

# One-to-one Feature Matching with Inaccurate Maps

Chatavut Viriyasuthee and Gregory Dudek

Centre for Intelligent Machines

McGill University

Montreal, Quebec, Canada H3A 2A7

{pvirie,dudek}@cim.mcgill.ca

**Abstract**—In the problems of localization using inaccurate maps, navigation agents have to match available information from sensors to maps in order to find their locations. A map contains a set of constraints that can be expressed in the form of a graphical model that matching algorithm has to satisfy. There are two generally categories of constraints: absolute and relative. We propose a relaxation-based algorithm for the NP-hard problem of one-to-one feature matching with absolute and relative constraints. The algorithm is a combination between relaxation labeling and the Kuhn-Munkres method where the former is known for its highly parallel structure imitated the human visual process. To test the performance, we applied the algorithm in a robotics application where the objective is to match range scanner features to those in inaccurate template maps provided by humans. Our experiments show that the proposed algorithm can achieve qualified matching results in artificial and real situations.

## I. INTRODUCTION

In this paper, we are interested in matching problem between inaccurate scenes that involves distortion and absence of some features. This is a sub-problem of navigation using inaccurate maps framework [1], [2] where the goal is to localize agents from environments into human-provided maps in order to retrieve proper actions. The human-provided maps are very convenient to create, for example from drawing, but they suffer from abstraction and from lacking of accuracy. These characteristics may hinder the task of localization into this type of maps. Examples of this idea are presented by Skubic et al.[2], [3] where they aim to use a hand-drawn map for controlling a robot. However, their assumptions may not support certain types of scenes, such as ones that are noisy and filled with redundant features. We propose an algorithm to address a more general version of the matching problem.

In the scope of matching, many approaches across various fields have been explored: optimization, algorithmic methods, computer vision, and machine learning [4]–[6]. Currently, representing and solving the matching problem using graphs has been particularly an interesting topic [7]–[9] especially when dealing with certain types of graphs, such as chains and trees, where matching can be done efficiently. A more general version of graph matching is the inference problem in graphical models [10]. While the exact inference algorithms can provide the optimal results, they can either only apply in certain cases of graphical models or suffer from intractable computation time. Approximate inference algorithms, such as Monte Carlo and Variational methods, have been proposed as

alternatives and have been proven successful in real applications [11]. However, if there is such a requirement of the one-to-one matching constraint, which seems to be general enough to appear in many matching situations, this constraint makes the inference problems become harder because it relates to every node in the graphs creating fully connected models.

We address the problem of one-to-one feature matching between a world and a given template map. The likelihood of a world feature to be matched with a template feature is defined by the similarity factor between the two features and also the support factors from potentially assigned neighbors. We call these factors absolute constraints and relative constraints respectively. This work proposes an algorithm for the matching problem based on both types of constraints that are provided in the map in form of a graph. It is based on relaxation labeling (RL) by Hummel and Zucker [12] and Kuhn-Munkres algorithm (KM) [13]. RL is a well-established bio-inspired framework based on the highly parallel neural structure of the human visual system. It has been widely adopted in computer vision problems [14], [15] and sometimes appears in graphical model inference [16]. However, there is only a limited number of existing literature that explore the combination of RL and KM algorithm [17]. The details of our algorithm are explained in section III. We demonstrate the effectiveness of our algorithm in a robotics application that involves matching detected range scanner features (or world features) to human-provided features in given template maps. The experimental results are provided in section IV.

## II. PROBLEM STATEMENT

Given a set of features  $X$  in a map, we seek to find an injective correspondence of these features to a set of extracted world features  $Y$  that satisfies provided map constraints. We first define each type of constraints and discuss its benefits. Then, we formulate the matching problem based on these constraints and derivation from graphical model inference.

### A. Constraints

The simplest way to design a map is by independently defining matching preference from each map feature to each individual world feature. This kind of preference is what we refer to as absolute constraint. An absolute constraint is a function that computes matching eligibility of a map-world feature pair,  $A : X \times Y \rightarrow \mathbb{R}$ .

