DRAFT VERSION - To appear in Robotics and Autonomous Systems Journal, Elsevier.

Subjective Local Maps for Hybrid Metric-Topological SLAM [1]

J.L. Blanco *, J. González, and J.-A. Fernández-Madrigal

Dept. of System Engineering and Automation, ETSII Campus de Teatinos, University of Málaga, E-29071 Málaga, Spain

Abstract

Hybrid maps where local metric sub-maps are kept in the nodes of a graph-based topological structure are gaining relevance as the focus of robot Simultaneous Localization and Mapping (SLAM) shifts towards spatial scalability and long-term operation. In this paper we examine the applicability of spectral graph partitioning techniques to the automatic generation of metric sub-maps by establishing groups in the sequence of observations gathered by the robot. One of the main aims of this work is to provide a probabilistically grounded interpretation of such a partitioning technique in the context of generating local maps. We also discuss how to apply it to different kinds of sensory data (landmarks extracted from stereo images and laser range scans) and how to consider them simultaneously. An important feature of our approach is that the partitioning takes into account the intrinsic characteristics of the sensors, such as the sensor field of view, instead of applying heuristics supplied by a human as in other works. Thus the robot builds "subjective" local maps whose size will be determined by the nature of the sensors. The ideas presented here are supported by experimental results from a real mobile robot as well as simulations for statistical analysis. We discuss the effects of considering different combinations of sensors in the resulting clustering of the environment.

Key words: Graph partitioning, Hybrid maps, Map building, Simultaneous Localization and Mapping (SLAM)

1. Introduction

The topic of Simultaneous Localization and Mapping (SLAM) has received a great attention by the robotics community in the last decade. According to the kind of world model chosen, proposed methods can be classified broadly into metric ones, which use geometrical information [9,11,13], and topological ones, which model the world as a graph whose nodes usually represent distinctive places [1,20]. Recently, hybrid models that combine both types of information have been proposed as a promising solution to deal with large and complex real robot environments. Typically, hybrid approaches attach local geometrical maps (suitable for metric robot localization) to the nodes of a graph-based world representation (suitable for task planning or reasoning) [5,7,22]. A crucial point then is to decide how to partition the whole map of the environment into local maps. From the different proposals reported in the literature, the following ones are of special significance in the context of this work: the Atlas framework [5], where a new local map is started when localization performs poorly in the previous one; and, more recently, the hierarchical SLAM presented in [7], where sensed features are integrated into the current local map until a given number of them is reached. However, none of these works provide a mathematically grounded solution

* Corresponding author
Email addresses: jlblanco@ctima.uma.es (J.L. Blanco), jgonzalez@ctima.uma.es (J. González), jafma@ctima.uma.es (J.-A. Fernández-Madrigal).

[1] Preliminary work was presented in the 2006 IEEE International Conference on Robotics and Automation, Orlando, USA [4].

for the problem or one that does not depend on strong human-provided heuristics that should be manually adjusted for each specific environment. Other interesting works pursue efficiency by hierarchically dividing metrical maps into local regions and subregions, or by exploiting the sparse nature of covariance matrices in the context of EKF-based SLAM [15,24,26]. All these are based on the properties of covariance matrices within maps of *landmarks*, hence they are not applicable to other types of observations (e.g. raw laser range scans).

The approach discussed in this paper consists of partitioning a graph-based representation of robot observations, usually called an *appearance-based representation* when using image sensors [16,17]. Here, the sequence of observations gathered by the robot (and the corresponding poses from which the robot takes each observation) are set as the nodes of an auxiliary weighted graph. By dividing this graph into disjoint clusters of highly connected nodes we can automatically determine a partitioning of the observed environment into "areas", what is required by hybrid approaches to SLAM. The semantics of these areas will be in general related neither to human concepts, such as rooms or a corridor, nor to operational needs. Rather, the distribution of the obtained sub-maps will be determined by the simultaneous visibility of landmarks from different robot poses: sensors with a wider field of view (FOV) will produce larger sub-maps since more overlap will be found between observations. This property that emerges naturally from the physical robot configuration is consistent with the ways of the biological world, where sensory capabilities definitively determine the spatial structure of world models.

An important contribution of this work is the discussion of a new interpretation of the above process in probabilistic terms, hence providing a mathematical basis that justifies its usage in the context of Hybrid Metric-Topological (HMT) SLAM [3]: the resulting partitions will minimize a given measure of the relation between adjacent sub maps (as explained in section 4) with the aim of obtaining sub-maps as much closer to conditional independence as possible.

Given an auxiliary graph of robot observations, there are two critical issues regarding its partitioning: the computation of the arc weights, and the criterion for performing the partitioning itself. As introduced elsewhere [4], we propose to set the weights according to the Sensed Space Overlap (SSO), a pairwise measurement between observations that reflects to what extent a pair of observations capture the same *entities* (points, landmarks, etc.) from the environment, as illustrated in Fig. 1. Regarding the criterion for partitioning the graph, we follow previous works ([4,3,27,28]) that employ the minimum normalized-cut (min-Ncut), originally introduced in [18]. The min-Ncut has the desirable property of generating *balanced* clusters of highly interconnected nodes, i.e. of observations that are very likely to correspond to the same part of the environment, under our definition for graph weights. Furthermore, it can be computed efficiently by means of an approximate solution based on spectral decomposition, which will be also reviewed in this article.

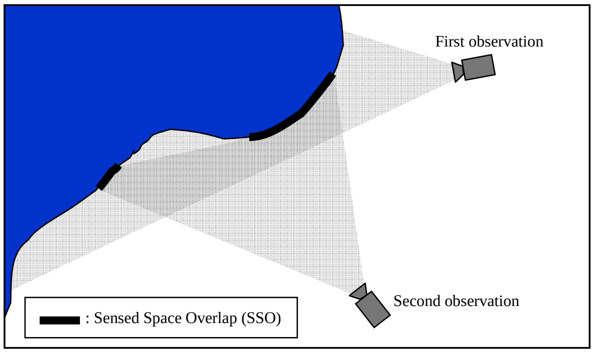


Fig. 1. The sensed-space overlap (SSO) is a measurement of to what extent a pair of observations catch the same part of the environment.

The remainder of the article is organized as follows. In section 2 we review the spectral approach to graph partitioning for generic graphs. Next, we will derive expressions for computing the arc weights of the auxiliary graph for different kinds of sensors. In section 4 , we present the motivations and the formal support for partitioning a sequence of observations within a SLAM framework, and in section 5 we explain how our method can be integrated into an online hybrid SLAM framework. Finally, we validate our ideas by presenting experimental results from real data.

2. Background on Spectral Graph Partitioning

In the following we review the definitions involved in the normalized cut, the basis for the *bisection* of a graph using the spectral approach, and how to extend it for generating any number of clusters.

