Making Difficult Decisions Autonomously: The Impact of Integrated Mapping and Navigation

Christopher M. Smith[1], John J. Leonard[2], and Hans Jacob S. Feder[2]
[1]C.S. Draper Laboratory, 555 Technology Square, Cambridge, MA 02139
[2]Massachusetts Institute of Technology, 77 Mass Ave., Cambridge, MA 02139
cmsmith@alum.mit.edu,feder@deslab.mit.edu,jleonard@mit.edu

Abstract

The role of navigation is changing. The forces of increased autonomy, less prior knowledge, and larger missions are extending the navigation problem from the requirement of absolute localization to the larger question of context determination. Current technologies are inadequate in the face of such circumstances. The key to an evolved navigation technology begins with the ability to reason, in an integrated way, about the models used to determine vehicle context: physical models, dynamic models, sensor models, and behavior models. The integrated mapping and localization (IMAN) algorithm provides a hybrid estimation scheme to integrate decision-making about navigation events with navigation and mapping. An overview of IMAN is presented, along with an initial analysis of its performance. While IMAN is sensitive to the complexity of ambiguous situations, the algorithm demonstrates superior performance when complexity does not lead to failure. These results are used to examine the emerging set of technological needs for advanced navigation and mapping applications, including map representation, multiscale modeling, map fusion, and cross-model correlation.

1 Introduction

Navigation is usually thought of in fairly simple terms. The vehicle is operating in some environment. Knowledge of the vehicle's location is used in specifying mission requirements and in selecting appropriate actions for the vehicle (i.e. the guidance and control problems). Because these facets of the vehicle software, guidance, navigation, and control, are usually considered as separate and monolithic modules, navigation performance is often defined on its own terms. The ability of the vehicle to navigate is specified in absolute localization accuracy. Other navigation tasks, such as mapping and target tracking, are presented in the same manner. The traditional role of navigation is to provide an absolute global estimate of the state of the vehicle and any aspects of its surroundings that are of interest [4].

Recently, considerable emphasis has been placed on the concurrent mapping and localization (CML) problem [13, 8, 11, 15]. The goal of CML is for a mobile robot to build a map of an unknown environment while simultaneously using that map to improve its navigation. The essential advance in CML has been in addressing the issue of data association, the uncertainty involved in correctly associating measurements with the objects in the environment from which they arise [3]. Initial approaches to CML, such as stochastic mapping (SM), either ignored possible data-association errors, matched measurements by hand, or used a nearest-neighbor approach [13, 8]. Alternative approaches have adopted a field-based representation of the environment [14]. While this reduces the assumptions made by prior models about the environment, it fails to capture the persistence of environmental objects to allow higher-level reasoning about the behavior of these objects through time. Feature-based approaches are typically implemented using some form of Kalman filter. Uncertainty in data association creates a problem for such implementations; there is now uncertainty about how to model the measurements. Additionally, when mapping unknown regions, the set of features which is present must be estimated in addition to the state of these features, creating further modeling problems [7].

In Section 2, we consider the expanding role of navigation for autonomous and semi-autonomous sysems and the structural requirements such changes place on navigation and mapping solutions. An initial implementation of such a solution, integrated mapping and navigation (IMAN), is presented in Section 3. After describing the algorithmic structure of IMAN, some comparative results are detailed in Section 4. Section 5 concludes the paper with a discussion of the technological needs pointed out by our consideration of the changing role of navigation and experiences implementing IMAN.

2 The Broader Role of Navigation

Concurrent mapping and localization is just one example of a navigation problem that stretches the conception of navigation beyond its traditional bounds. This expansion extends navigation from merely localizing the vehicle to the problem of determining the context of the robot, evaluating the situation in which it finds itself. There are additional forces pushing modern navigation techniques in this direction. Increased autonomy, resulting from either the need to operate robustly in the presence of unreliable communication channels or a desire to reduce the cognitive burden on the operator, requires the ability for the vehicle to handle more ambiguous situations by itself. This can be accomplished either by asserting some set of *a priori* assumptions for such decisions, a choice which can result in failure when such expected behavior does not obtain, or by allowing the vehicle to reason about the decisions being made. Operation in unknown or more varied environments can lead to an effective reduction in prior knowledge, requiring a greater effort to disambiguate navigational events in stride. Finally, increased mission length can be predicated on the ability to carry out additional decision-making tasks at the vehicle level. All of these trends point toward greater autonomy and increased decision-making capabilities, competencies that require a broadening of navigation solutions.