However, merely absolute constraints may not be suitable for some circumstances. For example, there may be several instances of a map feature type A in the world, but we wish to match A to one of the world instances that is nearby an instance of feature B. Grouping the two features to create a new unique feature is an ad-hoc way to solve this problem that we try to avoid. Consequently, we turn to relative constraints, which we define here to allow more flexibility when designing a map. A relative constraint is a function that computes matching eligibility based on neighbor assignments,  $R : X^{n+1} \times Y^{n+1} \rightarrow \mathbb{R}$  where  $n$  is the size of influential neighborhood. It is worth noting that the one-to-one constraint can also be regarded as a relative constraint that covers the entire set of features.

Although using only absolute constraints in a map has the mentioned weakness, it has been known that

*Lemma 2.1:* one-to-one matching with only absolute constraints is in the polynomial-time complexity class because it can be solved efficiently using the KM algorithm [13].

### B. Matching problem

We can create a map using a graphical model where each node belongs to a map feature and the relations between nodes (or edges) are represented by relative constraints. The matching problem itself can be regarded as the inference problem where the goal is to properly assign each node with a world feature. This problem is NP-hard in nature. In this section, we reformulate the inference problem in terms of optimization such that it can be used to derive a proper inference algorithm (which will be discussed in the next section).

Every graphical model is defined by a joint distribution of its nodes, which can be factorized into a product of local distributions  $\phi$  [11] if we temporarily disregard the one-to-one constraint. From the a joint probability  $P(\mathbf{x} = \mathbf{y})$ , let  $\mathbf{x}$  and  $\mathbf{y}$  present the lists<sup>1</sup> of map features and assigned world features respectively, we present a transformation using the following steps:

$$\begin{aligned} \max_{\mathbf{y}} P(\mathbf{x}) &\equiv \max_{\mathbf{y}} \prod_C \phi_C(\mathbf{x}_C = \mathbf{y}_C) \\ &\equiv \max_{\mathbf{q}} \sum_{C, \mathbf{y}_C} \log \phi_C(\mathbf{x}_C = \mathbf{y}_C) \prod_{\forall x_i \in \mathbf{x}_C} q(x_i = y_i) \end{aligned}$$

where  $\mathbf{x}_C$  and  $\mathbf{y}_C$  are the node list in a local group  $C$  and its current assignment respectively. We first apply logarithm to turn product into summation. Then we sum over every possible assignment and introduce indicators  $q$  (and their vector container  $\mathbf{q}$ ) as a way to choose specific assignments that have been considered. The indicators  $q$  have the following properties:

$$q(x = y) \in \{0, 1\}; \quad \sum_y q(x = y) = 1; \quad \forall x \quad (1)$$

where  $q(x = y) = 1$  if we assign node  $x$  with  $y$ .

<sup>1</sup>lists are sets with arranged elements.

In case that all relative constraints involve only two adjacency nodes  $x$  and  $\tilde{x}$ , let us define  $\log \phi_C(\mathbf{x}_C) = R(x, \tilde{x}) + A(x) + R(\tilde{x}, x) + A(\tilde{x})$  using both relative and absolute constraints.<sup>2</sup> This form allows us to bridge the graphical model and the formulation described below. Furthermore, we change our focus from summation over all local groups to all individual nodes instead. The transformation continues as follows:

$$\begin{aligned} &\equiv \max_{\mathbf{q}} \sum_{x, y} \sum_{\tilde{x}, \tilde{y}} (R(x, \tilde{x}) + A(x)) q(x = y) q(\tilde{x} = \tilde{y}) \\ &\equiv \max_{\mathbf{q}} \sum_{x, y} \left( \sum_{\tilde{x}, \tilde{y}} R(x, \tilde{x}) q(\tilde{x} = \tilde{y}) + |\tilde{x}| A(x) \right) q(x = y) \end{aligned} \quad (2)$$

where  $\tilde{x}$  is any node that shares a group with  $x$  and  $|\tilde{x}|$  is the total number of such groups.

Finally, we wish to find values of the indicators such that they maximize objective (2) with respect to the one-to-one constraint  $\sum_x q(x) = 1$  and properties (1). This problem is in the category of the quadratic assignment problem [19] and 0-1 quadratic programming where the complexity of solving for the optimal solution is NP-hard. However, it can be seen that if we treat the inner terms of objective (2) as new absolute constraints  $\dot{A}(x = y, \mathbf{q})$ , the objective becomes

$$\max_{\mathbf{q}} \sum_{x, y} \dot{A}(x = y, \mathbf{q}) q(x = y) \quad (3)$$

which is a linear summation of the indicator values where the optimal result can be computed in polynomial time according to Lemma 2.1. Next section will discuss an algorithm that utilizes this idea.

