Multiple Map Intersection Detection using Visual Appearance

Kin Leong Ho, Paul Newman
Oxford University Robotics Research Group
{klh,pnewman}@robots.ox.ac.uk

Abstract

It is difficult to detect intersections between maps using only geometric information. We propose a novel technique to solve this correspondence problem using a visual similarity matrix. Given sequences of images collected by robots, subsequences of visually similar images are detected. Since every image is time-stamped, we can extract from each robot the portion of the local geometric map that was built when the sequence of images was captured. Using standard scan matching, an alignment of corresponding geometric submaps is determined. The local maps can then be joined into a single global map. Crucially, the algorithm does not depend on the knowledge of relative poses between the robots or mutual observation. A statistical assessment of significance in the visual alignment score of the subsequence of images is used to prevent false triggering of joint map detection. We present results of combining four local maps into a single map over 180m in length traversed, through the detection of the intersections using the proposed algorithm.

1 Introduction

The approaches of current collaborative multi-robot map building algorithms can be broadly classified into three main categories: (1) merging sensory data from multiple robots with known data association between features in local maps built by different robots [1] (2) detecting other robots to determine relative position and orientation between local maps [2, 3] or assuming relative poses between robots are known [4] (3) deriving the transformation between robots' coordinate systems through the matching of landmarks [5, 6]. Generally, algorithms with strong assumptions about known data association or relative poses have been limited to theoretical experiments or highly engineered experiments. The algorithms that have worked with real world data with weaker assumptions have been limited to those that rely on detection of other robots. This approach means that the robots might duplicate each other's work by exploring the same environment for long periods of time without being aware of each others' poses. Otherwise, the robots have to hypothesize their relative positions and try to congregate at a hypothesized meeting point. This allows the robots to determine accurately each others' relative poses but distracts them from the task of exploration [3]. A more exploration efficient way of joining local maps is to detect similar intersections between local maps and align the local maps given the relative orientation of the similar intersections.

This work proposes the use of visual appearance to detect "similar" intersections between local maps built by multiple robots. These common intersections can be used to align the maps. A visual similarity matrix is constructed, which is composed of similarity scores between images taken from a camera on each robot. Each element $M(i, j)$ in the visual similarity matrix is a measure of similarity between image i and image j. No prior knowledge of the pose of individual robot is required. It builds upon work done in [8] to detect loop closure by image matching using visually salient features. Despite the highly discriminative nature of photometric information, false positives still exist because repetitive entities occur frequently in urban environments eg. windows. To resolve the issue of repetitive visual images, [9] takes into consideration spatial information as well as visual information. A contribution of this paper is to exploit the topological structure of the visual similarity matrix to enhance robustness in the detection of intersections between maps. The idea is simple; by matching subsequences of images captured from topologically linked locations, the probability of false positives is greatly reduced.

2 Related Work

As discussed in [6], a common problem overlooked by many papers on multi-robot mapping is data as-

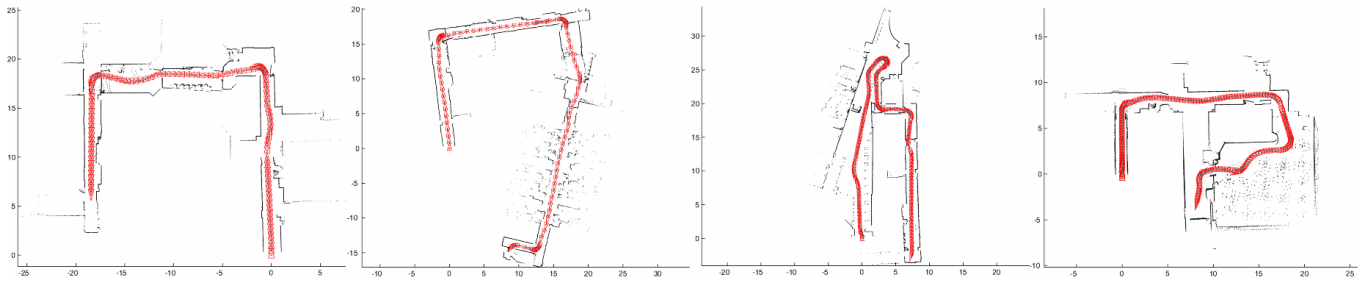

Figure 1: Local maps of different parts of the same building built by distributed robots. There is an overlap between each of the maps but as can be seen, it is very difficult to distinguish the overlap using only 2D geometric information.