2.1. *Normalized Cut of a Graph*

Let $G = \langle V, E, \Omega \rangle$ be an undirected, weighted graph, where V is the set of vertices or nodes and E is the set of weighted edges or arcs, using non-negative weight values; the symmetric $|V| \times |V|$ square matrix Ω is the weight matrix, where its element ω_{ij} is the weight of the arc between nodes i and j. According to the definition introduced by Shi and Malik in [18], the *normalized cut* (Ncut) is a measure associated to the partitioning of V into *two* disjoint subsets A and $\bar{A}$, such as $A \cup \bar{A} = V$ and $A \cap \bar{A} = \emptyset$, defined as:

$$Ncut(A, \bar{A}) = \frac{cut(A, \bar{A})}{assoc(A, V)} + \frac{cut(A, \bar{A})}{assoc(\bar{A}, V)} \qquad (1)$$

which in turn uses the standard definition of the *cut* between two disjoint sets of nodes A and $\bar{A}$:

$$cut(A, \bar{A}) \doteq \sum_{u \in A, v \in \bar{A}} \omega_{uv} \quad , A \cap \bar{A} = \emptyset \qquad (2)$$

and of the *association* of two non-disjoint sets of nodes:

$$assoc(A, V) \doteq \sum_{u \in A, v \in V} \omega_{uv} \quad , A \subset V \qquad (3)$$

The association of a given subgraph (A) with the whole graph (V) measures the intergroup "cohesion", that is, the connection "strength" between the two sets of nodes. Note that the definitions above fulfill:

$$assoc(A, V) = cut(A, \bar{A}) + assoc(A, A) \qquad (4)$$

as Fig. 2 illustrates with an example.

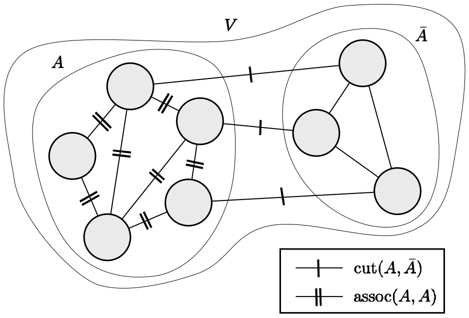


Fig. 2. An example that illustrates the concepts of *cut* and *association* for a pair of sets of nodes. It can be sen how the cut involves only the arcs between two disjoint sets (A and $\bar{A}$ in this case), whereas the association takes into account all the arcs between the non-disjoint sets (A with itself in this example). Observe how the association of a set with the whole graph, i.e. $assoc(A, V)$, can be decomposed into its cut with the rest and the association with itself.

Partitioning graphs under the criterion of minimizing the *cut* value tends to generate groups of no practical utility for some applications, since they have the least connected nodes of the graph. It is of much more interest to get subgraphs with a balance between both, the intergroup and the intragroup cohesion, which is achieved by minimizing the normalized cut (Ncut) defined in Eq. (1). Thus, the minimum normalized cut (min-Ncut) of a graph V is given by:

$$\{A, \bar{A}\} = \arg \min_{A, \bar{A}} Ncut(A, \bar{A}) \qquad (5)$$

The range of possible values for the Ncut can be derived from Eq. (4), which implies that, for the maximum value of the cut (which happens when the nodes in a group are connected only to the other group), the values of $assoc(A, A)$ and $assoc(\bar{A}, \bar{A})$ are zero, therefore:

$$assoc(\mathrm{A}, \mathrm{V})\big|_{\min} = assoc(\bar{\mathrm{A}}, \mathrm{V})\Big|_{\min} = cut(\mathrm{A}, \bar{\mathrm{A}}) \qquad (6)$$

Since the minimum value attainable from a cut is zero, corresponding to the case of no connections between the two groups, the minimum Ncut value is also zero. On the other hand, the maximum Ncut value is determined by the maximum values of each of the terms in the sum of Eq. (1). From Eq. (4) we see that the maximum value of each of these terms is:

$$\begin{aligned} max \, \frac{cut(A, \bar{A})}{assoc(A, V)} &= max \, \frac{cut(A, \bar{A})}{cut(A, \bar{A}) + assoc(A, A)} \\ &= \frac{cut(A, \bar{A})}{cut(A, \bar{A})} = 1 \qquad (7) \end{aligned}$$

Thus, the Ncut provides a numerically well defined measure of the quality of a partition that falls within the range $[0, 2]$.

As discussed in the work by Shi and Malik [18], finding the exact min-Ncut bisection is computationally intractable (a NP-complete problem), hence we follow their proposal for an approximate solution based on spectral decomposition, which leads to near-optimal cuts. Their method is summarized next for completeness, though it could be skipped by the reader since it is not a contribution of this work and is not necessary for following subsequent sections.

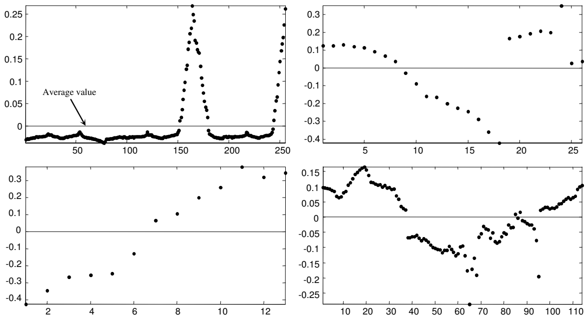


Fig. 3. Four real examples of graph spectral bisections. The plots show the components of the eigenvectors which are used to choose the bisection. The length of these vectors is given by the number of nodes in the graph. The average of the eigenvectors (horizontal lines) is used as the bisection threshold.

2.2. *Spectral Bisection*

Let x be the bisection indicator vector with dimension $N = |V|$, where each element x_i equals 1 or -1 depending of the node i falling into the group A or $\bar{A}$, respectively. Let d be the vector with the sum of the weights of the incident arcs for each node, that is $d_i = \sum\limits_j \omega_{ij}$. We build a diagonal matrix D with d as its diagonal. It can be shown that the min-Ncut problem can then be rewritten as:

$$\arg\min_x Ncut(x) = \arg\min_y \frac{y^t(D-\Omega)y}{y^t Dy} \tag{8}$$

where $y = (1+x) - b(1-x)$, with 1 being a $N \times 1$ vector of all ones, and

$$b = \frac{\sum\limits_{x_i>0} d_i}{\sum\limits_{x_i<0} d_i} \tag{9}$$

Ideally, the elements of vector $\mathbf{y}$ should take just two discrete values, since x_i can be either 1 or -1. However, if this condition is relaxed and $\mathbf{y}$ is allowed to be real valued (this is the approximation of the approach), then Eq. 8 is no longer discrete and can be minimized by solving the generalized eigensystem:

$$(D-\Omega)y = \lambda Dy \tag{10}$$

where $L(V) = D - \Omega$ is a well-known term, namely the Laplacian matrix of the graph ([4],[8]). The above equation can be rewritten as a standard eigensystem using $z = Dy$:

$$D^{-\frac{1}{2}}(D-\Omega)D^{-\frac{1}{2}}z = \lambda z \tag{11}$$

It can be shown that $z_0 = D^{\frac{1}{2}}1$, the eigenvector corresponding to the smallest eigenvalue in Eq. (11) ("the smallest eigenvector" from now on), is zero. Translating back this result to the original system in Eq. (10), we have that $y_0 = 1$ is the smallest eigenvector of Eq. (11). Since the fraction in Eq. (8) is a Rayleigh quotient [12], and its eigenvectors are orthogonal [2] , then both Eq. (10) and Eq. (11) are minimized with the next smallest eigenvector. Thus, we have that solving the min-Ncut expressed in Eq. (8) is equivalent to finding the second smallest eigenvector, y_1, of Eq. (10).