## III. METHODOLOGY

Solving the one-to-one matching problem that involves relative constraints is generally hard. Alternatively, we can convert all relative constraints into absolute ones and then perform the assignment on this new constraint set using KM algorithm. The whole idea of constraint conversion conforms to the concept of maximizing average local consistency in the relaxation labeling framework. It can be used to convert objective (2) into (3). In this section, we discuss the approach in detail.

### A. Relaxation Labeling and Kuhn-Munkres method

The objective (3) is in an inner product form of the new absolute constraint and the indicator vector. The main concept of maximizing average local consistency in RL is to seek a value for each individual  $q$  such that the two vectors are aligned thus yielding the maximum result. We start the conversion task by first disregarding the first constraint in (1)

<sup>2</sup>For appearances' sake, let any individual  $x$  inside functional quantities stand for  $x = y$  for some corresponding  $y$ .

Similar to many other approaches in matching, we assume that there is no high order constraint that has more than two nodes in the maps [4], [18]. Although the same concept can be used to transform high order constraints with slight changes in the structure of the final objective.

to allow  $q$  to have any value within range  $[0, 1]$ ; following this scheme is the concept of continuous RL.

There are several implementations of continuous RL [18]. For simplicity, we adopt the radial projection update rule from [20]. Let  $n$  be the index of RL's iterations, for every map feature  $x$  and every world feature  $y$

$$\begin{aligned}\ddot{A}(x = y, \mathbf{q}_n) &= \dot{A}(x = y, \mathbf{q}_n) - \min_{y'} \dot{A}(x = y', \mathbf{q}_n) \\ q_{n+1}(x = y) &= \frac{q_n(x = y) + \alpha \ddot{A}(x = y, \mathbf{q}_n)}{1 + \alpha \sum_{y'} \ddot{A}(x = y', \mathbf{q}_n)}\end{aligned}\quad (4)$$

This rule can be regarded as a version of gradient ascending where  $\alpha \in \mathbb{R}$  is used to control the rate of convergence. A proper starting point for  $\mathbf{q}_0$  is the set of probabilities that corresponds to the normalization of the original absolute constraints. It is important that we first subtract the minimum of all possible assignments from every new absolute constraint to avoid the rule from producing negative probabilities.

However, using continuous version of the indicator poses a new problem because any assignment can be tied with a small indicator value, and we use summation over all possible assignments of all surrounding nodes despite that we in fact want to assign only one world correspondence to each neighbor node. With respect to this contradiction, we redefine the new absolute constraints as follows:

$$\dot{A}(x = y, \mathbf{q}) = \sum_{\tilde{x}} R(x = y, \tilde{x} = \tilde{y}_{\max}) + |\tilde{x}|A(x = y) \quad (5)$$

Where  $\tilde{y}_{\max} = \arg \max_{\tilde{y}} R(x = y, \tilde{x} = \tilde{y})q(\tilde{x} = \tilde{y})$

This setting can mitigate the effect of *multiple supports* by selecting only a relative constraint from the most possible assignment of each neighbor instead of one that is strong from receiving supports from many weak assignments.

At the end of RL iterations, we reach objective (3) where the optimal one-to-one matching can be given by the KM algorithm. In order to address missing features in real circumstances, we augment the world space of KM with additional  $|X|$  dummy features. KM always chooses assign a map feature to a dummy if it can produce a better matching strength than a threshold  $\bar{A}_x$  defined as follows:

$$\bar{A}_x = \sum_{\tilde{x}} \bar{R}_{x, \tilde{x}} + |\tilde{x}| \bar{A}_x$$

where  $\bar{R}_{x, \tilde{x}}$  and  $\bar{A}_x$  are the threshold for classifying as missing of each relative and absolute constraint respectively. These thresholds must be provided into the process as parameters.

At the perfect situation where the indicator values  $q$  converge to the true optimal solution during the radial projection process, the newly converted absolute constraints must be equivalent to the ideal absolute constraints that perfectly encode the information of relative constraints; and therefore the results of KM must also be the optimal results for maximizing objective (2). However in general, there is no guarantee that solving relaxation labeling with radial projection can provide the optimal  $q$ ; and thus the results of KM might not correspond to the true optimal matching. Because of this, we additionally

present an improvement routine that helps enhance the quality of matching results.

