

Large Scale Sonarray Mapping using Multiple Connected Local Maps

Kok Seng CHONG Lindsay KLEEMAN

kok.seng.chong@eng.monash.edu.au lindsay.kleeman@eng.monash.edu.au

Intelligent Robotics Research Centre (IRRC)

Department of Electrical and Computer Systems Engineering

Monash University, Clayton Victoria 3168, Australia

<http://calvin.eng.monash.edu.au/IRRC/index.html>

Abstract

This paper presents a strategy for achieving practical mapping navigation using a wheels driven robot equipped with a sonarray (an advanced sonar array). The original mapping navigation experiment, carried out with the same robot configuration, builds a feature map consisting of commonplace indoor landmarks crucial for localisation, namely planes, corners and edges. The map exhaustively maintains covariance matrices among all features, thus presents a time and memory impediment to practical navigation in large environments. The new local mapping strategy proposed here breaks down a large environment into a topology of local regions, only maintaining the covariance among features in the same local region, and the covariance among local maps. This notion of two hierarchy representation drastically improves the memory and processing time requirements of the original global approach, while preserving the statistical details necessary for an accurate map and prolonged navigation. The new local mapping scheme also extends the endeavour towards reducing error accumulation made in the global mapping strategy by eliminating totally error accumulated between visits to the same part of an environment. This is achieved with a map matching strategy developed exclusively for the advanced sonar sensor employed. The local mapping strategy has been tested in large, real life indoor environments and successful results are reported here.

1 Introduction

This work is motivated by the need to extend the Julier Uhlmann Kalman Filter based mapping strategy developed by the authors [6] to large environments. The original strategy treats the robot's configuration and the environmental features as evolving **states**, and exhaustively maintains **covariance** matrices among them. It works well in a reasonably sized environment where the number of landmarks to be mapped are limited. However, to map a very large environment, the strategy is challenged by the need to store and process an ever expanding list of features. This also implies gradual degradation of processing **speed** and increased demand for computer random access **memory** (RAM). For example, when a map has n features and a new measurement arrives, there are n features to test out for the possibility of fusion before calling it a new feature. If fusion is possible, the robot's position and the state of all n existing map features, and as many as $(n+1)(n+2)/2$ covariance matrices must be updated. Even if fusion does not occur, to integrate it into the list of existing map features the new measurement must create a correlation matrix with every existing map feature. The requirement for memory grows quadratically and without bound. For this, [13] attempts to reduce the size of covariance matrix by declaring features with a small covariance as "confirmed targets", thereby deleting its cross-covariance with the robot's position and other features.

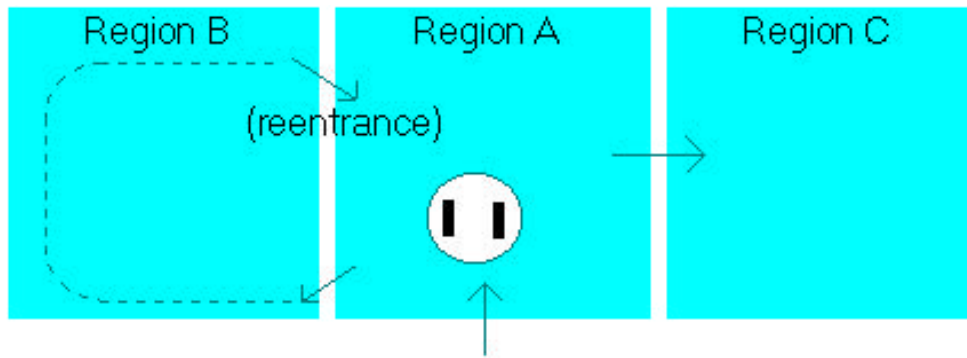


Figure 1: Robot exploring a large environment consisting of several regions

The original mapping strategy also cannot cope with the problem of **error accumulation** with complete success. It is well known that linearisation of non-linear equations introduces bias [2,10,13]. Bias also comes from the physical systems themselves such as the imperfect calibration of wheels and the placement offset of the sonar sensor from the true centre of the robot. In addition, noise characteristics such as that of odometry, speed of sound and target bearing are fairly hard to come by accurately. All these factors make it very difficult to continually produce a consistent map. The reader is referred to the example in Figure 1. The robot has just mapped region A, and its path planning strategy leads it to explore region B first and then region C by traversing region A again. Theoretically speaking, the robot would localise with the features in region A to recover from the error accumulated in region B. In practice, if region B is large, the error accumulated will be large, so the robot might not be able to recognise that it is re-entering region A if it relies on Kalman Filter only. It then proceeds to build a conflicting map on top of the original map of region A. By the time region C has been mapped, its actual spatial positioning relative to region A is grossly inaccurate. The example just given illustrates a classical mapping problem - if a robot returns to its starting position after making a long distance navigation, can the robot identify it?

This work intends to tackle the above problems using a local mapping strategy. With this strategy, the environment is represented as a set of interconnected local maps. A new local map is created as usual until the covariance of position becomes large. That occurs when the robot has travelled a reasonably long distance since it first created its current local map. The large position covariance is a good indication that the features in the current local maps are not highly correlated with the features in the new local map. By not maintaining the covariance matrices among features from two separated local maps, the strategy gains the memory and processing advantages. The authors also attempt to alleviate the inevitable problem of error accumulation by map matching [4]. Refer to Figure 1 again, the authors' local mapping scheme attempts to recognise region A when the robot reenters it by scrutinising the error in the robot's position relative to region A. After affirming that the possibility of reentrance is high, it performs a **map match** with its new measurements on the local map of region A. If map matching is successful, the latest position of the robot will be accurate with respect to the local map of region A. The local map of region C generated will then be independent of the position error accumulated in the process of mapping region B.

Local mapping is not a new concept. The work in [1] resorts to a graph of local feature maps that register the three dimensional information of the edges on the floor obtained from a camera. A new local map is created when none of the features in the current local map is visible from the next position. The arcs between local maps contain the robot's displacement information. To facilitate fast local mapping with sonar sensing, [9] stores a local map as a list of pre-processed sonar reflection points. Dynamic environments are effortlessly accommodated by deleting conflicting points of reflections. However, none of the above approaches exploit reobservation of features to enhance the accuracy of the maintained maps, and are preliminary in the sense that they have not been tested in the difficult task of exploring large environments. Local mapping is not confined to feature based environment models. The work proposed in [3], maintains a graph of local Bayesian distribution grid maps in its multi-sensory navigation scheme. Another version of the above approach is LOGnets [14], which adopts local occupancy grid with a special type of concentric, unevenly sized cells. Map update is a simple scheme of incrementing a cell frequency count if the returned sonar reading falls within the cells. In both grid based approaches, the methods for determining if an old local map has been revisited are purely heuristical. Other environmental models include principal eigenvector components [16].

The major contributions of this local mapping system are:

- With the advanced sonar sensor by [12], the robot can very quickly pick up the most salient natural landmarks useful for mapping and localisation.
- At any instant of navigation, the map, hence the processing demand, is very small. This gears mapping based navigation

towards a real time speed.

