

Data Association for Mobile Robot Navigation: A Graph Theoretic Approach

T. Bailey, E. M. Nebot, J. K. Rosenblatt and H. F. Durrant-Whyte

Australian Centre for Field Robotics
Department of Mechanical and Mechatronic Engineering, J07
University of Sydney, NSW 2006, Australia
{tbailey,nebot,julio,hugh}@mech.eng.usyd.edu.au

Abstract

Data association is the process of relating features observed in the environment to features viewed previously or to features in a map. Correct feature association is essential for mobile robot navigation as it allows the robot to determine its location relative to the features it observes.

This paper presents a graph theoretic method that is applicable to data association problems where the features are observed via a batch process. Batch observations (e.g., scanning laser, radar, video) detect a set of features simultaneously or with sufficiently small temporal difference that, with motion compensation, the features can be represented with precise relative coordinates.

This data association method is described in the context of two possible navigation applications: metric map building with simultaneous localisation, and topological map based localisation.

Experimental results are presented using an indoor mobile robot with a 2D scanning laser sensor. Given two scans from different unknown locations, the features common to both scans are mapped to each other and the relative change in pose (position and orientation) of the vehicle between the two scans is obtained.

1 Introduction

This paper presents a method for robust data association between two batch observations. A batch observation is defined here as a perception of the environment wherein the extracted set of features have accurately known relative geometry. For example, features extracted from a radar sweep, with appropriate motion compensation for any movement of the radar unit during the sweep, can be considered as a batch observation.

The basic philosophy of this data association method is as follows. If static features exist that are common to two batch observations, then the relative geometry between those features must be the same in both observations. It is assumed that in each observation there will be noise and dynamic obstacles, which will result in false features. There will also be valid features that are not common to both observations due to their different viewpoints. The aim, therefore, is to find a one-to-one

mapping of the features common to both batch observations. This is done by selecting the largest subset of features where the geometric constraints between features are mutually satisfied.

The problem of mapping the common features from two batch observations is equivalent to the graph theoretic problem of finding the Maximum Common Subgraph (MCS) between two graphs. In this paper, we transform each batch observation into a graph by equating features to graph vertices and the geometric relationships between features to graph edges. It is then a matter of solving the MCS problem to obtain the optimal common feature mapping.

The experimental data used in this paper was obtained from a 2D scanning range laser in an indoor environment. The feature types that were extracted were points and lines, and the relationships between these features were distances and angles. The graph-based representation used in this paper is not limited to these types of features, however, and one of the advantages of this method is that it is extendable to generic feature types and their invariant inter-feature relationships. It is also a simple matter to extend this method for use with a 3D sensor.

This paper is organised as follows. Section 2 discusses two possible navigation applications for this technique. Section 3 describes the vehicle and sensors used to collect experimental data. The next section describes the feature extraction methods used to generate static features from each laser scan. Section 5 presents the MCS algorithm, which is the core of this paper. Section 6 shows the results of the experimental data and section 7 discusses methods for detecting false data associations. The final section makes concluding remarks on the reliability of the matching process as demonstrated by the experimental data.

2 Navigation Applications

One application for this technique is *feature-tracking* which, for example, is required for building metric maps while simultaneously localising the robot from the map [9]. Presently feature observations are treated as isolated events and are mapped to a particular tracked feature if the location of the observed feature is sufficiently close to the predicted location of the tracked feature.

Validation of this data association is determined by the error between predicted and observed locations using a threshold bound or *observation gate*. The size of this gate is determined as a function of the tracked feature covariance and vehicle pose covariance – typically within a Kalman Filter framework. Any observation that falls outside of a tracked feature's observation gate is rejected as an observation of that feature [10]. In the case where an observed feature falls within the observation gate of more than one tracked feature, the observed feature must either be rejected entirely or some more sophisticated method, such as Multiple Hypothesis Tracking [6], may be used to maintain more than one possible data association.

Problems with the current feature tracking methods are mainly due to the difficulty in ensuring one-to-one feature mappings and to the propagation of the vehicle location error to the observation error. A tight observation gate is required to reject spurious observations but this, in turn, requires very stringent accuracy of the vehicle location. Therefore, if the vehicle pose estimate drifts too far from its true pose, feature tracking may fail through false data association or restricted true data association, thus resulting in localisation failure.