sociation. The paper attempted to tackle this question by introducing an algorithm that aligned local maps into a global map by a tree-based algorithm for searching similar looking landmark configurations. The landmark configuration consists of relative distances and angle between a triplet of adjacent landmarks. Another landmark-based algorithm for map matching was described in [5], which combined topological maps of indoor environments. Landmarks such as corners, T-junctions, ends-of-corridor and closed doors were stored in the search space for correspondences. However, spatial configuration of three landmarks or simple geometric primitives are not very discriminative features.

A vision-based approach was used in [7] to combine maps built by a team of robots in the same worksite. Images described by color histograms are compared against each other to find the best matching image pairs. In the experimental setup, only images of planar surfaces are captured. Therefore, an inter-image homography can be calculated for selected image pair. If the homography is supported by a sufficiently high number of corners, intersection is found and robot paths can be registered with respect to one another. However, the use of a single image pair for matching is prone to false positives - particularly in urban environment containing repetitive entities, as described in [9]. Importantly, none of the algorithms described above have any mechanism to determine that two local maps have no common overlap. They simply find the 'best' alignment possible between the two.

Figure (1) shows that the correspondence of overlaps between these local maps is very difficult to be determined using geometric information alone. In the next section, we propose a method using discriminative photometric information to solve the correspondence problem.

3 A Visual Similarity Matrix

A system to detect loop closure was developed in [8], by matching the most recently captured image against every single image the robot has captured previously and stored in a database. Each image is described by visually salient features, which are used for image similarity comparison. In contrast to [8], the only interest point detector that we adopt to extract features from images is the detector developed in [11], which finds "maximally stable extremal regions" or "MSERs". MSERs offer significant invariance under affine transformations. Having found image features, we encode them in a way that is both compact to allow swift comparisons with other features, and rich enough to allow these comparisons to be highly discriminatory. For this, we use the SIFT descriptor [12] which has become immensely popular in global visual localization applications [13].

3.1 Assignment of weights to descriptor

Previously in [8], a saliency detector [14] was used to assign binary weight to each SIFT descriptor based solely on local photometric information. In this work, various weights are assigned to different SIFT descriptors based on their frequency of occurrence or rarity within the image database. The underlying concept is the more rare a descriptor is within a database, the more significance or weight should be attached to matching the descriptor.

The vector space model [15] which has been successfully used in text retrieval is employed in this work. Each image can be considered as a document consist-

ing of visual words. In this case, each SIFT descriptor is a visual word. Construction of a visual vocabulary is achieved by clustering similar SIFT descriptors (in terms of euclidean distance) into visual words that can be used for inverted file indexing. An agglomerative clustering algorithm is used. Weights, W_i, are assigned to each SIFT descriptor, D_i, (word) according to the frequency of the occurrence of the visual word in the image database. This is based on the inverse document frequency [16] formulation: $W_i = \log_{10}(N/n_f)$ where N is the number of images stored in the image database and n_f is the number of occurrences of the descriptor in the database. The collection of images is represented by an inverted index for efficient retrieval. To further enhance the retrieval speed, we employ a k-d tree to search for the visual words.

3.2 Similarity Scoring Function

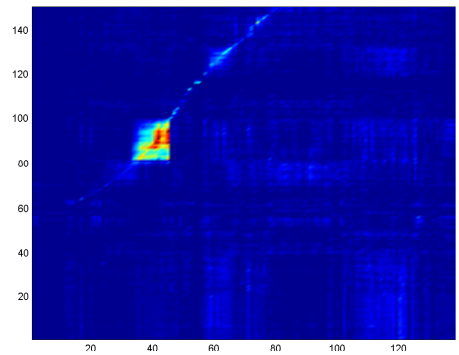

Figure 2: A visual similarity matrix constructed from comparison between two image sequences collected by two robots. Cells with high similarity scores are colored in bright red while cells with low similarity scores are colored in dark blue. The bright line highlights the sequence of images that are similar to each other - indicating that there is an overlap in the two environments explored. The bright "square" in the visual similarity matrix is a result of a visually similar region in the environment such as a long fence.

