We have selected a scenario within which we may evaluate the performance of the multiple cooperative agents system. As discussed in the previous section, the overall goal of the system is to transport a large object from one place to another while avoiding obstacles and passing through a doorway. Ultimately, this goal will be broken up into subgoals by the task planner [14, 15], though for the current work, we assign the following subtasks manually:

- It is the responsibility of observation agent A to check the prescribed path for obstacles, and to ensure that the pathway is wide enough at all points to accommodate the two manipulatory agents carrying the object. These two

subtasks are performed using the inverse perspective projection and ultrasound sensing modalities (see Section 2.2)

- It is the responsibility of observation agent B to follow the two manipulatory agents, keeping them in the field of view of a camera, thus allowing the human supervisor to monitor their progress.
- Obstacle avoidance is accomplished by treating the two manipulatory agents as one combined vehicle.

Within this context, the following parameters are variable, allowing for many variations on the basic scenario:

1. **Initial Localization:** In the simplest case, it is assumed that the initial locations and orientations of all agents within a given global coordinate system are known. A more complicated scenario involves knowledge of the locations and orientations of the agents *relative to one other*, but *not* relative to the global coordinate frame. In this case, some initial sensor-based localizations must be performed by one of the observation agents. In the most complicated case, no a priori knowledge of agent locations or orientations is assumed; it is the responsibility of the observation agents to localize themselves and the manipulatory agents within the global coordinate frame before path-planning and execution may begin. The various cases demand ever increasing levels of cooperation among the agents.

2. **Continuing Localization:** Continuing localization of mobile agents during execution of a task is a subject of extensive research. Most approaches employ some form of a dead-reckoning system based on wheel encoders, inertial navigation systems [5], gyroscopic information [4] or a combination of these. Dead-reckoning systems suffer from error accumulation; for long-term task-execution, they should be supplemented with some other form of localization. The environment may be modified to include man-made land marks which a mobile platform can locate with its sensors, or active beacons such as satellites in a global positioning system. The most difficult form of localization does not rely on modification of the external environment at all: the mobile platform uses its sensors to locate landmarks within a map of the environment. [7, 21, 37].

3. **Path generation:** Many alternatives are possible with regard to the generation of the initial way-points through which the manipulatory agents should transport the object. In the simplest case, the human supervisor may specify the way-points explicitly. A slightly more complicated case involves the human supervisor teleoperating one of the observation agents along a desired path, and recording way-points at various intervals. A still more sophisticated approach employs a path planner such as is described in [6] and [39].

4. **Knowledge of the environment:** The task may be performed in a *completely known* environment, in which case the locations of all objects (doorways, walls, etc.) are known in global coordinates. There are no unforeseen obstacles in such an environment. A more challenging scenario assumes a *partially known* environment. Here the

location of the doorway is known in global coordinates, but unforeseen obstacles may be encountered along the prescribed path, and would have to be avoided. In the case, *no a priori knowledge* of the environment is assumed, in which case extensive exploration must be performed by the observation agents before any path planning or traversal may begin.

5. **Obstacles:** The system may be used in environments which are obstacle-free, in which case no deviation from the precomputed plan is necessary. In environments which contain relatively few obstacles, a deviation may be necessary, though the distances between obstacles may permit the preferred side-by-side configuration of the manipulatory agents. Side-by-side configuration is preferred because the controllers for mobile platforms and robot manipulators can be decoupled. In environments containing a sufficiently high density of obstacles which prohibit the preferred side-by-side configuration, the manipulatory agents would not only have to deviate from the prescribed path, but also have to execute a configuration change to the leader-follower configuration (without dropping the object) in order to complete the task.

6. **Need for reconfiguration:** Even in an obstacle-free environment, the doorway through which the manipulatory agents must pass may be too narrow to permit the preferred side-by-side configuration, and a re-configuration during execution is necessary. In contrast to the case involving unforeseen obstacles such a reconfiguration may be planned and scheduled.

The number of combinations afforded by the above scenario parameters is prohibitively large. We limit ourselves to the following three illustrative and increasingly complex situations:

	Initial Localization Necessary	**Continuing Localization Method**	**Path Generation**	**Knowledge of Environment**	**Obstacle Density**	**Need for Reconfiguration?**
1	none	dead reckoning	human	complete	zero	yes: planned
2	partial	dead reckoning	human	partial	low	yes: planned
3	partial	dead reckoning	human	partial	high	yes: unforeseen

See Figure 1 for a graphical representation of the agents' motions subject to these three combinations.

By *partial initial localization* we mean that the agents' initial positions relative to one another is known. However, the local coordinate frame in which this is known must still be aligned with the global coordinate frame by one of the observation agents localizing itself. In the current scenario we employ dead reckoning only for continuing localization of the agents within the environment. *Human* path generation refers to the simplest case of way-point specification by

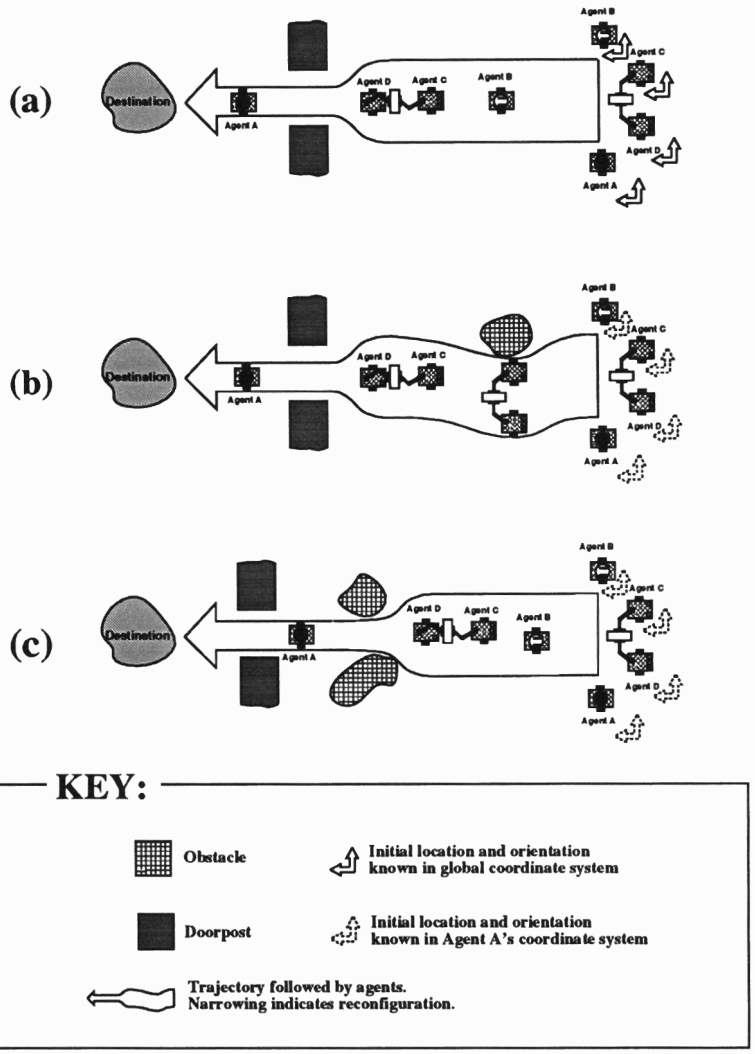


Figure 1: Graphical representation of three scenarios: (a) All four agents' initial locations and orientations are known; no obstacles encountered; manipulatory agents undergo scheduled reconfiguration to pass through doorway. (b) Agents' relative initial locations and orientations are known; Agent A performs global localization; an obstacle causes deviation from planned path; the manipulatory agents undergo a scheduled reconfiguration to pass through a doorway. (c) Agents' relative initial locations and orientations are known; Agent A performs global localization; multiple obstacles warrant unscheduled reconfiguration of the manipulatory agents.

the human supervisor. In example 3, the reconfiguration is unforeseen as it is a result of a high obstacle density and a correspondingly narrow pathway. See sections 4 and ?? for an evaluation of the success of our system in these three scenarios for the specified task.

2 Components

In this section we discuss

1. The hardware configuration and control architecture of the two manipulatory agents,
2. The observation agents and their capabilities,
3. The human agent.

2.1 Mobile Manipulatory Agents

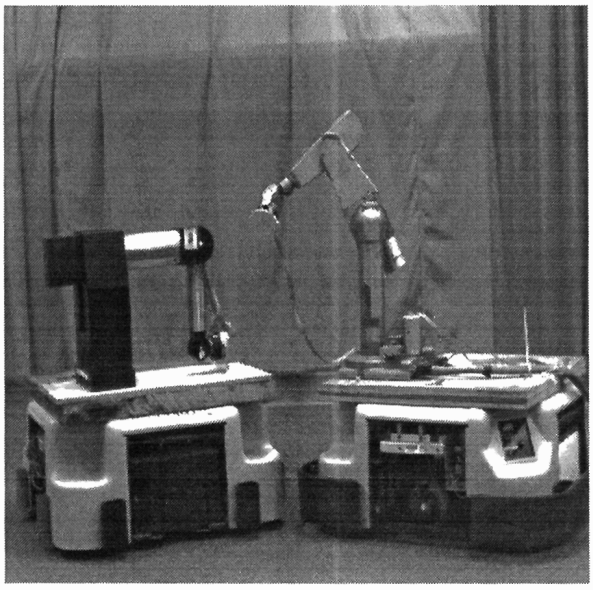


Figure 2: The two manipulatory agents.

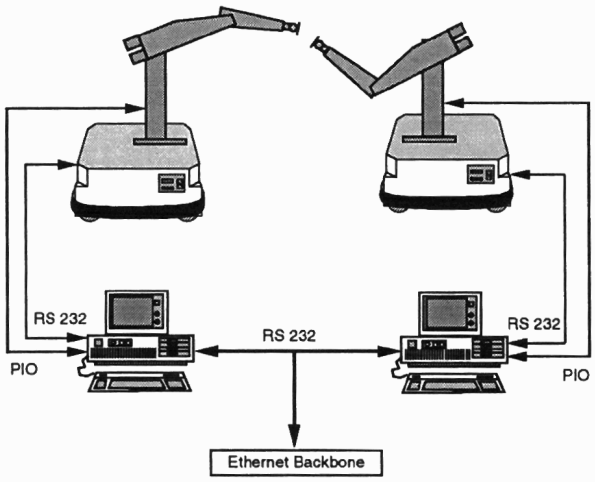


Figure 3: The Hardware architecture of the manipulatory agents C and D for the experimental setup.

Each of the manipulatory agents has a six degree-of-freedom manipulator mounted on a three degree-of-freedom car-like robot vehicle. Because of the excess degree-of-freedom to operate in three-dimensional Cartesian space, they are

defined as *redundant manipulators*. Both have a six degree-of-freedom force sensor to measure forces/moments applied at the end-effector. Figure 2 depicts the two agents: one with a Puma 260 manipulator (right) and the other with a Zebra-ZERO manipulator (left). A decentralized controller is implemented on a math-engine processor for each agent and input/output signals are relayed to the mobile robots and manipulators by a PC-AT 286 with serial and parallel ports. If needed, states of the agents (for instance joint angles, error states, etc) can be shared between agents through a network of communication. Figure 3 schematically describes the architecture mentioned above.

Within the prescribed scenario, the major subtask of the manipulatory agents is to cooperatively pick up the object prescribed by the human supervisor and transport it to the desired destination. Along the path, which may be preplanned or dynamically changed according to sensory information, stable grasping of the object is always required. In order to maneuver in a cluttered environment, the marching configuration may need to be changed from a serial formation to a parallel formation or vice versa. A more detailed discussion appears in Section 3.3.3.

2.2 Mobile Observation Agents

The two observation agents are equipped with various sensor modalities residing on a car-like robot vehicle. Each agent has a designated general purpose work station (SPARC 2) for processing of the sensory data and the mobile bases are controlled via a serial port. Within the prescribed scenario, the subtasks to be executed by the observation agents comprise:

1. Localization of each agent in global coordinates,
2. Iterative verification that proposed trajectories are obstacle-free, and
3. Monitoring of the progress of the manipulatory agents (agents C and D), allowing for possible intervention by the human supervisor.

The observation agents have five sensor modalities at their disposal. In this subsection we describe the physical model and basic operation of each modality, as well as the type of information which we have chosen to extract.

2.2.1 Ultrasound

Basic operation Observation agent A, depicted in Figure 4, is equipped with a ring of sixteen standard POLAROID™ sensors, with an inter-sensor radial separation of fifteen degrees.

We assume that each sonar return is associated with a single element of the set of basic features $\mathcal{P}$ = {planar reflective patches, outer diffractive corners, inner reflective corners}. Due to the large angle of the cone of ensonification [20, 22],

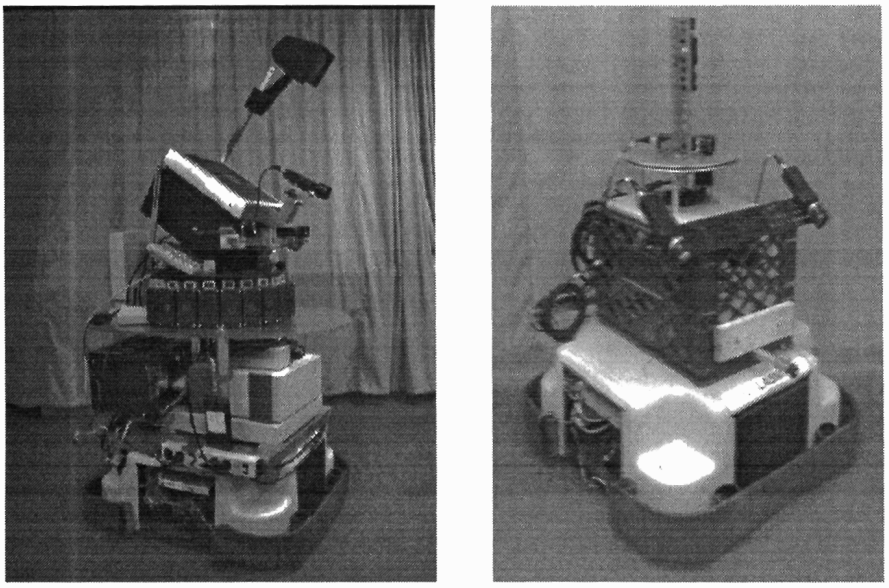


Figure 4: (a) Observation agent A with four sensor modalities: ultrasound, stereo pair, light-striping and odometry, and (b) Observation agent B with two sensor modalities: inverse perspective projection and odometry.

each ultrasound reading, though yielding relatively accurate *range*, provides scant information about the azimuthal or latitudinal location of the basic feature. All that one can infer from a *single* measurement is the existence of an element of $\mathcal{P}$ at the distance r somewhere along the boundary of the transmitted cone truncated at range r. See Figure 5.

Ultrasonic feature detection In order to extract azimuthal and orientation information from ultrasonic data, and hence detect features, multiple measurements are required. For *multiple* measurements generated by the same planar surface, all arcs, in the noise-free 2D case, share a common tangent; corners (both inner reflective and outer diffractive) induce measurements whose arcs intersect at the corner. In general, a continuous curve defined parametrically by $(s_x(t), s_y(t))$, where s_x, s_y are differentiable, induces measurements such that the arc corresponding to each measurement intersects the curve at a point $(s_x(\hat{t}), s_y(\hat{t}))$ where both curve and arc share the tangent vector $(\dot{s}_x(\hat{t}), \dot{s}_y(\hat{t}))$ [28, 29].

In the 2D noise-free case, ultrasonic data association is equivalent, therefore, to find subsets among the set of all measurements such that in each subset, all measurements correspond to arcs which either intersect at a point, or share a common tangent with a feature described parametrically by $(s_x(t), s_y(t))$; when the feature is a planar curve, all measurements in a cluster share a *common* tangent. The data from members of each cluster are used to estimate the parameters of the underlying feature [28, 29]. In general, the problem of finding arbitrary subsets is exponential in nature. A *polynomial* algorithm for ultrasound feature detection is described in [28] and [29]. The algorithm aggregates

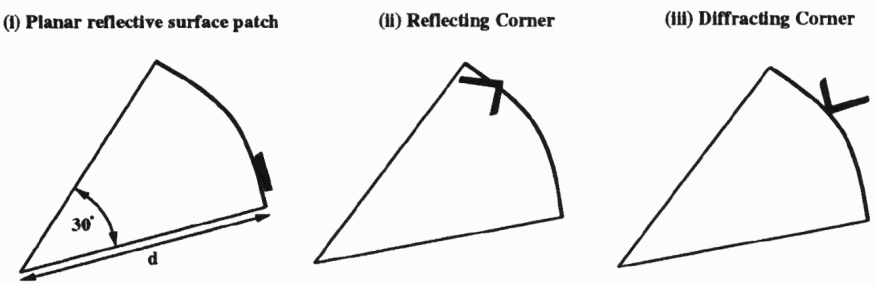


Figure 5: Possible inferences to be drawn from a single sonar measurement d: Primitive is located at the measured distance and is one of: (i) a planar reflective surface patch aligned with the impinging wavefront, (ii) a reflecting corner, or (iii) a diffracting corner.

sonic data accumulated from *arbitrary* transducer locations, performs the clustering *sequentially*, rather than in a batch fashion, is computationally *tractable* and *efficient* despite the inherent exponential nature of clustering, is *robust* in the face of noise in the measurements, and is *precise* in that it converges in a statistical sense to ground truth. The output of the algorithm comprises the parameters of features in space such as extended planar surfaces or corners.

Application The ultrasonic feature extraction algorithm may be used both for *exploration* and map-building of an unknown environment, and for *localization* and hence *navigation* within known surroundings. In the context of the present scenario, it is used primarily for the initial localization of observation agent A, both in Cartesian $X-Y$ position and in orientation.

2.2.2 Light-striping

Basic operation Observation agent A, depicted in Figure 4a, is equipped with a light-striping device consisting of a light source projecting three planes of light in front of the robot at an angle to the ground. A camera offset vertically from the light source uses elementary projective geometry to detect an object which intersects any of the light planes.

Light-striping feature detection It is possible to obtain 3D information about the object with the device, but with current hardware, processing time would be prohibitive. Instead, the algorithm we employ simply detects gaps in the expected locations of the stripes in the camera image and interprets these as segments of an object. Over time, these segments may be grouped to form a bitmap corresponding to the "shadow" of the object under the projected light. Registration and integration of several shadows extracted from multiple views yields the "footprint" of the object. See Figure 6.

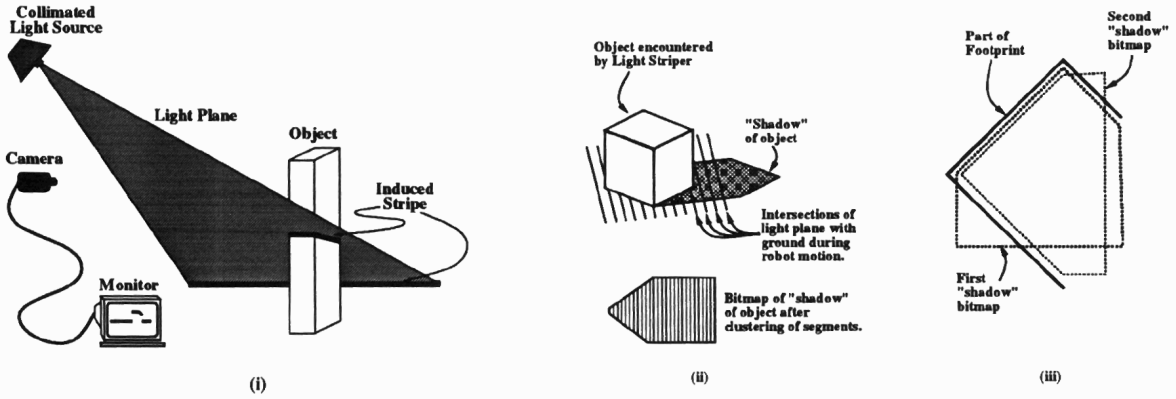


Figure 6: Basic operation of the Light-Striping device: (i) Overall setup showing extraction of single segment, (ii) Multiple segments grouped into "shadow" bitmap, and (iii) Registration and integration of multiple bitmaps for reconstruction of "footprint".

Define a *scan* to be a straight-line motion of length s at a fixed velocity. During such a motion, errors in dead reckoning (including slippage) are small and calibration errors are consistent over many segments. Thus, the grouping task is simplified. Footprint detection involves multiple scans from multiple viewpoints. After each scan, the bitmap extracted is analyzed for features. This facilitates registration of the bitmap with a *previous* bitmap sharing a subset of the features. It also aids in planning the *next* viewpoint such that several features will be common to this bitmap and the next. See [27] for a more detailed description of the motion planning and registration techniques used for footprint detection.

Application The light-striping footprint detector may be used to extract information regarding objects' extents, orientations and shapes. Such information is useful for exploration and map-building, for the disambiguation of landmarks during navigation, and for localization. In the current scenario, the light-striping modality is used to determine the extent of unexpected obstacles or the width of the zone between two obstacles.

2.2.3 Stereo

Basic operation Observation agent A, depicted in Figure 4a, is equipped with a stereo pair of cameras. The images from the two cameras are compared, and the azimuthal disparity of corresponding features is used to infer range.

Stereo feature detection Clustering of points of similar range facilitates segmentation of the camera images into regions approximately corresponding to objects in the cameras' common field of view. Analysis of the shape of each segment yields valuable information regarding object shape, height, and azimuthal position and extent. We employ

a stereo algorithm which reverses the data flow direction usually associated with stereo algorithms: instead of using disparity to determine range, the algorithm employs range information (from the ultrasound modality or knowledge of a map) for segmentation.

Application Information regarding objects' shape, azimuthal extent and height extracted by the stereo modality is useful for exploration and map-building, for the disambiguation of landmarks during navigation. Data regarding azimuthal position of landmarks may be used in conjunction with other modalities for agent localization. In the present scenario, the stereo modality is not used.

2.2.4 Inverse Perspective Projection (IPP)

Basic Operation Observation agents A and B, depicted in Figure 4, are each equipped with a stereo pair of cameras tilted with respect to the horizontal plane. The camera parameters and the geometry of the set up are known and static throughout the experiment.

Obstacle Detection Obstacles are detected through the difference between a pair of stereo images after applying the proper inverse perspective mapping proposed in [25]. Differences in perspective between left and right views are used to determine the presence of an obstacle and its' approximate location. The computed map of the free space in the common field of view of both cameras is used for obstacle avoidance maneuvers [18]. See Figure 2.2.4.