- The generation of local maps is done with careful mathematical criteria as in [6], so the speed advantage is not gained at the expense of map integrity.
- Through the process of map matching, the covariance of the robot's position and its correlation with all the features in the loaded local map can be correctly recovered. This facilitates further map building after re-entrance.
- The classical problem of 'going back to where it started' has been solved in two challenging large environments, where the collaborative strength of map building and map matching have been tested.

The paper is structured as follows. Section 2 describes the robot used in the experiment. Section 3 summarises the global mapping approach. Section 4 expatiates the mathematical details of the proposed local mapping strategy. Section 5 covers the map matching scheme. Experimental results are shown in section 6, followed by section 7 on conclusions and future work.

2 The Robot

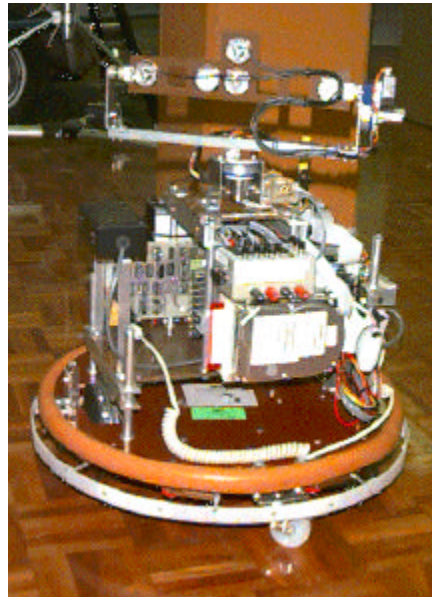


Figure 2: The mapping robot, Werrimbi, with a sonarray on the top

The description of various hardware modules of the mapping robot, Werrimbi (Figure 2), can be found in [5]. Werrimbi has a highly accurate wheel based odometry system. Exteroceptive sensing is performed with an advanced sonarray [12] which can classify reflectors into discrete planes, corners or edges, and accurately estimate their ranges, horizontal bearings and vertical bearings. The above modules are controlled by a 486DX2-66MHz board through a ISA AT Bus.

3 The Global Mapping Scheme

The discrete primitives generated by the sonarray are fused together using a Julier-Uhlmann Kalman Filter based mapping scheme [6]. The states of the filter are the robot's 2D coordinates, orientation, speed of sound, the line parameters of the planes and the 2D coordinates of the corners and edges. The extent of a plane is marked by the coordinates of its endpoints. Covariance blocks are maintained among these entities [6]. To produce a realistic representation of the environment, the collinearity and correspondence constraints among the features are exploited. For example, the filter is activated if a new corner measurement is found at the intersection of two planes. The extent of both planes are then adjusted accordingly.

4 The New Local Mapping Scheme

The fundamental rationale behind the new local mapping strategy is that if two features are very far apart, the update of one feature will not significantly enhance the estimation of another feature through their correlation. Hence there is no reason for the storage of their correlation matrix. Eliminating matrices of loose correlation not only saves memory, it also improves processing speed because there are less matrices to update.

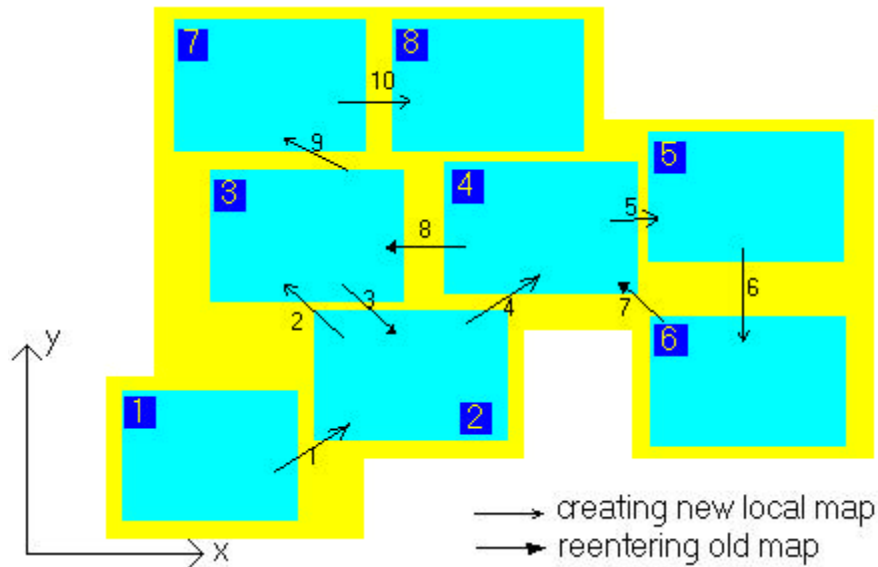


Figure 3 : Representation of a large environment as an interconnected set of local maps

With the new strategy, the map and measurements retained in the memory is exactly the same as the global mapping approach, except that they are much smaller in size because only a restricted set of objects surrounding the robot are present in the memory for processing at any one time. When the robot decides that there is a need for a new local map or suspects that an old local map has been revisited, the current local map is saved to hard disk. The memory is then cleared. If a new map is desired, the memory can be instantly put to use. If, on the other hand, an old map is to be revisited, it is retrieved from the hard disk. As a result only one small local map exists in the memory at one time, leaving more memory space for data processing. Throughout the navigation, the memory also constantly maintains, for each local map, a list of robot positions responsible for the formation of each of them. The purpose is to quickly test out if the robot has revisited an old local map before loading it into the memory and starts a rigorous map matching based validation. All positions are globally relative to each other. The reader can refer to Figure 3 for the two scenarios of creating a new local map and revisiting old local map as the robot maps a large environment. Adopting the terminology of graph theory, one can view each local map as a *node* of a graph, connected together by a set of *directed arcs* indicating the sequential relationship between two local maps.

The complete local mapping algorithm is shown as a flowchart in Figure 4. The robot executes its usual routine of sensing and mapping just like in the global approach until its position covariance signals it to switch to another local map. The criteria for the termination of the current local map is when the covariance of the robot's position exceeds a threshold, \mathbf{Pr}_{TH} . That is, when it is decided that the error in position might have become significant and the last few features in the current map are no longer significantly correlated with the initial few features. The robot then searches through the list of old positions to find out if it has revisited a local map. If one of the old positions is found to be sufficiently close to its current position, the corresponding local map is loaded from the disk and map matching is performed. The prior position information can be used to prune the map matching search space. The details of map matching are covered in section 5. If matching is successful, a revisit has occurred, so the robot carries on mapping with the loaded local map. Otherwise, another local map is tested until finally one, or none at all, is found. If the later is the case, a new local map is created for further mapping. Each local map is assigned an index to indicate the sequence of their generations.