The graph-based data association method presented here has several advantages for the feature tracking problem. First, the matching process depends only on the relative geometry of features observed in a batch and is therefore independent to vehicle pose errors. Second, because a feature must conform to the geometric constraints imposed by *all* the other features, the individual error thresholds can be more relaxed. Finally, as feature association is performed externally to the (Kalman) filter framework, the filter can simply use the results of this algorithm to update the corresponding tracked feature estimates without having to deal with data association.

A second application for this data association method is for *distinctive place* recognition for *topological maps* [13]. A distinctive place is a location in the environment that is distinguishable from other places on the basis of patterns observable in sensory data. A topological map is a graph-based description of the environment where each node is a distinctive place, and the connecting edges between nodes contain procedural information that will enable the vehicle to travel from one node to the next.

The primary weakness of topological map representations has been reliable distinctive place recognition in all but the most structured of environments. Most methods to date are based on simplistic models of common indoor features such as doorways and corridor intersections [2][12][13]. A more robust method is to describe a distinctive place by a batch observation taken at that place. Subsequent observations can then use the data association method described here to determine the vehicle's pose with respect to that distinctive place. More detail regarding using this data association concept for topological maps can be found in a previous paper [3].

3 Test Vehicle and Sensors

The data used in this paper was logged from an indoor scanning laser mounted on a three-wheeled mobile robot – SydNav, shown in Figure 1. SydNav is driven and steered from the front wheel only. The two rear wheels are fixed facing forwards and can rotate freely. The front wheel has very accurate encoder data for steering and drive. There are, however, quite substantial biases in the dead reckoning estimate due to relatively coarse approximations of the centre location of the three wheels, the front wheel radius and the vehicle wheelbase.

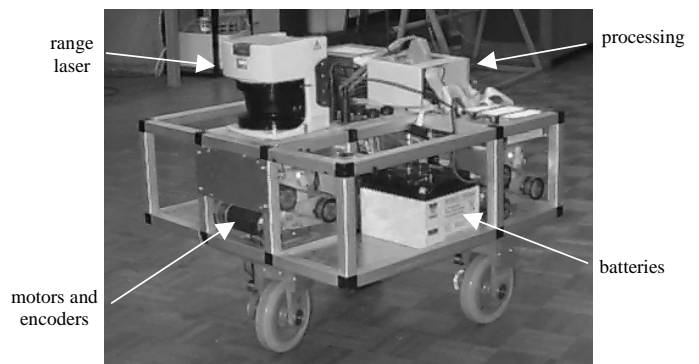


Figure 1: The SydNav mobile robot.

The scanning laser produces a 2D scan over 180° with a resolution of 0.5°. It has a maximum range of about 30m. A complete scan from the laser can be obtained at a rate of 2Hz on a 19200 baud serial link.

4 Feature Extraction

Feature extraction is performed on each individual laser scan, with the assumption that the relative geometry between features is fixed (i.e., the scan is an instantaneous snapshot of the environment). This is reasonable as the scan sweep takes about 0.04s and the indoor robot moves at less than 0.5m/s. For higher speed vehicles, some motion compensation would be required to obtain accurate relative geometry between features.

Presently, the feature extraction algorithms are very basic but they have been sufficient to produce reasonably stable features from indoor laser data. The first step in feature extraction is to cluster the range data. This is performed sequentially starting from the first range measurement in the scan. Each range measurement, R_i , is compared with the next adjacent range measurement, R_{i+1} , as shown below:

$$\begin{aligned} \Delta R &= \text{abs}(R_i - R_{i+1}) \\ \Delta R_{\text{max}} &= C1 + C2 * \min\{R_i, R_{i+1}\} \\ \text{if } \Delta R < \Delta R_{\text{max}} &\rightarrow \text{add to cluster} \\ \text{else } &\rightarrow \text{start new cluster} \end{aligned}$$

ΔR_{max} is the maximum distance two consecutive range measurements may vary if they are to be considered part of the same cluster. The constants C1 and C2 are tunable to the noise and resolution characteristics of the laser.

They were set to 0.07m and 0.04m/m, respectively, for the experimentation in this paper.