Figure 7: a) Left Image; b) Map of the free space in lower resolution (obstacles in white); c) Right Image

Application The obstacle detection modality monitors the free space ahead of the mobile base. At each instance of time a set of detected obstacles is registered with the obstacles seen previously. Obstacles are approximated by a covering ellipse. This modality then provides the necessary parameters for navigation in cluttered environments using

the artificial potential field approach. This information can also be used for updating the world model. In the present scenario, the IPP modality is used to "sweep" through the proposed trajectory of the manipulatory agents, ensuring that it is obstacle-free.

2.2.5 Odometry

Basic operation Both observation agents, depicted in Figure 4, as well as the manipulatory agents, depicted in Figure 3, are equipped with wheel encoders measuring the rotation of each wheel. Assuming no slippage occurs, knowledge of the wheel radii and the base-line distance between wheels permits computation of vehicle translation, rotation and linear and angular velocity. Such computation is performed by hardware on each agent.

Unfortunately, empirical results have shown that the effective base-line distance is both load- and terrain-dependent. Due to the cumulative nature of dead-reckoning errors, odometry by itself is highly unreliable as a positioning modality. For long-term operation, the conjunction of odometry with other landmark-detecting modalities is essential in order to keep position and orientation errors below acceptable bounds. In the current scenario, however, these difficulties are ignored. Various methods of sensor-based localization are evaluated and compared in [26].

2.3 Human Supervisory Control

In order to take advantage of the autonomous aspect of our system, and yet ensure successful completion of all tasks possible with teleoperation, we have developed our human-machine interface based upon the mediation hierarchy [1, 2]. This hierarchy permits the human supervisor to interact at any level of our robotic system. Robotic systems are not robust in handling unmodeled events. Reactive behaviors may, or may not, be able to guide the robot back into a modeled state i.e., error recovery may not be achieved. Reasoning systems may simply fail. Once a system has failed it is difficult to restart the task from the failed state. Rather, the rule base is revised, programs altered, and the task retried from the beginning.

Our interface, MASC - Multiple Agent Supervisory Control system permits the agents to work autonomously until the human supervisor is requested to take control or a problem is detected by the human supervisor. Our design strategy is to develop a general system which is applicable to various robotic systems. We combine the advantages of autonomous systems with the human's ability to control a system through a human-machine interface. MASC provides the human supervisor with tools to interact with all processing levels of the robotic system. These interactions may correct corrupted data or process decisions which would typically cause an autonomous system to enter an incorrect state. We desire to create a more comprehensive semi-autonomous system based on this interaction which will successfully complete the execution of task assignments.

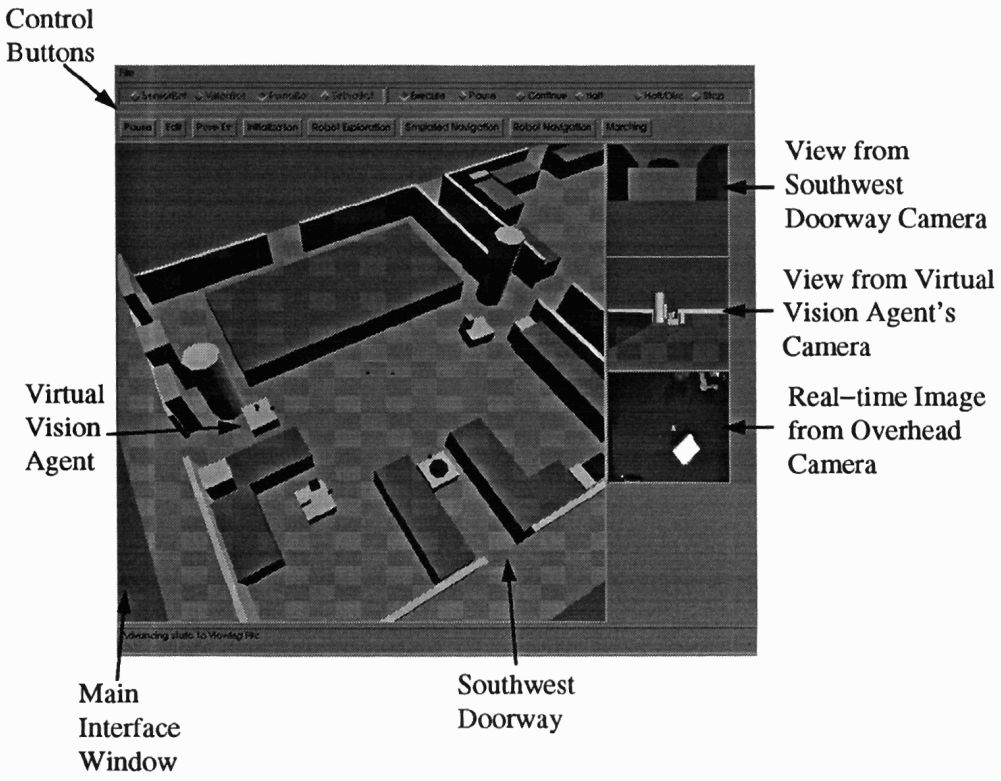


Figure 8: The MASC system interface.

The individual robotic agents, their associated manipulators and processes may be controlled by the human supervisor through MASC. The primary task of the human is to "supervise" the actions of the agents during execution [36]. Through MASC, the human supervises the system, observes sensory data and images. Each agent is composed of multiple control and processing levels. In order for the successful semi-autonomous execution of feasible tasks, MASC must permit the human supervisor to interact with these levels. We have organized the supervisor's interactions with the many system levels into a hierarchy of mediation.

The human supervisor communicates with the agents through MASC. We have provided display push buttons, termed control buttons, (see Figure 8). The control buttons allow the human supervisor to specify system information. The human supervisor may request any agent's sensory data while in any system state. The agents transmit odometry and heading readings, sensory readings, raw image data, and processed data. This information is employed by MASC to create various system displays. Image data may be displayed in windows to the right of the main interface window and images may be overlaid onto the virtual environment model, (see Figure 8). Processed data, such as the free space map originating from the visually guided obstacle avoidance process, may be displayed in a window or overlaid onto the

virtual model.

MASC combines autonomous and telerobotic control. While an agent proceeds autonomously it may petition assistance from the human. The human supervisor must acknowledge the request and then furnish the proper information.

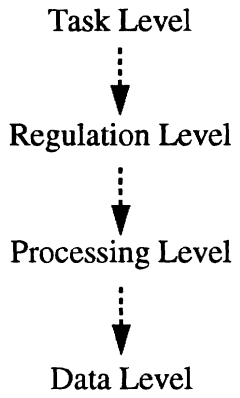


Figure 9: Hierarchical levels of human interaction.

We have defined four hierarchical levels of supervisory interaction with the various levels of a robotic system (see Figure 9). These levels define the various types of intervention into the differing levels of our robotic system. This interaction should permit the human supervisor to correct situations which would cause a fully autonomous system to fail its' task execution. It is important to note that the human supervisor only interacts with the agents when assistance is requested or when the supervisor detects a situation where s/he deems it necessary to intervene on behalf of an agent.

2.3.1 Task Level

The *task level* permits the human supervisor to specify the actions to be executed by an agent or a group of agents in order to complete an assigned task. Tasks may include exploration of the environment to assist with model building, following an assigned path to a goal, observing the execution of a task assigned to another agent, moving in a configuration, carrying items such as pipe, and the navigation necessary to transport items from one location to another. Since humans are better equipped to divide tasks into subtasks, we currently assign this undertaking to the human supervisor, however, we are incorporating the incremental task planner ItPlanS [14, 15] to assist the human supervisor.

Application The *task level* implementations are employed to determine the global plan for the task which is interpreted by the task description translation algorithm (as described in Section 3.2).

2.3.2 Regulation Level

There exist minimal interactions which are necessary between a human-machine interface and a robotic system. If an agent is on the verge of colliding either with another agent or an obstacle, the human supervisor should be able to prevent such a collision. The human supervisor possesses a means of monitoring an agent's actions. This monitoring may occur through video images, displays of sensory data or positional updates. It is essential that the interface provide a means for the human supervisor to choose such information for monitoring purposes. Also, in such a system, the agent's processes may require information from the human supervisor in order to begin processing. The interface must facilitate provision of this information. The *regulation level*, see Figure 10, couples these interactions into one mediation level. We have developed three types of interaction on this level, *control interaction*, *request interaction* and *specification interaction*.

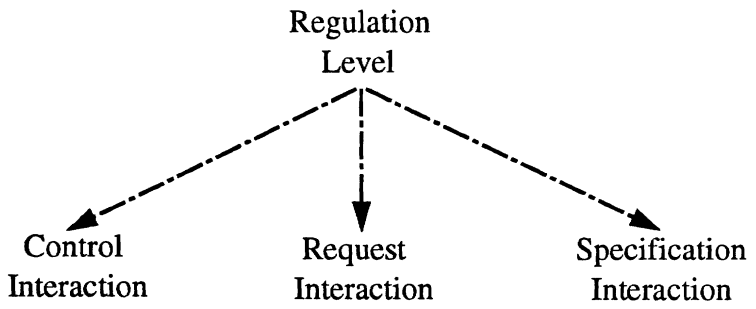


Figure 10: Interactions on the *regulation level*.

Control Interaction

MASC provides the human supervisor with the capabilities to cope with situations when an impending collision must be avoided, or if one agent should be instructed to wait for another agent to complete it's task assignment through *control interaction*. The human supervisor may also teleoperate the agent via *control interaction*. The *control interaction* provides the supervisor with the ability to control the progress of the agent while executing a task either for the purpose of halting or assisting progress.

Request Interaction

Systems possess various types of information which may be of use to the human supervisor at different times throughout the system execution but which are not necessary during the entire system execution. The objective is to avoid overloading the human supervisor with too much information [36, 40]. The *request interaction* permits the human supervisor to request the sensory data and processed information from the agent's only when needed

for error detection or monitoring purposes. Once the human supervisor no longer requires this information, s/he can inform the agent's processes to cease transmission.

Specification Interaction

Various processes require information from the human supervisor before they can begin processing. Such a process may be a path planning process for which the human supervisor must specify the starting, intermediary and goal points of the path. The *specification interaction* provides the human supervisor with the means to interactively specify information pertinent for a process' execution.

Application The *control interaction* and/or the *specification interaction* permits the human supervisor to create the general path for the agents to follow. The *control interaction* also permits the human supervisor to stop the progress of the agents when they are approaching a dangerous situation, for instance if the two manipulatory agents have not properly reconfigured, the human can stop them before they run into an obstacle. The *request interaction* permits the human supervisor to obtain raw sensory and image data as well as processed data from the agent's processes which is used to monitor the execution of the task.

2.3.3 Processing Level

There exist situations where a process may be incapable of reaching a satisfactory decision based on ambiguous information and must therefore request assistance from the human supervisor. There are also situations when a process will formulate a correct decision in a local context but the decision will be unsatisfactory in the global scheme, therefore the human supervisor should either assist with the decision making process or override the decision formulated by the process.

While observing an agent's actions based on a particular process, the human supervisor may determine the process is formulating an incorrect interpretation. The human supervisor may then intervene in the process to clarify the information, override a decision or allow it to continue with it's processing. The *processing level* permits the human supervisor to aid a process when it is unable to arrive at a decision and to rectify incorrect decisions deduced by a process. This level of interaction will protect the agents from entering failure states.

Application The *processing level* permits the human supervisor to interact with the processes during the task execution. This interaction may be requested by a particular process or the human supervisor may determine through the monitoring of information that assistance is necessary.

2.3.4 Data Level

It is known that mechanical devices fail from time to time, and that the automatic reconfigurations for such failures are not always successful. Therefore, the human supervisor should be provided with the means to reconfigure the system. The *data level* of the mediation hierarchy permits the human supervisor to reconfigure the system when automatic reconfigurations have failed and hence ensure correct data is passed up through the system.

Application The *data level* permits the human to identify hardware failures such as a stuck ultrasound sensor. When such a failure is detected, the process is notified to ignore that particular sensor. Also, it is through this type of interaction that the human is able to reset the odometry readings based upon the localization information from the observation agent.

3 Behaviours and Task Description Language

In this section we provide additional information regarding the basic control strategies associated with available components of the system. These strategies comprise a set of *elementary processes* associated with each sensor and actuator corresponding to the basic perceptual capabilities of the sensors and the basic motion modes of the actuators, some of which are described in the following section. Each motion mode corresponds to a particular control law describing the manner in which the commands are generated. Similarly, sensors have associated procedures for data acquisition and extraction. Behaviours are particular couplings between elementary processes. These are grouped together and modeled in terms of finite state machines (FSM), following the notation of Supervisory Control Theory of Discrete Event Systems (DES)[34].

3.1 Discrete Event Systems Model

We associate with each sensor and actuator a suite of elementary processes which are grouped together to form the so called *fundamental process.* The states of the fundamental process correspond to a basic motion/sensing strategy and the transitions between the states are created by external events (depending on the task) or by the successful completion or failure of the strategy. Each fundamental process is modeled as a DES system as described by its associated automaton $G = (Q, \Sigma, \delta, q_0, Q_m)$ in the fashion defined in [19]. Events are classified into two categories: *controllable events* Σ_c(the ones which may be prevented from occurring or forced to occur) and *uncontrollable events* Σ_u(those which may not be prevented from occurring). Controllable events identify basic strategies (which run continuously if invoked, until successful termination or failure) or just a single locus of computation. Tasks are expressed as networks of elementary

strategies/processes formed using a set of composition operators which may change dynamically during execution. The intuition behind these operators is identical to that in Lyons [24]. The basic operators used in our examples are defined as follows.

Composition operators

- **Sequential composition** $P = R\ ;\ S$. Process P behaves like R until R terminates and then P behaves like S. P terminates when S terminates with the termination status of S.
- **Concurrent composition** $P = R \parallel S$. Process P behaves like R and S running in parallel. P terminates successfully if both processes terminate successfully or fails if both processes fail. If R and S share events, a communication link is established between them.
- **Conditional composition** $P = R_{<v>} : S(v)$. Process P behaves like R until R terminates successfully computing value v which is used to initialize process S. If R fails the entire composition fails.
- **Disabling composition** $P = R \sharp S$ Disabling composition is similar to parallel but if one of the processes fails the other processes also fails.

Expressions where primitives are from the set of all controllable events, Σ_c, and operators are from the set of composition operators, the expressions are referred to as "sentences in a task specification language". The main objective of this approach is: given a sentence in the task description language, synthesize a finite state machine controller - DES supervisor[1]. The DES supervisor then monitors the execution of the task, invoking correct strategies and monitoring system responses. This DES supervisor operates in parallel with the human supervisor which may override any of the decisions made by the DES supervisor. A more in depth description of this approach can be found in [17].

3.2 Global task planner and Task Specification Language

The Task Specification Language and the DES supervisor model the discrete event interface and control between the physical components of the system which are characterized by the elementary continuous control and sensing strategies. However, for more complicated tasks such as occur in multi-agent systems, a task decomposition phase is necessary and subtask assignment must occur for individual agents. This stage is currently completed by the human supervisor, although we are currently incorporating a higher-level symbolic planner to assist the human.

Our approach is demonstrated through an in depth description of some of the elementary strategies and models of the *fundamental processes*. We demonstrate the idea of representing tasks as networks of processes in a few examples. The

[1] Supervisor in this section is a FSM operating in a closed loop with the system. This is to be distinguished from the human supervisor mentioned in previous sections.

following subsection describes the motion modes and perceptual strategies used during the execution of more complicated tasks.

3.3 Elementary Processes

3.3.1 Local Maneuvering

The central question for the control of a mobile base is how to move it from one location to another in a structured or unstructured environment. This problem involves issues of path planning, motion planning and localization given available sensory information and/or a priori knowledge. Within this work we explore both path planning and control issues, while assuming that the global goal/objective is determined a priori. In order to address the control issues properly, we must consider the unique property of the wheeled mobile platforms, captured by nonholonomic constraints.[2] In the presence of these constraints the design of a control law, which brings the robot from an arbitrary initial configuration to a final configuration is an intrinsically nonlinear control problem.

We have implemented two strategies for local maneuvering. One utilizes artificial potential fields to steer the mobile base in a closed loop fashion, while the other contructs an R-geodesic path which the base follows in an open loop manner.

Potential Fields In addressing control issues, one possibility is to adopt the artificial potential field method [16]. The method provides us with an incremental on-line generated holonomic path[3], which is modified using simple projection strategy for nonholonomic robots [11], alternatively we may use a feedback-linearized control law [42].

The role of the planner is to generate trajectory commands in order to reach a desired location from an arbitrary initial configuration. While doing so with additional sensory information the objective is to take into account unexpected occurrences of obstacles in the path and steer around them. We adopt an artificial potential field method originally introduced by where the goal $X_g = (x_g, y_g)$ is represented as an attractive potential field:

$$U_a(X) = \frac{k_p}{2}(X_g - X)^2$$

In order to achieve the desired goal we need to exert a force, which is proportional to the gradient of the given potential function $F = -\nabla_X (U_a(X))$. The obstacles are represented by a repulsive hyperbolic potential function:

$$U_{rep}(X) = \begin{cases} \frac{k_{rep}}{\gamma}(\frac{1}{\eta(X)} - \frac{1}{\eta_0})^\gamma & \text{if } \eta(X) \leq \eta_0 \\ 0, & \text{otherwise} \end{cases}$$

[2] An example of a nonholonomic constraints is a wheeled mobile robot moving on the plane under perfect rolling constraint. (i.e. robot are unable to move sideways).

[3] A holonomic planner does not consider the nonholonomic constraints while generating two successive configurations of the base, i.e. treating the base as omnidirectional.

where coefficient $\gamma > 2$, η is a distance function to the obstacle and η_0 is obstacle's influence range. The desired velocity $\dot{X}_d$ at each instance of time is derived from the holonomic path planner. The positional part $\dot{X}_d$ and rotational $\dot{\theta}_d$ part of the desired velocity with artificial potential field is

$$\begin{aligned} \dot{X}_d &= -\nabla_X \left(U_a(X) + U_r(X)\right) \\ \dot{\theta}_d &= \arctan(\dot{x}_d, \dot{y}_d) - \theta \end{aligned} \tag{1}$$

The platform can instantaneously execute only linear motion along it's main axis due to the nonholonomic constraints, while the superimposed turning rate forces this linear motion to be aligned with the field flow. The resulting commands representing desired linear and turning velocity settings then are:

$$\begin{aligned} u_1 &= k_p(\dot{x}_d \cos\theta + \dot{y}_d \sin\theta) \\ u_2 &= k_\theta(\arctan(\dot{x}_d, \dot{y}_d) - \theta) \end{aligned} \tag{2}$$

where $\dot{x}_d$ and $\dot{y}_d$ are given by equation (2) and gains k_p and k_θ are used to weight the two input commands. By adopting this control strategy for point to point motion, we associate the following motion mode for the mobile base.

GoTo This mode implements the above derived control law (1) provided that the desired velocity vector $\dot{X}_d = (\dot{x}_d, \dot{y}_d)$ is computed from the overall potential field.

GoToMarch This control law generates commands for the mobile base while marching in parallel formation (next to each other), while keeping the distance from the midpoint constant. The desired angular (u_{2m}) and linear velocity (u_{1m}) of the midpoint are computed based on control law (1). The linear velocity of the mobile base is then:

$$\begin{aligned} u_1 &= u_{1m} + u_{2m} * radius \\ u_2 &= u_{2m} \end{aligned} \tag{3}$$

where *radius* is the distance of the base from the reference midpoint.

GoToHeading Since we cannot guarantee the final goal configuration and the desired platform heading using the *GoTo* strategy, another basic strategy which we associate with the mobile platform is pure rotation so that the desired heading is reached θ_d. The desired angular velocity is in this case $u_2 = k_\theta.(\theta - \theta_d)$.

The "motion modes" use proportional feedback laws, servoing on goals while avoiding obstacles. The goal and obstacles can be supplied either by perceptual processes, by the DES supervisor process, or by the human supervisor.

R-geodesic Path Generation Strain energy stored in an ideal linear elastic string is proportional to the square of the deformation. With such an elastic system, minimum energy implies minimum deformation. Based upon this simple statement, we are going to use an imaginary elastic string and two pairs of circles with radius R as auxiliaries to find the shortest smooth R-geodesic path [13]. R is the minimum turning radius that the mobile robot can perform.

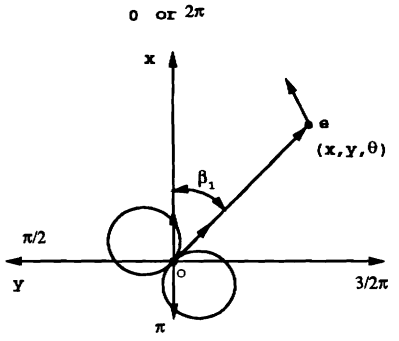


Figure 11: Undeformed elastic string

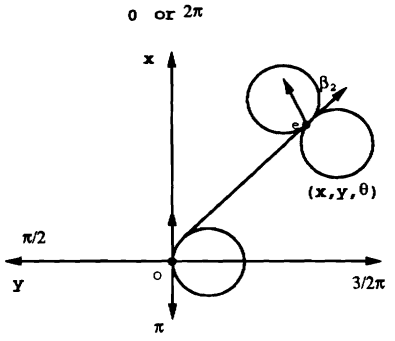


Figure 12: Starting arc

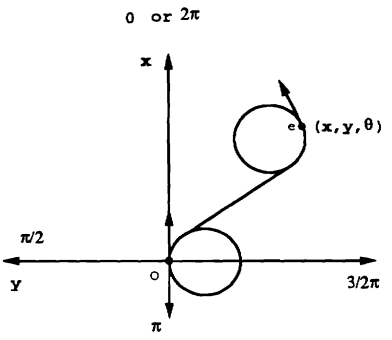


Figure 13: Ending arc

For convenience of illustration, we define $(x_i, y_i, \theta_i) = (0, 0, 0)$ as the starting configuration and (x_e, y_e, θ_e) as the ending configuration, as shown in Figure 11. The elastic string is initially unstretched and of length $l = \sqrt{x_e^2 + y_e^2}$. First, attach one pair of circles at point o with the string as the common tangent. These two circles are of radius R. Through a rotation $|\beta_1|$ $(< \pi)$ about point o, the direction of the tangent will match the starting heading angle. After this rotation, the string has been deformed into a new shape, namely an arc and a new straight line as shown in Figure 12. Another pair of circles can be attached to point e which share the straight line as common tangent. By the same token, a rotation $|\beta_2|$ $(< \pi)$ about point e is needed to match the direction of the tangent with the ending heading angle. Eventually, with these two rotations, the original straight string is deformed into three segments - one arc, one straight line and one arc.[4] All three segments are joined together smoothly. Figure 13 shows this R-geodesic path.