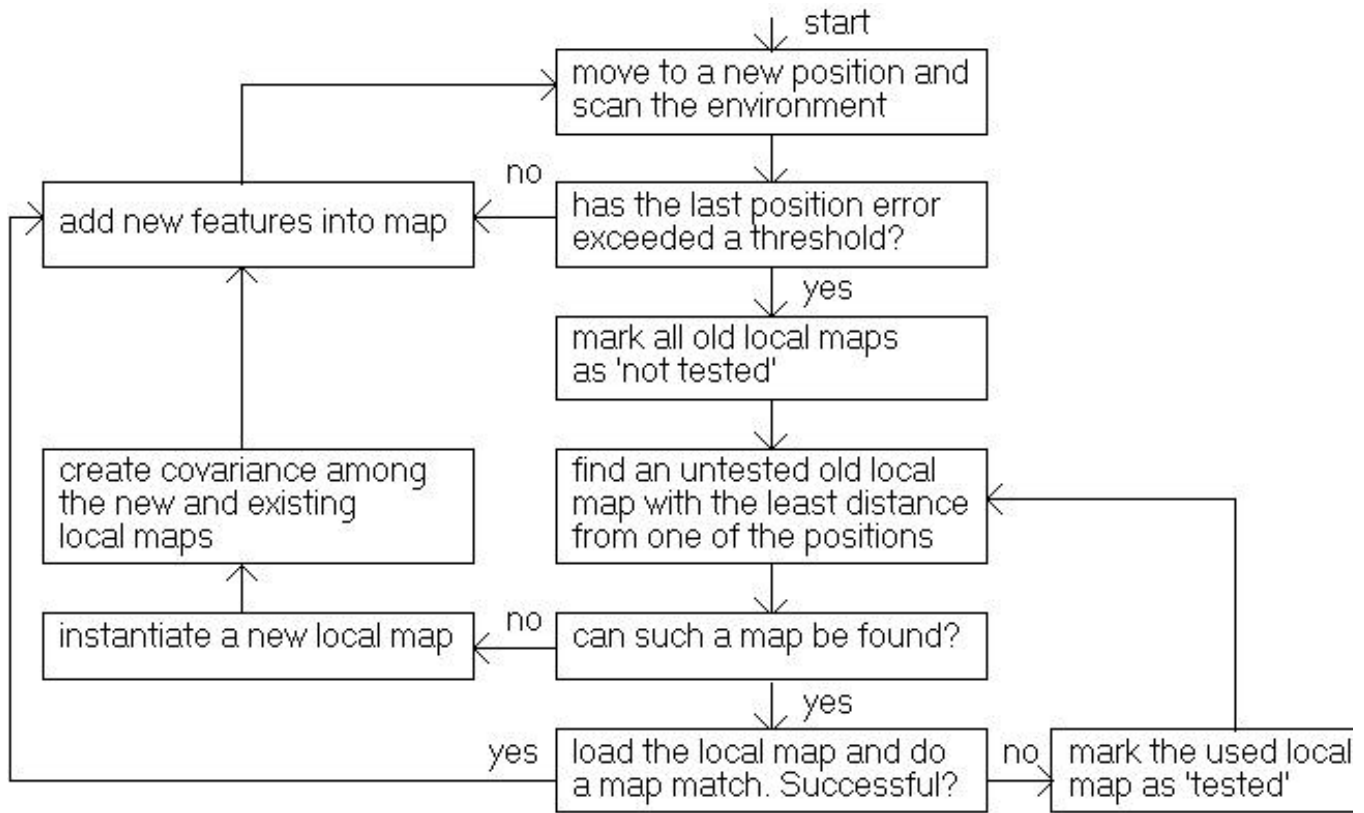


Figure 4 : Flowchart for the local mapping algorithm

The notion of map matching helps to alleviate the problem of error accumulation because the process of map matching almost completely discards the previous knowledge of the robot's position and orientation by re-estimating them with the loaded old map and its new measurements. Only the speed of sound estimation is carried over to the next local map. While the environment is still not perfectly mapped in a global sense, the robot can now practically remove all the error accumulated between visits to the same area. As shown later, this helps the robot tackle such challenging environments as loops and long corridors.

4.1 Relative Covariance Among Local Maps

When a new local map is instantiated, its **relative covariances** with other local maps are created. The relative covariance, \mathbf{Pr} , between two local maps is defined as the covariance between the first robot's positions at which the two local maps are first created. Relative covariance among local maps must be maintained in order to estimate the uncertainty of a robot's position in one local map relative to another local map. Such information is crucial for statistically determining if the robot is revisiting an old region (section 4.2), and for setting up a search bound for each measurement during map matching (section 4.3).

Referring to Figure 5, suppose that n_{lm} local maps have already existed. The covariance of the position at which a new local map (local map $n_{lm}+1$) is to be created is $\mathbf{Pr}_{n_{lm}+1,i}$ with respect to the current local map i , and the covariance of the current local map with respect to an old local map j ($j < i$) is $\mathbf{Pr}_{i,j}$, then the covariance of the new local map $n_{lm}+1$ with respect to the old local map j is

$$\mathbf{Pr}_{n_{lm}+1,j} = \mathbf{Pr}_{n_{lm}+1,i} + \mathbf{T}(d, \theta) \mathbf{Pr}_{i,j} \mathbf{T}^T(d, \theta) \quad (\mathbf{1})$$

where

$$\mathbf{T}(d, \theta) = \begin{bmatrix} 1 & 0 & -d \sin(\theta) \\ 0 & 1 & d \cos(\theta) \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

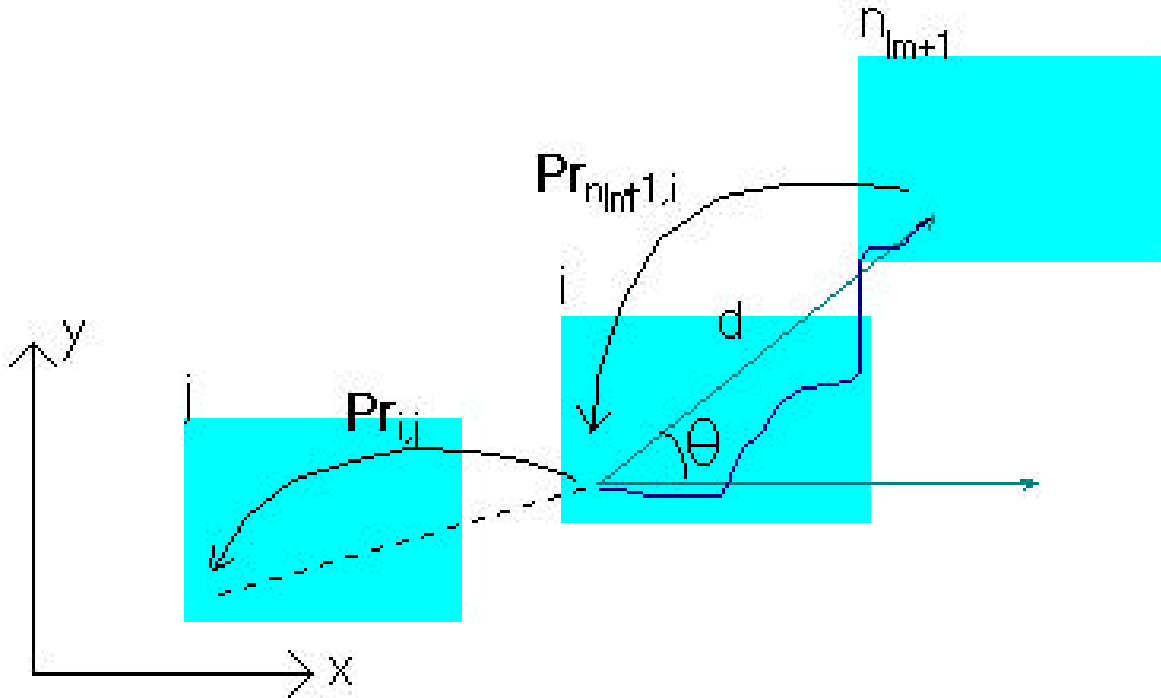


Figure 5 : Generating the relative covariance of the new local map with respect to an old local map