After clustering, three types of features are extracted: foreground edges, foreground points and lines (see Figure 2). Foreground edges and points are used because they remain stable in the presence of occlusions. They are obtained by checking the two edge points of each cluster and comparing them with the edge points of the two adjacent clusters. If *either* edge point has a shorter range than the edge on the adjacent cluster, then that point is classified as a foreground edge. If *both* edge points have shorter ranges than the edges of the adjacent clusters, and the cluster is small, then that cluster is a foreground point.

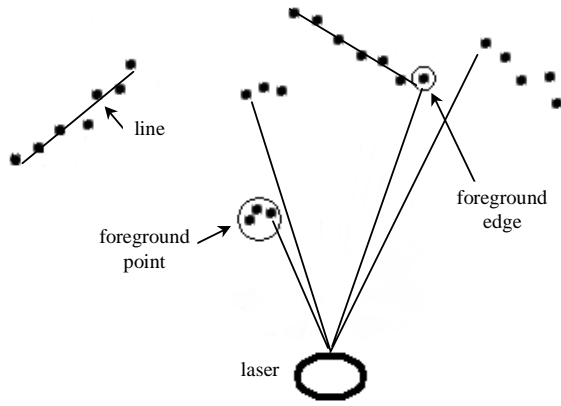


Figure 2: Feature extraction from a laser scan.

Lines are determined from the data points in each cluster using a bisection method as follows. The two end points of a cluster are connected by a line. Each point in the cluster is then tested for distance from this line. If the point of greatest distance from the line is sufficiently far away, the cluster is split at that point and the process is repeated. If, however, the point of greatest distance from the line is reasonably close to the line, a line is fitted to the valid point set using a least-squares based line-fitting algorithm, such as in [11].

5 Maximum Common Subgraph Algorithm

The features extracted from a batch observation can be represented as a graph where each node is a feature and each edge is an invariant relationship between two features. The data association problem (i.e., finding a one-to-one mapping of common features between two observations) can then be transformed to the problem of finding the Maximum Common Subgraph (MCS) between two graphs. Edges must be defined for every pair of features in a graph, so that every node is connected to every other node (i.e., a complete graph); thus the MCS is equivalent to finding all the nodes that are common to both graphs.

Some methods that have been used for finding the MCS include backtrack search [14], neural networks [15], and maximum clique [8]. The maximum clique based method has been implemented in this paper. A clique is defined as a complete subgraph within a graph (i.e., each

node is connected to every other node in the subgraph). The maximum clique is the largest of these complete subgraphs.

The maximum clique approach to finding the MCS is a three-step process described in the sections below. The first is to generate the two feature graphs for the laser scan data. The second step involves generating a correspondence graph between the two feature graphs. Finally, the maximum clique of the correspondence graph is found which, in turn, gives the mapping of common features from the two laser scans.

5.1 Feature Graph Generation

After feature extraction, each laser scan is described by a set of points and lines in a laser-centred coordinate frame. A feature graph is generated by defining each feature as a node and the geometric relationship between two features as an edge. The conversion of this geometric information to a graph representation is as follows:

- | | |
|--------|--|
| Nodes: | P = point |
| | L = line |
| Edges: | P-P \rightarrow distance |
| | P-L \rightarrow perpendicular distance |
| | L-L \rightarrow acute angle magnitude |

An example geometric representation of node and edge labeling is shown in Figure 3 and the equivalent graph representation is shown in Figure 4. Only a few of the edge labels are shown here to reduce clutter in the diagrams.

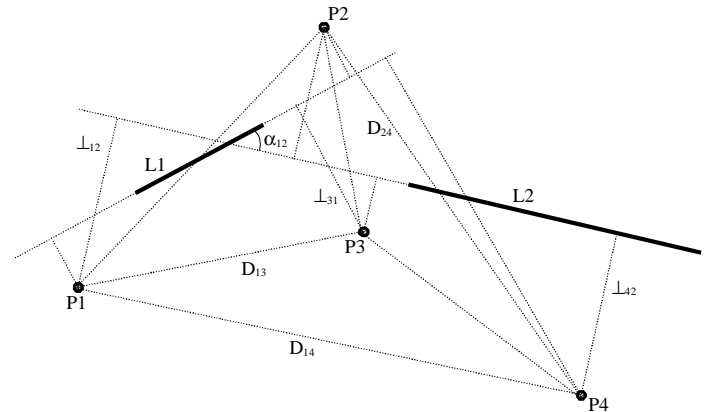


Figure 3 Geometric representation of features.