3.3.2 Simple active sensing strategies

Each subtask described in the introduction to section 2.2 necessitates its own *active sensing strategy*. We now describe the approaches adopted in the context of the current scenario.

Localization of agent A using ultrasound Assume the existence of a priori knowledge of two perpendicular walls w_1 and w_2 in the environment. Let l_{min} denote the minimum length of a planar feature detectable by the ultrasound

[4]For the cases which β_1 or β_2 is zero, there will be only one or two segments in the path

modality. Then localization of agent A warrants the traversal of a trajectory t such that the lengths of the projections of t on w_1 and w_2 exceed l_{min}. The simplest such trajectory is a straight line path of length l which intersects the extension of w_1 at an angle β such that $l\cos\beta > l_{min}$ and $l\sin\beta > l_{min}$. Assuming an initial coarse estimate of the agent's location and orientation, such a trajectory may be executed.

Upon completion of this motion, the two planar features extracted by the ultrasound modality are assumed to correspond to w_1 and w_2 respectively. A least squares fit yields an estimate for the location of agent A both in Cartesian $X-Y$ position and in orientation. The accuracy of this approach is under investigation.

Trajectory verification Once all agents have been localized, the human supervisor chooses the way points leading to the goal and then a preliminary trajectory is proposed by one of the path planning methods. The observation agents should verify that this path is unobstructed before the manipulatory agents traverse it.

In the current scenario, the trajectory is verified by agent A attempting a traversal and using the IPP modality for obstacle detection. In the event of obstacle detection, the light-striper is be recruited to scan to extract obstacle extent. Agent A also employs the ultrasound modality to ensure that the pathway is wide enough at all points to accommodate the two manipulatory agents carrying the object.

Progress monitoring Once an obstacle-free path has been identified, progress of the sensor-impaired agents should be monitored for possible intervention by the human supervisor. A sufficient active sensing strategy to accomplish this subtask comprises moving agent B behind manipulatory agents C and D, and then following their trajectory. The human supervisor may monitor progress by examining the image stream emanating from one of the cameras.

3.3.3 Coordination of the mobile manipulators

This subsection describes some preliminary results of the two mobile manipulators cooperating towards a common goal. In this case, we set a common goal as transporting a large object along a desired trajectory. Roughly speaking, there are two different approaches for achieving the coordinated task. One approach is to treat a closed kinematic chain involving both mobile manipulators plus the object as one system and to design a controller for the whole system so that the object follows a specified trajectory. A simulation example of such a scenario is shown in Figure 14 where a two degree-of-freedom planar manipulator is mounted on each mobile platform. In the example, we assume that the two end-effector points are rigidly connected via a grasped object which is represented as a point in the figure. The desired trajectory of the object is specified along a circular arc which is depicted by a dotted line. The whole system has eight degrees-of-freedom and is subjected to three kinematic constraints: two nonholonomic and one holonomic. In

general, by choosing five output equations one can achieve the input-output feedback linearization and decoupling of the system. In Figure 14, we have chosen the output equations as the position and orientation of the object, the orientation error between the two platforms, and the separation of the two platforms. The figure shows that the object successfully follows the desired path as both platforms move forward along concentric arcs.

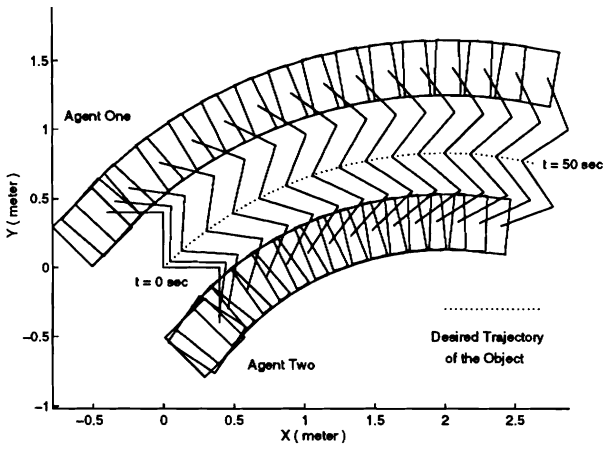


Figure 14: A simulation result of the coordinated mobile manipulators (Case I): Traversal of a trajectory in marching formation using centralized control.

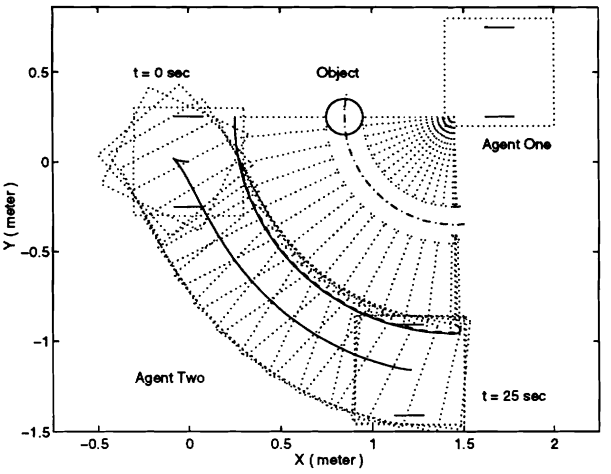


Figure 15: A simulation result of the coordinated mobile manipulators (Case II): Change of configuration from marching to serial.

An obvious disadvantage of this approach is that all measurements and controls are assumed to take place in the same bandwidth. This assumption is very difficult to fulfill in practice due to the following reasons. First, in general a manipulator and a mobile platform have different dynamic characteristics, *i.e.,* the manipulator is used to achieve a fine, fast positioning while the response of the mobile platform is fairly slow and is only suitable for gross motions. Secondly, this approach is susceptible to the complexity of the system. In other words, as the system becomes complex, *e.g.,* more joints or more mobile manipulators involved, the computational expense increases dramatically.

In order for the system to be flexible and computationally manageable, a more decentralized approach is desirable as opposed to the centralized approach illustrated above. A decentralized approach implies that the mobile manipulators should be able to execute the tasks based on a limited amount of information exchange between the agents. This may be achieved in a variety of ways. For instance, a reliable force sensor at the end-point will be helpful to infer the "intention" of the partner agent. Also the desired trajectory may be carefully designed so that it makes the coordination between the agents easier from various perspectives, *e.g.,* the nonholonomic constraints, the workspace of the manipulator, or the complexity of the controller.

The next simulation demonstrates execution of one of the subtasks mentioned in section 2.1, *i.e.*, change of the configuration of the two agents (Figure 15). In this example, the configuration change is executed by moving only one of the mobile manipulators (agent C), with the other platform (agent D) stationary and swinging its manipulator as agent C maneuvers[5]. With this scheme, the two mobile agents are more loosely coupled than in the previous case. A difficulty with this case is that, due to the presence of nonholonomic constraints, the controller has to be switched from one platform to the other in order to align the two platforms in parallel with each other; this causes a small drift of the end-point towards the end of the trajectory as shown in Figure 15. The small error at the end-point is assumed to be compensated by a stiffness type of control on one of the mobile agents.

3.4 DES models of fundamental processes

We now outline models for some of the fundamental processes as well as the human supervisor. Each fundamental process is modeled as a DES system, in the fashion described at the beginning of this section: having a set of controllable and uncontrollable events associated with it. The set of controllable events corresponds to the set of available control/perceptual strategies, while the set of uncontrollable events represents responses of the system, reporting successful completion, interruption of the strategy or a change of a set-point, or a global variable associated with the process. We distinguish two kinds of interrupts; an internal interrupt occurs upon failure or violation of some constraints, and an external one occurs when some other process triggers a change of a strategy (e.g. change of state of the fundamental process). The examples of models of the fundamental processes for the mobile base, the IPP stereo sensor and the human supervisor are shown in Figure 3.4. Any attempt from the human supervisor to intervene with the system is modeled as an uncontrollable event followed by a command, corresponding to the type of request. This is expressed in terms of a finite state machine as in Figure 3.4.

Since we are dealing with a distributed system, the communication between processes is modeled via events. If two invoked processes share an event, a communication link is established between them.

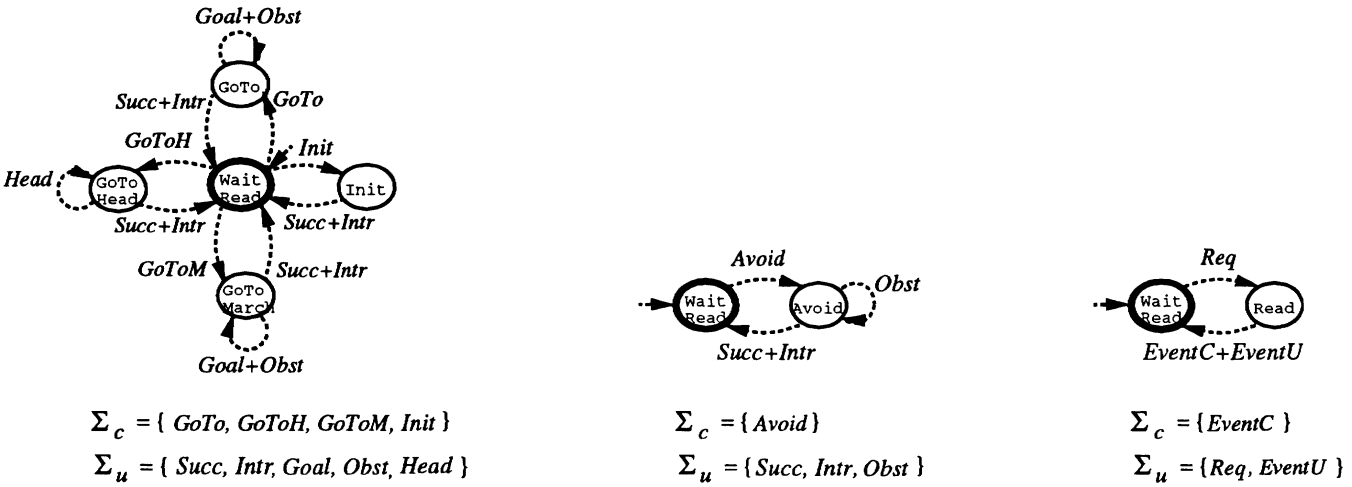
The following outlines a few examples of simple tasks which illustrate the idea of the task specification language.

Example 1. The task of reaching a prespecified goal, *Goal*, while avoiding obstacles may be described in the task specification language as follows:

$$GoTo(Goal) \parallel Avoid$$

The processes share an event *Obst*; therefore a communication link is established between them and the obstacle detection process sends the information about obstacles to the *GoTo* motion mode of the mobile base's fundamental

[5]The details of the coordination scheme employed for agent C can be found in [41, 42]



Σ_c = { *GoTo, GoToH, GoToM, Init* }
Σ_u = { *Succ, Intr, Goal, Obst, Head* }

Σ_c = { *Avoid* }
Σ_u = { *Succ, Intr, Obst* }

Σ_c = { *EventC* }
Σ_u = { *Req, EventU* }

Figure 16: Fundamental models of the mobile base, the IPP stereo sensor and the human supervisor.

process. In this case there is no need for DES supervisory control, since the composition of the two processes is parallel.

Example 2. The task of localizing the mobile base using ultrasound sensing and the initialization of the state of the mobile base is as follows:

$$Localize < (x, y, \theta) > \ : \ Init < (x, y, theta) >$$

The conditional composition is often used in cases where the computation of one process provides a parameter for a particular control strategy of another process. The parameter is transmitted to the other process via the DES supervisor. This effect can be alternatively modeled by a shared event between the two processes, requiring a communication link between them.

Example 3. The task of two mobile bases cooperatively marching to a given destination *Goal* in a parallel formation while avoiding obstacles is expressed in the task specification language as follows:

$$(GoTo_A(Goal) \parallel Avoid_A) \sharp (GoTo_B(Goal) \parallel Avoid_B)$$

where indexes A and B represent the two agents engaged in the task. A detailed description of the DES supervisor synthesis process can be found in [17].

4 Experiments

We are testing the scenario described in Section 1.4. The components which we have tested and evaluated include:

- The agents marching in formation while avoiding detected obstacles.
- The manipulatory agents carrying an object while marching.
- The use of the ultrasound to detect walls and corners as well as to localize the observation agent equipped with that sensing modality.
- The human supervisory aspects such as the ability of the human supervisor to monitor the actions and processes of the system, as well as to specify paths.
- The algorithm for the global task specification and the algorithm which translates the task specification into the task supervisor.

See [1, 17, 28, 29, 41, 42] for details of the workings of these components, as well as test results and evaluations.

The components currently being tested and evaluated include:

- The reconfiguration of the mobile manipulators without dropping an object.
- The localization of observation agent B and manipulatory agents C and D by means of observation agent A's ultrasound and light-striping modalities.

Once we have completed these final tests, the components will be combined and the full scenario tested.

5 Conclusions

Based on the experience gained in designing, testing, and integrating the modules in the experimental system described above, we cite the following conclusions:

- The human agent is a necessary component in the successful operation a system for multiagent cooperative material handling in an unstructured indoor environment. Our approach makes fundamental use of human agents' expertise for aspects of task planning, task monitoring, and error recovery.
- The partitioning of the robotic agents into two classes — mobile observers, and mobile manipulators — provides very useful degrees of freedom in the experimental design. This flexibility allows us to build systems which have more extensive ability to observe manipulatory agents working either in close quarters or near obstacles.

- Our application of potential functions has shown the value of this approach in controlling the motion of multiple vehicles in obstacle-laden environments. The motions obtained exhibit good stability and smoothness characteristics.
- One of the salient aspects of our DES supervisory control system is its ability to arbitrate between conflicting control subtasks within a given task requirement.
- At present there is no single sensing modality which is versatile enough to provide sufficient data about the environment for the execution of all tasks within our scenario. Sensory integration between multiple modalities is therefore essential.

References

[1] Julie A. Adams and Richard Paul. Human-managed, hierarchical control of multiple mobile agents. In *Proceedings of the 1994 IEEE Conference on Decision and Control.* IEEE, Dec. 1994.

[2] Julie A. Adams and Richard Paul. Human management of a hierarchical control system for multiple mobile robots. In *Proceedings of the 1994 IEEE International Conference on Systems, Man and Cybernetics.* IEEE, Oct. 1994.

[3] R. Arkin, T. Balch, and E. Nitz. Communication of behavioral state in multi-agent retrieval tasks. In *IEEE Intl. Conference on Robotics and Automation*, pages 588–593, 1993.

[4] B. Barshan. Orientation estimate for mobile robots using gyroscopic information. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, volume 2, pages 1615–1621, September 1994. Munich, Germany.

[5] B. Barshan and H. F. Durrant-Whyte. An inertial navigation system for a mobile robot. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pages 2243–2248, July 1993. Yokohama, Japan.

[6] Craig Becker. *Internet Path Planner Server Protocol.* Stanford University, January 1994.

[7] M. Betke and K. Gurvits. Mobile robot localization using landmarks. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 135–142, May 1994.

[8] R. A. Brooks, P. Maes, M. Mataric, and G. Moore. Lunar base construction robots. In *IROS-90, IEEE International Workshop on Intelligent obots and Systems*, 1990.

[9] P. Caloud, W. Choi, and J. Latombe. Indoor automation with many mobile robots. In *IROS-90*, pages 67–72, 1990.

[10] P. Chiacchio, S. Chiaverini, L. Sciavicco, and B. Siciliano. Task space dynamic analysis of multiarm system configurations. *The International Journal of Robotics Research*, 10(6):708–715, December 1991.

[11] A. De Luca and G. Oriolo. Local incremental planning for nonholonomic mobile robots. In *Proceedings of 1994 International Conference on Robotics and Automation*, pages 104–110, San Diego, CA, May 1994.

[12] K. S. Decker and V. R. Lesser. The analysis of quantitaive cooridination relationships. Coins 91-83, Department of Computer Science, University of Massachusets, Amherst, 1992.

[13] L. E. Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, (79):497–516, 1957.

[14] Christopher W. Geib. *Draft: Representing Actions for Planning.* PhD thesis, University of Pennsylvania, Philadelphia, PA, December 1994.

[15] Christopher W. Geib. The intentional planning system: Itplans. In *Proceedings of the Artificial Intelligence Planning Systems Conference*, pages 55 – 64, 1994.

[16] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, 5(1):90–98, 1986.

[17] J. Kosecka and R. Bajcsy. Alternatives for the supervisor design. In *submitted to IROS'95.*

[18] J. Košecka and R. Bajcsy. Cooperation of visually guided behaviors. In *Proceedings ICCV 93*, Berlin, Germany, May 1993.

[19] J. Košecka and R. Bajcsy. Discrete event systems for autonomous mobile agents. In *Workshop on Intelligent Robot Control, Zakopane*, July 1993.

[20] R. Kuc. A spatial sampling criterion for sonar obstacle detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7), July 1990.

[21] R. Kurazume and S. Nagata. Cooperative positioning with multiple robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1250–1257, May 1994.

[22] J. J. Leonard. *Directed Sonar Sensing for Mobile Robot Navigation.* PhD thesis, University of Oxford, 1990.

[23] G. Lucarini, M. Varoli, R. Cerruti, and G. Sandini. Cellular robotics: Simulation and hw implementation. In *IEEE Robotics and Automation, Intl. Conference*, 1993.

[24] Damian M. Lyons. Fundamentals of ∇f - part i: The basic ∇f model. Technical report, Philips Laboratories, 1989.

[25] H.A. Mallot, H.H. Bulthoff, J.J. Little, and S. Bohrer. Inverse perspective mapping simplifies optical flow computation and obstacle detection. *Biological Cybernetics*, 64:177–185, 1991.

[26] R. Mandelbaum. *Sensor Fusion for Mobile Robot Localization, Exploration and Navigation.* PhD thesis, University of Pennsylvania. In preparation.

[27] R. Mandelbaum and M. Mintz. Active sensor fusion for mobile robot exploration and navigation. In *Proceedings of the Workshop on Intelligent Robotic Systems: Sensor Fusion VI, Vol. 2059, Sponsored by SPIE*, September 1993.

[28] R. Mandelbaum and M. Mintz. Sonar signal processing using tangent clusters. Technical report, University of Pennsylvania, 1994. In preparation.

[29] R. Mandelbaum and M. Mintz. Sonar signal processing using tangent clusters. In *Proceedings of the OCEANS '94: special session on Automated Unmanned Vehicles*, September 1994. Brest, France.

[30] M. Mataric. Minimizing complexity in controlling a mobile robot population. In *IEEE Intl. Conference on Robotics and Automation, Nice*, pages 830 – 835, 1992.

[31] F. R. Noreils. An architecture for cooperative and autonomous mobile robots. *Proceedings of the IEEE International Conference on Robotics and Automation*, 3:2703 – 2710, May 1992.

[32] F. R. Noreils. Coordinated protocols: An approach to fromalize coordination between mobile robots. In *IEEE Intl. Conference on Intelligent Robots and Systems*, pages 717–724, 1992.

[33] L. Parker. Designing control laws for cooperative agents. In *IEEE Intl. Conference on Robotics and Automation*, pages 582–587, 1993.

[34] P. J. Ramadge and W. M. Wonham. The control of discrete event systems. *Proceedings of the IEEE*, 77(1):81–97, January 1989.

[35] Craig W. Reynolds. Flocks, herds and schools: A distributed behavioral model. In *SIGRAPH*, pages 25 – 33, 1987.

[36] Thomas Sheridan. *Telerobotics, automation, and human supervisory control.* MIT Press, Cambridge, Mass., 1992.

[37] W. B. Thompson, T. C. Henderson, T. L. Colvin, L. B. Dick, and C. M. Valiquette. Vision-based localization. In *Proceedings of the 1993 Image Understanding Workshop*, pages 491–498, April 1993.

[38] T. Ueyama and T. Fukuda. Self-organization of cellular robots using ramdom walk. In *IEEE Intl. Conference on Robotics and Automation*, pages 595–600, 1993.

[39] Chau-Chang Wang. Local path planner. University of Pennsylvania. Personal Communication, Sept. 1993.

[40] Susan Whalley. The human computer interface - designing for process control. In *IEEE International Conference on Systems, Man and Cybernetics*, pages 827 – 832. IEEE, 1992.

[41] Y. Yamamoto and X. Yun. Coordinating locomotion and manipulation of a mobile manipulator. *IEEE Transactions on Automatic Control*, 39(6):1326–1332, June 1994.

[42] Yoshio Yamamoto. *Control and Coordination of Locomotion and Manipulation of a Wheeled Mobile Manipulator.* PhD thesis, University of Pennsylvania, Grasp Laboratory, 1994.

Sonar Signal Processing using Tangent Clusters

Robert Mandelbaum *Max Mintz*

General Robotics and Active Sensory Perception (GRASP) Laboratory
Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104

Abstract - **We describe a novel approach to the extraction of geometric features from sonic data. As is well known, a single sonar measurement using a standard POLAROID^{TM} sensor, though yielding relatively accurate information regarding the *range* of a reflective surface patch, provides scant information about the location in *azimuth* or *elevation* of that patch. This lack of sufficiently precise localization of the reflective patch hampers any attempt at data association, clustering of multiple measurements or subsequent classification and inference. The problem is particularly apparent in uncertain environments with unknown geometry, such as is found underwater. Moreover, the underwater environment precludes the usual (office-environment) simplification of two-dimensionality. We propose a multi-stage approach to clustering which aggregates sonic data accumulated from *arbitrary* transducer locations in an *on-line* fashion. It is computationally *tractable* and *efficient* despite the inherent exponential nature of clustering, and is *robust* in the face of noise in the measurements. It therefore lends itself to applications where the transducers are *fixed* relative to the mobile platform, where remaining stationary during a scan is both impractical and infeasible, and where deadreckoning errors can be substantial. The approach may be used both for map-building during exploration and for feature identification during navigation.**

I. INTRODUCTION

A. Physics of Sonars

We describe a multi-stage approach to clustering of, and feature extraction from, ultrasound measurements obtained using standard POLAROID^{TM} sensors. We assume at the lowest level that each sonar measurement is generated by an element of the set of basic features $\mathcal{P}$ = {planar reflective patches, outer diffractive corners, inner reflective corners}. All that one can infer from a *single* measurement is the existence of an element of $\mathcal{P}$ at the distance r somewhere along the boundary of the transmitted cone truncated at range r. In the case of *planar* reflective patches, the patch has *orientation tangential to the acoustic wave fronts.* In 3D, the region of uncertainty of the location of the reflective patch forms a section of the surface of a sphere, centered at the transducer and of solid angle $2\pi(1-\cos\alpha)$ steradians, where α is the half-width of the ultrasonic emission cone. For standard Polaroid transducers, $\alpha \approx 15°$ [9, 11]. In 2D, the cone is a wedge and the region of uncertainty is an arc. See Figure 1. Note the distinction drawn between a reflective *patch* and the underlying reflective *surface* of which the patch is a part.