Equation (1) implicitly assumes that $\mathbf{Pr}_{i,j}$ and $\mathbf{Pr}_{n_{int}+1,j}$ are uncorrelated. Attention is drawn to the specification that only the relative covariance with the old local maps with indexes less than the index of the current local map (i.e. $j < i$) needs to be evaluated. Since local map indexes are assigned in the order of their formations, this applies to any two local maps which are **monotonously linked** together, that is, one can reach local map i from local map j by traversing (following the directed arcs) a set of local maps with monotonously increasing indexes. Refer to the example in Figure 3 again, the relative covariance between local map 3 and local map 4, $\mathbf{Pr}_{3,4}$ and $\mathbf{Pr}_{4,3}$ are not evaluated because one must pass through local map 2 to move from local map 3 to local map 4, and 2 is less than 3. Local map 2 is defined as the **common root** of local map 3 and local map 4. The common root of two monotonously linked local maps is the one with the smaller index. The reason for not evaluating the relative covariance among the non-monotonously linked local maps is that the uncertainty of any robot's position relative to map 4 can be easily computed with a simple equation involving $\mathbf{Pr}_{3,2}$ and $\mathbf{Pr}_{4,2}$, as demonstrated in the next section. It also has a secondary advantage of reducing storage space.

4.2 Reentrance of a Local Map

Supposed that the current local map has an index i , and the covariance of the latest robot's position, p_i , relative to the current local map, is $\mathbf{P}_{p_i,i}$. To check if an old map with index j is being revisited, the algorithm computes the distance between p_i and all robot's positions responsible for the generation of local map j . If the shortest among the computed distance, d_{min} , is less than a threshold d_{TH} , local map j is loaded from disk for a trial map matching (section 4.3). However, if d_{min} is not within the distance threshold, there is still a possibility that local map j is being revisited, if the uncertainty associated with this shortest distance is accounted for. We can begin by computing the covariance of p_i and the covariance of the position closest to it in local map j , p_j , with respect to their common root, k . With the covariance of the two positions, $\mathbf{P}_{p_i,k}$ and $\mathbf{P}_{p_j,k}$, the standard deviation of d_{min} can be subsequently determined.

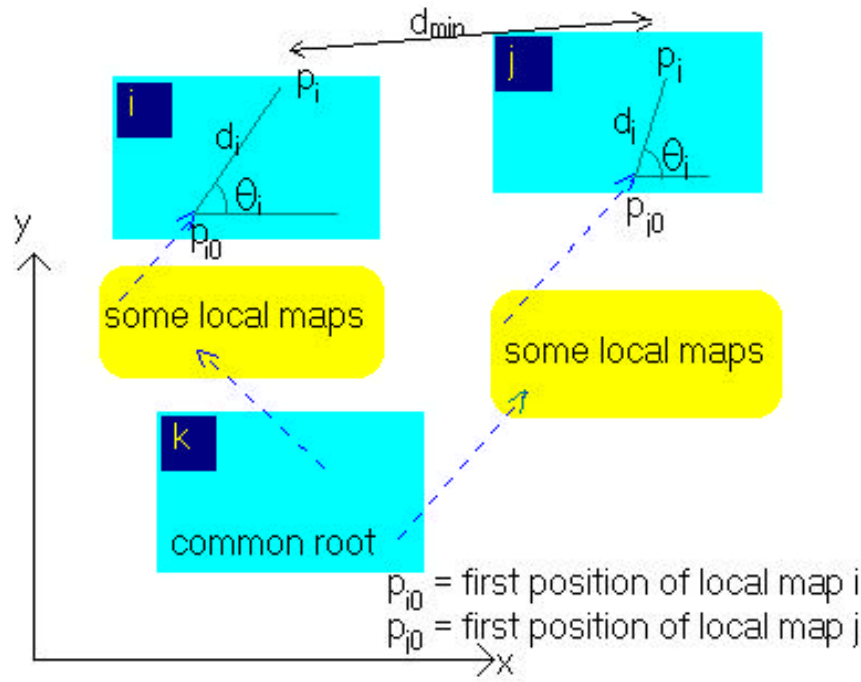


Figure 6 : Illustration of the computation of the standard deviation of d_{\min}

The final problem is that the covariance of p_j relative to local map j is not stored in order to save memory. Nevertheless, one knows that it is bounded by \mathbf{Pr}_{TH} , the threshold which triggers a new map. Here \mathbf{Pr}_{TH} is regarded as a worst case approximation to $\mathbf{P}_{p_j, j}$.

By referring to Figure 6,

$$\mathbf{P}_{p_i, k} = \mathbf{P}_{p_i, i} + \begin{cases} \mathbf{0} & \text{if } i = k \\ \mathbf{T}(d_i, \theta_i) \mathbf{Pr}_{i, k} \mathbf{T}^T(d_i, \theta_i) & \text{if } i, j > k \end{cases} \quad (3)$$

$$\mathbf{P}_{p_j, k} \leq \mathbf{Pr}_{TH} + \begin{cases} \mathbf{0} & \text{if } j = k \\ \mathbf{T}(d_j, \theta_j) \mathbf{Pr}_{j, k} \mathbf{T}^T(d_j, \theta_j) & \text{if } i, j > k \end{cases} \quad (4)$$

To compute the exact standard deviation of d_{\min} , the correlation between p_i and p_j with respect to local map k is required. Since such exact value is unimportant here, computational load can be reduced by approximating the standard deviation of the distance between p_i and p_j by

$$\delta_{d_{\min}} \approx \sqrt{\nabla_{p_i} d_{\min} \mathbf{P}_{p_i, k} \nabla_{p_i} d_{\min}^T + \nabla_{p_j} d_{\min} \mathbf{P}_{p_j, k} \nabla_{p_j} d_{\min}^T} \quad (5)$$

Local map j is declared **not revisited** at p_i if

$$d_{\min} > d_{TH} + 3 \delta_{d_{\min}} \quad (6)$$

4.3 Search Bound for a New Measurement

Map matching follows the loading of local map j . As will be explained in section 5.1, the map matching involves the search for the

correspondence of each new measurements in the local map. The search space can be pruned by exploiting the approximate knowledge of the robot's position provided by local map i . More specifically, the **search bound** of a new measurement should be set directly proportional to the accuracy of p_i relative to local map j .