To measure the similarity between two images, I_u and I_v, we employ the cosine similarity method. Since each image is represented as a vector of words with different weights, we can measure their cosine similarity by the inner dot product of the two image vectors as shown in equation 1. The scoring for a match of a term is based on the weights from the inverse document frequency. If the images have different number of visual words, imaginary visual words with no associated weights are inserted into the smaller image vector so that the sizes of both image vectors are equal.

$$S(I_u, I_v) = \frac{\sum_{i=1}^{n} u_i.v_i}{(\sum_{i=1}^{n} u_i^2)^{1/2}.(\sum_{i=1}^{n} v_i^2)^{1/2}} \qquad (1)$$

where $I_u = [u_1 \cdots u_n]$, $I_v = [v_1 \cdots v_n]$ and u_i and v_i are visual words from the respective images.

Consequently, we can construct a visual similarity matrix between two image sequences using the cosine similarity function. Each element $M_{i,j}$ of the similarity matrix is the similarity score between image i from robot 1 and image j from robot 2. Every image from robot 1 is compared with all images from robot 2. When there is an overlap between the local maps of the robots, there will be a connected sequence of elements with high similarity scores found within the visual similarity matrix. This is shown by the bright line in Figure 2. The next section will describe the method employed to find this local sequence alignment within the visual similarity matrix.

4 Local Sequence Alignment

Local sequence alignment is a widely used tool in computational molecular biology, which finds the best alignment from a similarity matrix constructed from comparing two DNA or protein sequences. We adopt a similar approach [17] in finding the best alignment between two image sequences. The algorithm finds regions of similarity between protein and nucleic acid sequences that may have little overall similarity but the shared pattern may have biological significance. Similarly, local maps built by distributed robots may have little overall similarity due to mapping of different areas as shown in figure (1) but the intersection is important. Finding a matching pair of subsequences of images between the robots' image sequences is a strong indicator of overlap of their maps. The detection of visually similar subsequences of images is therefore the precursor of map joining.

4.1 Local Alignment Algorithm

Our local alignment algorithm is a modified version of the Smith-Waterman algorithm [17], which is a dynamic programming algorithm. Given two sequences such that $A = a_1, a_2, \cdots, a_n$ and $B = b_1, b_2, \cdots, b_m$, a similarity function $S(a_i, b_j)$ gives a similarity score between sequence elements a_i and b_j. A similarity matrix of scores is calculated in [17] by comparing each element from one sequence to every other element in the other sequence - the same way our visual similarity matrix is constructed. In order for the Smith-Waterman algorithm to work, the similarity function

	I_{b1}	I_{b2}	I_{b3}	I_{b4}	I_{b5}	I_{b6}
I_{a6}	-10	0.35	-10	-10	-10	-10
I_{a5}	-10	-10	-10	-10	<u>0.32</u>	-10
I_{a4}	-10	-10	0.26	<u>0.37</u>	0.26	-10
I_{a3}	-10	0.21	<u>0.27</u>	<u>0.33</u>	-10	-10
I_{a2}	-10	<u>0.32</u>	0.25	0.18	-10	-10
I_{a1}	-10	-10	-10	0.15	-10	-10

	I_{b1}	I_{b2}	I_{b3}	I_{b4}	I_{b5}	I_{b6}
I_{a6}	0	0.35	0	0	0	0
I_{a5}	0	0	0	0	<u>1.61</u>	0
I_{a4}	0	0	0.85	<u>1.29</u>	1.55	0
I_{a3}	0	0.53	<u>0.59</u>	<u>0.92</u>	0	0
I_{a2}	0	<u>0.32</u>	0.57	0.75	0	0
I_{a1}	0	0	0	0.15	0	0

Table 1: The matrix above is an example of a visual similarity matrix where each cell is the similarity score between the corresponding images. The bottom matrix is the corresponding H-matrix calculated from the visual similarity matrix shown above. The sequence alignment selected is underlined.

must give a negative score when two elements are very dissimilar. In our implementation, image pairs with a similarity score that falls below a given threshold are deemed to be dissimilar and are rescored with a fixed negative value.

To find a pair of subsequences of images with high degrees of similarity, a matrix H is constructed. Each element, H_{ij}, is the cumulative similarity score of a subsequence starting at a_k and b_l and ending in a_i and b_j in the visual similarity matrix.