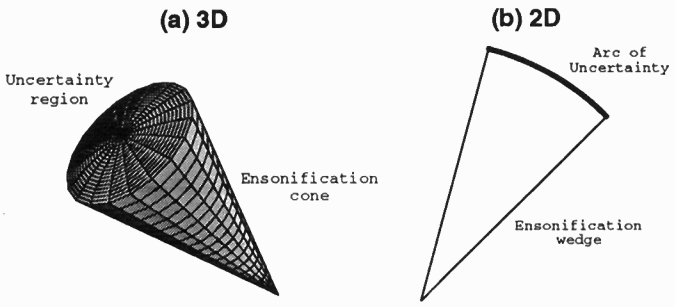


Figure 1. a) Region of azimuthal and latitudinal uncertainty for location of reflective patch. b) Region of azimuthal uncertainty in 2 dimensions

For *multiple* measurements generated by the same planar surface, all arcs, in the noise-free 2D case, share a common tangent; corners (both inner reflective and outer diffractive) induce measurements whose arcs intersect at the corner. In general, a smooth curve defined parametrically by $(s_x(t), s_y(t))$, s_x, s_y piecewise differentiable, will induce measurements such that the arc corresponding to each measurement intersects the curve at a point $(s_x(\hat{t}), s_y(\hat{t}))$ where both curve and arc share the tangent of orientation $(\dot{s}_x(\hat{t}), \dot{s}_y(\hat{t}))$ at that point.

B. Clustering and Parameter Estimation

In the 2D noise-free case, ultrasonic data association is equivalent, therefore, to finding subsets among the set of all measurements such that in each subset, all measurements correspond to arcs which either intersect at a point, or share a common tangent with a sought feature described parametrically by $(s_x(t), s_y(t))$; when the feature is a planar curve, all measurements in a cluster share a *common* tangent[1]. The data from members of each cluster are used to estimate the parameters of the underlying feature. In general, the problem of finding arbitrary subsets is exponential in nature. In order to reduce the problem to the polynomial domain, and hence render it computationally tractable, other information must be utilized or some compromises made.

[1] In 3D, the feature being sought is described parametrically by $(s_x(u,v), s_y(u,v), s_z(u,v))$. All truncated cones in a cluster must either intersect along a line or share common tangential planes with this surface; when the feature is a planar surface, all truncated cones share a *common* tangent plane.

C. Specifications of an algorithm

We would like an algorithm which

- aggregates sonic data accumulated from *arbitrary* transducer locations,
- performs the clustering *on-line*, rather than in a batch fashion,
- is computationally *tractable* and *efficient* despite the inherent exponential nature of clustering,
- is *robust* in the face of noise in the measurements,
- is *precise* in that it converges in a statistical sense to ground truth, and
- has high *soundness* and *completeness* factors, where a *soundness* factor, in this setting, indicates what fraction of detected features really exist, and a *completeness* factor indicates the proportion of extant features which are detected by use of the algorithm.

D. The Issue of Representation

The representation chosen for the accrued sonar measurements is crucial for the attainment of the above specifications. In general, it is advantageous for the algorithm to transform all data to a vector space representation in which features are dense clusters of some easily recognizable shape; in such a space, the search for features is greatly facilitated. The naive representation using truncated cones or arcs in space is *not* amenable to this type of clustering [12].

E. Previous approaches to clustering

A common approach to ultrasound data processing uses each measurement to update some form of *occupancy grid* [1, 2, 7, 8, 10, 14]. While this approach is useful for low level tasks such as obstacle avoidance, feature extraction involves the application of edge detection or similar procedures to the grid; much inherent orientation information has been lost and must subsequently be recovered. In [4, 5] Kalman filtering is used to extract geometric features, though the sensor model employed for ultrasonic sensors does not reflect the underlying physics of the devices very well. Reference [3] uses a multiple hypothesis framework for dynamic world modeling. Instead of avoiding the exponential growth, the approach relies rather on pruning to bring the growth to within manageable limits. In [13], the sensor itself is modified and explicit use is made of orientation information. The results are impressive. However, though the approach taken there is similar, in the 2D case, to the early stages of clustering described here, [13] does not address the full generalized clustering problem nor do they address issues of computational efficiency. Generalization to 3D would require further modifications to the sensor.

II. CLUSTERING USING COMMON TANGENTS

A. Overview

A common deficiency of many ultrasound data clustering algorithms is that they do not make sufficient use of the *orientation* information inherent in each measurement emanating from a planar surface: When a planar reflective patch returns ultrasonic energy to the transducer, we know not only that the patch is located somewhere within the cone of ensonification at range r, but also that *the planar patch is oriented tangential to the acoustic wave front.*

We propose a multi-stage approach to clustering. We assume at the lowest level that each sonar measurement is generated by an element of the set of basic reflective patches $\mathcal{P} = \{$planar reflective patches, outer diffractive corners, inner reflective corners$\}$.

The first stage of clustering consists of finding groups of 3 measurements whose truncated cones all share a common tangential plane-segment. In 2D, we look for pairs of measurements whose arcs share a common line-segment. Section C. delineates this procedure and describes the representation used to facilitate the tripleting or pairing operation.

The second stage of clustering, described in Section D., consists of associating multiple plane-segments (line-segments in 2D) into groups corresponding to larger underlying features such as planar surfaces and corners in the environment.

Finally, Section E. describes approaches adopted for the parameter estimation phase: recursive least total squares (RLTS) fitting for planar surfaces, and recursive center of gravity estimation for corners.

B. Representation

The fundamental representational unit used by the algorithm, then, is the plane-segment or the line-segment. For simplicity of exposition, the remainder of this section will focus on the 2D case; the representation and algorithm are easily extended to the 3D case.

We use a redundant *normal parametrization* representation ([6], page 336) for line-segments: the line of which the segment forms a part is represented by the pair (ρ, ϕ) where ρ is the perpendicular distance of the line from some fixed $(0,0)$ point; ϕ is the inclination of the normal to the line relative to the x-axis in a counter-clockwise direction[2]. This pair is supplemented with parameters to specify the locations of the endpoints. Only two extra parameters are necessary, but for simplicity we use three: two for the location of the center-point of the line-segment (c_x, c_y) and one for the length of the segment l. Hence, each line-segment is represented by the 5-tuple $(\rho, \phi, c_x, c_y, l) \in \Gamma \times \Phi \times C_x \times C_y \times L$.

C. The first clustering stage: Finding common tangents

In this section we discuss the problem of finding a matching arc for a new datum arc from among an already existing database of previously unmatched arcs. A brute force search through the entire database is inefficient not only because of the high *expected number* of matches to be attempted, but also because *each* test for a common tangent is computationally expensive (consider, for example, the computation involved in evaluating ϕ below).

Let an arbitrary arc be represented by the 4-tuple $(x, y, \theta, r) \in X \times Y \times \Theta \times R$ where (x, y, θ) represents the pose of the transducer and r represents the range measurement obtained at this pose. The angle subtended by the arc is assumed to be a constant 2α, where α is the half-angle of the cone of ensonification.

Consider two circles (x_1, y_1, r_1) and (x_2, y_2, r_2). Without loss of generality, let $r_1 \geq r_2$. Then the circles share exactly two common tangents provided $r_2 + d > r_1$, where d is distance between centers $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$; the parameters of the common tangents are

[2] In 3D, the plane of which the plane-segment forms a part may be similarly represented by the triplet (ρ, ϕ, ω), where ρ is the perpendicular distance of the plane from the $(0,0,0)$ point, and ϕ and ω are respectively the azimuth and latitude of the normal to the plane.

$$\phi = \arctan\left[\frac{y_1 - y_2}{x_1 - x_2}\right] \pm \arcsin\left[\frac{r_1 - r_2}{d}\right] - \frac{\pi}{2} \quad (1)$$
$$\rho = r_2 + x_2\cos\phi + y_2\sin\phi.$$

Hence, in order to test whether two arcs $a_1 = (x_1, y_1, \theta_1, r_1)$ and $a_2 = (x_2, y_2, \theta_2, r_2)$ share a common tangent, we calculate the ϕ parameters of the common tangents of the two *circles* of which the arcs form a part; if $\phi \in \Theta_\cap = ([\theta_1 - \alpha, \theta_1 + \alpha] \cap [\theta_2 - \alpha, \theta_2 + \alpha])$, a match has been found, and the remaining parameters of the common line-segment supported by a_1 and a_2 may be computed:

$$l = \sqrt{d^2 - (r_1 - r_2)^2}$$
$$\begin{pmatrix} c_x \\ c_y \end{pmatrix} = \begin{pmatrix} (x_1 + x_2)/2 \\ (y_1 + y_2)/2 \end{pmatrix} + \left[\frac{r_1 + r_2}{2}\right]\begin{pmatrix} \cos\phi \\ \sin\phi \end{pmatrix} \quad (2)$$

Consider an alternative, less computationally intensive approach: A trigonometric argument shows that the tangents, in normal parametric form, to the arc drawn with center (x, y), radius r, orientation θ and half-angle α form the set

$$\mathcal{T}_{x,y,\theta,r} = \{(\rho, \phi) \in \Gamma \times \Phi \ :$$
$$\rho = r + x\cos(\theta + \gamma) + y\sin(\theta + \gamma), \ \ \gamma \in [-\alpha, \alpha]\}$$

For two arcs $(x_1, y_1, \theta_1, r_1)$ and $(x_2, y_2, \theta_2, r_2)$, let the associated sets of tangents be $\mathcal{T}_1$ and $\mathcal{T}_2$. Let $\mathcal{T}_\cap = \mathcal{T}_1 \cap \mathcal{T}_2$. Then, for $x_1 \neq x_2$ or $y_1 \neq y_2$, $\mathcal{T}_\cap$ is non-empty and singleton or at most doubleton if and only if the arcs share a common tangent. Now, in general, finding the intersection point(s) of two arbitrary $\mathcal{T}$ sets is no less computationally intensive than finding common tangents to two arbitrary circles. However, if we let $\mathcal{T}_{\text{rect}} = (\rho_{\min}, \rho_{\max}, \phi_{\min}, \phi_{\max})$ represent the smallest rectangle in $\Gamma \times \Phi$ which contains $\mathcal{T}_{x,y,\theta,r}$, then the comparison of two arcs may be expedited by checking the intersection of the associated $\mathcal{T}_{\text{rect}}$ sets; an empty intersection evidences the incompatibility of the arcs and no further resources need be wasted. Checking for intersection of rectangular regions is computationally trivial; moreover, the limits of each $\mathcal{T}_{\text{rect}}$ set need be computed only once per ultrasonic measurement, regardless of which other arcs the measurement is to be matched with[3].

Further, note that for all $\mathcal{T}$ sets, $\phi_{\max} - \phi_{\min} = 2\alpha$, a constant. Hence, if we tessellate transducer pose space $X \times Y \times \Theta$ into a rectangular grid, and record each arc $(x, y, \theta, r, \rho_{\min}, \rho_{\max})$ by entering the triple $(r, \rho_{\min}, \rho_{\max})$ at the grid location containing (x, y, θ), then a *new* arc $(x', y', \theta', r', \rho'_{\min}, \rho'_{\max})$ need be matched only against those arcs stored in the region $\theta' - \alpha \leq \theta \leq \theta' + \alpha$; a linked list of all currently unmatched arcs in each θ-plane may be used to expedite the search.

In addition, the list of potential matching arcs may be shortened by rejecting arcs whose transducer position was either too far or too close from the current arc's transducer position. The reason for rejecting measurements taken too *far away* may be clear: by finding the common line-segment, we are inferring the possible existence of a planar reflecting surface at the location of that line-segment; inferring a *large* intervening world structure from two physically remote pieces of evidence may reduce the *soundness* factor of the overall algorithm (see Section C.), especially if no mechanism is included for *negating* such inference. Odometric errors accrued while traversing the long distance between transducer locations calls into question also the *precision* of the inferred line-segment parameters.

The reason for rejecting measurements taken too *close together* is the high sensitivity of the parameters of common tangents to small perturbations in either (x_1, y_1) or in (x_2, y_2) when both $y_1 - y_2$ and $x_1 - x_2$ are close to 0. See, for instance, equation 1. Odometric error will introduce such perturbations. The difficulty is exacerbated if in addition $r_1 \approx r_2$ (as is often the case for measurements taken in close proximity).

D. *The second clustering stage: Looking for specific features*

The first stage of clustering produces descriptions of line-segments each of which is the common tangent to two arcs. In the second stage of clustering, we search within the set of line-segments for groups which belong to the same large-scale feature of the environment. In this exposition, the set of environmental features is limited to $\mathcal{F} = \{\text{planar surfaces, corners}\}$; in Section D.1. we show how to extend the set $\mathcal{F}$ to features of more general shape.

It would seem at first glance that we are now forced to extract suitable clusters of 5-dimensional vectors in $\Gamma \times \Phi \times C_x \times C_y \times L$, a formidable task. However, in the present stage of clustering, the precise locations of the *vertices* of individual line-segments are not as important as the *center-point* location and *orientation*. Hence, we project each 5-tuple onto the $C_x \times C_y \times \Phi$ subspace; each line-segment is now represented by the triple (c_x, c_y, ϕ), and clustering may be performed in the rather more amenable 3-dimensional subspace[4].

D.1. *Feature detection: General case*

Feature detection now consists of finding subsets of suitable shape of the data points in $C_x \times C_y \times \Phi$. The desired shape is dependent on the nature of the feature to be detected. Consider a feature F described parametrically in world coordinates by $(s_x(t), s_y(t))$, $t_{\min} \leq t \leq t_{\max}$, s_x, s_y piecewise differentiable[5]. Then, for any $\hat{t}$, a line tangent to F at $(s_x(\hat{t}), s_y(\hat{t}))$ is parallel to the vector $(\dot{s}_x(\hat{t}), \dot{s}_y(\hat{t}))$; a *normal* to the tangent has orientation $\arctan\left[-\dot{s}_x(\hat{t})/\dot{s}_y(\hat{t})\right]$ For the purposes of ultrasonic detection, consider F to consist of reflective patches, each coincident with F for some t and co-linear with a tangent to F at $(s_x(t), s_y(t))$. In $C_x \times C_y \times \Phi$ the set $\mathcal{P}_F$ of

[3] As opposed to equation 1 where explicit use is made of parameters of *both* arcs in a coupled fashion.

[4] In 3D plane-segments require 9 parameters for unambiguous description; for clustering purposes, we can limit our consideration of each plane-segment to the triple $(c_x, c_y, c_z) \in C_x \times C_y \times C_z$ for location of the center-point, and the pair $(\phi, \omega) \in \Phi \times \Omega$ representing orientation of the normal to the plane. Clustering is performed in a 5-dimensional subspace of the rather foreboding 9-dimensional full parameter space.

[5] This is a 2D approximation of the the feature; we are actually assuming F to be the cylinder

$$\begin{pmatrix} s_x(u) \\ s_y(u) \end{pmatrix}, \ u_{\min} \leq u \leq u_{\max}, \ -\infty < v < \infty$$

reflective patches associated with F describes the shape

$$\mathcal{P}_F = \left\{ \begin{pmatrix} s_x(t) \\ s_y(t) \\ \arctan\left[\frac{-\dot{s}_x(t)}{\dot{s}_y(t)}\right] \end{pmatrix} : t_{\min} \leq t \leq t_{\max} \right\} \quad (3)$$

This, then, is the shape of the clusters we extract from the set of data points (line-segments) in $C_x \times C_y \times \Phi$ in order to detect the feature F. Such clustering is, in general, a difficult and potentially computationally intensive task, requiring the application of some form of convolution with a mask of shape $\mathcal{P}_F$, a Hough Transform, or some other method of point-to-curve transformation. However, for a certain class of features, the shape described by $\mathcal{P}_F$ is quite simple and the clustering problem is greatly facilitated. $\mathcal{F}$ is a subset of this class; for each element $F \in \mathcal{F}$ in 2D, the set $\mathcal{P}_F$ is 1-dimensional with special properties.

D.2. Feature detection: Planar surfaces

Consider a vertical planar surface feature described parametrically by $(t,\ mt + b)$, $t_{\min} \leq t \leq t_{\max}$ for some m and b. By equation 3,

$$\mathcal{P}_F = \left\{ \begin{pmatrix} t \\ mt + b \\ -\arctan\left(\frac{1}{m}\right) \end{pmatrix} : t_{\min} \leq t \leq t_{\max} \right\}$$

Thus, all reflective patches lie in a hyperplane of constant ϕ in $C_x \times C_y \times \Phi$; within this hyperplane, the patches lie on the line $(x(t),\ y(t)) = (t,\ mt + b)$. Further, the orientation of the line within the hyperplane is precisely the ϕ-value of the hyperplane.

Hence, given the stream $\Psi \subseteq C_x \times C_y \times \Phi$ of line-segments from the first stage of clustering, the present clustering task is to search for groups $\Psi'_i \subseteq \Psi$ such that $\forall i\ \forall \psi \in \Psi'_i,\ \ \psi_\phi = \phi_i \wedge \psi_{c_y} = b_i + \psi_{c_x} \tan \phi_i$, where ϕ_i and b_i are constants, and $\psi_\phi, \psi_{c_x}, \psi_{c_y}$ are the Φ, C_x and C_y components of the line-segment ψ respectively.

The current implementation used for the experiments described in Section III. performs clustering of line-segments on this basis. In order to deal with noise, we tessellate $C_x \times C_y \times \Phi$ space and quantize all data point (line-segment) parameters: each cell is a rectangular bar enclosing a region of constant (quantized) ϕ, and orientated at angle ϕ (see Figure 2). In effect, this is simply a grid tesselation of $\Gamma \times \Phi$. Hence, we maintain a 2-dimensional grid of linked-lists, one for each quantized $(\rho,\ \phi)$ pair[6]. Each list records the clusters so far detected with those $(\rho,\ \phi)$ parameters. There may be multiple distinct clusters in each list since the merging of distantly separated clusters is avoided in order to prevent the *soundness* factor of the overall algorithm from being compromised.

D.3. Feature Detection: Corners

A corner feature F_c cannot, by definition, be described by a differentiable pair of functions $(s_x(t),\ s_y(t))$. We note that for an *acute, inner* corner, acoustic energy will be reflected back to the transducer as long as the corner lies within the cone of ensonification. Orientation issues, such as arose in the case of planar reflective surfaces, are irrelevant here since we will receive sonar returns from a corner for a great variety of impinging angles. Similarly, *outer* corners will diffract acoustic energy back to the transducer as long as the corner lies within the acoustic cone. Hence, the set of reflective (or diffractive) patches $\mathcal{P}_{F_c}$ may be considered to comprise all patches centered at the corner with orientation perpendicular to some ray emanating from within the angle enclosed by the corner[7]. Hence, for a corner at (c_x, c_y) defined by "walls" at angles ϕ_1 and ϕ_2, $\phi_2 > \phi_1$, relative to the C_x axis,

$$\mathcal{P}_{F_c} = \left\{ \begin{pmatrix} c_x \\ c_y \\ \phi \end{pmatrix} : \phi_1 \leq \phi \leq \phi_2 \right\}$$

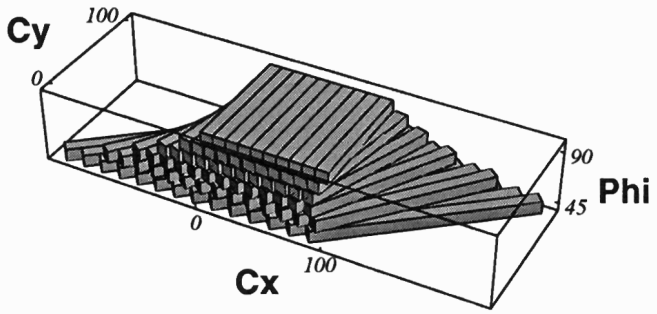


Figure 2. A partial tesselation of $C_x \times C_y \times \Phi$ space for detection of planar surfaces. The tesselation shown here is for $0 \leq C_x \leq 100$, $0 \leq C_y \leq 100$, $40° \leq \phi \leq 100°$; it is coarser than would be used in practice.

As in the case of planar surfaces, $\mathcal{P}_{F_c}$ lies on a line in $C_x \times C_y \times \Phi$; in this case, the line is perpendicular to $C_x \times C_y$.

Given the stream $\Psi \subseteq C_x \times C_y \times \Phi$ of line-segments from the first stage of clustering, then, the clustering task for detection of corners in the noise-free case is to search for groups $\Psi'_i \subseteq \Psi$ such that $\forall i\ \forall \psi \in \Psi'_i,\ \ \psi_{c_x} = c_{x_i} \wedge \psi_{c_y} = c_{y_i}$, where c_{x_i} and c_{y_i} are constants.

Clustering in the presence of noise, thus, consists of identifying groups of line-segments which mutually intersect in a relatively highly localized region of space. A grid tesselation of $C_x \times C_y$ suffices.

E. The third stage of clustering: Parameter estimation

E.1. Parameter estimation: Planar surfaces

We describe the method of *recursive least total squares* (RLTS) for the parameter estimation of a cluster Ψ' of approximately co-linear line-segments. The most significant drawback of a least squares approach is its low breakdown point (that is, its great sensitivity to even a single outlier). In our particular case, however, this is *not* a great concern, since outliers are effectively filtered out in the clustering stage. RLTS boasts computational efficiency and a certain elegance of simplicity.