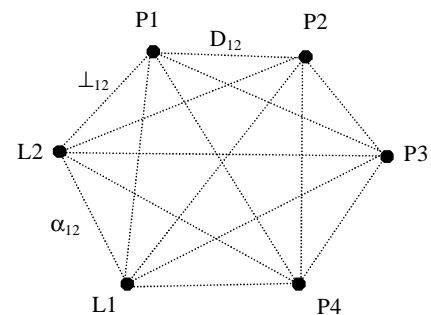


Figure 4 Graph representation of features.

5.2 Correspondence Graph Generation

Having generated two feature graphs, with their node and edge labels, the creation of a correspondence graph can be considered at a level of abstraction beyond the laser scan matching problem, and is valid for any two labeled undirected graphs. Figure 5 shows two graphs with node labels {I, J, K} and edge labels {a, b, c, ...}, and the resulting correspondence graph is shown in Figure 6.

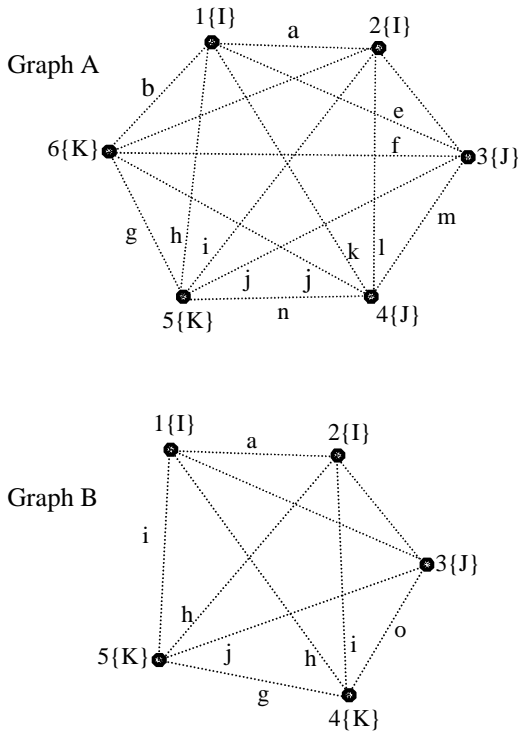


Figure 5 Feature graphs A & B.

The method for defining a correspondence graph is based on [4] (also described in [8]). First, create the correspondence graph nodes from all the possible pairings of same-labeled nodes from the two graphs. For example, for the node type I, there is {A1,B1}, {A1,B2}, {A2,B1}, and {A2,B2}. Next, look for edge matches between the two graphs. If the edge connects two same-labeled nodes (e.g., 'a' connects I and I) then this symmetry results in two edges being added to the correspondence graph (e.g., {A1,B1} to {A2,B2} and {A1,B2} to {A2,B1}). If, however, the edge connects two different types of node label (e.g., 'e' connects I and J) then only one edge is added to the correspondence graph (e.g., {A1,B2} to {A3,B3}).

Essentially, an edge in the correspondence graph implies the possibility of a mapping between the two sets of points described at the nodes it connects. By finding the maximum clique of this graph, the best point mapping can be found.

5.3 Maximum Clique Search

Finding the maximum clique in an arbitrary undirected graph is known to be an NP complete problem. A computationally tractable solution is attainable, however, because the correspondence graph is generally sparse and there exists efficient exact or approximate methods for finding the maximum clique when the search space is not too large [5] [7] [8].

The algorithm used in this paper is an exact method based on [7]. The maximum clique for our example correspondence graph is shown in Figure 7. This result means that feature 1 of graph A maps to feature 2 of graph B (i.e., {A1, B2}) and so forth (i.e., {A2,B1}, {A3,B3}, and {A5,B5}).