### B. Improvement steps

Improvement steps are required to improve the matching quality as it may be spoiled during the RL process. The intuition behind these steps is analogous to the concept of molecular lattice formation in chemistry where we try to destroy weak bonds and reform stronger ones. It can be seen that each matching result from KM can be visualized using a graph by overlaying relative constraints on matched world features. The nodes of the graph are the world features and the edges are the relative constraints. We can treat the graph as a lattice structure; and by selectively splitting the graph and passing the result components into the matching process again, we acquire a better graph. The steps can be described in detail as follows:

- 1) Identify and remove the edges that are worse than their thresholds  $\bar{R}$ ; this may produce some disjoint components.
- 2) Remove assignments from all features in small components. The small components are the components that have the number of feature nodes less than a criterion  $\in \mathbb{N}$ ; this criterion has to be given as a parameter.
- 3) For all unassigned features that are adjacent to assigned features, let the relative constraints act as additional absolute constraints. This step is required to ensure that the new features are assigned correspondingly with the previous ones.
- 4) Repeat the matching process on the unassigned features.

The number of repetition can be given. Alternatively, the algorithm may run until there is no unassigned feature left or when all the new formed edges are worse than  $\bar{R}$ .

### C. Summary and Complexity Analysis

In summary, the algorithm can be described as follows:

- 1) Initialize the setting  $\mathbf{q}_0$  from the original absolute constraints.
- 2) Convert constraints using relation (5) and radial projection rule (4).
- 3) Perform one-to-one matching on objective (3) using KM algorithm.
- 4) Improve the output result and repeat the process from the second step.

The asymptotic running time of RL in a single thread process is  $O(N|X|^2|Y|^2)$  where  $N \in \mathbb{N}$  is the number of RL's iterations. The complexity of KM method with dummy features is bounded within  $O((|X| + |Y|)^3)$  [13]. The improvement steps can be achieved in  $O(|X|^2 \log(|X|))$  using a disjoint set; the  $|X|^2$  part comes from the maximum possible number of the relative constraints. Let  $K \in \mathbb{N}$  be the total number of repetition, the overall complexity of our algorithm is  $O\left(K\left(N|X|^2|Y|^2 + (|X| + |Y|)^3\right)\right)$ . However, because RL was originally designed to be highly parallel, we can utilize this characteristic of RL to help reduce computation time.

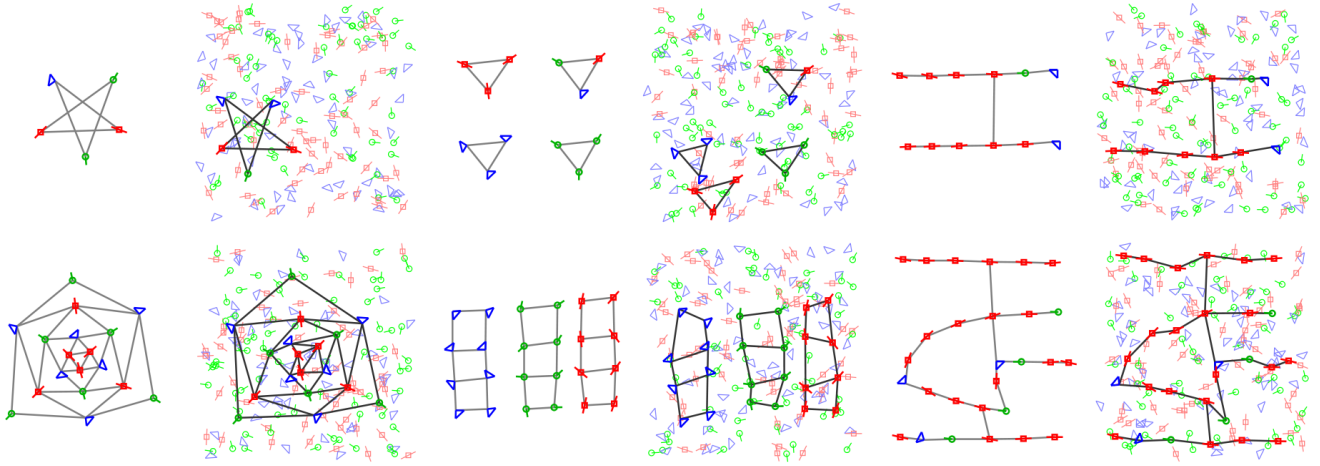


Fig. 1. The artificial maps and their matching results in the world data with 200 irrelevant features and seeds generated with  $0.2 \text{ m}^2$  position variance. Each pair contains a map (left) and its matching result overlays on the noisy world data (right).