As described in Section B., each line-segment $\psi \in \Psi'$ is represented redundantly as a 5-tuple $(\rho, \phi, c_x, c_y, l) \in \Gamma \times \Phi \times C_x \times C_y \times L$. Similarly, the parameters we wish

[6] This clustering phase, for this particular case, then, is similar to the Hough method, though only to identify *potential* members of a cluster; other parameters such as physical separation and length are still taken into account.

[7] It should be also noted that since diffraction is far more dispersive of energy than reflection, the practical *range* at which outer corners may be detected is significantly smaller than that of reflective inner corners; the detection range falls off as the angle of the corner increases. See [9] for more detail.

to estimate are those of some underlying line-segment $\hat{\psi} = (\hat{\rho}, \hat{\phi}, \hat{c_x}, \hat{c_y}, \hat{l})$ which in a least total squares (LTS) sense "summarizes" the contents of Ψ'. Total rather than ordinary least squares are used in order to ensure independence of orientation relative to the coordinate system. Moreover, examination of equation 2 shows that each center-point (c_x, c_y) will be displaced in approximately the direction of θ (the normal) under the influence of noise in measurements r_1 and r_2 [12].

We estimate the pair $(\hat{\rho}, \hat{\phi})$ by means of a LTS fit on the *center-points* of members of Ψ'; we describe here a *recursive* variant on standard LTS fitting which allows parameters to be updated on-line as new members of ψ are found. Let the number of members of Ψ' be n. Hence, the set of points to which we wish to fit a LTS line is $C = \{(c_{x_i}, c_{y_i})^T \ : \ 1 \leq i \leq n\}$. Let the mean (center of gravity) of C be the point $\mathbf{g} = (g_x, g_y)^T$. Let the set of displacements of the data points relative to $\mathbf{g}$ be $V = \{\mathbf{v}_i = (c_{x_i}, c_{y_i})^T - \mathbf{g} \ : \ 1 \leq i \leq n\}$. Then a normal to the least total squares fit on C is the eigenvector $\mathbf{e}$ associated with the smallest eigenvalue of the scatter matrix $S = \sum_{i=1}^{n} \mathbf{v}_i \mathbf{v}_i^T$ (see [6], page 334).

In order to transform the LTS estimator into a *recursive* form, consider the merging of two sets of points C_1 and C_2, with means $\mathbf{g}_1$, $\mathbf{g}_2$, displacement sets V_1, V_2 and scatter matrices S_1, S_2 respectively. Let $n_1 = |C_1|$ and $n_2 = |C_2|$. Let $C_\cup$ denote $C_1 \cup C_2$. It can be shown [12] that the mean $\mathbf{g}_\cup$ and scatter matrix $S_\cup$ of $C_\cup$ are given by

$$\mathbf{g}_\cup = \mathbf{g}_1 + \left(\frac{n_2}{n_1 + n_2}\right)(\mathbf{g}_2 - \mathbf{g}_1) \tag{4}$$

$$S_\cup = S_1 + S_2 + \left(\frac{n_1 n_2}{n_1 + n_2}\right)(\mathbf{g}_2 - \mathbf{g}_1)(\mathbf{g}_2 - \mathbf{g}_1)^T.$$

A new member $\tilde{\mathbf{c}} = (\tilde{c}_x, \ \tilde{c}_y)^T$ to a cluster C_1 may be considered to comprise its own cluster $C_2 = \{(\tilde{c}_x, \ \tilde{c}_y)^T\}$ with $n_2 = 1$, $\mathbf{g}_2 = \tilde{\mathbf{c}}$ and $S_2 = \mathbf{0}$. As a measure of goodness of the RLTS fit, we consider the ratio of variances in the directions of the eigenvectors. Since distinct eigenvectors of symmetric matrices (such as a scatter matrix S) are orthogonal, this ratio is equivalent to the *eccentricity* of the ellipse ubiquitously used to represent covariance matrices in 2D. By a straightforward algebraic argument, it may be shown that

$$\text{Eccentricity} = \frac{\mathbf{Trace}\left[S \cdot (\mathbf{e}_1 \mathbf{e}_1^T)\right]}{\mathbf{Trace}\left[S \cdot (\mathbf{e}_2 \mathbf{e}_2^T)\right]} \tag{5}$$

where $\mathbf{e}_1$ and $\mathbf{e}_2$ are the eigenvalues of S, and $\mathbf{e}_2 \geq \mathbf{e}_1$. This eccentricity measure may be used in certain cases to disambiguate genuine planar surfaces (which generally have eccentricity very close to 0) from corners and spurious clusterings (for which eccentricity is close to 1).

We find the endpoints of $\hat{\psi}$ by finding the extremes of the projections of the endpoints of elements of Ψ' onto the line described by $(\hat{\rho}, \hat{\phi})$; the center-point $(\hat{c_x}, \hat{c_y})$ and length $\hat{l}$ of $\hat{\psi}$ are then easy to deduce.

E.2. Parameter estimation: Corners

At present, the heuristic used to estimate the location of a corner associated with a cluster Ψ' is simply the mean of the *center-points* of members of Ψ'. In augmenting a set of points C_1 of size n_1 with a new point $(\tilde{c}_x, \tilde{c}_y)^T$, equation 4 may be used to update the mean recursively.

III. EXPERIMENTS

In order to test the efficacy of the approach, we test it both in simulation and on real data. Figure 3 shows typical results for an office environment consisting of vertical walls, tables and workstations. In this case, the analysis can be simplified to 2D.

The top left figure shows the layout of the test environment. Black rectangles represent desks and workstations. The lower wall contains a power outlet which appears as an inner reflective corner to ultrasound. The top right figure shows the raw sonar data: each arc represents the region of uncertainty of the location of a reflective patch. The grey area shows the trajectory of the mobile vehicle and the loci of the transducers. Note the concentration of arcs around the power outlet on the lower wall. The bottom right figure depicts the common tangents extracted by the first stage of clustering, while the bottom left shows the planar features extracted by the second and third stages. Some thresholding has been applied to suppress features of minimal supporting evidence. The numbers represent the reciprocals of the eccentricity measures calculated for each cluster according to equation 5. Note the high numbers for most clusters indicating eccentricities close to 0.

Comparison of Figures 3(b) and (d) illustrates the dramatic data reduction afforded by the clustering method. Due to misallignment of typesetting, perhaps less clear from the figure is the precision of the feature extraction: the distance between features representing the edges of the enclosure are 353 cm (width) and 378 cm (height) as against ground truth of 354 cm and 380 cm respectively, or about 0.5% error despite significant odometric error accumulated over the course of the experiment.

See [12] for reports and analyses of simulation as well as other experiments using real data.

IV. CONCLUSION

We have delineated a multi-stage approach for the extraction of geometric features from sonic data. In this exposition, the set of environmental features was limited to $\mathcal{F} = \{\text{planar surfaces, corners}\}$ though we showed how to extend the set $\mathcal{F}$ to features of more general shape.

The technique affords great data reduction in an *on-line*, *tractable*, and *computationally efficient* manner, despite the inherent exponential nature of clustering. The parameter estimation phase was seen to be *precise*. No model for noise was assumed.

The algorithm was seen in the example to be fairly *sound* in that most extracted features corresponded to real structures in the environment. It was also reasonably *complete* in that most features in the rather simple environment were detected. The soundness and completeness of the algorithm in more complex environments are yet to be ascertained.

The method is suitable for systems employing *fixed* transducers as well as rotating ones. It lends itself to applications where remaining stationary during a scan is both impractical and infeasible, and where deadreckoning errors can be substantial. The approach may be used both for map-building during exploration and for feature identification during navigation.

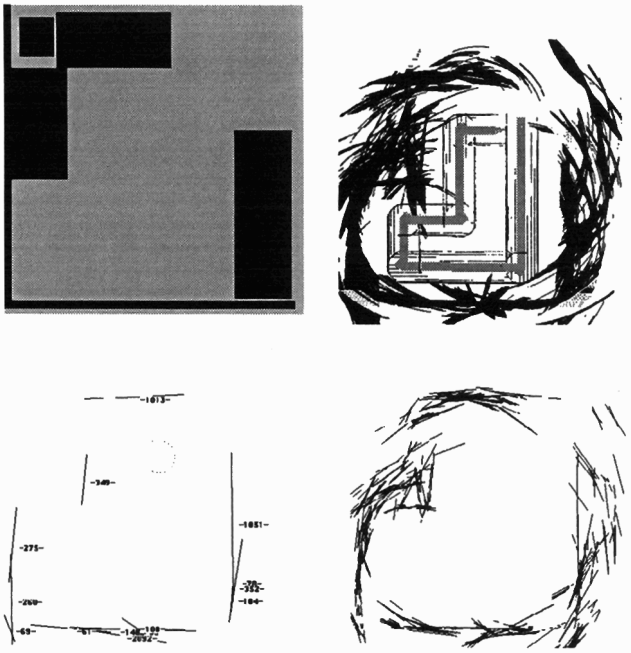


Figure 3. Experiment in office environment. Beginning at top left, in clockwise order, (a) Layout of the environment, (b) Raw data, one arc per measurement, (c) Common tangents extracted by the first stage of clustering, and (d) Planar features extracted by the second and third stages. The numbers are reciprocals of the eccentricity measures for each cluster.

V. FUTURE WORK

We intend to extend the paradigm to include a mechanism for *negative evidence*: When sections of an inferred feature are found to be absent (e.g. doorways in a planar surface), the parameters of the feature should be updated accordingly. Currently only *positive* evidence is utilized. Inclusion of such a mechanism would also render the algorithm suitable for (minimally) dynamic environments.

We intend to enhance the testbed to incorporate the ability to *localize* the vehicle based on a comparison of extracted features with some model of the environment. This is essential for landmark-based navigation.

Finally, we intend to integrate the data extracted by the algorithm with data from other sensor modalities to furnish a richer, more complete, robust and accurate description of the environment.

REFERENCES

[1] R. G. Brown, L. P. Chew, and B. R. Donald. Mobile robots, map-making, shape metrics, and localization. *Submitted to IEEE ICRA*, 1993.

[2] T. Camus, J. Monsarrat, and T. Dean. Planning and selective perception for target retrieval. In *Proceedings of the 1993 Image Understanding Workshop*, pages 593–597, April 1993.

[3] I. J. Cox and J. J. Leonard. Probabilistic data association for dynamic world modeling: A multiple hypothesis approach. In *Proceedings of the 1991 International Conference on Advanced Robotics (ICAR)*, June 1991.

[4] J. L. Crowley. Dynamic modeling of free-space for a mobile robot. *IEEE/RSJ International Workshop on Intelligent Robots and Systems '89*, September 1989.

[5] J. L. Crowley. World modeling and position estimation for a mobile robot using ultrasonic ranging. *1989 IEEE International Conference on Robotics and Automation*, May 1989.

[6] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis.* Wiley-Interscience, 1973.

[7] A. Elfes. Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation*, RA-3(3):249–265, June 1987.

[8] A. Elfes. Multi-source spatial fusion using bayesian reasoning. In M. A. Abidi and R. C. Gonzalez, editors, *Data Fusion in Robotics and Machine Intelligence*, pages 137–163. Academic Press, 1992.

[9] R. Kuc. A spatial sampling criterion for sonar obstacle detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7), July 1990.

[10] R. Kuc and V. B. Viard. A physically based navigation strategy for sonar-guided vehicles. *International Journal of Robotics Research*, 10(2), April 1991.

[11] J. J. Leonard. *Directed Sonar Sensing for Mobile Robot Navigation.* PhD thesis, University of Oxford, 1990.

[12] R. Mandelbaum and M. Mintz. Sonar signal processing using tangent clusters. Technical report, University of Pennsylvania, 1994. In preparation.

[13] A. Ohya, Y. Nagashima, and S. Yuta. Exploring unknown environment and map construction using ultrasonic sensing of normal direction of walls. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, May 1994.

[14] S. B. Thrun. Exploration and model building in mobile robot domains. In *Proceedings of the IEEE International Conference on Neural Networks*, 1993.

ACKNOWLEDGMENTS

Portions of this research were supported by the following grants and contracts: ARPA Contracts N00014-92-J-1647, and DAAH04-93-G-0419; ARO Contracts DAAL03-89-C-0031PRI, and DAAL03-92-G0153; NSF Grants CISE/CDA-88-22719, IRI92-10030, IRI92-09880, IRI93-03980, and IRI93-07126.

We would like to thank Jean-Marc Vezien of INRIA for the French translation of the abstract.

Feature-Based Localization using Fixed Ultrasonic Transducers *

Robert Mandelbaum and Max Mintz

General Robotics and Active Sensory Perception (GRASP) Laboratory
Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104

December 14, 1994

Abstract

We describe an approach for mobile robot localization based on geometric features extracted from ultrasonic data. As is well known, a single sonar measurement using a standard POLAROID™ sensor, though yielding relatively accurate information regarding the *range* of a reflective surface patch, provides scant information about the location in *azimuth* or *elevation* of that patch. This lack of sufficiently precise localization of the reflective patch hampers any attempt at data association, clustering of multiple measurements or subsequent classification and inference.

In previous work [15, 16] we proposed a multi-stage approach to clustering which aggregates sonic data accumulated from *arbitrary* transducer locations in an *sequential* fashion. It is computationally *tractable* and *efficient* despite the inherent exponential nature of clustering, and is *robust* in the face of noise in the measurements. It therefore lends itself to applications where the transducers are *fixed* relative to the mobile platform, where remaining stationary during a scan is both impractical and infeasible, and where deadreckoning errors can be substantial.

*Portions of this research were supported by the following grants and contracts: ARPA Contracts N00014-92-J-1647, and DAAH04-93-G-0419; ARO Contracts DAAL03-89-C-0031PRI, and DAAL03-92-G0153; NSF Grants CISE/CDA-88-22719, IRI92-10030, IRI92-09880, IRI93-03980, and IRI93-07126.

In the current work we apply this feature extraction algorithm to the problem of localization in a partially known environment. Feature-based localization boasts advantages in robustness and speed over several other approaches. We limit the set of extracted features to planar surfaces. We describe an approach for establishing correspondences between extracted and map features. Once such correspondences have been established, a least squares approach to mobile robot pose estimation is delineated. It is shown that once correspondence has been found, the pose estimation may be performed in time *linear* in the number of extracted features. The decoupling of the correspondence matching and estimation stages is shown to offer advantages in speed and precision.

Since the clustering algorithm aggregates sonic data accumulated from *arbitrary* transducer locations, there are no constraints on the trajectory to be followed for localization except that sufficiently large portions of features be ensonified to allow clustering. Preliminary experiments indicate the usefulness of the approach, especially for accurate estimation of orientation.

1 Introduction

1.1 Overview

In [11], a distinction is drawn between *continuous localization* and *relocation*. The difference "rests on the use of *a priori* knowledge of the vehicle position estimate in achieving correspondence, and the weighted inclusion of this *a priori* position in the updated estimate" [11]. *Continuous* localization is seen as "the normal mode of operation, with relocation used for initialization and error recovery".

In this paper both forms of localization are addressed. However, unlike [11], our approach does not employ extended Kalman filtering for the estimation phase; for this reason, no weighted inclusion of a priori position information in the updated estimate is necessary in either type of localization.

The problem of *geometric* model-based mobile robot localization (whether relocation or continuous) using the ultrasound modality can be subdivided into the following subproblems:

1. Extraction of geometric features,
2. Localization of extracted features within the local coordinate frame (calculation or updating of location and orientation parameters of extracted features within the local coordinate frame),
3. Establishment of correspondences between extracted features and model features, and
4. Estimation of robot location and orientation within the global coordinate frame.

Many approaches in the literature omit stages 1 and 2 above completely. Instead, they rely on establishing correspondences directly between actual sensor measurements and map features. The two primary difficulties with such an approach are

- For certain sensor modalities, a single measurement is insufficient to allow correspondence matching. An example is the ultrasound modality: the wide-beam nature of wave propagation results in large uncertainty in azimuth and orientation of the reflecting surface associated with a single measurement. This hampers direct correspondence matching with map features.
- Even for sensor modalities where direct correspondence matching between raw data and map features is possible, this correspondence has to be established *on every cycle*. While the process may be expedited with the aid of the previous pose estimate, the essential difficulty of correspondence matching cannot be averted.

By matching extracted *features* with map *features*, we avert these difficulties in this work. While stages 1 and 2 are necessary for the *relocation* problem, once correspondences have been established, the same correspondences may be used for many cycles, with new data readings being used to update the parameters of the extracted features. It may be argued that establishing correspondences between new measurements and extracted features is no less difficult a task. While this is true in essence, at least the correspondence problem has been decoupled from the localization problem. This decoupling allows for pose estimation techniques which are both fast and precise. Further, since localization is now performed using extracted *features*, it is much less sensitive to individual noisy or spurious measurements; each extracted feature contains the combined information from *many* measurements over *relatively long* periods. In theory, false matches become less frequent, and localization benefits from the resultant improved robustness.

We employ the algorithm described in [15, 16] for the first two subproblems above. The algorithm accepts a stream of transducer locations, orientations and corresponding measurements, and outputs a list of planar and corner features extracted from the input data. The location and orientation parameters of the extracted features are given in the local coordinate frame. For a more detailed exposition of the operation of the feature extraction algorithm, the reader is referred to [15, 16].

The current paper focuses on an approach to subproblems 3 and 4 for both continuous localization and relocation. In section 2 we describe our approach to correspondence matching, while the pose estimation problem is addressed in section 3. We show that once correspondence has been achieved, the pose estimation phase may be performed in time *linear* in the number of extracted features.

1.2 Notation

Define the *pose* $p \in \Re^2 \times [0, 2\pi]$ of a mobile platform to be a combination of its current position and orientation within a global coordinate frame (GCF). Denote the set of possible poses by $\mathcal{Q}$. If the extent of the platform's motion is confined to the region $[0, X] \times [0, Y]$, then $p \in \mathcal{Q} = [0, X] \times [0, Y] \times [0, 2\pi]$.

Let $\Re^+$ represent the non-negative real numbers. $\mathcal{P}(A)$ denotes the power set of A. $\lfloor x \rfloor$ represents the largest integer less than or equal to x. $\mathcal{Z}$ is the set of integers. $c[A] \subseteq \mathcal{B}$ denotes the image of set $A \subseteq \mathcal{A}$ under the mapping $c : \mathcal{A} \rightarrow \mathcal{B}$. Similarly, $c^{-1}[B]$ denotes the inverse image of set $B \subseteq \mathcal{B}$ under the mapping c.

$\mathbf{A}^\sharp$ denotes the pseudo-inverse of a non-square matrix $\mathbf{A}$.

1.3 Related work

The issue of localization (geometric as well as other) has been addressed extensively in the literature. Approaches may be divided into the following four broad categories according to the basic primitives used for correspondence matching between the local and global coordinate frames:

1. **Rastor-rastor** localization: Many approaches which use an *occupancy grid* representation of the environment [8, 9] attempt localization by finding the optimal match of the local bitmap of occupied space with a global occupancy grid. The goodness of a match is a function of three parameters x, y, and θ: It is a measure of the (discrete) correlation between the 2D function approximated by the occupancy grid and the function approximated by the local bitmap translated by (x, y) and rotated by θ. Drawbacks of this approach are the computational complexity of correlation as well as the trade-off between efficiency and precision, embodied in the choice of grid resolution. In [18] it is concluded that even for systems using an occupancy grid representation, more reliable position estimates are obtained by extracting segments from the grids and performing segment-segment matching. Of course, extracting features such as segments from an occupancy grid can be computationally expensive. It is also often conceptually inefficient: in constructing the occupancy grid, one discards relevant data such as surface orientation. One then attempts to re-extract this information with some operator on the grid.

2. **Rastor-feature** localization: This category comprises approaches which represent the environment parametrically in features; Localization is performed by establishing correspondences between individual unclustered sensor measurements and the map features. Once correspondence has been established, some form of optimization is performed to minimize some function of the vector of spatial discrepancies between measurements and features. Approaches differ in how the correspondences are established, the definition of spatial discrepancy and the function to be minimized.

   In [5], the entire environment space of the robot is searched for locations which would yield sensor readings consistent with the measured range readings. It is recognized that due to noise, there may be no location at which *all* readings are consistent; the location which yields the greatest number of consistent readings is selected. While mention is made of the need for a metric over pose space $\mathcal{Q}$, no such metric is described. Despite numerous tools borrowed from computational geometry, the algorithm is computationally prohibitively expensive: it is stated to be $O(m^2n^2\log(mn))$ where m is the number of range readings taken and n is the complexity of the map (compare with $O(n)$ for pose estimation using the approach described in section 3. The rastorized version runs in time $O(mr^2e^2l)$ where the rastorized environment is r by r, e is the radius of an "error ball", and l is the size of the greatest range reading. Hence, this algorithm is not practical for *continuous*

localization. Moreover, the approach does not address orientation uncertainty at all. Inclusion of this dimension would make it even more computationally infeasible, so that the utility of the approach is questionable even for the less constrained problem of *relocation*. Furthermore, the algorithm is based more on intuitive arguments than mathematical rigor, nor does it deal with uncertainty and sensor error in a satisfactory manner.

In [13], the pose estimation problem is formulated as an iterative optimization in terms of the extent to which the map explains the observed measurements. The approach assumes a coarse initial position estimate is available, and estimates the correct position *assuming the orientation is known*. Each data point, representing the position of a reflecting point in the environment, is classified to a target line segment. A *correction vector* is then associated with each data point, and *weighted voting* of the correction vectors yields an overall translation vector to correct position. Various estimators are introduced to indicate whether a calculated pose is probable. For pose calculations deemed reliable, orientation is then corrected by maximizing one of the estimators with respect to rotation. The new orientation estimate is used as an initial orientation estimate for the next iteration.

Though the general approach presented in [13] seems valid, it suffers from a few drawbacks. A pencil-beam model for ultrasound propagation is assumed and forms an integral part of the entire approach; this is an inadequate model, glossing over the issues introduced by the wide-beam nature of the propagation. Further, many of the estimators used are heuristic and ad hoc. Though "correct" behavior is shown for a single example, no general behavior is proved, either theoretically or empirically. The orientation estimation procedure is particularly deficient in this regard: Global maximization is performed on a poorly characterized function.

Methods employing Kalman filters to model both robot pose and map features abound [1, 4, 3, 6, 11, 12]. These approaches may be categorized as rastor-feature localization methods, since each unclustered measurement is used individually to update the various Kalman filters. Though features are extracted from the data, these features are not used for correspondence.