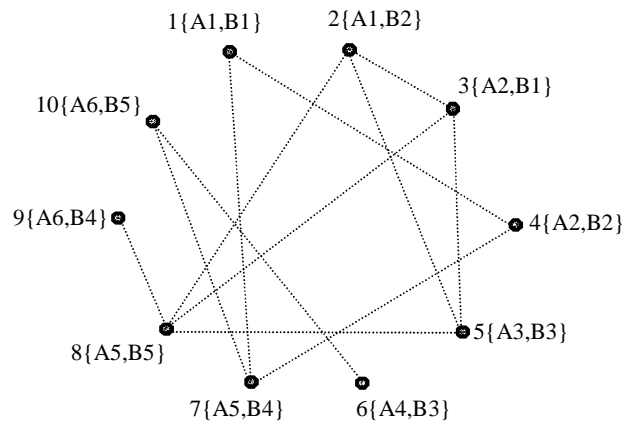


Figure 6 Correspondence graph of feature graphs A & B.

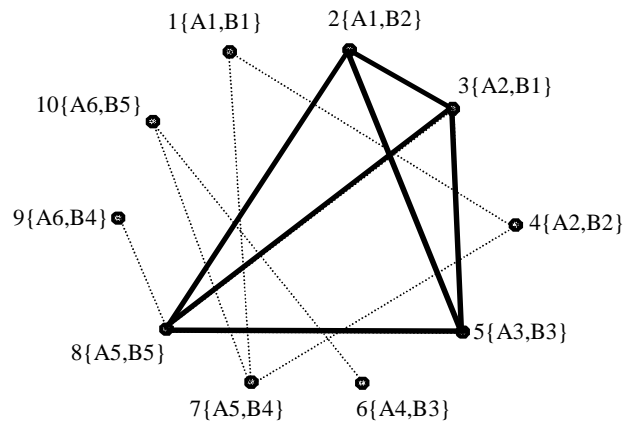


Figure 7 Maximal clique of correspondence graph.

6 Experimental Results

Experimental results were based on data logged from the 2D scanning range laser on board the indoor mobile robot, SydNav. Shown below (see Figures 8 and 9) are two laser scans taken from different locations in a region of a static corridor environment. These scans are representative of this particular environment (with a more complete description of this area being presented in [3]). The location of the vehicle in these scans is depicted

by a triangle with the right-most point designating the front.

The two scans were converted to feature graphs with point and line node labels as described above. The correspondence graph was generated with some variance allowed in the edge matching process. These variances were $\pm 30\text{cm}$ for distances between two points, $\pm 40\text{cm}$ for perpendicular distances between a point and a line, and ± 0.1 radians for the acute angle magnitude between two lines. The maximum clique was then found using the algorithm referenced above.

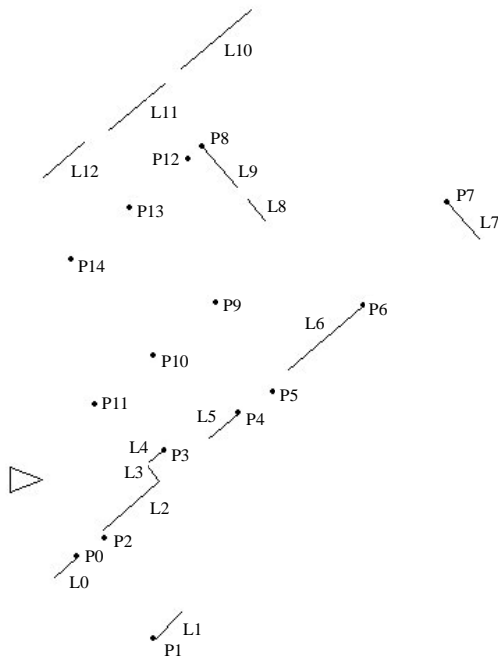


Figure 8 Features extracted from scan A.

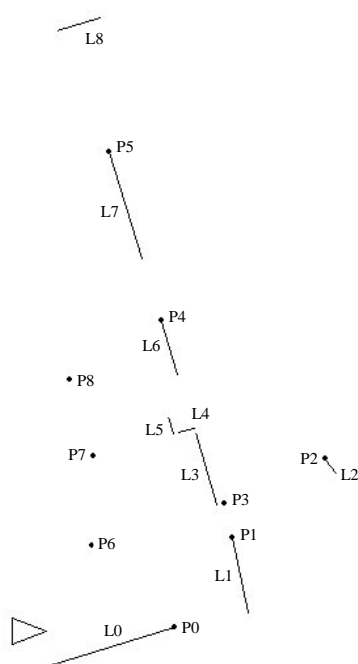


Figure 9 Features extracted from scan B.