#### IV. EXPERIMENTS

In this section, we show quantitative and qualitative performance of our algorithm in feature matching tasks from human-provided maps to artificial and real world datasets that have the characteristics of range scanner data.

We proposed three types of features including corner, line, and ending. Every feature had three properties which were type, location in  $\text{m}^2$ , and orientation in radian. Each absolute constraint was a likeness between a world and a map feature which is defined by an inner product of the orientation vectors from the features. But when the features were different in type, we penalized the likeness by a constant factor. A line and an ending are similar; therefore, the likeness was three quarters times the inner product. The factor for a line or an ending against a corner was one half, but the orientation of either the line or the ending was tested with the closest of the two lines, in angular distance, that form the corner. Furthermore, we normalized all absolute constraints by the size of its neighborhood to make them all equally important. Each relative constraint was defined by the negative of square distance between neighbors in the world and the map,  $R(x = y, \tilde{x} = \tilde{y}) = -||x_l - \tilde{x}_l - (y_l - \tilde{y}_l)||^2$  where the subscript  $l$  denote features' locations.

We used average relative constraint strength (ARCS) as the evaluation metric where the goal is to maximize it. The ARCS provides direct information for how well the relative constraints are formed and provides indirect information about the amount of features that have been classified as missing; in particular, when all features are missing, the ARCS must be equal to the constraint thresholds  $\bar{R}$ .

##### A. Artificial Dataset

In the first two experiments, we numerically evaluated the performance of our algorithm by comparing the result ARCS with statistical average. The statistical average was computed by matching map features to certain world features called *seeds*. The seeds were generated from map features with normally distributed positions and orientations around the original ones at different variance magnitudes. They were added

into the world feature sets and would be chosen when none other alternatives produce better matching results. For the first experiment, we varied the variance magnitude of the seeds' positions. In the second experiment, we varied the amount of irrelevant world features. The irrelevant world features were randomly generated features that had been inserted into the world apart from seeds and we introduced them into the experiment using irrelevant/relevant ratio where the number of relevant features was  $|X|$ .

We used six testing maps shown in Figure 1. For each map the minimum component criteria were 2,1,3,3,3,3 respectively. Each seed feature had orientation variance at 0.05 radians. The default value of position variance and irrelevant/relevant ratio are  $0.05\text{m}^2$  and 1.0 respectively. In each experiment, we fixed one value to its default and allowed another to change. Our algorithm involved 5 steps of radial projection with rate factor  $\alpha = 0.5$ . For each configuration, we run the experiment 50 times on each map and the results were reported on the average. Figure 3a and 3b show the results.

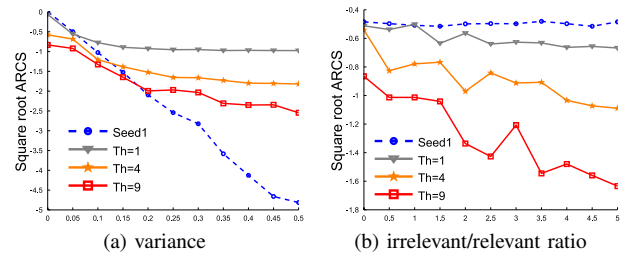


Fig. 3. Quantitative performance of our algorithm when compared with direct matching to the seeds across three constraint thresholds. The plots show square root magnitude of the ARCS against increasing a) seeds' variance b) irrelevant/relevant ratio.