Carrying forward the symbols defined in section 4.2, let the covariance of p_i relative to local map j be $\mathbf{P}_{p_i,j}$. If local map i and local map j are monotonously linked,

$$\mathbf{P}_{p_i,j} = \mathbf{P}_{p_i,i} + \mathbf{T}(d_i, \theta_i) \mathbf{P}_{r_{i,j}} \mathbf{T}^T(d_i, \theta_i) \quad \text{if } j < i \quad (7)$$

$$\mathbf{P}_{p_i,j} = \mathbf{P}_{p_i,i} + \mathbf{T}(d_j, \theta_j) \mathbf{P}_{r_{j,i}} \mathbf{T}^T(d_j, \theta_j) \quad \text{if } j > i \quad (8)$$

if local map i and local map j are not monotonously linked and has a common root k ,

$$\begin{aligned} \mathbf{P}_{p_i,j} = & \mathbf{P}_{p_i,i} + \mathbf{T}(d_j, \theta_j) \mathbf{P}_{r_{j,k}} \mathbf{T}^T(d_j, \theta_j) \\ & + \mathbf{T}(d_i, \theta_i) \mathbf{P}_{r_{i,k}} \mathbf{T}^T(d_i, \theta_i) \end{aligned} \quad (9)$$

for each measurement $\mathbf{m} = [r \ \rho]^T$ with covariance $\mathbf{Cov}(\mathbf{m})$ at p_i , which comprises the range and bearing to a classified reflector, relative to p_i . The Cartesian coordinates of the reflector are

$$\mathbf{x} = \begin{bmatrix} x_m & y_m \end{bmatrix}^T = \mathbf{H}(p_i, \mathbf{m}) \quad (10)$$

and its covariance relative to local map j is

$$\mathbf{Cov}(\mathbf{x}) = \nabla_{p_i} \mathbf{H} \mathbf{P}_{p_i,j} \nabla_{p_i}^T \mathbf{H}^T + \nabla_{\mathbf{m}} \mathbf{H} \mathbf{Cov}(\mathbf{m}) \nabla_{\mathbf{m}}^T \mathbf{H}^T \quad (11)$$

The search bound for \mathbf{m} is defined as a circle centred at \mathbf{x} , with a radius r_{BOUND} given by

$$r_{\text{BOUND}} = \beta \sqrt{\text{trace}(\mathbf{Cov}(\mathbf{x}))} \quad (12)$$

Since the position covariance is meant for establishing a search bound for a measurement, **not** map building, the value of β are not critical. If the bound is so large that it captures more than one map feature for a measurement, the map matching algorithm will eliminate the unsuitable ones with its measure of discrepancy function (section 5.2). In the current implementation, β is set to 4.

5 Map Matching

The map matching strategy localises a robot in a local map based on a combinatorial search scheme and a measure of discrepancy function. This best estimate of position is further enhanced with a least square formulation. Since the navigation mode is continuous, an approximate knowledge of the robot's position is available to guide the map matching process.

5.1 The Combinatorial Search for Positions

To calculate both the Cartesian coordinates and orientation $\mathbf{x}_0 = [x \ y \ \theta]^T$ of the robot, it is necessary to associate **two** of its new measurements to **two** of the map features. This can be achieved using an exhaustive matching scheme. That is, for every combination of two new measurements, a pair of features of the same types are extracted from the stored map (within the search bound of each measurement) and fix computation is attempted. Since there could be more than one association depending on the size of the search bound, a few position candidates might be obtained in this stage.

5.2 Generation of Sub-map

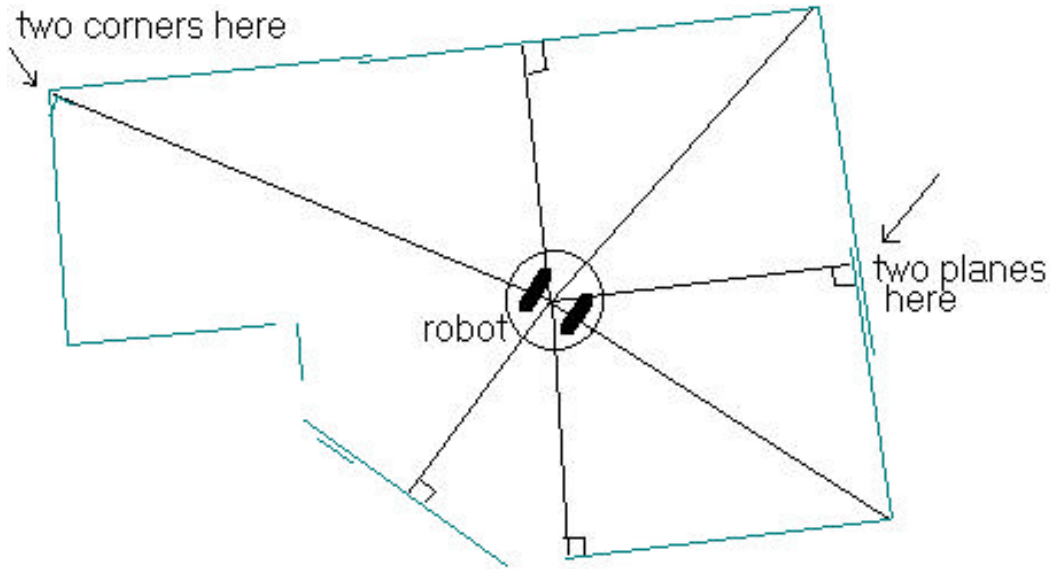


Figure 7: A typical imperfect stored local map and the visible features at the position candidate

Once a rough initial estimate of the robot's position has been established, a sub-map of the stored local map is then generated, which consists of all **visible** map features at the estimated position. They comprise the coordinates of the normal glints of planes, and the coordinates of both the standalone corners and the fused corners. All coordinates are Cartesian.

The stored map is not always perfect as it might include approximately overlapping planes and corners. To account for this, if the line of sight to a feature is blocked by a plane at a point very close to that feature, the feature is considered not blocked. In this case, the sub map will include conflicting features as illustrated in Figure 7. The matching cost function has been designed such that it is versatile enough to accommodate this problem.

5.3 Matching Cost Function - The Measure of Discrepancy

A **measure of discrepancy** has been designed to assess the quality of matching. It has been envisioned that the measure should take the role of a cost function which possess the following properties:

1. Every local feature must contribute a value to the total cost so that inadequate matching does not lead to a low cost.
1. In an imperfect map, as mentioned, overlapping features might present. This function should pick the one among them which gives rise to the least cost.
1. The unobserved map features should have no effect on the quality of matching. Only those features that match successfully to the local features should contribute to the total cost.