The key to this broadening of the role of navigation is to understand that navigation, at a fundamental level, is not so much about determining where the vehicle is as it is about determining the context of the vehicle. Context determination depends, of course, on what is important about the environment in which the vehicle is operating. This situational assessment is the result of fusing two types of knowledge. The first is sensor data, which can itself come in multiple modalities. Second, there is prior knowledge. The *a priori* knowledge that the vehicle has is not necessarily restricted to specific knowledge about the environment (there is a hill at these coordinates) but can also include ontological information, that is, information about what to expect when interpreting the data (a hill looks like this and there may be hills in this area). While there are certainly situations in which knowledge of the vehicle's absolute position is crucial, there are a wide variety of situations in which relative localization is sufficient to achieve the mission objectives. Navigation should never become an end in itself, rather it should provide the appropriate contextual information within which the guidance and control (or, more broadly, the requirement specification and action selection) problems can be attacked. This broader interpretation of navigation as context determination becomes critical as more extensive modeling of the environment and decision-making regarding navigational events come to form the basis of the vehicle's contextual knowledge.

These observations bear directly on the question of how navigation solutions should be structured. Two points are particularly relevant given the current state-of-the-art in mobile robot navigation. First, in order to gain leverage from ontological prior knowledge, a feature-based representation of the environment is necessary at some level. It is only at the feature level that persistence and behavior of environmental objects can be captured. While field-based (or grid-based) methods may have some use in low-level processing, such as obstacle avoidance, decision-making activities are best situated posterior to feature extraction. Second, the role of navigational events is important for feature-based navigation solutions. While field-based systems skirt the necessity of contextual assumptions by foregoing persistent models, feature-based solutions depend on these assumptions unless some higher level of event-based reasoning is available. Vanward solutions to the navigation problem should be structured to handle reasoning about navigational events within a feature-based representation scheme. These structural constraints are by no means the last word on the structure of the navigation problem. Aspects of map representation, decision granularity, and information fusion remain to be determined within particular solutions, but these traits are indicative of solutions that can handle the increased autonomy and broader responsibilities that modern navigation implementations require.

3 Integrated Mapping and Navigation

We have developed an initial implementation to concurrent mapping and localization that both uses feature-based representation of the environment and explicitly models decision-making regarding navigational events. In IMAN, individual measurement returns are considered the navigational events of interest. It is assumed that individual features can be extracted from the sensor data. For the current implementation, point-like features are considered. Each given return may come from a known feature or an object which is not yet being modeled, or it may be spurious, e.g. the result of clutter or multipath effects.

Stochastic mapping uses a single state vector and error covariance matrix. In IMAN, the vehicle model and each proposed feature are represented by a distinct model. This model separation is introduced to provide increased flexibility in modeling as well as to account for the multiple simultaneous estimates which are allowed within a multiple hypothesis framework. At any given time step, IMAN produces a *set* of state estimates and estimate error covariances for the vehicle and each feature. These estimates represent the possibilities depending on which of the proposed hypotheses are ultimately accepted as valid.

Figure 1 summarizes the IMAN algorithm, which consists of the following steps [10]:

1. **state projection**: possible vehicle and feature states from the previous cycle are projected using the individual vehicle and feature models;
2. **match hypothesis formation**: measurements are compared to existing feature models to determine possible data association matches;
3. **additional hypothesis formation**: additional hypotheses concerning measurement source, existing feature disposition, and new feature identification are formed in cases where data association is ambiguous;
4. **feature track updating**: feature track trees are updated to reflect the feature disposition hypotheses which have been formed;
5. **vehicle track updating**: the vehicle track tree is updated to reflect the possible navigational events for the current cycle; and
6. **pruning**: the vehicle and feature track trees are pruned to remove estimates reflecting unlikely events and to enforce consistency.

State projection for IMAN occurs analogously to SM. Each updated state is projected according to the model for that track (the vehicle or one of the features). The essential difference is that each proposed track will in general have multiple updated and, therefore, projected state estimates.

When measurements are taken, hypotheses are formed to explain the underlying situation. A hypothesis, in general, is the association of a feature track and a measurement, asserting that a given measurement arises from a specific proposed feature. This is somewhat in contrast to the concept of target-to-track assignments in multiple target tracking [7]. In that case,

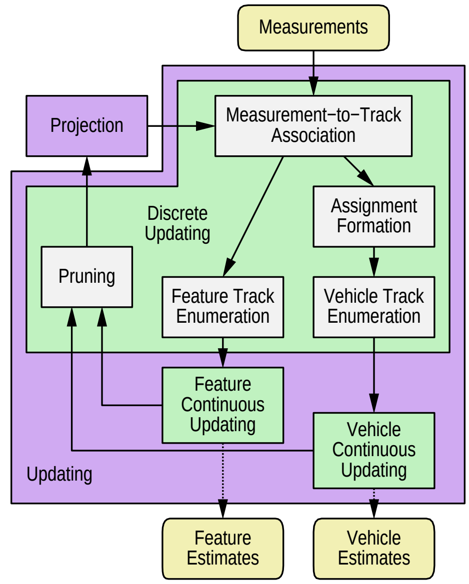


Figure 1: Process flow in IMAN.