The only approximation assumed in the above derivation is that the components of the eigenvector y_1 will not take just two discrete values, but any real number. Obviously, this complicates the bisection criterion; still, in many situations there will be a clear distinction between the two clusters of nodes A and $\bar{A}$, as can be observed in the real examples shown in Fig. 3. Three different criteria seem plausible for assigning each node a group: (i) look at the sign of each component of the eigenvector; (ii) take the mean value of the eigenvector as a threshold for the partition, and (iii) sweep over the different threshold values looking for the minimum Ncut. The second one is the criterion we have used in our implementation due to its compromise between efficiency and good results.

2.3. *Partitioning into k-groups*

The method presented above provides a solution to the graph bisection problem; however, this method must be generalized to divide a graph into a variable number of subgraphs. An easy and effective way of achieving that is to recursively apply bisection to any of the generated subgraphs as long as two clearly differentiated groups are obtained. The resulting min-Ncut value for a given bisection is a well-grounded measure of the goodness of the cut. The Ncut of a graph, with values in the range $[0, 2]$, measures the inter-group cohesion of the resulting subgraphs, inversely scaled by the intra-group cohesion. Values close to zero indicate almost no connection between groups (a good partition), while values near 2 indicate that the groups are more strongly connected to each other than with themselves (the

[2] Since the Laplacian matrix $D - \Omega$ is positive semidefinite, $D^{-\frac{1}{2}}(D-\Omega)D^{-\frac{1}{2}}$ is symmetric positive semidefinite, thus its eigenvectors are orthogonal

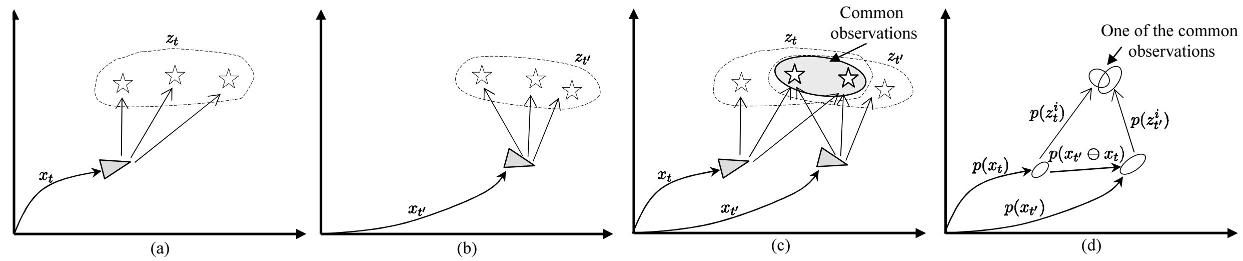


Fig. 4. A graphical representation of the variables involved in computing the SSO for a pair of observations. (a)–(b) The robot pose and corresponding observations at two time steps t and t'. (c) Some of the sensed elements (landmarks or points) may be common to both observations. (d) A representation of the probabilistic distributions taken into account when determining whether two elements from different observations correspond to the same map element or not. Refer to section 3.1 for further explanation.

partition should not be done). Therefore, an intermediate value must be established as a threshold $0 < \tau < 2$ to decide accepting the bisection or not. This value is typically chosen heuristically ([18,25]), and in practice a value in the range $[0.2, 1]$ will give good results. The procedure for k-ways partitioning is summarized in Table 1 for clarity.

3. The Similarity Function SSO

In this section we address the problem of assigning weights to our graph of observations. To this purpose, we introduce the Sensed Space Overlap (SSO), a pairwise similarity measure for observations that reflects how much space in common sense a pair of observations.

Let the map variable m be comprised of the set of variables for each individual map element, such that $m = \{m^1, ..., m^i\}$. Similarly, the observations at a certain time step t, z_t, will be considered as composed of the observations of individual map elements, that is, $z_t = \{z_t^1, ..., z_t^k\}$. The nature of these "map elements" is not relevant for the generic definition of the SSO [3] . Furthermore, we will denote the set of map elements sensed in a given observation z_t as $M(z_t)$. Put mathematically:

$$M(z_t) = \{m^i : z_t^k \text{ observes } m^i, \forall\, k\} \tag{12}$$

Table 1
Recursive Algorithm for Spectral Partitioning in k-groups

```
algorithm RecursivePart($G$) → {$P$}
    ({$A, B$}, $N_{cut}$) = SpectralBisection($G$)
    if $N_{cut} < \tau$
        $P$ = {RecursivePart($A$), RecursivePart($B$)}
    else
        $P = G$
    end-if
end-algorithm
```

At this point, we can define the SSO function in general for any pair of observations z_a and z_b as the ratio of commonly observed map elements relative to the overall number of observed elements. Using the notation defined above, we can write down the generic expression for the SSO as:

$$SSO(z_a, z_b) \doteq \frac{|M(z_a) \cap M(z_b)|}{|M(z_a) \cup M(z_b)|} \tag{13}$$

where $|\cdot|$ stands for the cardinal (the size) of a set. The SSO can be implemented differently for each kind of sensor. Next we derive expressions for two kinds of sensory data: landmarks and range scans. A similar definition that could be employed for monocular images has been proposed in [27].

3.1. *Landmark Observations*

We are interested in computing the similarity $SSO(z_t, z_{t'})$ between the observations z_t and $z_{t'}$ taken at time steps t and t' from the robot poses x_t and $x_{t'}$, respectively. Recall that each observation z comprises a set of individual features $\{z^i : i = 1..k\}$, one for each sensed map element. An individual feature z^i represents the observed spatial position of the i'th map element (a landmark in this case) relative to the robot pose x at the corresponding time step. All the involved variables are represented graphically in Fig. 4(a)–(b) for clarity. The only problematic step in computing the SSO is determining the number of matches from common elements between the pair of observations z_t and $z_{t'}$, as the ones shown in the example in Fig. 4(c). Concretely,

[3] They would be cells for an occupancy grid map, individual features in a landmark-based map, etc.

to decide if a pair of individual features z_t^i and $z_{t'}^j$ correspond to the same map element, we have to check whether the following equality holds:

$$x_t \oplus z_t^i = x_{t'} \oplus z_{t'}^j \tag{14}$$

being $\oplus$ the pose composition operator [19].

Within a probabilistic SLAM framework there is uncertainty in measures and poses, thus we do not know the exact value of the variables, but only their associated probability distributions. Therefore, we can only decide for correspondences within a given certainty bound. This can be visualized with the uncertainty ellipses shown in Fig. 4(d), where it can be seen how the position of the same landmark is represented by different (although overlapping) Gaussians according to each of the observations.