The considerations give rise to the following measure. Let n_m be the total number of measurements. $\mathbf{m}_i = [r_i \quad \rho_i]^T$ is the time of flight and direction of the i^{th} local measurement, relative to the estimate of the robot's state, $\bar{\mathbf{x}}_0$, where $i=1,2,\dots,n_m$. $\mathbf{f}_{\mathbf{x}_0}$ is a function which transforms the polar representation into Cartesian coordinates according to $\bar{\mathbf{x}}_0$. M is the set of all features in the sub map. For each \mathbf{m}_i , its contribution to the total is

$$C_i = \begin{cases} \min_{s \in M, \text{type}(s) = \text{type}(\mathbf{m}_i)} \|\mathbf{f}_{\mathbf{x}_0}(\mathbf{m}_i) - \mathbf{s}\|^2 & \text{if } < C_{TH} \\ C_{TH} & \text{otherwise} \end{cases} \quad (13)$$

where C_{TH} is a threshold. C_{TH} ensures that whilst the first property mentioned is satisfied, the cost contributed by an unmatched feature (indicated by a large square norm) will not overshadow the successful matching of other features. The **measure of discrepancy** D_m (units of metres) can be defined as

$$D_m = \sqrt{\sum_{i=1}^{n_m} C_i / n_m} \quad (14)$$

The chosen position candidate is the one which has the **minimum** D_m .

5.4 Least Square Formulation (LS)

Once the position candidate with the minimum discrepancy has been chosen, further enhancement can be sought by exploiting all matched feature pairs using a least square formulation. The strength of this formulation is that it makes immediate use of the position estimate previously generated and fully exploits the cross covariances among map features \mathbf{P}_{ij} and the covariances of the local measurements. The cross covariances among map features have been provided by Kalman Filter during the map generation stage [6].

Firstly, the linear least square formulation [17] is summarised. Suppose that a system has the following *observation equation*:

$$\mathbf{z} = \mathbf{H}\mathbf{x}_0 + \mathbf{n} \quad (15)$$

where \mathbf{z} is the observation, \mathbf{H} is the state transformation matrix, and \mathbf{n} is the noise with a covariance of $E(\mathbf{n}\mathbf{n}^T) = \mathbf{C}$. Under the notion of least square estimation, the best estimate of \mathbf{x}_0 is one which minimises the following weighted least square function

$$W = (\mathbf{z} - \mathbf{H}\mathbf{x}_0)^T \mathbf{C}^{-1} (\mathbf{z} - \mathbf{H}\mathbf{x}_0) \quad (16)$$

The solution to the above problem is well known [17]. The best estimate $\bar{\mathbf{x}}_0^{LS}$, and its covariance $\mathbf{C}_{\mathbf{x}_0}^{LS}$, in a least square sense, can be solved as

$$\hat{\mathbf{x}}_0^{LS} = (\mathbf{H}^T \mathbf{C}^{-1} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{C}^{-1} \mathbf{z} \quad (17)$$

$$\mathbf{C}_{\mathbf{x}_0}^{LS} = (\mathbf{H}^T \mathbf{C}^{-1} \mathbf{H})^{-1} \quad (18)$$

Before proceeding, the state of the robot \mathbf{x}_0 is expanded to include the speed of sound, $\mathbf{x}_0 = [x \quad y \quad \theta \quad c_s]^T$. Supposed that at the position with the least cost $\bar{\mathbf{x}}_0$, the matching strategy obtains a total of n_r matched measurement-feature pairs. Each matched pair forms a non-linear relationship \mathbf{G} with \mathbf{x}_0 , the measurement vector \mathbf{m}_i , and a set of map features $\{\mathbf{x}_{i_1}, \mathbf{x}_{i_2}, \dots\}$ which contribute to the corresponding sub map feature:

$$\mathbf{G}_i(\mathbf{x}_0, \{\mathbf{x}_{i_1}, \mathbf{x}_{i_2}, \dots\}, \mathbf{m}_i) = \mathbf{0} \quad i = 1, 2, \dots, n_r \quad (19)$$

To apply the least square equations, linear expansion about the mean of the function parameters is required. Let's assume, without a loss of generality, that two map features are involved. After linear expansion, equation (19) can be expressed as

$$\mathbf{z}_i = \nabla_{\mathbf{x}_0} \mathbf{G}_i \mathbf{x}_0 + \mathbf{n}_i \quad (20)$$

where

$$\mathbf{z}_i = -\mathbf{G}_i(\hat{\mathbf{x}}_0, \{\hat{\mathbf{x}}_{i_1}, \hat{\mathbf{x}}_{i_2}\}, \hat{\mathbf{m}}_i) + \nabla_{\mathbf{x}_0} \mathbf{G}_i \hat{\mathbf{x}}_0 \quad (21)$$

$$\begin{aligned} \mathbf{n}_i = & \nabla_{\mathbf{m}} \mathbf{G}_i(\mathbf{m}_i - \hat{\mathbf{m}}_i) + \nabla_{\mathbf{x}_{i_1}} \mathbf{G}_i(\mathbf{x}_{i_1} - \hat{\mathbf{x}}_{i_1}) \\ & + \nabla_{\mathbf{x}_{i_2}} \mathbf{G}_i(\mathbf{x}_{i_2} - \hat{\mathbf{x}}_{i_2}) \end{aligned} \quad (22)$$

By stacking all n_r of measurements together,

$$\mathbf{z} = [\mathbf{z}_1 \quad \mathbf{z}_2 \quad \cdots \quad \mathbf{z}_{n_r}]^T \quad (23)$$

$$\mathbf{H} = [\nabla_{\mathbf{x}_0} \mathbf{G}_1 \quad \nabla_{\mathbf{x}_0} \mathbf{G}_2 \quad \cdots \quad \nabla_{\mathbf{x}_0} \mathbf{G}_{n_r}]^T \quad (24)$$

$$\mathbf{C} = \begin{bmatrix} E\{\mathbf{n}_1 \mathbf{n}_1^T\} & \cdots & E\{\mathbf{n}_1 \mathbf{n}_{n_r}^T\} \\ \vdots & \ddots & \vdots \\ E\{\mathbf{n}_{n_r} \mathbf{n}_1^T\} & \cdots & E\{\mathbf{n}_{n_r} \mathbf{n}_{n_r}^T\} \end{bmatrix} \quad (25)$$

where

$$\begin{aligned} E\{\mathbf{n}_i \mathbf{n}_i^T\} = & \begin{bmatrix} \nabla_{\mathbf{x}_{i_1}} \mathbf{G}_i & \nabla_{\mathbf{x}_{i_2}} \mathbf{G}_i \end{bmatrix} \begin{bmatrix} \mathbf{P}_{i_1 i_1} & \mathbf{P}_{i_1 i_2} \\ \mathbf{P}_{i_2 i_1} & \mathbf{P}_{i_2 i_2} \end{bmatrix} \begin{bmatrix} \nabla_{\mathbf{x}_{i_1}} \mathbf{G}_i^T \\ \nabla_{\mathbf{x}_{i_2}} \mathbf{G}_i^T \end{bmatrix} \\ & + E\{(\mathbf{m}_i - \hat{\mathbf{m}}_i)(\mathbf{m}_i - \hat{\mathbf{m}}_i)^T\} \end{aligned} \quad (26)$$

and

$$\begin{aligned} E\{\mathbf{n}_i \mathbf{n}_j^T\} = & \begin{bmatrix} \nabla_{\mathbf{x}_{i_1}} \mathbf{G}_i & \nabla_{\mathbf{x}_{i_2}} \mathbf{G}_i \end{bmatrix} \begin{bmatrix} \mathbf{P}_{i_1 j_1} & \mathbf{P}_{i_1 j_2} \\ \mathbf{P}_{i_2 j_1} & \mathbf{P}_{i_2 j_2} \end{bmatrix} \begin{bmatrix} \nabla_{\mathbf{x}_{j_1}} \mathbf{G}_j^T \\ \nabla_{\mathbf{x}_{j_2}} \mathbf{G}_j^T \end{bmatrix} \\ & \quad i \neq j \end{aligned} \quad (27)$$