vehicle navigation uncertainty is not taken into account and multiple feature and/or measurement models are not permitted, leading to the conjecture that all target-to-track assignments are equi-probable. These assumptions are not made by IMAN, making target-to-track assignment inseparable from data-to-data assignment, and, in fact, central to the project of track estimation. The hypotheses formed represent the logical possibilities of the sensor physics based on the vehicle's current understanding of its context.

In the development of multiple hypothesis tracking, it was observed that some data association decisions did not affect each other at all [9]. If the full set of possibilities was considered, many duplicate estimates would result from these orthogonal decisions. To address this, methods for clustering dependent decisions (and more importantly, separating independent decisions) were developed [6]. When vehicle navigational uncertainty is introduced, uncertainty about what has happened to a feature is spread to the vehicle track during match hypothesis formation. Subsequently, this uncertainty is passed to other feature tracks as well. Thus, even features in separate 'clusters' quickly come to depend on decisions about each other's feature disposition. Clustering is no longer useful in maintaining sets of independent decisions. However, a full enumeration of the possibilities still contains many duplicates. Each branching in a feature track tree can be thought of as a decision to be made at a later time. While multiple branches still exist, the decision is active; once only one possibility remains, the decision is resolved. We want to make sure that (1) there will only be one state left for each track tree once all decisions are resolved, and (2) all states that are tracked are *possible* given some decision calculus. Each state maintains a set of the active decisions it depends on and the necessary resolution required for that state to hold. States are compatible if there are no active decisions upon which they both depend for which they require different hypotheses (decision resolutions) [12]. Checking state compatibilities serves to minimize the number of state combinations considered and, to the degree possible, separate orthogonal decisions.

Many data association decisions can be made unambiguously. This accounts for the decent performance of SM with nearest neighbor data association resolution in a wide variety of circumstances. The goal of data association resolution in IMAN (steps 2 and 3) is to reduce to this simple, but effective, strategy when it works, but to recognize ambiguous situations and to adapt to them as they occur. This approach leads to a two-step data association resolution strategy. First, match hypotheses are formed. Each measurement is compared with a predicted measurement based on combinations of possible projected vehicle states and compatible possible projected feature states for existing feature tracks. If the Mahalanobis distance of the innovation falls within a specified threshold, then a match hypothesis is formed associating the measurement with that feature track. In other words, a match is hypothesized if the actual innovation falls within a specified highest density region (HDR) of the estimated innovation process. The gating threshold is typically characterized by the number of standard deviations in the HDR using a Gaussian model. This approach is typical for existing multiple hypothesis algorithms [3]. For the second stage of data association resolution, additional hypotheses are generated for those cases where data association remains ambiguous. An unambiguous association is assumed to occur if a measurement matches only one track and that track is matched only by that measurement. Ambiguous cases occur, for example, when more than one measurement gates with a track or when a measurement gates with multiple tracks. In these ambiguous cases, additional hypotheses are formed. For each feature track involved, an additional hypothesis is made to account for the possibility that that feature was missed, or did not produce a measurement. For each measurement involved, the additional possibilities that it represents a new feature track and that it did not arise from any feature (i.e., it is spurious) are hypothesized.

Feature track updating can occur in conjunction with hypothesis formation. Each feature disposition hypothesis produces zero or one updates to each possible projected feature estimate. Miss hypotheses are used to update each projected feature estimate. For each combination of projected vehicle state, projected feature state, and measurement, an update will be generated if the states have compatible dependencies and the measurement gates with the vehicle and feature state estimates. These match updates occur in the typical Kalman sense, but cross-model correlations (those between the feature model and the vehicle model) are ignored at present to reduce the overall complexity of the algorithm. See Uhlmann et al.[15] for a discussion of the impact of correlations in CML estimation.

While each feature is restricted to one of a set of hypotheses about its disposition, the vehicle takes part in all of the measurements in a given time step. Because of this, vehicle track updating cannot take place on the basis of individual hypotheses, but rather must be based on sets of hypotheses spanning the decision space of the tracks and measurements for that time step. An assignment is an exhaustive, consistent set of hypotheses spanning this decision space. Each assignment will explain the disposition of each proposed feature and the origin of each measurement in a consistent manner. The first step in updating the vehicle track tree is the formation of all such possible assignments from the feature gating and measurement origin decisions. All assignments must be considered (in contrast to traditional multiple hypothesis tracking [3]), because (1) target-to-track associations are not equiprobable for IMAN and (2) the assignment likelihood calculation process requires consideration of all possible assignments. The efficient generation of possible assignments from a set of decisions has been addressed in some depth [3].