Intuitively, changes in variance must have a direct influence toward the seed results; the higher the variance, the lower ARCS is produced from seed matching. Therefore, an optimal algorithm must perform as good as seed matching when the position variance is low and better when the variance increases. Figure 3a shows that our algorithm tends to follow this behavior. At low variance, ARCS from our algorithm followed

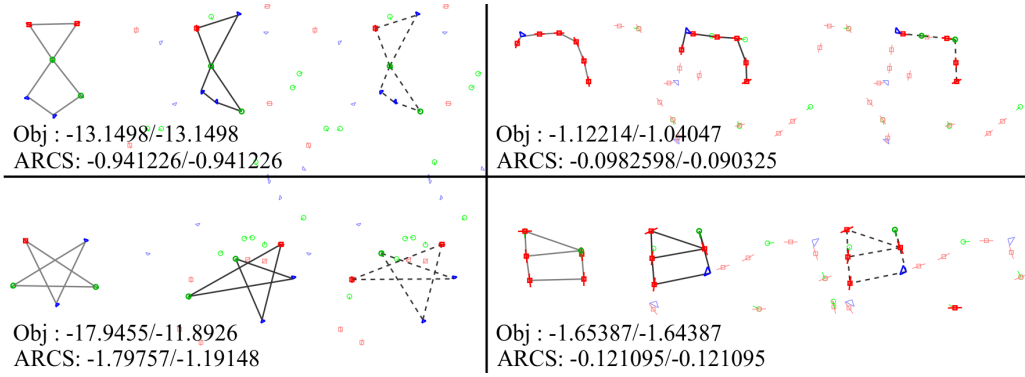


Fig. 2. Comparison between our algorithm's and optimal results; For each tuple from left to right, there are the map, the result from our algorithm, and the optimal result respectively. We also show the final ARCS and objective values from (our algorithm)/(optimal) in each map.

the seed's tightly or loosely depending on the threshold value. And when the variance was high, our algorithm could maintain ARCS and never allowed it to become lower than the threshold value. Since the threshold controlled the elimination process, lower threshold allowed only strong assignments to survive leading to higher ARCS closer to the seed's<sup>3</sup>.

The increasing irrelevant/relevant ratio reduced performance of our algorithm as opposed to the seed performance which was unaffected by both the ratio and the threshold because matching to seeds did not take them into account. To see where it ended, we solely tested matching performance in the extreme condition where each world set has 200 irrelevant features. The qualitative results are shown in Figure 1; it can be seen that our algorithm still managed to find proper correspondence between the map and the world domains.

Next, we show comparison between matching results of our algorithm and the optimal solutions from combinatorial search. We fixed constraint threshold to 4. The minimum component criteria for every map was 1. Again, our algorithm involved 5 iterations of both 0.5- $\alpha$  radial projection and improvement steps. Results are shown in Figure 2.

It can be seen our algorithm could produce close-to-optimal matching results and even the optimal results themselves sometimes. It is worth pointing out that our algorithm computed each result in less than 1 second which was clearly much more efficient than adopting combinatorial search, which took several hours.

### B. Real Dataset

In addition, we tested our algorithm on actual range scanner datasets<sup>4</sup> to show that it also works well in real scenes. World features were extracted from snapshots of these datasets by a process briefly described as follows: 1) We scanned for large intervals, continuous points, and sharp turns between the output data points from range scanners. These landmarks indicate promising places where corner, line, and ending should appear respectively. 2) From these landmark locations, we extended

the scanning forward and backward to count number of points that form straight lines and structurally support appearance of features. 3) To become a line or an ending, we evaluated the least squares error when we tried to fit a straight line to the supporting points. As for a corner, we used the least squares errors from both lines that form corner and also the magnitude of cross product, measuring how close the two lines form the right angle. The complexity of the algorithm is  $O(hk)$  where  $h$  is the number of the output data point and  $k$  is the size of supports.

Maps were created from a user-interface that enables human operators to draw scenes in range data format (each drawing point is presented in polar coordinate system from an arbitrarily chosen viewing point). The map features were extracted from the drawings using the same extraction routine for the world features. Then, using the same interface, the operators were allowed to create links between the extracted features in the maps in order to define relative constraints.

Matching results with ARCS are shown in Figure 4. The constraint threshold was 4 and the minimum component criteria was 3 for all maps. Our algorithm involved 5 steps of radial projection with  $\alpha = 0.5$  in the constraint conversion process and 2 improvement steps. The results were computed in real-time manner and it can be seen that all ARCS results are better than the threshold.