If we rewrite Eq. (14) as[4] ,

$$\epsilon = (x_{t'} \ominus x_t) \oplus z_{t'}^j - z_t^i = 0 \tag{15}$$

with $\ominus$ being the inverse pose composition operator [19], we can then state the probability *density* of actually having a correspondence through $p(\epsilon = 0)$. If we assume that the probability distributions of both the robot pose and the landmarks can be appropriately approximated by Gaussians (which is acceptable in a hybrid mapping framework since coordinates into local maps are always relative to a near reference [3,5,21]), then $p(\epsilon)$ can also be approximated by a Gaussian by first-order uncertainty propagation (we omit the straightforward calculations). As a result, $p(\epsilon)$ would be centered near the origin if there is a correspondence, or far away otherwise. A robust criterion for deciding the correspondence is then to compute the Mahalanobis distance from the origin to the mean of the Gaussian, and to accept the correspondence if the distance is below, for example, a value of 3 (which represents a 99.7% confidence interval).

Notice that we are assuming that all the landmarks are indistinguishable and the matching must be determined solely from the spatial information, but there are some situations where landmarks have some sort of descriptor, which can then be integrated into the calculation of the Mahalanobis distance. In concrete, for the experiments shown at the end of this work, we have used SIFT [10] visual descriptors in addition to the spatial distance, as described in more detail in [14].

3.2. *Range Scans Observations*

In principle, we could apply the same process as in section 3.1 to compute the SSO of a pair of observations comprising raw range scans. However, there are subtle differences in the nature of the sensory data which make desirable the introduction of a slight modification: due to the discrete set of scanning directions it is very unlikely that exactly the same *point* (not landmark) is measured while scanning from different poses. We present a solution for accounting for this fact when considering uncertainties. This is the only difference with the process described above for landmarks.

If we denote as $\mathbf{C_t^i}$ the covariance of the 2D point $x_t \oplus z_t^i$ (please, refer to Fig. 4(d)), we can model the uncertainty due to the discrete sampling of the environment by summing an additional term σ_f to the diagonal of the covariance:

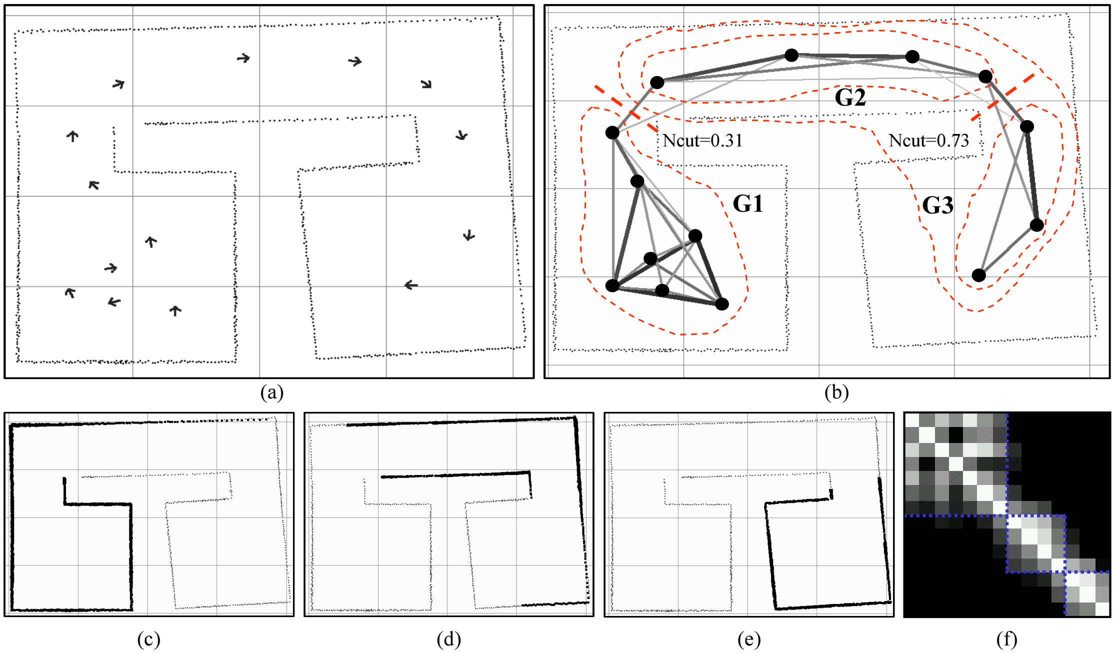
$$\mathbf{C_t^i} = \mathbf{J} \begin{pmatrix} \Sigma_\mathbf{p} & 0 \\ 0 & \Sigma_\mathbf{s} \end{pmatrix} \mathbf{J}^t + \begin{pmatrix} \sigma_f^2 & 0 \\ 0 & \sigma_f^2 \end{pmatrix} \tag{16}$$

where Σ_p and Σ_s are the covariances of the robot pose and sensor measurement, and $\mathbf{J}$ is the 2×5 Jacobian matrix of the pose composition operator. Since the spatial uncertainty due to the discrete sampling of surfaces is proportional to the sensed range (r) at each scanning direction, the value σ_f can be set to $r\beta$, being β a constant of the order of the discrete angular steps between the scan ranges.

3.3. *An Example*

It is illustrative at this point to consider an example to show how our overall method works off-line for partitioning the graph of observations shown in Fig. 5 (simulated raw range scans in this case). Here, the graph is firstly divided into the groups $\{G1\}$ and $\{G2, G3\}$ in the first execution of the partitioning algorithm. Going on recursively, the latter group is partitioned again due to its low Ncut value. The so resulting groups $\{G2\}$ and $\{G3\}$ are no longer bisected since the corresponding minimum Ncut values are above the threshold (set to 1 in this example), i.e. it is better not to separate the observations between each group. Notice that the final observation groups do roughly correspond to each of the

[4] Using the matrix form of Eq. (14) in homogeneous coordinates, $\mathbf{X}_t \mathbf{z}_t^i = \mathbf{X}_{t'} \mathbf{z}_{t'}^j$, we can operate to obtain $\mathbf{X}_t^{-1} \mathbf{X}_{t'} \mathbf{z}_{t'}^j - \mathbf{z}_t^i = 0$, which is stated in Eq. (15) using pose composition operators.



(a) (b)

(c) (d) (e) (f)

Fig. 5. An illustrative example of the graph partitioning method applied to a 2D laser map. (a) The global map obtained from 14 observations – arrows indicate the poses where observations were taken from. Notice that the map presents some orientation errors. (b) The auxiliary graph of observations. Each node contains the sensed space data (scan), and an estimate for its pose. The darker the arc, the higher the SSO between the observations. The observation graph is recursively partitioned into three groups: firstly, it is divided into two groups $\{G1\}$ and $\{G2, G3\}$, then, the latter group is partitioned again because it has a minimum Ncut below the threshold. The local maps obtained from these groups are shown in (d), (c) and (e), respectively. In (f) the weight matrix of the associated graph is shown as an image with dotted squares for the three partitions.

natural "rooms" that can be observed in the figure, although, as commented in the introduction, our partitioning method does not aim for "human-like" semantics, but for a "subjective" perception of the world by the robot.