Vehicle track tree updates originate from either the most recent set of projected possible states or the current set of updated possible states (just in case some other form of measurement, such as INS, has already been processed during the current time cycle). Each of these originating states is given a child state corresponding to each of the possible assignments. For each match hypothesis in an assignment, child states are produced by considering possible matches with projected possible states from the specified feature track. These updates are accomplished in the usual Kalman sense.

During vehicle update tree growth, state likelihood is calculated conditioned on the decision to accept the assignment root state, but not on the truth of that state. Forming the assignment root state results in a base likelihood derived from the number of features detected and missed and the number of measurements which are false alarms and real targets. Match hypothesis updates are multiplied by their gating probability (as in feature track updating). The leaves of the vehicle track then form an exhaustive, mutually exclusive set of possible events. The *a posteriori* probability of each assignment is then the normalized sum of the leaf states which depend on it. The second stage of vehicle track update tree likelihood calculation begins by setting the assignment root state likelihoods to the product of the parent state likelihood and the *a posteriori* assignment likelihood. The update portion of the tree is then renormalized so that the leaves of any branch sum to the probability of the root state for that branch.

The consideration of multiple hypotheses necessarily leads to rapid growth in the number of states to be considered. The basic multiple hypothesis tracking problem (when vehicle position is assumed to be known) grows as the exponential of the number of measurements received [2]. The IMAN algorithm would, unchecked, grow even more quickly. The viability of this algorithm depends on the timely resolution of delayed decisions. The rapid growth in algorithmic complexity means a fine line between increased explanatory power in decision-making and computational efficiency. Thus the efficient pruning of track trees to eliminate unlikely or unsupportable state estimates is important. Fortunately, a number of effective pruning strategies have been developed for multiple hypothesis tracking, including n-backstep, k-best, and threshold pruning [6, 3]. IMAN can, in large part, adopt these pruning methods directly, although the presence of a vehicle track tree can complicate implementation. A 1-backstep pruning based on assignments is performed. In other words, at one time step before the current cycle, the remaining possible assignments are ranked by total likelihood among current vehicle updated states. The most likely (or, in the case of a tie, several most likely) assignment is accepted, while the remaining assignments are rejected. Hypotheses from the set of decisions for the time cycle being pruned which are not included in the accepted hypotheses are rejected. Rejecting a hypothesis amounts to removing any state estimates form vehicle and feature tracks which depend on that hypothesis. After pruning, a recursive cleanup and consistency check is required to remove any additional hypotheses which lose all support due to removed state estimates. The consistency check assures that any unsupported dispositional hypotheses for feature tracks are rejected. Cleanup and consistency checking occur recursively until all inconsistent states have been removed.

Table 1: Simulation parameters for IMAN experimental results.

Parameter	Value
Depth measurement variance	25 cm^2
Speed measurement variance	0.25 $\frac{\mathrm{m}^2}{\mathrm{s}^2}$
Pitch measurement variance	0.26 deg^2
Yaw measurement variance	0.26 deg^2
Probability of detection	0.9
Probability of false alarm	0.3
Range measurement variance	0.5 m^2
Bearing measurement variance	2.6 deg^2
North coordinate process noise variance	0.5 m^2
East coordinate process noise variance	0.5 m^2
Depth process noise variance	0.0025 m^2
Speed process noise variance	0.01 $\frac{\mathrm{m}^2}{\mathrm{s}^2}$
Pitch process noise variance	2.6 deg^2
Yaw process noise variance	2.6 deg^2
North coordinate process noise variance	2 m^2
East coordinate process noise variance	2 m^2

4 Results

The IMAN algorithm described above has been completely implemented in C++. Comparisons have been performed in simulation with both dead reckoning and augmented versions of stochastic mapping. The simulation parameters are given in Table 1. Dead reckoning measurements (depth, speed, pitch, and yaw) are taken every second. Forward look sonar measurements are obtained each second from a ± 40 degree cone looking forward from the vehicle out to 300 meters range. Clutter is generated by assuming the number of spurious measurements obtained for each ping has a Poisson distribution over time. The angle and range values of the spurious measurements are uniformly distributed in the sonar measurement space. Localization is complicated by the presence of an unknown cross-current.