## V. CONCLUSION AND DISCUSSION

The main contributions of this paper are the formulation of the one-to-one matching problem from graphical models based on the use of relative and absolute constraints and a relaxation-based algorithm to solve it. The algorithm is a combination of the RL and KM methods with the additional improvement steps to enhance matching quality. We evaluated the algorithm on the matching problems between human-provided maps and range scanner data. The results show effectiveness of our algorithm in term of matching quality and speed.

This work is related to the graduated assignment algorithm [8], but with an additional elaboration whereby we enforce the one-to-one matching constraint using the KM method. Also, our objective (2) is in form of a quadratic assignment problem; therefore, we can use any quadratic

<sup>3</sup>If the threshold was too low, only few assignments could survive the elimination process; sometimes this might not be satisfactory.

<sup>4</sup>The data used in this experiment were acquired from the Robotics Data Set Repository (Radish) and Datasets of Cyrill Stachniss [21], [22]; we would like to thank D. Fox, N. Roy, and D. Haehnel for providing them.

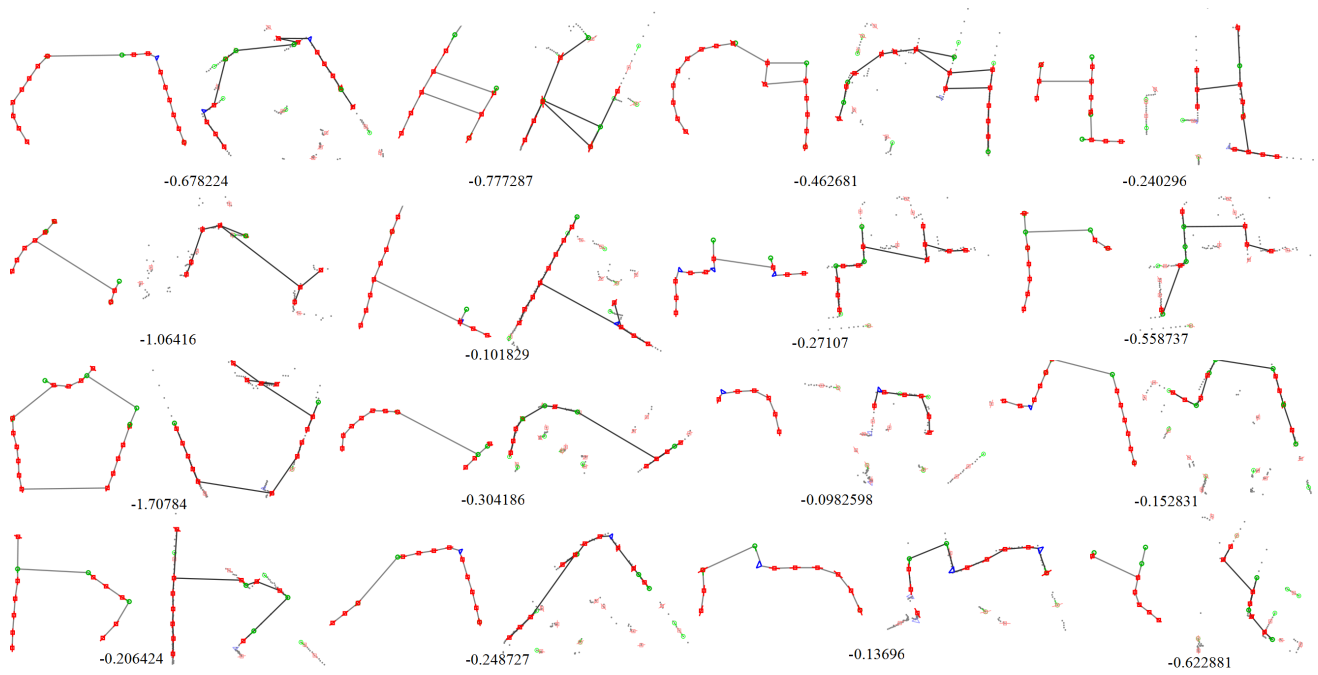


Fig. 4. Sixteen pairs of real experiment results are shown. Each pair contains a map (left) and the matching result with ARCS overlays on the extracted features from a snapshot of range scanner data (right). The raw range scan datasets were obtained from Radish and Datasets of Cyrill Stachniss [21], [22].

assignment algorithm to solve it. Thus, our proposed algorithm has some distinct features. Not only quadratic assignment problems, but RL can also be extended to non-quadratic versions, which is when the objective (2) includes higher order terms [12]. Though this extension comes with an increase in run-time complexity, the highly parallel structure of RL allows it to be computed efficiently on multithreading systems.