The formula for $H_{i,j}$ follows by considering the possibilities for ending the subsequence at a_i and b_j. Amongst $S(a_{j-1}, b_{j-1}), S(a_j, b_{j-1})$ and $S(a_{j-1}, b_j)$:

- if $S(a_{j-1}, b_{j-1})$ has the greatest similarity score: $H_{i,j} = H_{i-1,j-1} + S(a_i, b_j)$
- if $S(a_j, b_{j-1})$ has the greatest similarity score: $H_{i,j} = H_{i,j-1} + S(a_i, b_j)$
- if $S(a_{j-1}, b_j)$ has the greatest similarity score: $H_{i,j} = H_{i-1,j} + S(a_i, b_j)$
- A zero is included to prevent negative cumulative similarity score, indicating no similarity up to a_i and b_j

The maximum value in the H-matrix, the maximal alignment score, is therefore the endpoint of a subsequence of images with the greatest similarity. No other pair of subsequences has greater similarity. From the H-matrix at the bottom of figure 4.1, the maximal alignment score is $H(I_{a5}, I_{b5})$, which is an accumulation of similarity scores of the subsequence of underlined elements from $S(I_{a2}, I_{b2})$ to $S(I_{a5}, I_{b5})$.

To take into account that the robots might have traversed through the same area in opposite directions, the order of one robot's image sequence is reversed and the algorithm is repeated for that sequence order. The larger of the two maximal alignment scores is chosen. To determine which images have contributed to the maximal alignment score, the algorithm stores a pointer at each cell in the H-matrix, to indicate which previous cell contributed to its value. From the matrix element of H with the maximal alignment score, we are able to sequentially trace back the path of the other matrix elements that contributed to this maximum value. This yields the best matching pair of image subsequences.

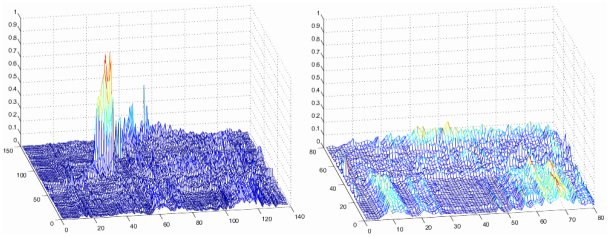

Figure 3: Left: A 3D visual similarity matrix between two image sequences that have overlapped intersection. Right: A 3D visual similarity matrix between two image sequences that have no overlapped intersection.

5 Statistical Significance of Local Alignment

Given a visual similarity matrix, the local alignment algorithm will produce the maximal alignment score along with the pair of image subsequences for that particular matrix. When the maximal alignment score exceeds an experimentally set threshold, potential map overlap is detected. The key question is whether the selected pair of subsequence of images is really due to an overlap of local maps. In other words, what value should a maximal alignment score be so that it is statistically significant enough to suggest that there is actually an overlap. On the left-hand side of Figure 3 shows the similarity matrix for image sequences that have significant overlap and the right-hand side shows

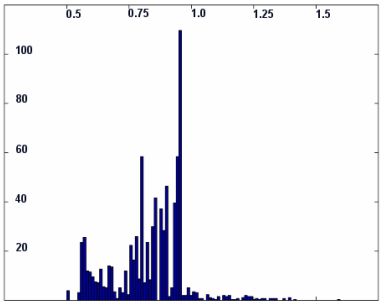

Figure 4: Typical distribution of maximal segment scores from 1000 random shuffles of similarity matrix.

a similarity matrix with no overlap between image sequences. Maximal alignment scores resulting from searching with a query subsequence against a whole sequence can be well described by the extreme value distribution [18] expressed in equation 2. This p.d.f can be used to judge the significance of the maximal alignment score for the pair of matching image subsequences.

$$P(x) = \frac{1}{\beta} \exp^{\frac{x-\mu}{\beta}} \exp^{-\exp^{\frac{x-\mu}{\beta}}} \quad (2)$$

where x is the maximal alignment score, μ is the mean of the distribution and β is the standard deviation of the distribution.