Dead reckoning is navigation using only the navigational system estimates of vehicle kinematics to estimate vehicle position. No external feature measure-

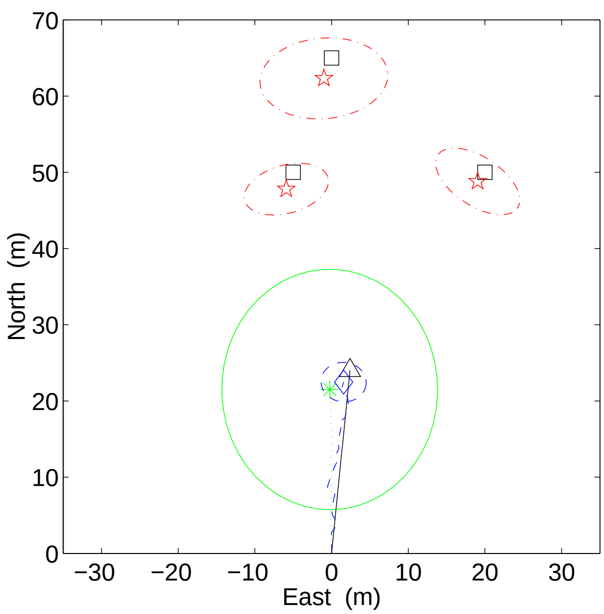


Figure 2: Comparison of IMAN and dead reckoning. The actual vehicle track, shown by a solid line and ending at a triangle, starts at (0,0). The vehicle is assumed to head directly north, but is pushed east by a cross-current. Three features, indicated by squares, are located at (50,-5), (50,20), and (65,0). The dead-reckoned vehicle path is shown by a dotted line, with the final position estimate marked with an asterisk. The three-sigma highest density region (HDR) is shown with a solid line. The IMAN estimated vehicle path is shown by a dashed line. The final estimate is indicated by a diamond, with the three-sigma HDR also shown by a dashed line. The final IMAN feature estimates are marked by stars. Their three-sigma HDRs are shown by dashed-dotted lines. Clutter is added to the data; 27% of the returns received are spurious. The IMAN algorithm is able to estimate vehicle drift with the aid of sonar measurements. IMAN also provides a more accurate error estimate. Accuracy in calculating the estimate error is important when making decisions based on the quality of the vehicle navigation estimate.

ments are incorporated. Dead reckoning measurement updates are performed in the same way (and using the same models) as for IMAN and the other algorithms. Because of the simulated current, dead reckoning performs consistently worse than the other algorithms. Also, since relative measurements are not incorporated, there is little difference in dead reckoning performance over the scenarios tested. A representative comparison of IMAN and dead reckoning is shown in Figure 2. The dead reckoning algorithm fails to recover the drift of the vehicle in the cross-current. While the dead-reckoned estimate is robust to this drift, this robustness comes at the price of extremely uncertain estimates.

To illustrate the performance of the implementation, we show the results of a series of Monte Carlo simulation runs with two features spaced at varying distances apart from one another. Three performance metrics provide the primary characterization of algorithm performance: completion rate, global error, and relative error. Completion rate is the percentage of simulations which run to completion. During some runs, the decision process becomes too complex for the integrated mapping and navigation algorithm. A complexity fault is a failure to complete a simulation due to decision-making complexity. When a complexity fault occurs, the number of state estimates being considered rapidly increases until available memory is exhausted. The completion rate is a measure of the robustness of the algorithm to the complexity of a given scenario. In general, complexity increases as ambiguous situations become more common.

Global error is the mean-square error of the mapping and navigation portions of the estimation process (errors in the navigational system are not included). The relative error represents the success of the algorithm at estimating the relative positions of the vehicle and features. If the navigational objects are considered pairwise, the components of the relative error are the squared differences between the estimated and actual vectors connecting each pair, as shown in Figure 3.

In order to provide a benchmark for IMAN performance, additional navigational algorithms are also considered in these performance analysis simulations. While these algorithms do not capture the complexity of the hybrid estimation problem to the extent that IMAN does, they provide a sanity check regarding the validity of IMAN and the effects of taking this more complex approach to concurrent mapping and localization.