4. Theoretical Support for Hybrid SLAM

In this section we derive a justification for the usage of the SSO as a metric within a min-Ncut partitioning in the context of hybrid SLAM, which is one of the contributions of this work.

Following the standard notation in the SLAM literature [23], the ultimate goal of any probabilistic localization and mapping method is to compute the joint posterior density of the robot path $x_{1:t}$ and the map m given the sequence of observations $z_{1:t}$ up to time step t:

$$p(x_{1:t}, m | z_{1:t}) \qquad (17)$$

where the robot actions have been dropped for clarity. Stated as a sequential Bayesian estimation problem, the statistical structure of the variables is the one illustrated in Fig. 6(a) as a dynamic Bayesian network. A critical issue in this model is that any observation z_t obtained by the robot depends on the *whole* map, represented by m. Although this condition is rigorously true, in practice the observations capture only a limited part of the map. Based on this idea, the authors proposed in [3] a factorization of the SLAM problem into a unified hybrid metric-topological (HMT) Bayesian estimation problem. Basically, the map m is divided into a set of metric sub-maps which can be estimated from (ideally) conditional independent sequences of observations. Here we discuss why the min-Ncut using the SSO for arc weights is a good choice for generating these sub-maps. The following reasonings are also applicable to other hybrid (or "hierarchical") approaches to SLAM [5,7], since the approximations introduced by any approach that divides the map into sub-maps are more negligible as the observations between the different clusters become closer to conditional independent (given the robot path).

As illustrated in Fig. 6(b), the SSO captures the fact that there may be some map elements m^i sensed

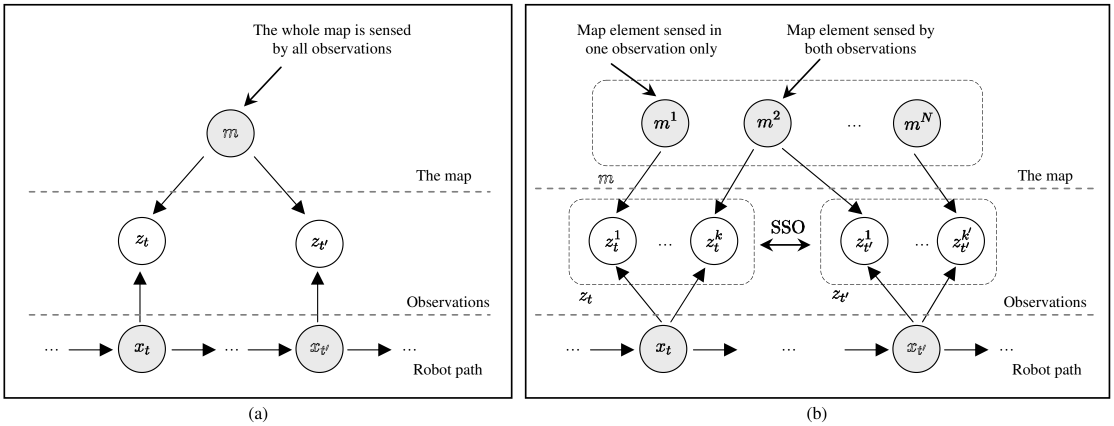


Fig. 6. (a) The common structure of the SLAM problem as a dynamic Bayesian network. (b) The same model if we split up the map and the observations into its separate variables for the individual map elements. Under this perspective, the SSO can be defined as the ratio of common map elements sensed by a pair of observations. As discussed in the text, this quantity is related to the "degree of Independence" between the involved variables.

by just one of the two observations, and other map elements sensed by both of them. It is straightforward to verify from Eq. (13) that the SSO function gives values in the range $[0,1]$, suitable to be used as weights in the arcs of the auxiliary observation graph: observations sensing exactly the same map elements are assigned a value of 1 and observations from totally different parts of the environment have a null SSO value.

Consider now the particular case of a null *cut* value between two clusters of observations A and $\bar{A}$ using the SSO as the arc weights, which in turn implies a null Ncut value (see Eq. (1)), that is:

$$cut(A,\bar{A}) = \sum_{a \in A, b \in \bar{A}} SSO(z_a, z_b) = 0 \tag{18}$$

Since the SSO is non-negative, it follows that:

$$SSO(z_a, z_b) = 0 \;\; \forall (a,b) \in A \times \bar{A} \tag{19}$$

which, given Eq. (13), implies that:

$$M(z_a) \cap M(z_b) = \emptyset \tag{20}$$

This relation can also be proved in the opposite direction: two sets of observations that do not contain any common map element have a null cut (and Ncut) value. In other words, using the min-cut or the min-Ncut criteria for partitioning is *equivalent* to finding sets of observations with the least map elements in common. Our choice for the min-Ncut (rather than the min-cut) criterion is due to its desirable property of producing more balanced groups (refer to [18]).

To sum up, for the case of a null Ncut value we can rigorously factorize the SLAM problem into statistically-independent estimations. In practice, it is virtually impossible to obtain such strict independence between difference areas of a large map. In the scope of hybrid SLAM [3], we propose to settle a threshold value τ for the maximum admissible Ncut value for actually partitioning the map into areas, whose value coincides with that employed in the recursive algorithm in Table 1. Note that this threshold value does not depend on the kind of sensors employed since Ncut values will represent relative SSO values.

5. Sequential Operation within Hybrid SLAM

In previous sections we have stated the problem of partitioning a sequence of observations as an off-line process, assuming a static and complete sequence of observations and associated robot poses. In the following we describe the issues raised when our method is applied to online SLAM.

Firstly, we must remark that there may exist several ways of integrating the partitioning technique into a hybrid metric-topological SLAM framework. In concrete, we will discuss here the approach taken in [3], where the sequence of the last robot observations is partitioned to check whether the robot has entered into a new area or not at each time step. More generally, the recursive partitioning algorithm would reveal the different areas in which the robot observations nearby its current position can be grouped into.

Within this context, a new node containing the last observation and the current probabilistic estimate of the robot pose is attached to the auxiliary observation graph for each time step of the SLAM algorithm. The only difference to an off-line (batch) version of our method lies in the computation of the weight matrix Ω, which is to be built sequentially as new nodes are added to the graph. Let Ω_t denote the weight matrix for the sequence of observations gathered up to time step t. For each new observation z_t this matrix can be updated just by expanding the previous one (Ω_{t-1}) with a new row and a new column:

$$\Omega_t = \left(\begin{array}{c|c} \Omega_{t-1} & \omega_{1:t-1,t} \\ \hline \omega_{1:t-1,t}^T & 0 \end{array}\right) \tag{21}$$

Since the weight of the reflexive arc that connects each node with itself is not employed in the calculations of the min-Ncut, the corresponding elements in the Ω matrix (the diagonal) can be set arbitrarily to zero, as in the equation above. Each update of the weight matrix implies the evaluation of the SSO for $t-1$ pairs of observations, corresponding to the column $\omega_{1:t-1,t}$ in Eq. (21). Thus, updating the Ω matrix at each time step has a computational complexity that increases linearly with time t. After updating the matrix, the bisection eigenvector must be computed (recall section 2.2), which can also be achieved in $\mathcal{O}(t)$ by applying the Lanczos algorithm [8]. To sum up, the overall complexity remains linear with the number of previous observations. This growth in complexity over time is not a problem as long as eventually the robot moves to a different area and a new matrix is created. Thus, in practice there is an upper bound to the size of this matrix, although it will depend on the specific structure of the environment and on the robot path.