A consistent shortcoming of systems employing Kalman filters is that often the system in question is not shown to meet the underlying assumptions of the Kalman filter. One of these assumptions is the Gaussian nature of both process and measurement noise: many researchers *assume* this noise model with no empirical or theoretical justification at all.

3. **Feature-feature** localization: This category consists of approaches in which measurements are clustered into features. Correspondence is established between extracted features and map features. A new estimate for location is obtained by minimizing a function on the vector of feature-feature discrepancies. These discrepancies may be based not only on *spatial* distances between extracted and map features, but also on *any* of the features' other parameters such as size, shape or orientation.

New data is incorporated into already existing clusters, and is used to update the clusters' parameters, prior to another phase of feature-feature localization. In this way, the correspondence and localization problems are decoupled.

The main advantages of feature-feature localization are twofold: Firstly, it is robust in the face of noise since each feature's parameters are derived from multiple measurements. This reduces the sensitivity of the localization to noisy or spurious measurements. Secondly, the same correspondence match may be used over many cycles, reducing overhead and speeding up the localization procedure. The major disadvantage of feature-feature localization is the longer start-up period during which features are extracted. During this period, some other form of localization such as odometry must be employed.

The approach described in this paper is a member of the feature-feature category.

In [10], regions of common depth (RCD's as defined in [11]) are extracted from ultrasonic data from a rotating scanning transducer. Each RCD is classified into one of four types according to neighboring RCD's and the width of the RCD. Pairs of RCD's are matched with pairs in the map; each match of a pair has an associated transformation under which the local coordinate frame is brought into correspondence with the global frame. The set of transformations is analyzed for clusters, and the centroid of the largest cluster selected as the "optimal" transformation. Though at first glance this approach appears to belong to the feature-feature category of localization algorithms, in essence it belongs more to the rastor-feature category: though features (RCD's) are extracted from the data before correspondence is attempted, new features are extracted during each cycle. New data is not analyzed for correspondence with previously found clusters. One of the main advantages of feature-feature localization is not exploited in that previous correspondence matches are not re-used. In effect, rastor-feature localization is being performed, though the "rastor" data is slightly more refined than individual sonar measurements.

The are several other difficulties with the approach described in [10]. First of all, the search for clusters is performed on a 4-dimensional space. Each coordinate is assumed to be independent of the remaining three coordinates; in fact, two of the coordinates are non-linearly dependent, and related to each other by a trigonometric relationship. Points within the space are assumed to be uniformly distributed, though no justification for the assumption is presented, either theoretic, intuitive or empirical. Similarly, no justification is given for the assumptions that the largest cluster in the space of transformations is the "correct" one, or that the centroid of this cluster is a reasonable choice for the localization transformation. Finally, the approach is suitable only for systems using scanning rotating sonars.

4. **Landmark-landmark** localization. Many systems base the localization procedure on *landmarks* detected in the environment. The distinction drawn here between a feature and a landmark lies in the amount of knowledge

about identity: A *feature* is a summary description of a cluster of data points; its identity within a large class of similar features is not known. A landmark, on the other hand, is a feature with a unique identity based on some distinguishing characteristic such as spatial location or some sensed property.

Landmark-based localization, then, differs from feature- or rastor-based localization in that *no correspondence matching is necessary.* This phase is rendered superfluous by a more comprehensive *recognition* phase.

Various approaches to landmark localization are explored in [2, 19, 20, 21] among others.

Localization based on the detection of beacons — easily recognizeable features placed in the environment — falls into the landmark-landmark or feature-feature category depending on whether detection of a beacon uniquely determines its identity or whether correspondence must still be established between extracted and map beacons.

2 Establishing correspondences

2.1 Problem statement

In this section we describe an approach to establishing correspondences between extracted features and model features. In the present work, extracted features comprise planar surfaces only.

In 2D, a planar feature is a line-segment. We use a redundant *normal parametrization* representation ([7], page 336) for line-segments: the line of which the segment forms a part is represented by the pair (ρ, ϕ) where ρ is the perpendicular distance of the line from some fixed $(0, 0)$ point; ϕ is the inclination of the normal to the line relative to the x-axis in a counter-clockwise direction. This pair is supplemented with parameters to specify the locations of the end-points. Only two extra parameters are necessary, but for simplicity we use three: two for the location of the center-point of the line-segment (c_x, c_y) and one for the length of the segment l. Hence, each line-segment is represented by the 5-tuple $(\rho, \phi, c_x, c_y, l) \in \Gamma \times \Phi \times C_x \times C_y \times L$.

Let $\mathcal{M} = \{m_i = (\rho_i, \phi_i, c_{xi}, c_{y_i}, l_i) \mid 1 \leq i \leq |\mathcal{M}|\}$ be the set of planar features in the given map. Let $\mathcal{F} \subseteq \mathcal{M}$ be the set of planar features within the mobile robot's current ultrasonic "field of view". Let $\mathcal{E}$ be the set of features extracted by the clustering algorithm described in [16]. Let $\mathcal{C}$ be the class of functions $\mathcal{C} = \{c : \mathcal{E} \rightarrow \mathcal{F} \cup \bar{0}\}$. Each element $c \in \mathcal{C}$ represents an assignment of correspondences: for a specific $e \in \mathcal{E}$, $c(e)$ is the map feature to which e corresponds under this assignment c. Note that we do not insist that elements of $\mathcal{C}$ are either injective or surjective: multiple elements of $\mathcal{E}$ may correspond to the same feature in $\mathcal{F}$ since the clustering algorithm may extract multiple

sections of the same underlying planar surface; further, not all elements of $\mathcal{F}$ will necessarily be detected. Note also the augmentation of $\mathcal{F}$ with the zero element $\bar{0}$ to form the range of elements of $\mathcal{C}$. The zero element $\bar{0}$ may be made the image of spurious extracted features. In this way, phantom extracted features need not to be mapped to elements of $\mathcal{F}$, though elements of $\mathcal{C}$ are still well-defined.

In essence, these properties of elements of $\mathcal{C}$ highlight some of the difficulties with which we are faced in the correspondence problem: Some features may be detected multiple times, while others are not detected at all, and the set of extractions may include phantom features which should not be associated with any real feature. The set $\mathcal{E}$ must be mapped to $\mathcal{F}$ despite these structural differences in the sets.

Let $q : \mathcal{C} \rightarrow \Re^+$ be a quality measure of assignments. The correspondence problem may be then be formulated as a search over $\mathcal{C}$ for an element $\hat{c} \in \mathcal{C}$ which maximizes q. A combinatorial analysis shows that $|\mathcal{C}| = (|\mathcal{F}| + 1)^{|\mathcal{E}|}$. Hence the correspondence problem is inherently exponential. We describe here an approach to find $\hat{c}$ in polynomial time.

2.2 The 1-dimensional case

Consider the special case where no two line-segments in $\mathcal{F}$ are parallel. We deal with more general cases in subsections 2.3 and 2.4. As a working example, let $\mathcal{F}$ be the set of line-segments making up an arbitrary triangle, as shown in figure 1. Let the normals to the three segments have orientations α_1, α_2 and α_3 relative to the global coordinate frame (GCF). Then the histogram of Φ-values of $\mathcal{F}$ has value 1 at α_1, α_2 and α_3, and is 0 elsewhere on the interval $[0, 2\pi)$. Denote this histogram $\mathcal{H}_{\mathcal{F}}^{\Phi}$. Similarly, let $\mathcal{H}_{\mathcal{E}}^{\Phi}$ denote the (discrete) histogram of Φ-values of $\mathcal{E}$. However, instead of each element of $\mathcal{E}$ contributing to the histogram equally, let the contribution of $e \in \mathcal{E}$ be the *number of ultrasound readings* which were clustered to form e (see [16] for an exposition of the clustering algorithm). The width of the "bins" in $\mathcal{H}_{\mathcal{E}}^{\Phi}$, w, represents the "slop" in orientation of extracted features which is to be tolerated. Examples of $\mathcal{E}$ and $\mathcal{H}_{\mathcal{E}}^{\Phi}$ are shown in figure 1. Note that the intervals between local modes of $\mathcal{H}_{\mathcal{E}}^{\Phi}$ are invariant under translations and rotations of the local coordinate frame (LCF) relative to the GCF. It is this invariance we exploit in order to find a good correspondence assignment $c \in \mathcal{C}$.

We now attempt to find the best correlation of $\mathcal{H}_{\mathcal{F}}^{\Phi}$ with a shifted version of $\mathcal{H}_{\mathcal{E}}^{\Phi}$. Let $\mathcal{H}_{\mathcal{E}}^{\Phi}(\beta)$ denote $\mathcal{H}_{\mathcal{E}}^{\Phi}$ shifted in the positive Φ-direction by β, with wrap-around at 2π. Assuming that the pairwise separations of α_1, α_2 and α_3 are all greater than the bin width w, the product $\mathcal{H}_{\mathcal{F}}^{\Phi} \cdot \mathcal{H}_{\mathcal{E}}^{\Phi}(\beta)$ will consist of at most three non-zero bins for each value of β. Let $\bar{p}(\beta)$ represent the three-vector of bin values of $\mathcal{H}_{\mathcal{E}}^{\Phi}(\beta)$ "picked out" by $\mathcal{H}_{\mathcal{F}}^{\Phi}$ for each $\beta = iw$, $0 \leq i \leq \lfloor \frac{2\pi}{w} \rfloor$. Examples of $\bar{p}(\beta)$ are shown in figure 1.

The final step consists of evaluating all the vectors $\bar{p}(\beta)$, $\beta = iw$, $0 \leq i \leq \lfloor \frac{2\pi}{w} \rfloor$, and selecting the "optimal" shift β. In other words, if $h : \mathbb{Z}^{|\mathcal{F}|} \rightarrow \Re$ represents an evaluation function of vectors of length $|\mathcal{F}|$ of integers, we search for a value of β, call it $\hat{\beta}$, which maximizes $h(\bar{p}(\beta))$. The choice of h depends on which properties of a correspondence assignment we choose to emphasize. In our case, we chose to stress two properties:

1. We wish to account for the greatest possible number of ultrasound readings. In other words, we wish to reward a high *mean* value of components of $\bar{p}(\beta)$.
2. We prefer the correspondence matches to be spread *evenly* over the elements of $\mathcal{F}$. In other words, we wish to reward low *standard deviation* among components of $\bar{p}(\beta)$.

Thus, the function h we selected for our experiments has the form $h(\bar{p}) = \lambda_1 \cdot \boldsymbol{m}(\bar{p}) - \lambda_2 \cdot \boldsymbol{s}(\bar{p})$ where $\boldsymbol{m} : \mathbb{Z}^{|\mathcal{F}|} \rightarrow \Re$ and $\boldsymbol{s} : \mathbb{Z}^{|\mathcal{F}|} \rightarrow \Re$ are, respectively, functions giving the mean and standard deviation of the components of a vector of length $|\mathcal{F}|$, and λ_1 and λ_2 are non-negative weighting factors.

Once $\hat{\beta}$ has been found, the correspondence matching is complete: use that $c \in \mathcal{C}$ which makes the following assignments:

- Tor each element $f \in \mathcal{F}$ with orientation ϕ_f in the GCF, map all elements in $\mathcal{E}$ which contributed to the bin in $\mathcal{H}_{\mathcal{E}}^{\Phi}(\hat{\beta})$ picked out by f. In other words, let the inverse image of f under c be

$$c^{-1}[\{f\}] = \{e \in \mathcal{E} \mid iw \leq \phi_e \leq (i+1)w,\ i = \lfloor \frac{\phi_f + \hat{\beta}}{w} \rfloor\}$$

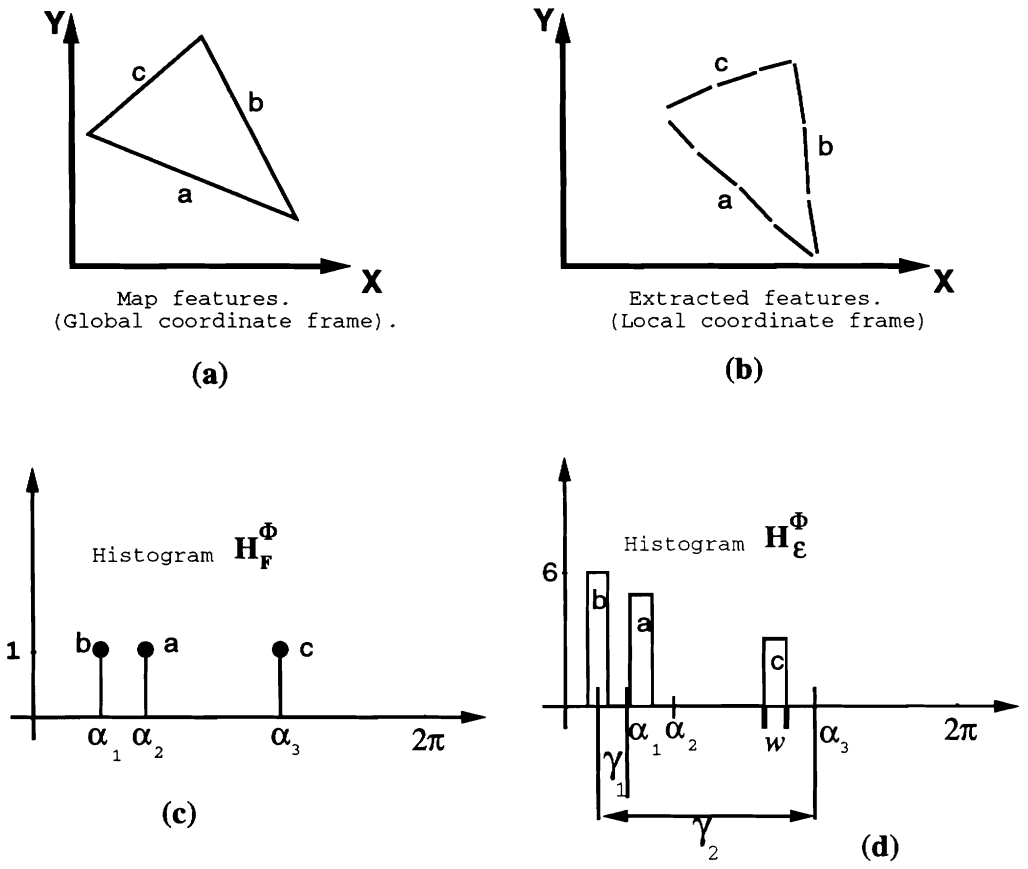
  where ϕ_e is the orientation of element e in the LCF.

- Map all remaining elements of $\mathcal{E}$ to the zero vector in $\mathcal{F} \cup \bar{0}$. i.e.

$$c\left[\mathcal{E} - c^{-1}[\mathcal{F}]\right] = \bar{0}$$

2.3 The 2-dimensional case

A difficulty arises when either $\mathcal{F}$ or $\mathcal{E}$ contains parallel line-segments (or at least line-segments whose angular separation is smaller than w so that they fall into the same bin in $\mathcal{H}_{\mathcal{F}}^{\Phi}$ or $\mathcal{H}_{\mathcal{E}}^{\Phi}$) but whose perpendicular spatial separation is substantial. In this case, projecting $\mathcal{F}$ and $\mathcal{E}$ into the Φ axis in the construction of $\mathcal{H}_{\mathcal{F}}^{\Phi}$ and $\mathcal{H}_{\mathcal{E}}^{\Phi}$, erases the distinction between these line-segments. This, in turn, leads to ambiguous correspondence matching if $\mathcal{F}$ contains the parallel segments, or at least one mismatched "phantom" segment if it is $\mathcal{E}$ that contains the parallel segments.



(e)

$$\bar{p}(0) = (0,0,0)$$
$$\bar{p}(\gamma_1) = (6,5,4)$$
$$\bar{p}(\gamma_2) = (0,0,6)$$

Figure 1: The 1-dimensional histogram case: (a) Map features in the GCF, (b) Extracted features in the LCF, (c) The histogram $\mathcal{H}_{\mathcal{F}}^{\Phi}$ with three non-zero values, one for each feature in $\mathcal{F}$. The labels represent correspondences between map features and histogram points. (d) The histogram $\mathcal{H}_{\varepsilon}^{\Phi}$. The labels represent correspondences between map feature clusters and histogram points. (e) Sample values of $\bar{p}(\beta)$.

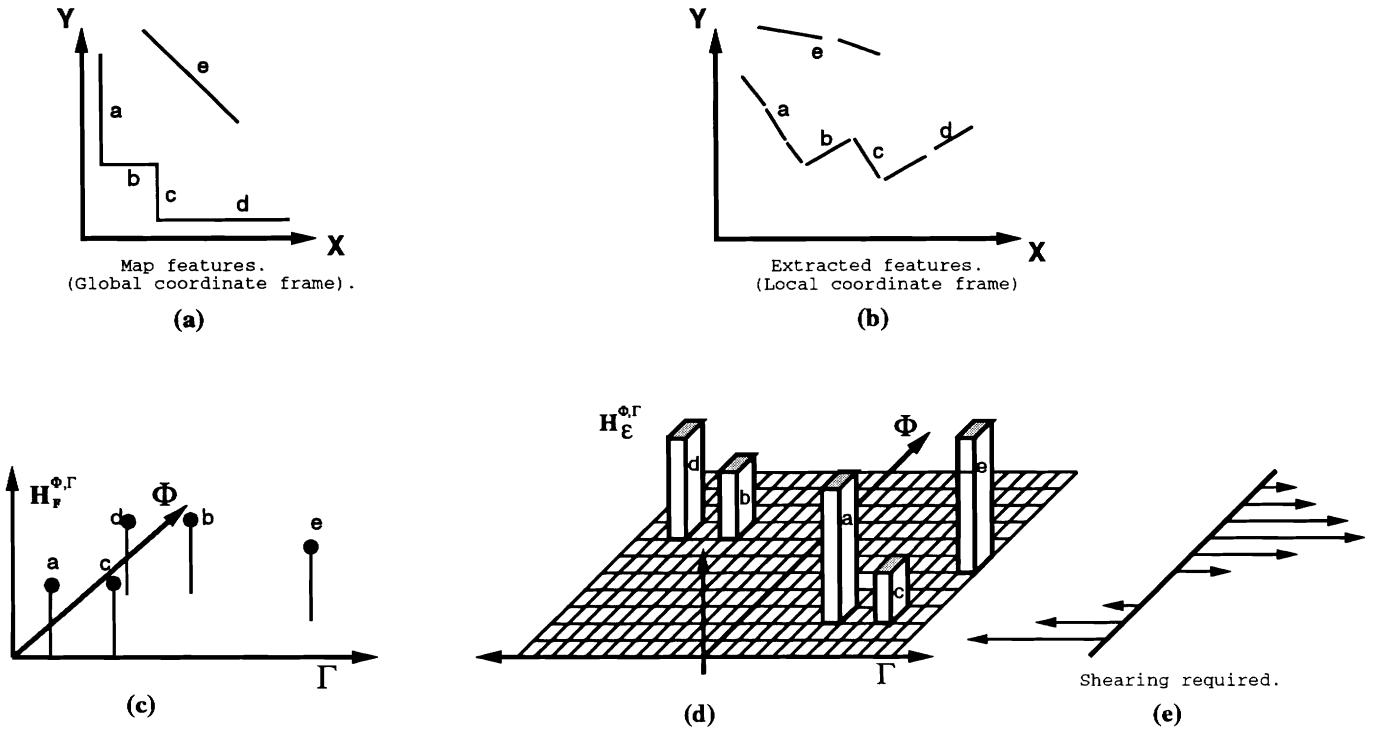


Figure 2: The 2-dimensional histogram case: (a) Map features in the GCF, (b) Extracted features in the LCF, (c) The histogram $\mathcal{H}_{\mathcal{F}}^{\Phi,\Gamma}$ with five non-zero values, one for each feature in $\mathcal{F}$. The labels represent correspondences between map features and histogram points. (d) The histogram $\mathcal{H}_{\mathcal{E}}^{\Phi,\Gamma}$. The labels represent correspondences between map feature clusters and histogram points. (e) Example shearing necessary to bring $\mathcal{H}_{\mathcal{E}}^{\Phi,\Gamma}$ into good correlation with $\mathcal{H}_{\mathcal{F}}^{\Phi,\Gamma}$ after a suitable (rigid) shift in the Φ direction.

In order to distinguish between parallel line-segments, we extend the histogram matching approach to the 2-dimensional case: Let $\mathcal{H}_{\mathcal{F}}^{\Phi,\Gamma}$ be the 2-dimensional histogram of $\mathcal{F}$ with respect to Φ and Γ. A point f in $\mathcal{F}$ contributes to the bin in $\mathcal{H}_{\mathcal{F}}^{\Phi,\Gamma}$ at location $\left(\lfloor\frac{\phi_f}{w_\Phi}\rfloor, \lfloor\frac{\rho_f}{w_\Gamma}\rfloor\right)$, where ϕ_f and ρ_f are the Φ- and Γ-values of f, and w_Φ and w_Γ are the resolutions of $\mathcal{H}_{\mathcal{F}}^{\Phi,\Gamma}$ in the Φ and Γ directions respectively. Similarly, let $\mathcal{H}_{\mathcal{E}}^{\Phi,\Gamma}$ denote the 2-dimensional histogram of $\mathcal{E}$. See figure 2 for a graphical depiction of the current running example.