Stochastic mapping is unable to handle data asso-

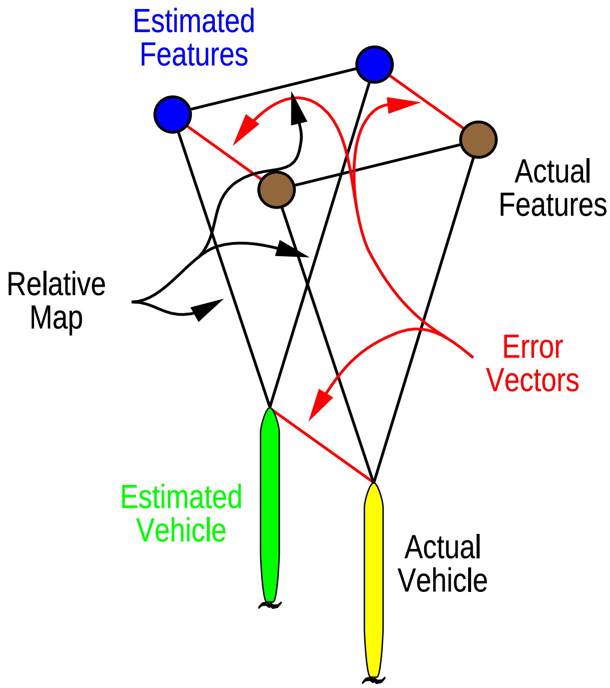


Figure 3: Global and relative error metrics.

ciation uncertainty due to the assumption that feature matching is provided along with measurements. This is particularly relevant when considering clutter, as no ready basis exists for explaining such a phenomenon within traditional stochastic mapping. An augmented stochastic mapping algorithm that is capable of operation in clutter is used as an alternative algorithm for comparison with IMAN.

The performance of IMAN is compared against two versions of the augmented stochastic mapping [5]. Data association is resolved using a nearest-neighbor algorithm [1]. Thus discrete decisions are completed instantaneously from the point of view of the continuous estimation problem; delayed decision-making is not possible. This separable approach to hybrid estimation is similar to, but less robust than, the probabilistic data association filter developed by Bar-Shalom [1, 7]. The first augmented stochastic mapping algorithm, ASM1, provides the complete estimate, including a full account of the cross-model correlations, which are ignored by IMAN. The second version, ASM2, drops the cross-model correlations. The result is a block diagonal covariance matrix for the state estimates.

Two features, at $(75, \pm\frac{\rho}{2})$, are used; the feature separation ρ is varied between 10 and 40 meters. Clutter is also added, with an expected 53% of measurements

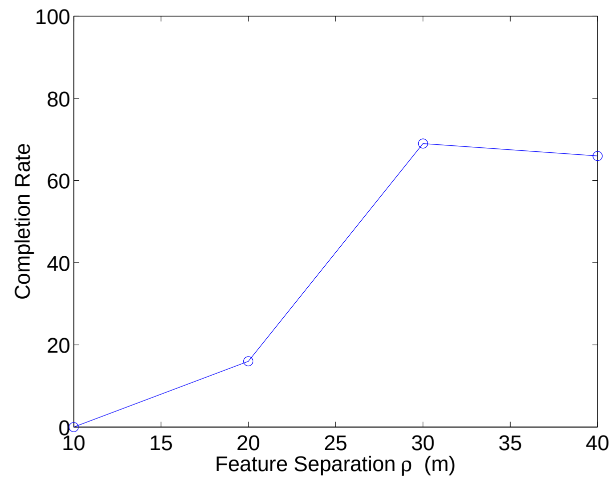
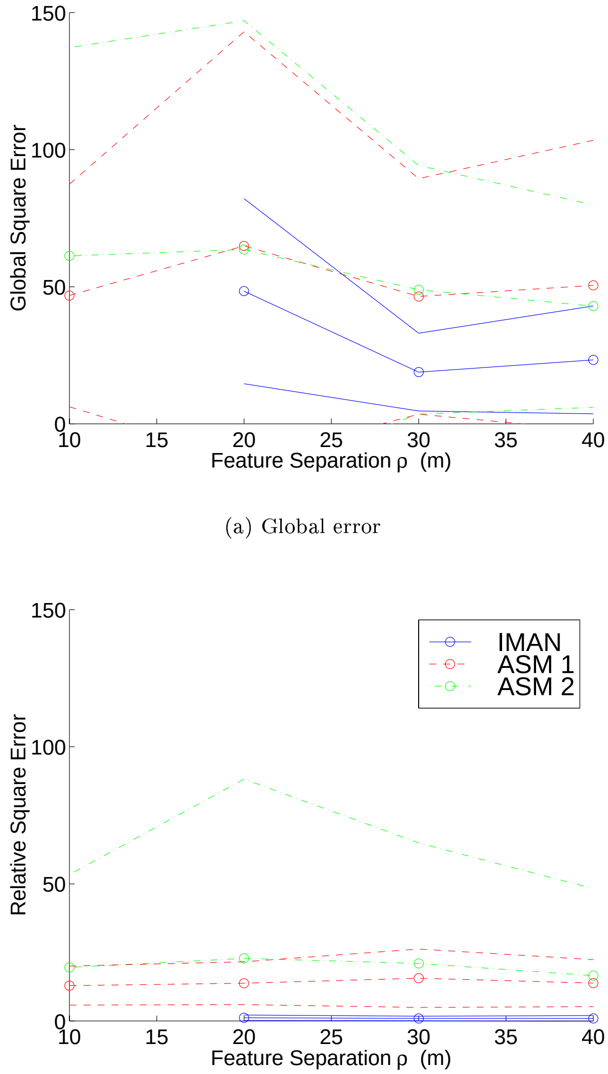