Having found a one-to-one mapping of the features common to both scans, the coordinate transform between the two scans can then be calculated. This is done by first finding the median rotation between the two scans, rotating the second scan by this amount, and then calculating the median translation required so that corresponding features overlay each other. An alternative method, which describes the calculation of a least-squares transformation between two point sets, can be found in [1].

Results of data association between the two example scans can be seen in Figure 10, with the feature mappings given in Table 1. This mapping meant that the pose of the vehicle in Scan B with respect to the vehicle pose in Scan A was found to be:

$$(x, y, \phi) = (-2.85\text{m}, 1.24\text{m}, -1.16\text{rad}).$$

It should be noted that, because line features were modeled here as infinite lines, mappings between collinear line segments were not properly defined or constrained. This is easily rectified by merging collinear lines during the feature extraction process but, in any case, a valid mapping is always obtained so this poses no significant problems.

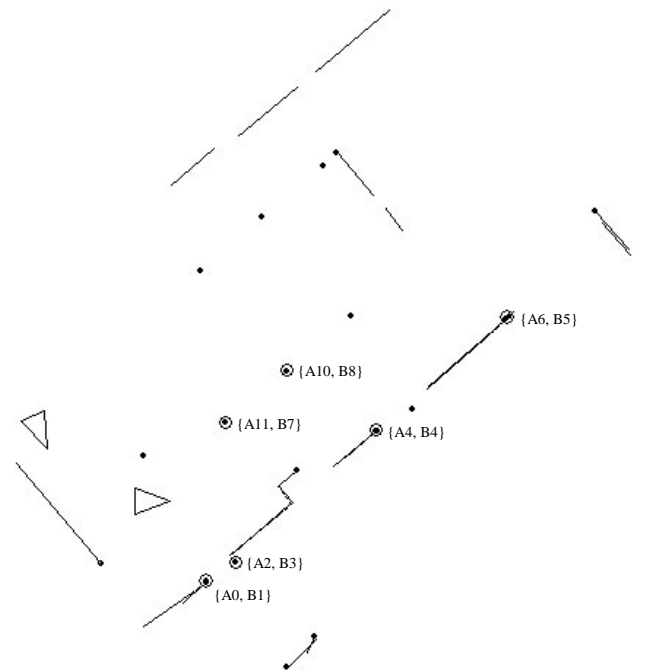


Figure 10 Scan B transformed and overlaid on scan A.

Points	Lines
{A0, B1}	{A0, B3}
{A2, B3}	{A2, B6}
{A4, B4}	{A3, B4}
{A6, B5}	{A4, B5}
{A10, B8}	{A5, B7}
{A11, B7}	{A7, B8}

Table 1 Mapping between the two feature graphs.

7 Error Detection

The only avenue for false mappings when using this data association technique arises from the existence of feature symmetries in the environment. That is, it is possible for a set of features to be aligned such that they coincidentally conform to the geometric constraints of a different set of features. The probability of this occurring decreases rapidly with the number of features mapped between the two scans [3]. Generally false mappings are detectable as they tend to result in a significant discrepancy between the calculated vehicle pose and the predicted vehicle pose. This discrepancy may be detected using an additional pose estimating method (such as odometry) or by checking the consistency of a particular scan match with other scan matches.

When validating a particular scan mapping, the size of the error threshold can be based on the precision of the pose estimate available from the set of mapped features (i.e., the variance of the vehicle pose estimate due to the geometric configuration of the features). This variance calculation is similar to the degree of precision available from a particular constellation of satellites with GPS navigation [16].

8 Conclusion

This paper presents a method for performing data association. This method is valid when features are observed as a batch observation such that they have accurate relative geometric information. The mapping of common features between two feature sets is transformed into the graph theoretic problem of finding the maximum common subgraph (MCS) which, in turn, can be represented as a maximum clique problem. There exists substantial literature on efficient methods for solving the maximum clique problem so that it is a viable method for real-time navigation systems.