As in the 1-dimensional case, the task is now to find shifts in $\mathcal{H}_{\mathcal{E}}^{\Phi,\Gamma}$ which will bring it into greatest correlation with $\mathcal{H}_{\mathcal{F}}^{\Phi,\Gamma}$. Unfortunately, however, in the 2-dimensional case, the relative locations of points in $\mathcal{H}_{\mathcal{E}}^{\Phi,\Gamma}$ are *not* invariant under translations and rotations of the LCF relative to the GCF. In fact, a straightforward trigonometric argument (see figure 3) shows that a rotation of the LCF by β followed by a spatial translation of (x, y) transforms a line with

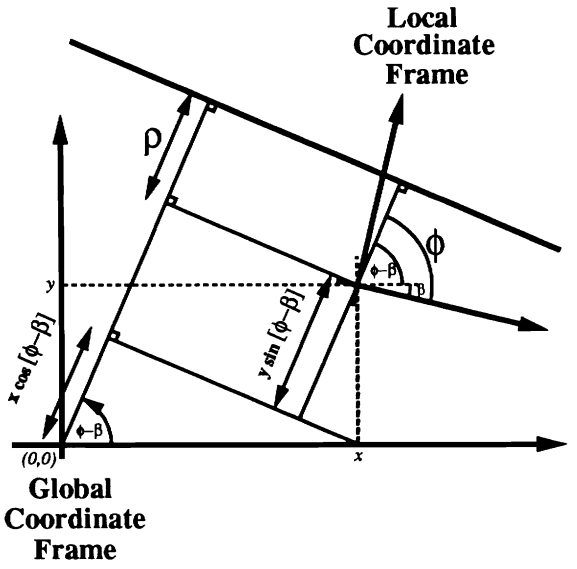


Figure 3: Rotation of the Local Coordinate Frame by β followed by spatial translation of (x, y) transforms a line with normalized parameters (ρ, ϕ) into the line with parameters $(\rho + x\cos(\phi - \beta) + y\sin(\phi - \beta), \phi - \beta)$.

normalized parameters (ρ, ϕ) into the line (ρ', ϕ'), where

$$\begin{pmatrix} \rho' \\ \phi' \end{pmatrix} = \begin{pmatrix} \rho + x\cos(\phi - \beta) + y\sin(\phi - \beta) \\ \phi - \beta \end{pmatrix} \tag{1}$$

Hence, relative differences in Φ are preserved. Similarly, for any fixed value of $\phi \in \Phi$, relative differences in Γ are preserved, since all lines with orientation ϕ are shifted in Γ by a constant factor $x\cos(\phi - \beta) + y\sin(\phi - \beta)$. However, lines with different Φ values are shifted by different amounts in the Γ direction. In other words, a **rotation** of the LCF corresponds to a rigid shift of $\mathcal{H}_{\mathcal{E}}^{\Phi,\Gamma}$ in the Φ direction; a *translation* of the LCF with respect to the GCF, however, corresponds to a *non-linear shearing* of $\mathcal{H}_{\mathcal{E}}^{\Phi,\Gamma}$. See figure 2 for a graphical depiction.

Let $\mathcal{H}_{\mathcal{E}}^{\Phi,\Gamma}(\beta, x, y)$ denote the 2-dimensional histogram of $\mathcal{E}$ after a rotation of β and a translation of (x, y). We describe here an approach to find the triple $(\hat{\beta}, \hat{x}, \hat{y})$ for which $\mathcal{H}_{\mathcal{E}}^{\Phi,\Gamma}(\hat{\beta}, \hat{x}, \hat{y})$ is in best correlation which $\mathcal{H}_{\mathcal{F}}^{\Phi,\Gamma}$. Once this has been done, we deduce the correspondence assignment $c \in \mathcal{C}$ in much the same way as in the 1-dimensional case.

Our approach to finding the triple $(\hat{\beta}, \hat{x}, \hat{y})$ is to decouple the searches for the rotation $\hat{\beta}$, and the translation $(\hat{x}, \hat{y})$.

The search for rotation $\hat{\beta}$: After a pure rotation, $\mathcal{H}_{\mathcal{E}}^{\Phi,\Gamma}(\beta,\, 0,\, 0)$ is related to $\mathcal{H}_{\mathcal{E}}^{\Phi,\Gamma}(0,\, 0,\, 0)$ by a rigid shift of β in the Φ direction. A heuristic approach to finding $\hat{\beta}$ consists of projecting $\mathcal{H}_{\mathcal{F}}^{\Phi,\Gamma}$ onto $\mathcal{H}_{\mathcal{F}}^{\Phi}$ and $\mathcal{H}_{\mathcal{E}}^{\Phi,\Gamma}$ onto $\mathcal{H}_{\mathcal{E}}^{\Phi}$ and then finding the best correlation of $\mathcal{H}_{\mathcal{F}}^{\Phi}$ with a shifted version of $\mathcal{H}_{\mathcal{E}}^{\Phi}$ as in the 1-dimensional case. We note that it is possible for the highest correlation estimate of $\mathcal{H}_{\mathcal{F}}^{\Phi}$ with $\mathcal{H}_{\mathcal{E}}^{\Phi}$ to result in a *non*-optimal correlation of $\mathcal{H}_{\mathcal{F}}^{\Phi,\Gamma}$ with $\mathcal{H}_{\mathcal{E}}^{\Phi,\Gamma}$. Cases for which this occurs usually involve multiple elements in $\mathcal{F}$ being regularly spaced in the Φ and Γ directions; we address this issue in section 2.4. For many cases, however, the computational efficiency afforded by the heuristic outweighs the sacrifice of a guarantee of optimality.

The search for translation $(\hat{x},\, \hat{y})$: Once a suitable rotation of the LCF relative to the GCF has been found, it is possible to use a spatial representation of elements of $\mathcal{F}$ and $\mathcal{E}$ to find the translation necessary for a good correspondence match. However, the rotation of all elements of $\mathcal{E}$ may be computationally prohibitive. We describe here an alternative approach which finds the parameters of shearing of $\mathcal{H}_{\mathcal{E}}^{\Phi,\Gamma}(\hat{\beta},\, 0,\, 0)$ necessary to bring it into best correspondence with $\mathcal{H}_{\mathcal{F}}^{\Phi,\Gamma}$. These parameters correspond to the translation $(\hat{x},\, \hat{y})$.

The approach consists of finding the amount of shearing of $\mathcal{H}_{\mathcal{E}}^{\Phi,\Gamma}(\hat{\beta},\, 0,\, 0)$ for each value of γ, $\gamma = iw_{\Phi}$, $0 \leq i \leq \lfloor \frac{2\pi}{w_{\Phi}} \rfloor$. Let $\mathcal{H}_{\mathcal{F}}^{\Phi,\Gamma}\,|_{\phi=\gamma}$ denote the 1-dimensional histogram "slice" of $\mathcal{H}_{\mathcal{F}}^{\Phi,\Gamma}$ at $\phi = \gamma$. Similarly, let $\mathcal{H}_{\mathcal{E}}^{\Phi,\Gamma}(\hat{\beta},\, 0,\, 0)\,|_{\phi=\gamma}$ denote the corresponding 1-dimensional histogram in $\mathcal{H}_{\mathcal{E}}^{\Phi,\Gamma}(\hat{\beta},\, 0,\, 0)$. For each $\gamma = iw_{\Phi}$, $0 \leq i \leq \lfloor \frac{2\pi}{w_{\Phi}} \rfloor$ for which $\mathcal{H}_{\mathcal{F}}^{\Phi,\Gamma}\,|_{\phi=\gamma}$ is non-zero, we find the best correlation between the 1-dimensional histograms $\mathcal{H}_{\mathcal{F}}^{\Phi,\Gamma}\,|_{\phi=\gamma}$ and $\mathcal{H}_{\mathcal{E}}^{\Phi,\Gamma}(\hat{\beta},\, 0,\, 0)\,|_{\phi=\gamma}$ using the method described in section 2.2. Denote by d_i the Γ-shift in $\mathcal{H}_{\mathcal{E}}^{\Phi,\Gamma}(\hat{\beta},\, 0,\, 0)\,|_{\phi=\gamma}$ necessary to achieve greatest correlation with $\mathcal{H}_{\mathcal{F}}^{\Phi,\Gamma}\,|_{\phi=\gamma}$.

We note from equation 1 that

$$\begin{aligned} d_i &= (\rho_{\mathrm{GCF}} - \rho_{\mathrm{LCF}}) \text{ at } \phi = iw_{\Phi} \\ &= \hat{x}\cos(iw_{\Phi} - \hat{\beta}) + \hat{y}\sin(iw_{\Phi} - \hat{\beta}) \\ &= a_i\hat{x} + b_i\hat{y} \end{aligned}$$

where $a_i = \cos(iw_{\Phi} - \hat{\beta})$ and $b_i = \sin(iw_{\Phi} - \hat{\beta})$. We may therefore estimate $\hat{x}$ and $\hat{y}$ by a least squares fit:

$$\begin{aligned} \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} &= \mathbf{A}^{\natural}\bar{d} \\ &= \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\bar{d} \end{aligned}$$

where $\mathbf{A} = [\bar{a},\, \bar{b}]$, $\bar{a} = [a_{i_1},\, a_{i_2},\, \ldots]^T$, $\bar{b} = [b_{i_1},\, b_{i_2},\, \ldots]^T$, $\bar{d} = [d_{i_1},\, d_{i_2},\, \ldots]^T$, and $i_1,\, i_2,\, \ldots$ are the values of i for which $\mathcal{H}_{\mathcal{F}}^{\Phi,\Gamma}\,|_{\phi=iw_{\Phi}}$ is non-zero. As a caveat, we note that a least squares approach is sensitive to outliers. For this

reason, least squares estimation may be replaced here with truncated least squares, least median of squares, or some other more robust estimator.

Once the triple $(\hat{\beta}, \hat{x}, \hat{y})$ has been found, the correspondence matching is complete: use that $c \in \mathcal{C}$ which makes the following assignments:

- For each element $f \in \mathcal{F}$ with Φ and Γ values ϕ_f and ρ_f in the GCF, map all elements in $\mathcal{E}$ which contribute to the bin in $\mathcal{H}_{\mathcal{E}}^{\Phi,\Gamma}(\hat{\beta}, \hat{x}, \hat{y})$ picked out by f. In other words, if f picks out bin (i, j) in $\mathcal{H}_{\mathcal{E}}^{\Phi,\Gamma}(\hat{\beta}, \hat{x}, \hat{y})$ (i.e. $i = \lfloor \frac{\phi_f + \hat{\beta}}{w_\Phi} \rfloor$ and $j = \lfloor \frac{\rho_f - \hat{x}\cos(\phi_f + \hat{\beta}) - \hat{y}\sin(\phi_f + \hat{\beta})}{w_\Gamma} \rfloor$), let the inverse image of f under c be

$$c^{-1}[\{f\}] = \{e \in \mathcal{E} \mid iw_\Phi \leq \phi_e \leq (i+1)w_\Phi \wedge jw_\Gamma \leq \rho_e \leq (j+1)w_\Gamma\}$$

  where ϕ_e and ρ_e are the Φ and Γ values of an element $e \in \mathcal{E}$ in the LCF.

- Map all remaining elements of $\mathcal{E}$ to the zero vector in $\mathcal{F} \cup \bar{0}$. i.e.

$$c\left[\mathcal{E} - c^{-1}[\mathcal{F}]\right] = \bar{0}$$

2.4 The general case

Even in the 2-dimensional case, ambiguity may arise in the search for the triple $(\hat{\beta}, \hat{x}, \hat{y})$. Instances where ambiguity may arise include the following cases:

1. For certain values of $\gamma = iw_\Phi$, $\mathcal{H}_{\mathcal{F}}^{\Phi,\Gamma} \mid_{\phi=\gamma}$ has regularly spaced non-zero bins, but not all the features corresponding to these bins are detected. In this case, the goodness of a match of $\mathcal{H}_{\mathcal{F}}^{\Phi,\Gamma} \mid_{\phi=\gamma}$ with $\mathcal{H}_{\mathcal{E}}^{\Phi,\Gamma}(\hat{\beta}, 0, 0) \mid_{\phi=\gamma}$ may be approximately equal for multiple values of Γ-shift d_i. An example of such an $\mathcal{F}$ is a set of parallel line segments, not all of which are detected. In this case a "mismatch" would be rejected as an outlier by a robust estimator of translation $(\hat{x}, \hat{y})$.

2. $\mathcal{F}$ contains features regularly spaced in Φ. In this case, if not all features in $\mathcal{F}$ are detected, the goodness of match of $\mathcal{H}_{\mathcal{E}}^{\Phi,\Gamma}(\beta)$ with $\mathcal{H}_{\mathcal{F}}^{\Phi,\Gamma}$ may be approximately equal for multiple values of β. An example of such an $\mathcal{F}$ is the set of line-segments representing a square room. In this case, the correlation between $\mathcal{H}_{\mathcal{F}}^{\Phi,\Gamma}$ and $\mathcal{H}_{\mathcal{E}}^{\Phi,\Gamma}(\beta)$ is equally good for four distinct values of β.

In the first case, the ambiguity may be resolved, or at least reduced, by using a different method for finding the best translation $(\hat{x}, \hat{y})$. As suggested in section 2.3, once $\hat{\beta}$ has been found, it is possible to use a spatial representation of

elements of $\mathcal{F}$ and $\mathcal{E}$ to find the translation necessary for a good correspondence match. Translations by (x, y) are evaluated according to the amount by which each element $e \in \mathcal{E}$, rotated by $\hat{\beta}$ and translated by (x, y) *overlaps* its counterpart in $\mathcal{F}$. For the second case, each candidate rotation can be evaluated by finding the best possible translation by the afore-mentioned spatial approach. The drawback of the spatial correlation approach is its computational inefficiency. This is exacerbated in the second case, where rotation and translation are no longer decoupled.

3 Pose estimation

Once correspondences have been established between extracted features and map features, a transformation must be found between the local and global (map) coordinate frames which maximizes some matching evaluation function. This decoupling of the correspondence matching phase from the pose estimation phase improves the precision of localization: correspondence matching is rastor-based and therefore computationally efficient but accurate only to the resolution of the $\mathcal{H}_{\mathcal{F}}^{\Phi,\Gamma}$ and $\mathcal{H}_{\mathcal{E}}^{\Phi,\Gamma}$ histograms; we can now afford to use a more precise approach to pose estimation with the knowledge that only "correct" correspondences are contributing to the final pose estimate. As far as orientation estimation is concerned, we describe an approach which is *linear* in the number of extracted features, combining precision with efficiency.

We divide the approach to aligning the LCF with the GCF into three phases: feature merging, translation estimation and rotation estimation.

3.1 Feature merging

The correspondence matching approach described in section 2 produces as output a many-to-one function $c : \mathcal{E} \rightarrow \mathcal{F} \cup \bar{0}$ where $\mathcal{F}$ is the set of planar features within the mobile robot's current ultrasonic "field of view," and $\mathcal{E}$ is the set of features extracted by the clustering algorithm described in [16].

In order to reduce the computational intensity of the translation and rotation estimation phases described in sections 3.2 and 3.3, we merge into a single feature all extracted features in $\mathcal{E}$ which have been deemed to correspond to the *same* underlying feature in $\mathcal{F}$. In the notation of section 2, we construct a new set of features $\mathcal{E}'$ and a new correspondence function $c' : \mathcal{E}' \rightarrow \mathcal{F} \cup \bar{0}$ as follows: Let $Merge : \mathcal{P}(\mathcal{F}) \rightarrow \Gamma \times \Phi \times C_x \times C_y \times L$ be a function which takes a set of extracted line-segments as an argument and produces the line-segment resulting from a merger of scatter matrices, as described in [16]. Begin with $\mathcal{E}' = c^{-1}[\{\bar{0}\}]$ and $\forall e' \in \mathcal{E}'$, $c'(e') = c(e')$. Then, for each feature

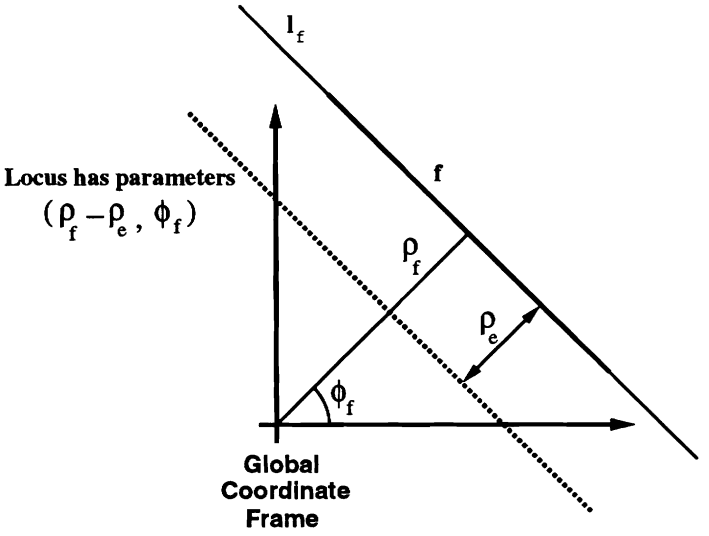


Figure 4: The locus (dotted) of positions (x_p, y_p) of the mobile robot within the GCF for which $l_{e'} = (\rho_{e'}, \phi_{e'})$ (not shown) lies on $l_f = (\rho_f, \phi_f)$.

$f \in \mathcal{F}$, add to $\mathcal{E}'$ the line-segment $e' = Merge\left(c^{-1}[\{f\}]\right)$ and define $c'(e') = f$.

Computational complexity: The weighted summation of 2×2 matrices is *linear* in the number of matrices. Computation of the eigenvectors of the resultant 2×2 scatter matrix requires a constant number of operations. Computation of endpoints of a line-segment representing a cluster of features may be performed in time *linear* in the size of the cluster. Hence, feature merging is *linear* in the number of extracted features.

3.2 Translation estimation

Let e'_i be the ith element of $\left(\mathcal{E}' - c'^{-1}[\{\bar{0}\}]\right)$. Denote the infinite line of which e'_i is a segment by $l_{e'_i}$. The normal parameters of $l_{e'_i}$ *in the LCF* are $\rho_{e'_i}$ and $\phi_{e'_i}$. Denote the underlying feature corresponding to e'_i by $f_i = c'(e'_i) \in \mathcal{F}$, and the infinite line of which it is a segment by l_{f_i}. The normal parameters of l_{f_i} *in the GCF* are ρ_{f_i} and ϕ_{f_i}.

If the LCF is rotated counter-clockwise by $\phi_{f_i} - \phi_{e'_i}$, then $l_{e'_i}$ can be made to lie on l_{f_i} for a suitable translation of the LCF. The locus of positions (x_p, y_p) of the mobile robot within the GCF for which $l_{e'_i}$ lies on l_{f_i} is the line with normal parameters $\rho_{f_i} - \rho_{e'_i}$ and ϕ_{f_i}. See figure 4. Hence, x_p and y_p are such that

$$x_p \cos\phi_{f_i} + y_p \sin\phi_{f_i} = \rho_{f_i} - \rho_{e'_i}$$

Hence, the set of constraints governing the position (x_p, y_p) of the mobile robot within the GCF may be expressed as

$$\mathbf{A}\begin{pmatrix} x_p \\ y_p \end{pmatrix} = \bar{d}$$

$$\text{where } \mathbf{A} = \begin{bmatrix} \cos\phi_{f_1} & \sin\phi_{f_1} \\ \cos\phi_{f_2} & \sin\phi_{f_2} \\ \vdots & \vdots \\ \cos\phi_{f_n} & \sin\phi_{f_n} \end{bmatrix} \text{ and } \bar{d} = \begin{pmatrix} \rho_{f_1} - \rho_{e'_1} \\ \rho_{f_2} - \rho_{e'_2} \\ \vdots \\ \rho_{f_n} - \rho_{e'_n} \end{pmatrix}$$

where $n = |\left(\mathcal{E}' - c'^{-1}[\{\bar{0}\}]\right)|$. A least squares estimate of the necessary translation of the LCF is, therefore,

$$\begin{aligned} \begin{pmatrix} x_p \\ y_p \end{pmatrix} &= \mathbf{A}^{\natural}\bar{d} \\ &= \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\bar{d} \end{aligned} \tag{2}$$

Note the similarity in structure with the least squares approximation of translation described in section 2.3. The difference is that in section 2.3, *rastorized* histograms of $\mathcal{F}$ and $\mathcal{E}$ were used to obtain a *coarse* approximation to translation, and to eliminate mismatched features; in the present section, we assume knowledge of correspondence between features, and use *non-discretized* data to obtain a refined estimate of translation.

Once again, least squares estimation may be replaced here with truncated least squares, least median of squares, or some other robust estimator.

Computational complexity: The number of operations required for the evaluation of the matrix $\mathbf{A}$ is *linear* in n. The various matrix multiplications are also *linear* in n. The matrix inversion is performed on a 2×2 matrix, requiring a constant number of operations. Hence, the translation estimation phase is, overall, *linear* in n.

3.3 Rotation estimation

Once a suitable translation of the LCF has been found, all that remains is to find the rotation θ_p of the translated LCF which minimizes some disparity criterion. The criterion we have selected is the sum of squared perpendicular

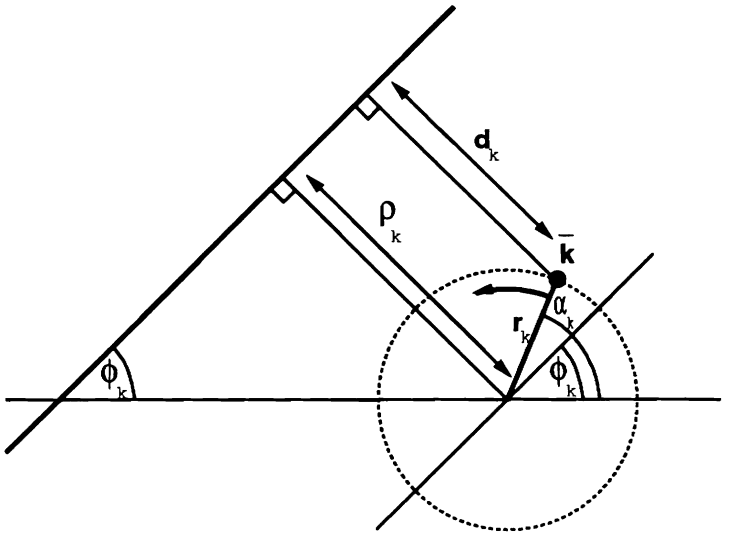


Figure 5: Perpendicular distance d_k of an arbitrary point $\bar{k} = (r_k, \alpha_k)$ (polar coordinates) from a line with with parameters ρ_k, ϕ_k) as a function of rotation θ of $\bar{k}$: $d_k(\theta) = \rho_k - r_k \sin(\alpha_k - \phi_k + \theta)$

distances between the endpoints of extracted features and the infinite lines to which these features have been deemed to correspond.