Figure 4: The effect of feature separation on completion rate. Track interaction remains a research issue for IMAN and is the largest factor in precluding a robust implementation. When features are 10 meters apart, the IMAN algorithm cannot disambiguate the tracks. As feature separation increases, IMAN fares better. These scenarios included an expected 53% of measurements being spurious. The baseline performance for this clutter density is reached with a feature separation of 30 meters, indicating that at this distance, features are essentially distinct to the algorithm.

being spurious. The completion rates for IMAN in these four scenarios are shown in Figure 4. IMAN had significant difficulties overcoming the complexity of reasoning about feature separation. A completion rate of 0% is obtained for feature separation of 10 meters. As ρ is increased to 40 meters, a 75% completion rate is obtained. The difficulties of separating feature tracks are complicated by the fact that the initial ranges are identical and clutter is present. The significant failures of IMAN in this case underscore the need for improved robustness, for example through the implementation of default hypothesis and recovery from complexity faults.

Figure 5 shows the global and relative error metrics for the feature separation scenarios. When complexity faults do not cause failure, IMAN demonstrates considerable performance gains in both global and relative mapping. Delayed decision-making and the consideration of discrete events allows better estimation choices to be made.



(a) Global error

(b) Relative error

Figure 5: The effect of feature separation on mapping and navigation performance. While IMAN is unable to complete highly ambiguous measurement sets, its performance for completed sets shows substantial gains. IMAN outperforms the augmented stochastic mapping algorithms in both global and relative mapping.

5 Technical Needs

This implementation of IMAN serves to illustrate some of the issues of incorporating a greater degree of decision-making into the navigation task. There remain several technological needs to be addressed before a navigation solution focusing on features and events can be practically implemented. These needs include four major questions: feature ontology specification, map representation, multiscale modeling, and the handling of cross-model correlations.

Feature ontology is the specification of a set of feature classes about which the vehicle can reason, that is, the kinds of features for which the vehicle has models. Humans often have a highly interpreted view of the world. Minute details are merged into a higher-level decomposition of the sensoria; for example, one can easily pick composed objects like people, cars, buildings, and trees from video imagery, even when there is significant occlusion and ambiguity. Feature extraction for robotic platforms focuses on much more primitive elements, such as edges and intersections. It is at this primitive element level that feature extraction can be made with any degree of reliability and robustness. In order to reason about more abstract or composed features, a hierarchical ontology is necessary. Increasingly complex feature models are used to build up knowledge of more complex feature instances from the primitive elements that can be reliably extracted from the sensor stream. The appropriate development of such an ontological hierarchy, while key to the advancement of feature-based navigation methods, has received little attention.

The issue of map representation has been clouded by the variety of conceptions of what constitutes a map. Of particular interest here are two of these conceptions. First, consider the project of providing some absolute global positioning information about the vehicle and objects being mapped. This capability may indeed be necessary for a variety of navigation tasks. However, there are additional, and sometimes more convenient, representational possibilities. A natural choice for feature-based navigation is a relative navigation strategy, in which the relationships between objects are constrained by the measurements taken. The problem arises when one attempts to gain the benefits of both these methods. Errors in the absolute position of objects can hide the fact that their relative positions are known quite accurately. Conversely, focusing on local relationships can lead to over-confidence in the absolute locations of the objects being tracked. Existing navigation techniques overwhelmingly choose one representational scheme and stick with it, losing the benefits of alternative representations. There does not exist an efficient method of combining these alternative representations.

As the experiments with IMAN have shown, the resolution of the navigational events being considered has a great impact on the efficiency of multiple-hypothesis methods. When too fine-grained a decomposition is adopted, the complexities inherent in multiple-hypothesis methods limit the practicality of the technique. Too course-grained a decomposition, on the other hand, can lead to discrete modeling errors as incorrect *a priori* decisions are made. To complicate matters, the choice of appropriate level of event decomposition is closely interwoven with the questions of feature ontology and specifics of the environmental context. These complications suggest a multiscale modeling approach. In such an approach, feature dynamic models and behavioral or event models can be represented at their own most appropriate time scales. The problem arises in devising an appropriate filtering mechanism when these disparate time-scale models are used together to describe a system. The problem is exasperated by the presence of both continuous and discrete elements in a hybrid estimation system.