Experimental results using the MCS method showed that, given two laser scan observations, the largest subset of features common to both scans could be obtained. From these mapped features, it was simply a matter of transforming the scans so that one overlaid the other to find the relative change in pose of the vehicle between the two observations.

The nature of the inter-feature geometric constraints was such that the only possibility of false mappings was from environmental symmetry. Mappings due to environmental symmetry were generally detectable by large discrepancies in the vehicle pose estimate.

A further advantage of the MCS approach is that it can be extended generically to include different feature types, including those extracted from other sensors, provided that they have some invariant relational properties to all the other feature types. Similarly, it is trivial to extend this method to a 3D sensor, as the graph representation simply requires feature types and the relationships between them.

References

- [1] Arun K. S., Huang T. S., Blostein S. D., "Least Squares Fitting of Two 3-D Point Sets", *IEEE Transactions on PAMI*, vol. 9, no. 5, pp. 698-700, 1987.
- [2] Aycard O., Charpillet F., Fohr D., Mari J. F., "Place Learning and Recognition using Hidden Markov Models", *Proceedings of the 1997 IEEE International Conference on Intelligent Robots and Systems*, pp. 1741-1746, 1997.
- [3] Bailey T., Nebot E. M., Rosenblatt J. K., Durrant-Whyte H. F., "Robust Distinctive Place Recognition for Topological Maps", *Proceedings of the 1999 International Conference on Field and Service Robotics*, pp. 347-352, 1999.
- [4] Barrow H.G., Burstall R.M., "Subgraph Isomorphism, Matching Relational Structures and Maximal Cliques", *Information Processing Letters*, vol. 4, no. 4, pp. 83-84, 1976.
- [5] Battiti R., Protasi M., "Reactive Local Search for Maximum Clique", *Proceedings of the Workshop on Algorithm Engineering*, pp. 74-82, 1997.
- [6] Blackman S. S., *Multiple-Target Tracking with Radar Application*, Artech House, Norwood, MA, pp. 83-107, 1986.
- [7] Bron C., Kerbosch J., "Algorithm 457 – Finding all Cliques of an Undirected Graph", *Communications of the ACM*, vol. 16, pp. 575-577, 1973.
- [8] Chen C. K., Yun D. Y. Y., "Unifying Graph Matching Problems with a Practical Solution", *International Conference on Systems, Signals, Control, Computers*, Sept. 1998.
- [9] Dissanayake M. W. M. G., Newman P., Durrant-Whyte H. F., Clarke S., Csorba M., "An Experimental and Theoretical Investigation into Simultaneous Localisation and Map Building", *6th International Symposium on Experimental Robotics*, pp. 171-180, March 1999.
- [10] Durrant-Whyte H. F., "An Autonomous Guided Vehicle for Cargo Handling Applications", *International Journal of Robotics Research*, pp. 407-441, 1996.
- [11] Kahn P., Kitchen L., Riseman E. M., "A Fast Line Finder For Vision-Guided Robot Navigation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 11, pp. 1098-1102, November 1990.
- [12] Kortenkamp D., Weymouth T., "Topological Mapping for Mobile Robots using a Combination of Sonar and Vision Sensing", *Proceedings of the 12th National Conference on Artificial Intelligence*, pp. 979-984, July 1994.
- [13] Kuipers B. J., Byun Y. T., "A Robot Exploration and Mapping Strategy Based on a Semantic Hierarchy of Spatial Representations", *Journal of Robotics and Autonomous Systems*, pp. 47-63, 1991.
- [14] McGregor J. J., "Backtrack Search Algorithms and the Maximal Common Subgraph Problem", *Software-Practice and Experience*, vol. 12, pp. 23-34, 1982.
- [15] Shoukry A., Aboutabl M., "Neural Network Approach for Solving the Maximal Common Subgraph Problem", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 26, no. 5, pp. 785-790, 1996.
- [16] Spilker J. J., "Satellite Constellation and Geometric Dilution of Precision", *Progress in Astronautics and Aeronautics*, vol. 163, pp. 177-208, 1996.