The least square estimate of the robot's position, orientation and the speed of sound can then be obtained using equation (17) and their covariance, \mathbf{P}_{00} , can be obtained using equation (18). The next step is to evaluate the cross covariance between \mathbf{x}_0 and all map features \mathbf{x}_k .

$$\begin{aligned} \mathbf{P}_{0k} = & E\{(\hat{\mathbf{x}}_0 - E(\hat{\mathbf{x}}_0))(\hat{\mathbf{x}}_k - E(\hat{\mathbf{x}}_k))^T\} \\ = & (\mathbf{H}^T \mathbf{C}^{-1} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{C}^{-1} \begin{bmatrix} \vdots \\ \nabla_{\mathbf{x}_{i_1}} \mathbf{G}_i \mathbf{P}_{i_1 k} + \nabla_{\mathbf{x}_{i_2}} \mathbf{G}_i \mathbf{P}_{i_2 k} \\ \vdots \end{bmatrix} \end{aligned} \quad (28)$$

Since the true mean of all equation parameters are not known, all matrix terms in the above equations are evaluated at the

measured values (for \mathbf{m}_i) or the estimates (for $\mathbf{x}_0, \mathbf{x}_{i_1}, \mathbf{x}_{i_2}, \mathbf{x}_{j_1}, \mathbf{x}_{j_2}$). For a plane and a standalone corner [6], the observation equation involves only one map feature state (i.e. \mathbf{x}_{i_2} is not present). Adjustment is then made to the above equations by setting all $\nabla_{\mathbf{x}_{i_1}} \mathbf{G}_i = \mathbf{0}$, with appropriate dimension. The measurement equation involves two map states only if a fused corner is involved [6], in which case the two states come from the two planes which intersect at that corner. The observation equations and the Jacobian matrices for all possible scenarios can be found in [6].

6 Experimental Results

The local mapping algorithm has been tested in two large, real life environments but only one will be shown here. It is composed of a long corridor and a loop into a laboratory. The robot starts at the position indicated with a 'Start' on Figure 9. It then moves right, turns right and enters a laboratory through a doorway, exits the lab at its starting position, moves left, makes a turn, travels back to its starting position again, enters the lab and moves towards the other doorway. The path was planned by a human operator. The tiny circles along the path indicate sensing points, which are uniformly separated by 1m. The total journey is 93m. The tiny line segments are plane measurements; the dark pointy objects are corner measurements and the light pointy objects are edge measurements. Typical landmarks the sonar sensor array pick up as map features are wall mouldings, table legs, wall discontinuities at doors, fire extinguishers, cupboards, concave corners between two walls and between a wall, trolleys and boxes. For the complete navigation, six local maps are produced. They are shown superimposed together in Figure 9.

The average size of a local map is only 41kbytes which is about 4% of the global map. The position clusters for all old local maps contributes another 0.3%. Figure 8 compares the time to process all measurements collected at every sensing position for the global and the local mapping algorithms. The growth rate of the processing time of the global approach is a lot higher than that of the local approach. The global mapping process has to be terminated at the 55th position because the number of features have reached 200 so the number of covariance matrices maintained are $200 \times 201 / 2 = 20,100$. The spikes on the global mapping graph in Figure 8 are proportional to the number of measurements fused into the map at various sensing points but some of which on the local mapping graph are caused by saving local maps to disk. The results for another 'long corridor' environment [7] are similar.

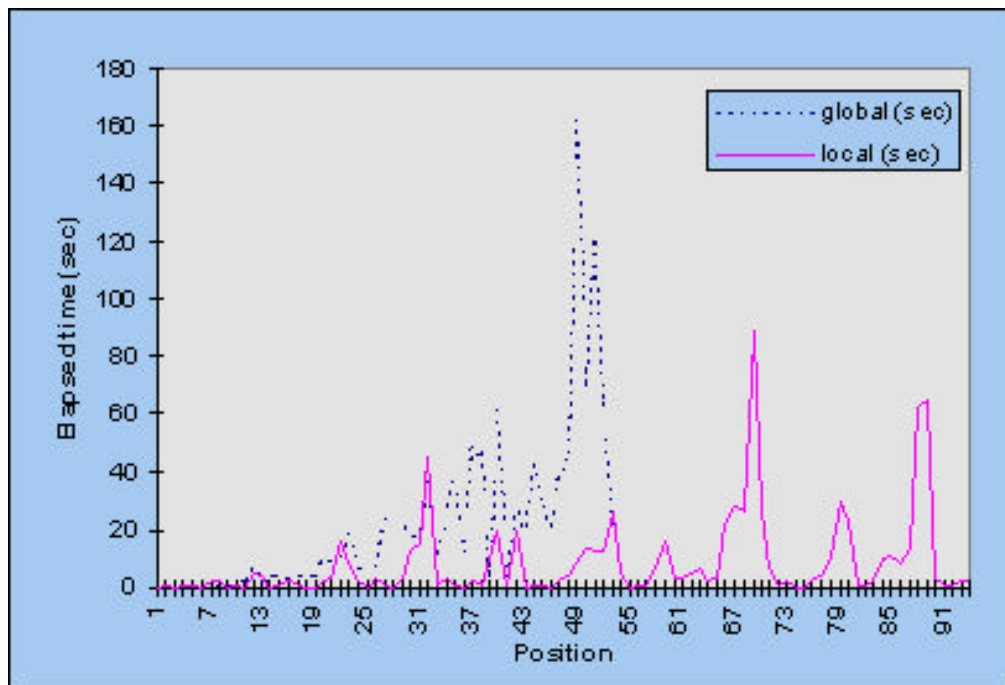


Figure 8 : Time to process all measurements collected at every position

7 Conclusion

The new local mapping strategy has been tested in large environments and is proven to be memory friendly, time efficient, and capable of dealing with the problem of error accumulation. In the future, on the fly navigation can be integrated with the stored local maps by predicting ahead which landmarks to sense to facilitate rapid localisation. For even larger environments, another layer of hierarchy can be mounted to group several adjacent local maps into a higher level local map. Another proposal worth researching is a way of fusing all local maps (including the overlapping ones), possibly using a least square approach, to generate one globally consistent map.

8 Acknowledgments

Mr. Greg Curmi's assistance in the design of the robot, the funding of a large ARC grant, the financial assistance provided by a MGS scholarship and an OPRS scholarship are all gratefully acknowledged.