Finally, the issue of cross-model correlations must be addressed, for both stochastic-mapping and concurrent-mapping-and-localization solutions. Cross-model correlations are an important factor in maintaining realistic estimates of navigation uncertainty, whether or not multiple hypotheses are maintained regarding navigation events. However, retaining this information is extremely expensive in terms of both storage and computational requirements, particularly when multiple hypotheses are entertained. The role of cross-model correlations with regard to map representation and multiscale modeling remains unclear. The desire is to limit the amount of correlation information to be stored and processed while maintaining enough robustness to prevent overconfidence in filter estimates (or even divergence).

In summary, the role of navigation is expanding from a simple question of localization to the broader task of context determination and situational assessment due to factors such as increased autonomy, effective reduction in prior knowledge, and extended mission durations. The result of this is an emerging need for increased focus on feature-based navigation solutions that offer reasoning about navigational events. The IMAN algorithm is an initial implementation of such a solution and has achieved limited success in improving navigational decision-making through a multiple-hypothesis approach to the problem of data association. Despite such advances, a number of technological needs remain and will require substantial effort to resolve. These technological needs include feature ontology, map representation, multiscale modeling, and the handling of cross-model correlations.

Acknowledgment

This research has been funded in part by the Naval Undersea Warfare Center, Newport, RI, U.S.A. H.J.S.F. acknowledges the support of NFR (Norwegian Research Council) through grant 109338/410. J.J.L. acknowledges the support of the Henry L. and Grace Doherty Assistant Professorship in Ocean Utilization and NSF Career Award BES-9733040. The authors thank C. Shaw and R. N. Carpenter for many helpful discussions.

References

[1] Y. Bar-Shalom and T. E. Fortmann. *Tracking and Data Association*. Academic Press, 1988.

[2] K. Chang, S. Mori, and C. Chong. Performance evaluation of a multiple hypothesis multitarget tracking algorithm. In *IEEE Int. Conference on Decision and Control (CDC)*, pages 2258–2263, 1990.

[3] I. J. Cox and J. J. Leonard. Modeling a dynamic environment using a Bayesian multiple hypothesis approach. *Artificial Intelligence*, 66(2):311–344, April 1994.

[4] I. J. Cox and G. T. Wilfong. *Autonomous Robot Vehicles*. Springer-Verlag, 1990.

[5] H. J. S. Feder, J. J. Leonard, and C. M. Smith. Adaptive mapping and navigation using sonar. Technical Report 98-3, Marine Robotics Laboratory, Massachusetts Institute of Technology, 1998. Available from http://oe.mit.edu/~jleonard/pubs/tech_report983.pdf.

[6] T. Kurien. Issues in the design of practical multitarget tracking algorithms. In Y. Bar-Shalom, editor, *Multitarget-Multisensor Tracking: Advanced Applications*, pages 43–83. Boston: Artech House, 1990.

[7] S. Mori, C. Chong, E. Tse, and R. Wishner. Tracking and classifying multiple targets without a priori identification. *IEEE Trans. on Automatic Control*, AC-31(5), May 1986.

[8] P. Moutarlier and R. Chatila. An experimental system for incremental environment modeling by an autonomous mobile robot. In *1st International Symposium on Experimental Robotics*, Montreal, June 1989.

[9] D. B. Reid. An algorithm for tracking multiple targets. *IEEE Trans. on Automatic Control*, AC-24(6), Dec. 1979.

[10] C. M. Smith. *Integrating Mapping and Navigation*. PhD thesis, Massachusetts Institute of Technology, 1998.

[11] C. M. Smith and J. J. Leonard. A multiple hypothesis approach to concurrent mapping and localization for autonomous underwater vehicles. In *International Conference on Field and Service Robotics*, Sydney, Australia, 1997.

[12] C. M. Smith, J. J. Leonard, A. A. Bennett, and C. Shaw. Feature-based concurrent mapping and localization for autonomous underwater vehicles. In *IEEE Oceans*, 1997.

[13] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In I. Cox and G. Wilfong, editors, *Autonomous Robot Vehicles*. Springer-Verlag, 1990.

[14] W. K. Stewart. *Multisensor Modeling Underwater with Uncertain Information*. PhD thesis, Massachusetts Institute of Technology, 1988.

[15] J. K. Uhlmann, S. J. Julier and M. Csorba. Nondivergent Simultaneous Map Building and Localisation using Covariance Intersection. Navigation and Control Technologies for Unmanned Systems II, 1997.