6. Experimental Results

We firstly provide some statistical results aimed to compare our method to other previous proposals, and next we show some typical partitions obtained for an indoor scenario for different combinations of sensors [5].

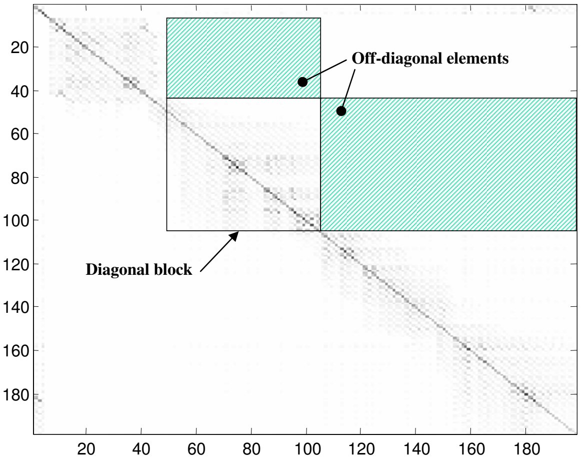


Fig. 7. The information matrix of a EKF solution to SLAM represents the shared information between landmarks in the map. Partitioning a map implies assuming conditional independence between different sub-maps: the matrix elements out of the diagonal block matrices are forced to zero. We employ this matrix to measure the information loss due to different partitioning methods.

6.1. *Statistical experiments*

As discussed in section 4, separating the map into clusters will incurs in approximation errors for most practical situations. In order to compare the loss caused by our method to other alternatives, we will define a measure of how much information is lost by performing any arbitrary partitioning of a map. The context for this comparison is EKF-based SLAM [6] rather than the RBPF-based approach employed in the next section. The reason for using an EKF with a map of landmarks is that the cross covariances between map elements are explicitly kept in the filter covariance matrix, whereas in grid-mapping with a RBPF they are not available.

The covariance matrix is important for our purposes since its inverse, the *information matrix*, represents the amount of shared information between the different landmarks in the map. Under a sub-map approach to SLAM, where the aim is to partition the map into statistically independent clusters, the information is maintained in block diagonal sub-matrices only, while the off-diagonal elements are discarded (i.e. assumed to be zero); this is illustrated with an example matrix in Fig. 7. In

[5] The datasets and C++ source code for these experiments are available in http://babel.isa.uma.es/mrpt/papers/.

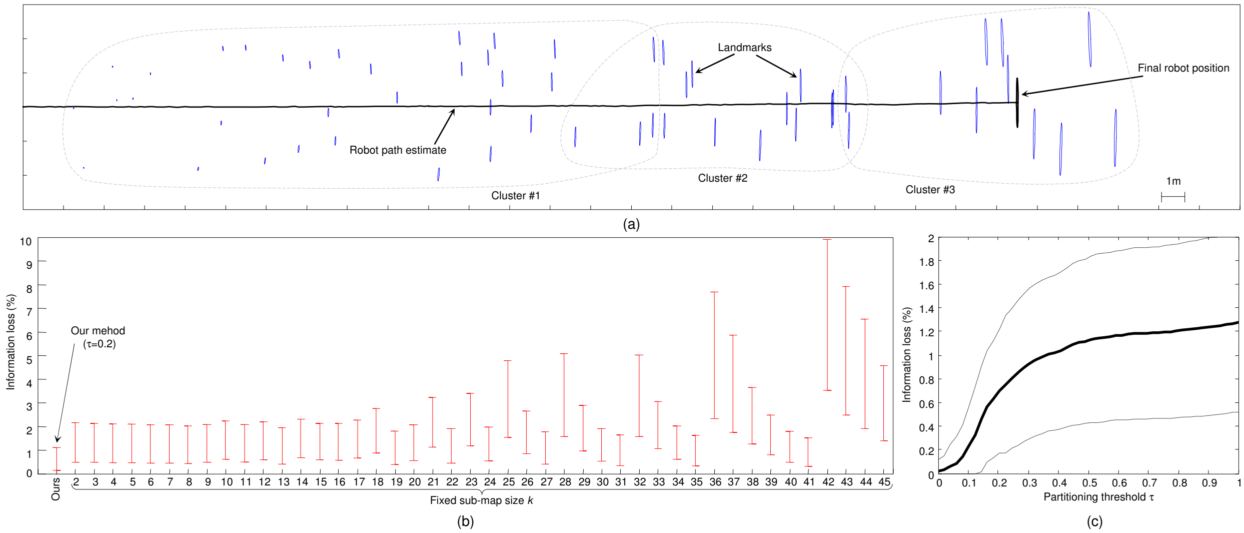


Fig. 8. Results for the statistical analyses of the information loss due to map clustering. (a) The final state of an EKF after one of the simulations. Observations are grouped (in grey) according to the three clusters produced by our method in this particular run. (b) The average and 68% confidence interval for the information loss e, calculated for both our method (the left-hand value) and for fixed size sub-maps comprised of k observations each. (c) The information loss and the 68% confidence interval for our method using different values of the threshold τ.

fact, approaches to SLAM based on a Sparse Extended Information Filter (SEIF) explicitly set to zero some entries of the matrix (which were not zero previously), therefore assuming a certain loss of information [24,26]. We must remark that the method presented in this paper does not rely on the information matrix as those based on a SEIF, hence its applicability to other kinds of mapping frameworks such as grid mapping with RBPFs.

We propose to measure the effects of forcing conditional independence (i.e. partitioning the map) through the information loss ratio e, defined as:

$$e = \frac{\sum\limits_{(i,j):C(i)\cap C(j)=\emptyset} |H(i,j)|}{\sum\limits_{a,b} |H(a,b)|} \tag{22}$$

Here $H(i,j)$ stands for the information matrix entries for landmarks i and j, and $C(i)$ represents the set of clusters which the landmark i belongs to. Put in words, the loss ratio e is proportional to the sum of all the information matrix elements out of the block diagonal matrices for each cluster in the partition (please refer to Fig. 7).

We have carried out simulations by running 150 times a simple EKF [2] for a planar path consisting of a straight trajectory of 50 meters long through an environment with 60 point features uniformly distributed, placed at random positions for each run. The final state of the filter after one of the runs is shown in Fig. 8(a), including the three clusters in which our method divides the observations in this case. Loops in the robot path have been intentionally avoided to keep the experiment simple and to obtain generic results independently of the implementation framework.