Let $\bar{k} = (r_k, \alpha_k)$ be an arbitrary point in the LCF expressed in polar coordinates. Let d_k denote the perpendicular distance of $\bar{k}$ from some line in the LCF with parameters ρ_k and ϕ_k. As can be seen from figure 5, after a counter-clockwise rotation of $\bar{k}$ by angle θ,

$$d_k = \rho_k - r_k \sin(\alpha_k - \phi_k + \theta) \tag{3}$$

Now, once again let e'_i be the ith element of $(\mathcal{E}' - c'^{-1}[\{\bar{0}\}])$. Let the endpoints of e'_i be $\bar{k}_i$ and $\bar{j}_i$. Let the polar coordinates of $\bar{k}_i$ and $\bar{j}_i$ be (r_{k_i}, α_{k_i}) and (r_{j_i}, α_{j_i}) respectively. Denote the underlying feature corresponding to e'_i by $f_i = c'(e'_i) \in \mathcal{F}$, and the infinite line of which it is a segment by l_{f_i}. The normal parameters of l_{f_i} *in the GCF* are $\rho_{f_i}^{GCF}$ and $\phi_{f_i}^{GCF}$. By equation 1, the parameters of l_{f_i} *in the unrotated LCF* are

$$\begin{pmatrix} \rho_{f_i}^{LCF} \\ \\ \phi_{f_i}^{LCF} \end{pmatrix} = \begin{pmatrix} \rho + x_p \cos\phi_{f_i}^{GCF} + y_p \sin\phi_{f_i}^{GCF} \\ \\ \phi_{f_i}^{GCF} \end{pmatrix}$$

For ease of notation, in what follows denote $\rho_{f_i}^{LCF}$ simply by ρ_i and $\phi_{f_i}^{LCF}$ by ϕ_i. Hence, for $n = |(\mathcal{E}' - c'^{-1}[\{\bar{0}\}])|$, the sum of squared perpendicular distances after a counter-clockwise rotation of endpoints by θ is $\sum_{i=1}^{n} (d_{k_i}^2 + d_{j_i}^2)$,

where, by equation 3,

$$\begin{aligned}
\sum_{i=1}^{n} d_{k_i}^2 &= \sum_{i=1}^{n} \left(\rho_i - r_{k_i} \sin\left(\alpha_{k_i} - \phi_i + \theta\right)\right)^2 \\
&= \sum_{i=1}^{n} \left(\rho_i^2 - 2\rho_i r_{k_i} \sin\left(\alpha_{k_i} - \phi_i + \theta\right) + r_{k_i}^2 \sin^2\left(\alpha_{k_i} - \phi_i + \theta\right)\right) \\
&= \sum_{i=1}^{n} \rho_i^2 - 2\sum_{i=1}^{n} \rho_i r_{k_i} \sin\left(\alpha_{k_i} - \phi_i + \theta\right) + \sum_{i=1}^{n} r_{k_i}^2 \sin^2\left(\alpha_{k_i} - \phi_i + \theta\right) \\
\text{and since } \sin^2\gamma &= \frac{1 - \cos 2\gamma}{2}, \\
\sum_{i=1}^{n} d_{k_i}^2 &= \sum_{i=1}^{n} \rho_i^2 - 2\sum_{i=1}^{n} \rho_i r_{k_i} \sin\left(\alpha_{k_i} - \phi_i + \theta\right) + \sum_{i=1}^{n} r_{k_i}^2 \left(\frac{1 - \cos\left(2\alpha_{k_i} - 2\phi_i + 2\theta\right)}{2}\right) \qquad (4)
\end{aligned}$$

$\sum_{i=1}^{n} d_{j_i}^2$ has similar form. As a function of θ, the second term in equation 4 is the sum of sinusoids of equal frequency 1. It is, hence, also a sinusoid of frequency 1. Similarly, the third term in equation 4 is the sum of sinusoids of frequency 2. It is, therefore, also a sinusoid of frequency 2. Hence, we may write

$$g(\theta) = \sum_{i=1}^{n} \left(d_{k_i}^2 + d_{j_i}^2\right) = B_1 + B_2 \sin\left(\theta + \mu_1\right) + B_3 \sin\left(2\theta + \mu_2\right) \qquad (5)$$

for some constants B_1, B_2 and B_3, μ_1 and μ_2. We seek the value of θ which minimizes $g(\theta)$. This function has at most two troughs in the interval $\theta \in [0, 2\pi)$. It can be shown that if we sample g at four points equally spaced in θ, call them θ_1, θ_2, θ_3 and θ_4, $\theta_j - \theta_{j-1} = \frac{\pi}{2}$, $2 \leq j \leq 4$, then $\min_{1 \leq j \leq 4} g(\theta_j)$ lies in a trough containing a *global* minimum of $g(\theta)$. Hence, an efficient approach to finding θ_p for which $g(\theta_p)$ is a global minimum is

1. Evaluate $g(0)$, $g(\frac{\pi}{2})$, $g(\pi)$ and $g(\frac{3\pi}{2})$ using equation 4.
2. Find the minimum of these four values.
3. Use the value of θ corresponding to this minimum as an initial estimate in a Newton-Raphson (NR) iterative approximation of θ_p.

Since the NR algorithm converges rapidly, only a few evaluations of the function $\sum_{i=1}^{n} \left(d_{k_i}^2 + d_{j_i}^2\right)$ by way of the right hand side of equation 4 will be necessary. Nevertheless, for large values of n, evaluations of $\sum_{i=1}^{n} \left(d_{k_i}^2 + d_{j_i}^2\right)$ may become prohibitively computationally intensive. For this reason, we use the following approach:

We evaluate $\sum_{i=1}^{n} \left(d_{k_i}^2 + d_{j_i}^2\right)$ at five points in order to solve for the five unknown parameters of $g(\theta)$ in equation 5. Let the five values of θ at which we evaluate $\sum_{i=1}^{n} \left(d_{k_i}^2 + d_{j_i}^2\right)$ be 0, $\frac{\pi}{4}$, $\frac{\pi}{2}$, π and $\frac{3\pi}{2}$. Then, substitution into equation 5

yields

$$g(0) = B_1 + B_2 \sin\mu_1 + B_3 \sin\mu_2 \quad (6)$$

$$g\left(\frac{\pi}{2}\right) = B_1 + B_2 \cos\mu_1 - B_3 \sin\mu_2 \quad (7)$$

$$g(\pi) = B_1 - B_2 \sin\mu_1 + B_3 \sin\mu_2 \quad (8)$$

$$g\left(\frac{3\pi}{2}\right) = B_1 - B_2 \cos\mu_1 - B_3 \sin\mu_2 \quad (9)$$

$$g\left(\frac{\pi}{4}\right) = B_1 - B_2 \cos(\frac{\pi}{4} + \mu_1) - B_3 \cos\mu_2 \quad (10)$$

(11)

$$(6)+(7)+(8)+(9) \Rightarrow \quad B_1 = \frac{g(0) + g(\frac{\pi}{2}) + g(\pi) + g(\frac{3\pi}{2})}{4} \quad (12)$$

$$(6)-(8) \Rightarrow \quad B_2 \sin\mu_1 = \frac{g(0) + g\left(\frac{\pi}{2}\right)}{2} \quad (13)$$

$$(7)-(9) \Rightarrow \quad B_2 \cos\mu_1 = \frac{g\left(\frac{\pi}{2}\right) + g\left(\frac{3\pi}{2}\right)}{2} \quad (14)$$

$$(13)\ /\ (14) \Rightarrow \quad \mu_1 = \tan^{-1}\left(\frac{g(0) + g(\pi)}{g\left(\frac{\pi}{2}\right) + g\left(\frac{3\pi}{2}\right)}\right) \quad (15)$$

$$(13) \Rightarrow \quad B_2 = \frac{g(0) + g(\pi)}{2\sin\mu_1} \quad (16)$$

$$(6)+(8) \Rightarrow \quad B_3 \sin\mu_2 = \frac{g(0) + g(\pi) - 2B_1}{2} \quad (17)$$

$$(10) \Rightarrow \quad B_3 \cos\mu_2 = B_2 \sin\left(\frac{\pi}{4} + \mu_1\right) - g\left(\frac{\pi}{4}\right) \quad (18)$$

$$(17)\ /\ (18) \Rightarrow \quad \mu_2 = \tan^{-1}\left(\frac{g(0) + g(\pi) - 2B_1}{2\left(B_2 \sin\left(\frac{\pi}{4} + \mu_1\right) - g\left(\frac{\pi}{4}\right)\right)}\right) \quad (19)$$

$$(17) \Rightarrow \quad B_3 = \frac{g(0) + g(\pi) - 2B_1}{2\sin\mu_2} \quad (20)$$

From equations 12, 15, 16, 19 and 20, we obtain an explicit formulation by which $g(\theta)$ may be evaluated iteratively at low computational cost.

Computational complexity The five evaluations of $\sum_{i=1}^{n}\left(d_{k_i}^2 + d_{j_i}^2\right)$ may be performed in time *linear* in the number of matched extracted features n. The time required for the iterative application of the NR algorithm is negligible compared to this. Thus, rotation estimation may be performed in time *linear* in n.

In this way, a precise estimate of the pose (x_p, y_p, θ_p) is obtained. The overall computational complexity of the pose estimation phase is *linear* in the number of extracted features.

4 Experiments

We are currently involved in the empirical testing and evaluation of the algorithm, both in simulation and on our physical testbed agents. Simulation results are very encouraging. A thorough analysis of the localization algorithm's properties on a real testbed system requires a measure of ground truth against which pose estimates may be compared. A system for establishing this ground truth is currently under development. See [17] for an analysis of the algorithm's empirical performance in terms of speed, precision and region of attraction[1]. See [14] for a comparison of localization characteristics using this modality with localization using various combinations of other modalities.

Two preliminary testbed experiments illustrate the usefulness of our approach:

Region of attraction: In this experiment we show the large region of attraction of the localization algorithm, both for *relocation* and for *continuous localization*, at least for the simple case ($|\mathcal{F}| = 2$) shown in figure 6.

Ultrasound localization versus odometry In this experiment we compare odometry-based against ultrasound feature-based localization. In the absence of a method of establishing ground truth, we proceed as follows:

1. Begin from some marked point in space. Identify the GCF with the LCF.
2. Steer the robot in a loop, and return to the original point in space.
3. Compare the estimates of final location and orientation according to odometry and the ultrasound feature-based localization algorithm.

We note that in our case, *no* continuous localization was performed; *all* data was subject to odometric error, so that the feature-based localization algorithm was at a distinct disadvantage. If continuous localization or some form of ground truth positioning system were to be used instead of odometry to keep track of the robot's pose for the duration of the experiment, we would expect an improvement in the quality of the parameters of the extracted features. This would lead to a corresponding improvement in relocation at the conclusion of the experiment.

Nevertheless, as is shown in figure 7, the relocation algorithm succeeds in reducing odometric error by at least 50%.

[1] The region of attraction for a known pose $q \in \mathcal{Q}$ is defined as the region in pose space $\mathcal{Q}' \subseteq \mathcal{Q}$ such that, for any $q' \in \mathcal{Q}'$, a robot with initial LCF origin at q' is be able to localize itself to within some threshold distance of q. The region of attraction depends on the nature of the environment, the point q, and the features that have been extracted thus far.

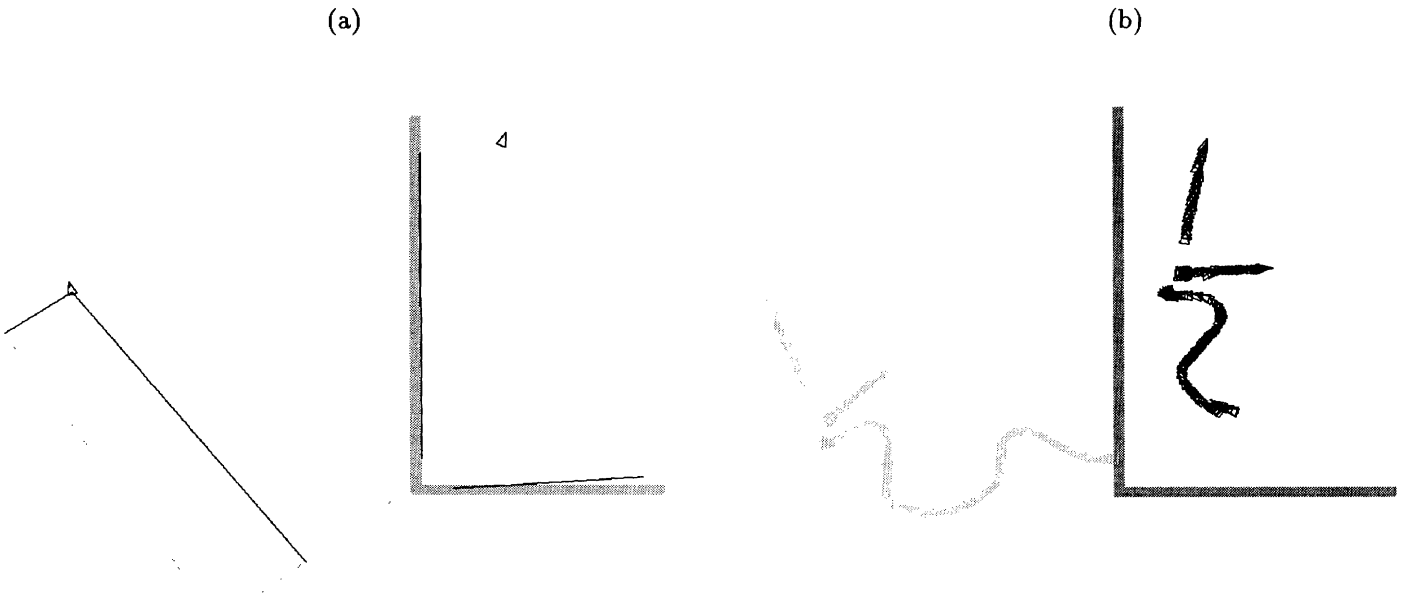


Figure 6: Experiment for which region of attraction is large. (a) Relocation. Light grey represents clusters of extracted features. Darker grey denotes features in the map. Dark lines on left show ρ values of extracted features in the LCF. Robot location in the LCF is shown on left. Dark features on right hand side represent the extracted features cast into the GCF after localization. The relocated robot is shown in its new pose in the upper right corner. (b) Continuous localization. Light grey represents the trajectory followed by the robot in the LCF. Darker grey represents environment structures (features in $\mathcal{F}$). Black represents the trajectory of the robot in the GCF as a result of continuous localization.

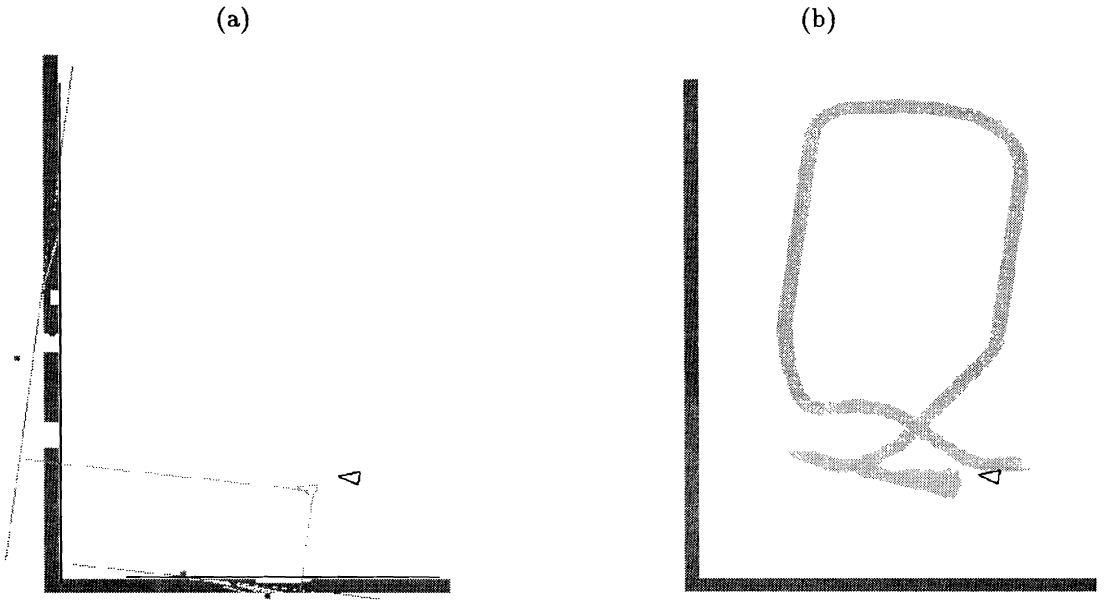


Figure 7: Experiment to compare odometry-based against ultrasound feature-based relocation. (a) Comparison of final orientation estimate. Light grey represents clusters of extracted features in the LCF, ρ values for these features, and the final pose of the robot according to odometry. Darker grey denotes features in the map. Dark features represent the extracted features cast into the GCF after relocation. Note the goodness of fit with map features. The relocated robot is shown in its new pose as a dark triangle. Note that localization algorithm reduces the odometric error in orientation by about 50%: in reality, final orientation was due west. This is in spite of the corruption of each individual ultrasound measurement by odometric error. (b) Comparison of final translation estimate. Light grey represents the trajectory followed by the robot in the LCF (i.e. according to odometry readings). Starting point is at the right. Endpoint cannot be clearly seen as the robot was maneuvered back and forth to ensure that, in reality, it ended at its starting point. Odometric slippage is clearly visible. Darker grey represents environment structures (features in $\mathcal{F}$). Black represents the final estimated pose of the robot in the GCF as a result of relocation. The localization algorithm is seen to reduce translation error by about 50%, placing the robot closer to its known starting point. Once again, this is in spite of the corruption of each individual ultrasound measurement by odometric error.

5 Conclusion

We have described a *feature-based* localization algorithm for mobile robots equipped with fixed ultrasonic transducers. We do not assume the presence of beacons, nor do we require the modification of the environment in any way.

We employ the method delineated in [15, 16] for the extraction of planar features from ultrasound data in the local coordinate system of the mobile robot. The advantages of using extracted features rather than unclustered ultrasonic measurements for localization include

1. Greater robustness and noise immunity: each feature represents the combined information from *many* measurements over *relatively long* periods.
2. Greater speed since the same feature matches may be used over long periods.

We address the issues of correspondence matching and pose estimation. The algorithm described here decouples these two phases. The matching stage is histogram-based, yielding a coarse estimate of pose and a function mapping extracted features to features in the map. Using this function, the pose estimation stage makes use of least squares estimation to yield a refined estimate of translation and rotation. The pose estimation stage is shown to be *linear* in the number of extracted features.

The decoupling of the matching and pose estimation phases allows the speed of rastorized techniques to be combined with the precision and finer resolution of non-discretized estimation. Furthermore, the complete correspondence matching phase need only be invoked for the purposes of *relocation*. For the case of *continuous localization*, the same correspondence matches may be used over long periods; only the pose estimation phase need be invoked during these periods.

The approach we delineate minimizes a sum of squared errors expression. Rapidly convergent gradient-descent techniques are employed for this minimization in the case of orientation estimation; we avoid local minima by selecting a initial estimate which *guarantees* convergence to the *global* minimum. The overall result is a localization algorithm which is both computationally efficient and accurate. Such an algorithm is a key component for the tasks of navigation, exploration of partially known environments, and cooperative material handling by multiple agents.

We are currently involved in the empirical testing and evaluation of the algorithm, both in simulation and on our physical testbed agents. As part of this effort, a system for establishing ground truth is under development. The accuracy and region of attraction of the localization algorithm are under analysis. Further, the algorithm is

being extended to incorporate point-type ultrasound features (corners), as well as features extracted by other sensor modalities. See [14] for details.

References

[1] R. Bauer. Active manoeuvres for supporting the localisation process of an autonomous mobile robot. In *Proceedings of the International Workshop on Intelligent Robotic Systems '94*, July 1994. Grenoble, France.

[2] M. Betke and K. Gurvits. Mobile robot localization using landmarks. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 135–142, May 1994.

[3] S. Borthwick and H. F. Durrant-Whyte. Simultaneous localisation and map building for autonomous guided vehicles. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pages 761–768, September 1994. Munich, Germany.

[4] S. Borthwick, M. Stevens, and H. F. Durrant-Whyte. Position estimation and tracking using optical range data. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pages 2172–2177, July 1993. Yokohama, Japan.

[5] R. G. Brown, L. P. Chew, and B. R. Donald. Mobile robots, map-making, shape metrics, and localization. *Submitted to IEEE ICRA*, 1993.

[6] J. L. Crowley. World modeling and position estimation for a mobile robot using ultrasonic ranging. *1989 IEEE International Conference on Robotics and Automation*, May 1989.

[7] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis.* Wiley-Interscience, 1973.

[8] A. Elfes. Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation*, RA-3(3):249–265, June 1987.

[9] A. Elfes. Multi-source spatial fusion using bayesian reasoning. In M. A. Abidi and R. C. Gonzalez, editors, *Data Fusion in Robotics and Machine Intelligence*, pages 137–163. Academic Press, 1992.

[10] A. A. Holenstein, M. A. Müller, and E. Badreddin. Mobile robot localization in a structured environment cluttered with obstacles. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2576–2581, May 1992. Nice, France.

[11] J. J. Leonard. *Directed Sonar Sensing for Mobile Robot Navigation.* PhD thesis, University of Oxford, 1990.

[12] J. J. Leonard and H. F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7(3), June 1991.

[13] P. MacKenzie and G. Dudek. Precise positioning using model-based maps. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 1867–1874, May 1994.

[14] R. Mandelbaum. *Sensor Fusion for Mobile Robot Localization, Exploration and Navigation.* PhD thesis, University of Pennsylvania. In preparation.

[15] R. Mandelbaum and M. Mintz. Sonar signal processing using tangent clusters. Technical report, University of Pennsylvania, 1994. In preparation.

[16] R. Mandelbaum and M. Mintz. Sonar signal processing using tangent clusters. In *Proceedings of the OCEANS '94: special session on Automated Unmanned Vehicles*, September 1994. Brest, France.

[17] R. Mandelbaum and M. Mintz. Testing and evaluation of a feature-based localization algorithm for fixed ultraonic transducers. Technical report, University of Pennsylvania, 1994. In preparation.

[18] B. Schiele and J. L. Crowley. A comparison of position estimation techniques using occupancy grids. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1628–1634, May 1994.

[19] K. T. Sutherland. Landmark selection for accurate navigation. In *Proceedings of the 1993 Image Understanding Workshop*, pages 485–490, April 1993.

[20] K. T. Sutherland and W. B. Thompson. Inexact navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, May 1993.

[21] W. B. Thompson, T. C. Henderson, T. L. Colvin, L. B. Dick, and C. M. Valiquette. Vision-based localization. In *Proceedings of the 1993 Image Understanding Workshop*, pages 491–498, April 1993.