## REFERENCES

- [1] B. Takács and Y. Demiris, "Multi-robot plan adaptation by constrained minimal distortion feature mapping," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, 2009, pp. 742–749.
- [2] G. Chronis and M. Skubic, "Robot navigation using qualitative landmark states from sketched route maps," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 2. IEEE, 2005, pp. 1530–1535.
- [3] G. Parekh, M. Skubic, O. Sjahputera, and J. Keller, "Scene Matching between a Map and a Hand Drawn Sketch Using Spatial Relations," in *Robotics and Automation, 2007 IEEE International Conference on*. IEEE, 2007, pp. 4007–4012.
- [4] G. Roig, X. Boix, and F. De la Torre, "Optimal feature selection for subspace image matching," in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*. IEEE, 2010, pp. 200–205.
- [5] W. Li and T. Lee, "Image registration and object recognition by affine invariant matching," in *Intelligent Multimedia, Video and Speech Processing, 2001. Proceedings of 2001 International Symposium on*. IEEE, 2002, pp. 56–59.
- [6] H. Ai, L. Liang, and G. Xu, "Face detection based on template matching and support vector machines," in *Image Processing, 2001. Proceedings. 2001 International Conference on*, vol. 1. IEEE, 2002, pp. 1006–1009.
- [7] L. Torresani, V. Kolmogorov, and C. Rother, "Feature correspondence via graph matching: Models and global optimization," *Computer Vision–ECCV 2008*, pp. 596–609, 2008.
- [8] S. Gold and A. Rangarajan, "A graduated assignment algorithm for graph matching," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 18, no. 4, pp. 377–388, 2002.
- [9] M. Pelillo, K. Siddiqi, and S. Zucker, "Matching hierarchical structures using association graphs," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 21, no. 11, pp. 1105–1120, 1999.
- [10] T. Caelli and T. Caetano, "Graphical models for graph matching: Approximate models and optimal algorithms," *Pattern Recognition Letters*, vol. 26, no. 3, pp. 339–346, 2005.
- [11] C. Bishop, *Pattern recognition and machine learning*. Springer New York, 2006, vol. 4.
- [12] R. Hummel and S. Zucker, "On the foundations of relaxation labeling processes," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 3, pp. 267–287, 1983.
- [13] J. Edmonds and R. Karp, "Theoretical improvements in algorithmic efficiency for network flow problems," *Journal of the ACM (JACM)*, vol. 19, no. 2, pp. 248–264, 1972.
- [14] W. Christmas, J. Kittler, and M. Petrou, "Structural matching in computer vision using probabilistic relaxation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 17, no. 8, pp. 749–764, 2002.
- [15] A. Rangarajan, "Self annealing: Unifying deterministic annealing and relaxation labeling," in *Energy minimization methods in computer vision and pattern recognition*. Springer, 1997, pp. 229–244.
- [16] S. Li, H. Wang, K. Chan, and M. Petrou, "Minimization of MRF energy with relaxation labeling," *Journal of Mathematical Imaging and Vision*, vol. 7, no. 2, pp. 149–161, 1997.
- [17] F. Tang and H. Tao, "Probabilistic object tracking with dynamic attributed relational feature graph," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 18, no. 8, pp. 1064–1074, 2008.
- [18] S. Li, "Parallel distributed relaxation labeling," *Recent Developments in Computer Vision*, pp. 363–371, 1996.
- [19] P. Hahn, T. Grant, and N. Hall, "A branch-and-bound algorithm for the quadratic assignment problem based on the Hungarian method," *European Journal of Operational Research*, vol. 108, no. 3, pp. 629–640, 1998.
- [20] P. Parent and S. Zucker, "Radial projection: An efficient update rule for relaxation labeling," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 11, no. 8, pp. 886–889, 1989.
- [21] A. Howard and N. Roy, "The robotics data set repository (radish)," 2003. [Online]. Available: <http://radish.sourceforge.net/>
- [22] C. Stachniss, U. Frese, and G. Grisetti, "Openslam," 2011. [Online]. Available: <http://openslam.org/>