The first result from the simulations is the comparison of the average information loss (e) for the different methods for partitioning the sequence of observations, summarized in Fig. 8(b). The left-hand value corresponds to our proposal, in this case using a fixed value of 0.2 for the recursive bisection threshold τ. It can be seen that the largest part of the confidence interval for e is below 1%, which is in contrast to the other alternative method, which consists of starting a new map after a fixed number k of observations, with k varying from 2 to 45. Although the loss of information is not drastic (less than 3% for most values of k), it is clear that our method not only implies a more reduced approximation error, but it is also more predictable as revealed by the lower variance of e. We must note that the proposal of starting a new sub-map every fixed number of features was proposed in [7], but other heuristics, such as starting a new sub-map when localization performs poorly [5], can be also ultimately expected to start new maps at a regular rate if landmarks are distributed uniformly through the environment.

A second statistical result is the characterization of our approach with respect to its parameter – the threshold τ. As expected, lower values of τ lead to lower approximation errors e, as represented in Fig. 8(c). Though this may seems to suggest to always use a threshold close to zero, in practice a compromise should be found between admissible errors and the size of the sub-maps, which grows as τ decreases. It is noteworthy that the expected error for $\tau = 0$ should be zero since sub-maps will be created only when no observation senses two landmarks from different clusters. Instead, our simulations give an average information loss of $e = 0.022\%$ (negligible, but not null). The reason of this result is that we integrate odometry in the simulations, which is known to create correlations between landmarks even when they have been not observed simultaneously (refer to [26] for a enlightening discussion on this topic).

6.2. *Partitioning a real indoor map*

To demonstrate the partitioning obtained by our method we have applied it to a sequence of observations gathered by one of our mobile robots, which is equipped with a front SICK laser scanner, a rear HOKUYO laser scanner, and a stereo camera pointing forward. SIFT image features are extracted from the stereo images at each time step to generate the set of 3D features (more details can be found in [14]) that we will consider landmark-like observations. To reveal the differences in the obtained partitions for each kind of sensor and their possible combinations, we have performed the partitioning four times: using the front laser scanner only, both laser scanners (providing a field of view of almost 360° – except for small lateral dead angles), visual landmarks only, and the three sensors simultaneously. When several sensors are combined into the same observation simultaneously, the corresponding SSO is computed by averaging the individual SSO functions, as can be easily derived from the definition in Eq. (13). The effects are discussed below.

The results are summarized in Fig. 9, where each row represents one set of sensors, and the left column shows the resulting groups of observations on an occupancy grid map of the environment (visual landmarks are not shown for clarity). The middle column shows the final SSO (arc weight) matrix for the time-ordered sequence of observations through the environment. Since the robot revisit the same places several times, it is remarkable that many off-diagonal elements contain high SSO values, i.e. they correspond to close areas. In the right column in the figure we can observe the rearranged matrices, built up by making the observations within the same cluster to have consecutive indexes: after clustering, the output matrix should be block diagonal ideally. In the figure we can see how the rearranged matrices for the experiments are clearly not block diagonal, but certainly most of the non-zero elements are approximately contained within the block diagonal matrices. The elements outside these diagonal blocks are the information that would be lost in the hybrid SLAM approach.

It is interesting to note the differences in the partitions obtained from the different sensors. Firstly, for the case of just one 180° FOV laser scanner (the top row in the figure) there exists overlap between different detected areas. For example, the groups #1 and #2, or the groups #4 and #5. However, actually each overlapping group contains observations with opposite robot headings, that is, since the FOV is 180° there is almost no overlap between the observations taken by the robot going in one direction and in the opposite. This does not occur in the second case, when two laser scanners are considered simultaneously covering almost 360° around the robot. For these sensors, we obtain the clustering closest to the "human" concept of rooms, with almost no overlap between the resulting areas. This is in contrast to the results from a stereo camera (third row in Fig. 9), with a narrow FOV of roughly 65°. We preprocess the images from the camera to obtain a set of 3D landmarks, which we consider as the camera observations themselves (please refer to our previous work [14] for a description of the process). More different areas are detected in this case (10 areas), whereas they were just 4 for the case of two laser scanners. The reason is that the narrow FOV leads to many groups of a few observations each, specially if the robot rotates.

In the case of using all the three sensors (two laser scanners and the stereo camera) we obtain the clusters shown in the bottom row of the figure. Here the weight matrix Ω is the average of those from the individual sensors (notice how this matrix, in the central column of the figure, is a mixture from the matrices at the second and the third row). We obtain 6 areas, which is an intermediary value between that obtained from the two laser scanners and the camera independently. Therefore, mixing sensors with largely different FOVs (360° vs. 65°) could be seen as having one single sensor with an intermediate FOV, a natural consequence of computing SSO by