9 References

1. Asada, M. et al "Representing Global World of a Mobile Robot with Relational Local Maps", IEEE Transaction on Systems, Man, and Cybernetics, vol. 20, no. 6, November/December 1990, pp.1456-1461.
2. Bar-Shalom, Y. and Li, X.R. "Estimation and Tracking: Principles, Techniques and software", Boston, London: Artech House Inc., 1993.
3. Cassandra, A.R. et al "Acting under Uncertainty: Discrete Bayesian Models for Mobile Robot Navigation", IEEE International Conference on Intelligent Robots and Systems, April 1996, pp.963-972.
4. Chong, K. S. and Kleeman, L. "Map Matching with Sonar Features", Technical Report MECSE-1996-13, Department of Electrical and Computer Systems Engineering, Monash University.
5. Chong, K. S. and Kleeman, L. "Accurate Odometry and Error Modelling for a Mobile Robot", Proceeding 1997 IEEE International Conference on Robotics and Automation.
6. Chong, K. S. and Kleeman, L. "Sonar Feature Based Map Building for a Mobile Robot", Proceeding 1997 IEEE International Conference on Robotics and Automation.
7. Chong, K. S. and Kleeman, L. "Sonar Based Map building in Large Indoor Environments", Technical Report MECSE-1997-1, Department of Electrical and Computer Systems Engineering, Monash University.
8. Cormen, T.H. et al "Introduction to Algorithms", MIT Press, 1990.
9. Firby, R.J. et al "Fast Local Mapping to Support Navigation and Object Localisation", Proceedings of SPIE - The 1992 International Society for Optical Engineering, vol. 1828, pp.344-352.
10. Jazwinski, A.H. "Stochastic Processes and Filtering Theory", New York: Academic Press, 1970.
11. Julier, S., Uhlmann, J. and Durrant-Whyte, H. "A New Approach for Filtering Non-linear Systems", Proc 1995 American Control Conference, Seattle, Washington, USA.
12. Kleeman, L. and Kuc, R. "Mobile Robot Sonar for Target Localization and Classification", The International Journal of Robotics Research, Vol. 14, No 4, August 1995, pp.295-318.
13. Leonard, J.J. and Durrant-Whyte, H.F. "Simultaneous Map Building and Localisation for an Autonomous Robot", IEEE/RSJ International Workshop on Intelligent Robots and Systems, Nov 3-5 1991, pp.1442-1447.
14. Malkin, P.K. and Addanki, S. "LOGnets: A Hybrid Graph Spatial Representation for Robot Navigation", AAA-90 Proceedings Eighth National Conference on Artificial Intelligence, 1990, vol. 2, pp.1045-1050.
1. Moutarlier, P. and Chatila, R. "Stochastic Multisensory Data Fusion for Mobile Robot Location and Environment Modeling", 5th International Symposium on Robotics Research, Tokyo, 1989, pp.85-94.
2. Nakamura, T. et al "Behavior-Based Map Representation for a Sonar-based Mobile Robot by Statistical Methods", IEEE International Conference on Intelligent Robots and Systems, April 1996, pp.276-283.
3. Whalen, A.D. "Detection of Signals in Noise", Academic Press, 1971, Chap 10 & 11.

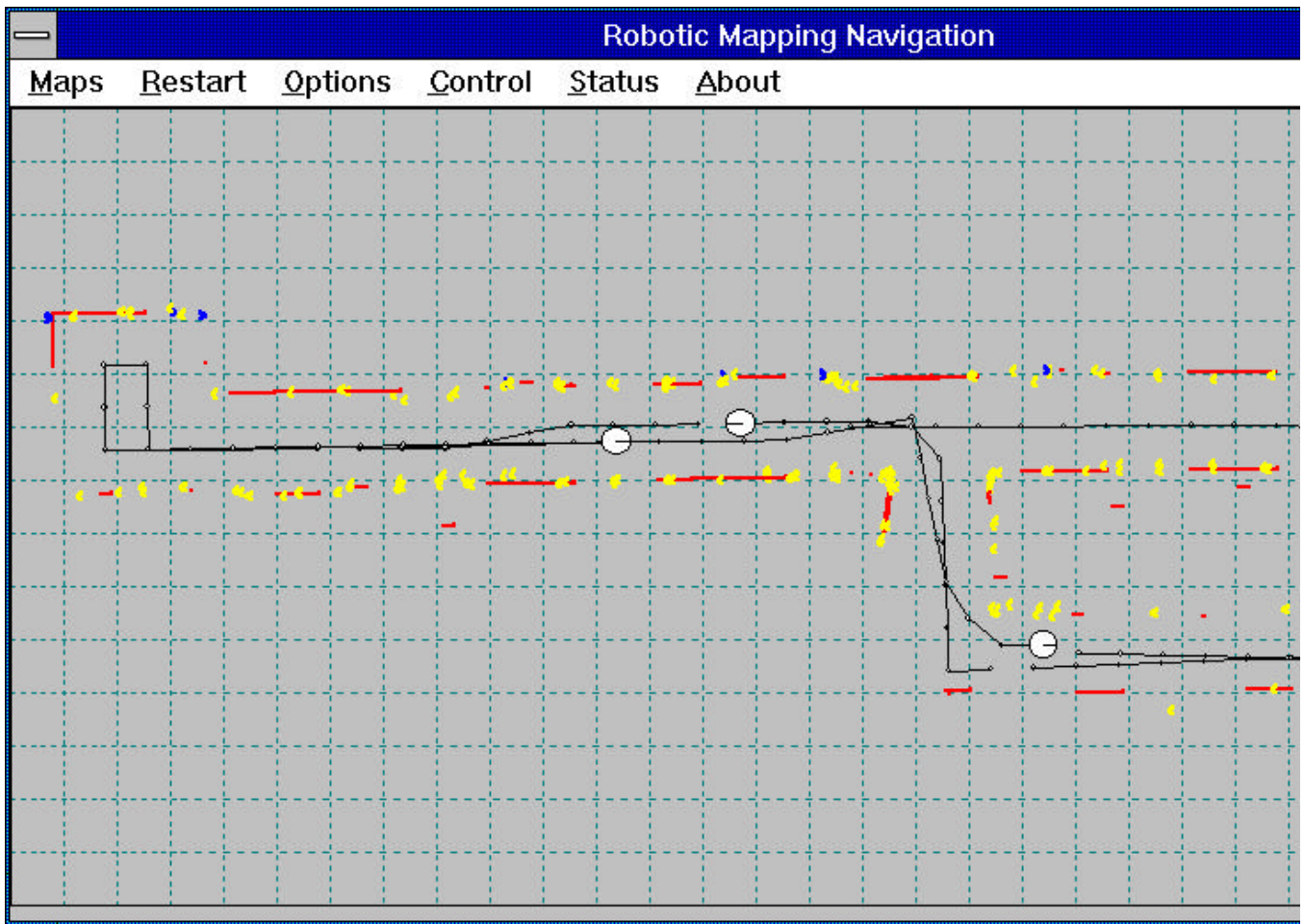


Figure 9 : Local mapping result with all local maps (dashed boxes) superimposed together