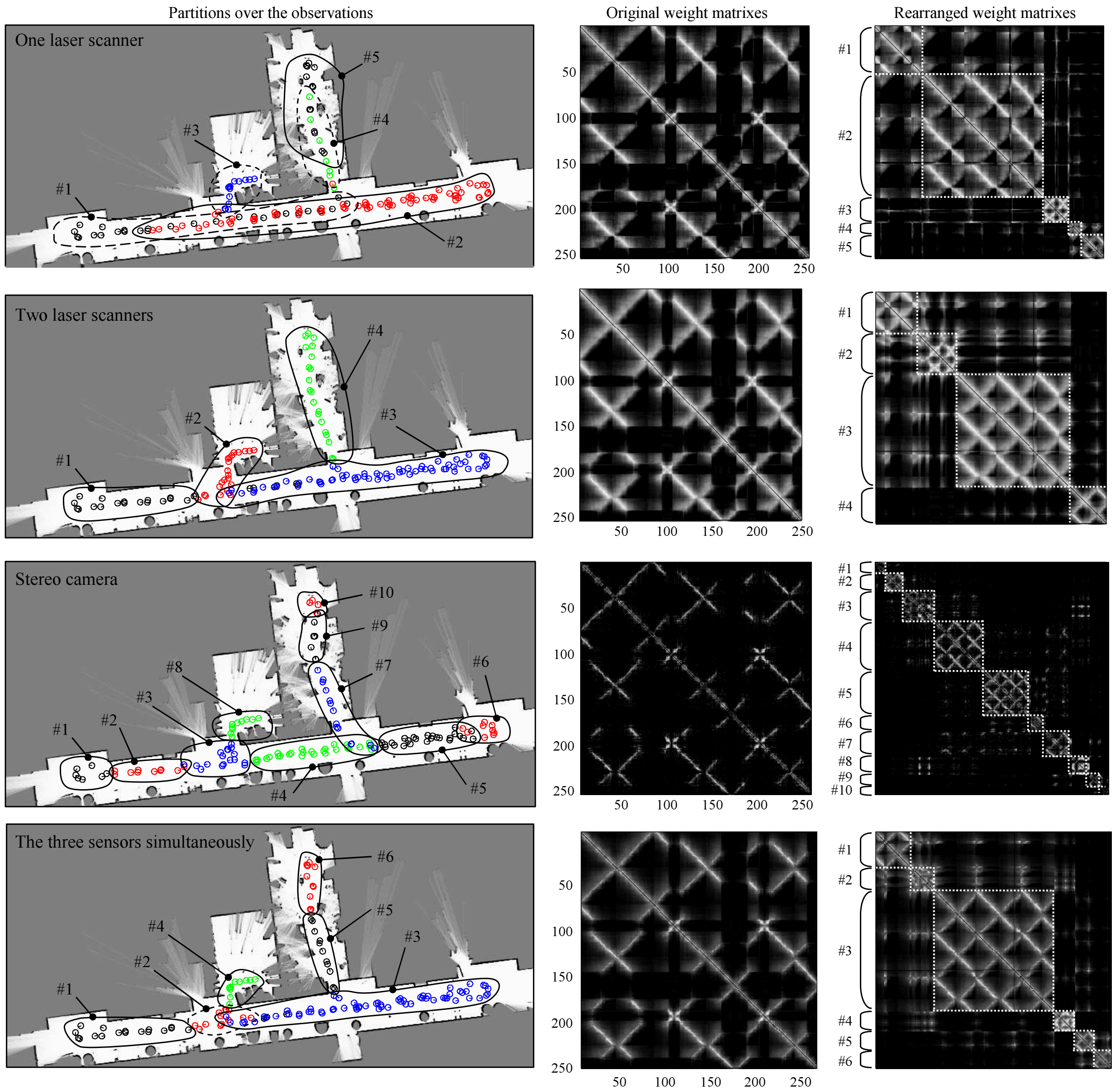


Fig. 9. Results from partitioning a map gathered by a real mobile robot. The left column shows the resulting clusters of observations obtained by the proposed method, while the middle and right columns contain the weight matrices Ω before and after rearranging the elements according to the partitioning, respectively. It can be seen how after the rearrangement most of the high values in Ω are within the diagonal blocks. Each row of graphs presents the results for a different set of sensors: one 180° laser scanner, two laser scanners, a stereo camera, and the three sensors simultaneously. See section 6.2 for a discussion of the results.

averaging over the different sensors.

Finally, we can observe in the rearranged weight matrices (at the right in Fig. 9) how most of the SSO high values are within the diagonal blocks corresponding to the clusters computed by our method. However, a few values are left out of these blocks. Within a hybrid SLAM method this would mean that the information out of the block-diagonal approximation of this matrix would be lost.

7. Conclusions

In this paper we have used spectral techniques for efficient graph partitioning in the generation of submaps for hybrid SLAM frameworks. Through the introduction of the SSO function, we have provided a general formulation for partitioning sequences of observations from different sensory data, illustrated with both range scans and landmarks. An important contribution of this work is the discussion of a probabilistically grounded interpretation of the usage of min-Ncut and the SSO function to probabilistic hybrid SLAM. Moreover, we have derived expressions for applying our ideas to several kind of sensors that ultimately can be modeled either as range sensors or landmark-detectors. We provide a statistical analysis of our method compared to other alternatives, as well as analyzed how the use of sensors with different FOVs affects the resulting clusterings obtained from a real data set. Thus, our approach produces robot "subjective" local maps. The discussed methods have been successfully integrated in a framework for hybrid mapping [3].

References

[1] P. Beeson, N.K. Jong, and B. Kuipers. Towards Autonomous Topological Place Detection Using the Extended Voronoi Graph. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4373–4379, 2005.

[2] J.L. Blanco. Derivation and Implementation of a Full 6D EKF-based Solution to Bearing-Range SLAM. Technical report, 2008.

[3] J.L. Blanco, J.A. Fernández-Madrigal, and J. Gonzalez. Towards a Unified Bayesian Approach to Hybrid Metric-Topological SLAM. *IEEE Transactions on Robotics*, (in press), 2008.

[4] J.L. Blanco, J. Gonzalez, and J.A. Fernández-Madrigal. Consistent Observation Grouping for Generating Metric-Topological Maps that Improves Robot Localization. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 818–823, 2006.

[5] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, and S. Teller. An Atlas framework for scalable mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 1899–1906, 2003.

[6] M. Dissanayake, P. Newman, S. Clark, HF Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, 2001.

[7] C. Estrada, J. Neira, and J.D. Tardos. Hierarchical SLAM: Real-Time Accurate Mapping of Large Environments. *IEEE Transactions on Robotics*, 21(4):588–596, 2005.

[8] G.H. Golub, L. Van, and F. Charles. *Matrix Computations*. Johns Hopkins University Press, 1996.

[9] J.S. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 318–325, 1999.

[10] DG Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, 1999.

[11] F. Lu and E. Milios. Globally Consistent Range Scan Alignment for Environment Mapping. *Autonomous Robots*, 4(4):333–349, 1997.

[12] B. Mohar. The Laplacian spectrum of graphs. *Graph Theory, Combinatorics, and Applications*, 2:871–898, 1991.

[13] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. *Proceedings of the AAAI National Conference on Artificial Intelligence*, pages 593–598, 2002.

[14] F.A. Moreno, J.L. Blanco, and J. Gonzalez. A probabilistic observation model for stereo vision systems: Application to particle filter-based mapping and localization. In *Pattern Recognition and Image Analysis*, volume 4477 of *LNCS*, pages 346–353. Springer, 2007.

[15] M.A. Paskin. Thin Junction Tree Filters for Simultaneous Localization and Mapping. *Computer*, 2002.

[16] JM Porta and BJA Kröse. Appearance-based concurrent map building and localization. *Robotics and Autonomous Systems*, 54(2):159–164, 2006.

[17] PE Rybski, SI Roumeliotis, M. Gini, and N. Papanikolopoulos. Appearance-based minimalistic metric SLAM. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 194–199, 2003.

[18] J. Shi and J. Malik. Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[19] R. Smith, M. Self, and P. Cheeseman. A stochastic map for uncertain spatial relationships. *The fourth international symposium on Robotics Research*, pages 467–474, 1988.

[20] T. Sogo, H. Ishiguro, and T. Ishida. Acquisition and propagation of spatial constraints based onqualitative

information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):268–278, 2001.

[21] J.D. Tardos, J. Neira, P.M. Newman, and J.J. Leonard. Robust Mapping and Localization in Indoor Environments Using Sonar Data. *The International Journal of Robotics Research*, 21(4):311–330, 2002.

[22] S. Thrun and A. Bucken. Integrating grid-based and topological maps for mobile robot navigation. *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, pages 944–950, 1996.

[23] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press, September 2005.

[24] S. Thrun, Y. Liu, D. Koller, A.Y. Ng, Z. Ghahramani, and H. Durrant-Whyte. Simultaneous Localization and Mapping with Sparse Extended Information Filters. *The International Journal of Robotics Research*, 23(7-8):693, 2004.

[25] O. Veksler. Image segmentation by nested cuts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 339–344, 2000.

[26] M.R. Walter, R.M. Eustice, and J.J. Leonard. Exactly Sparse Extended Information Filters for Feature-based SLAM. *International Journal of Robotic Research*, 26(4):335–359, 2007.

[27] Z. Zivkovic, B. Bakker, and B. Krose. Hierarchical map building using visual landmarks and geometric constraints. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2480–2485, 2005.

[28] Z. Zivkovic, B. Bakker, and B. Krose. Hierarchical map building and planning based on graph partitioning. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 803–809, 2006.