Adopting the approach in [19], we randomly shuffle the visual similarity matrix 1000 times and obtain the maximal alignment score each time. This results in a distribution such as that shown in Figure 4. This process is time consuming but it is only triggered when the alignment score exceeds an experimentally set threshold. The distribution parameters μ and β are estimated from the histogram of maximal alignment scores and we can calculate the probability of the particular alignment score happening by chance. It is a topic of future research to determine the number of shuffles required as a function of the size of the similarity matrix, to produce an accurate estimate of the distribution parameters. In our implementation, an alignment is only considered significant if the maximal alignment score lies outside 5 standard deviations from the mean.

6 Results

To illustrate the effectiveness of our approach we choose to employ a simple delayed state [21], laser based scan matching [20] SLAM algorithm for each robot. This choice is made entirely without prejudice - any SLAM algorithm could have been used. We proceed as follow. For every 0.5m the robot traverses and for every 15 degrees change in heading of the robot, an image is captured. The camera orientation toggles after capturing an image, between 60 degrees left of robot's heading and 60 degrees right of robot's heading. Every image and laser scan captured is time stamped.

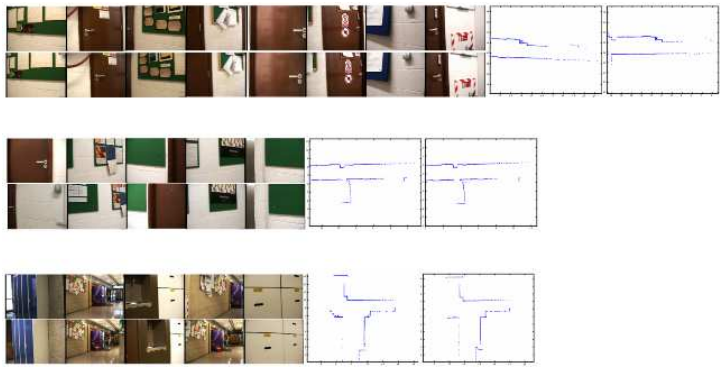

Figure 5: Pairs of image subsequences that match. Along with a pair of subsequence of images are the corresponding local submaps that should match each other as well.

In our experiment, four robots start exploring from different locations of the same building. Each robot builds its own local map as shown in figure (1). By comparing the image sequences collected by robot 1 and robot 2, a 114 by 146 visual similarity matrix is constructed. The time complexity of the local alignment algorithm is $O(nm)$ where n and m are the lengths of the respective sequences. For the size of this particular similarity matrix, the local alignment algorithm takes less than 0.3 second to find the optimal alignment using a Pentium 4, 2.40GHz CPU. The time complexity of comparing an image from one sequence against all the images in the other sequence is $O(log(p))$ where p is the number of visual words stored in the database. The average time to compare one image against a sequence of 146 images is 0.269 second. When the maximal alignment score as described in subsection 4.1 exceeds an experimentally set threshold, joint map detection is triggered. The statistical significance of the alignment score is assessed according to the approach described in section 5 to prevent false detection.

Figure 5 shows typical pairs of image subsequences

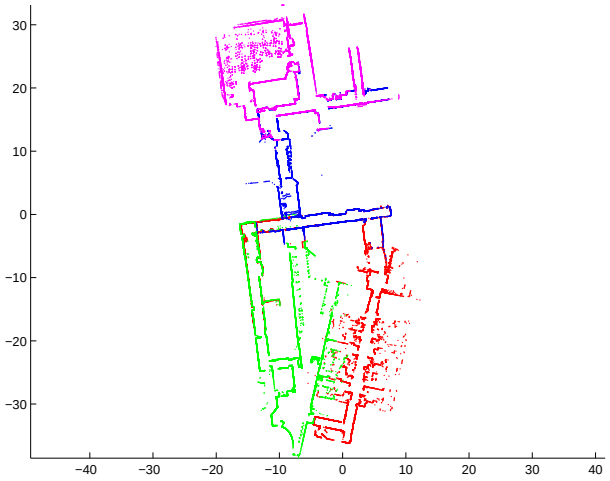

Figure 6: A combined map in a single coordinate frame that was formed by aligning the four local maps shown in figure 1

found by the local alignment algorithm. Since each image and laser scan is time-stamped, we can extract the portion of the local map that correspond to when the images were taken as shown in figure (5). A crude estimated transformation is used to bring the two local geometric submaps into close proximity. From here, scan matching produces accurate map to map transformations, allowing the four maps to be fused together as shown in figure (6). Without relying on detection of other robots, we have successfully aligned four local maps into a single, big map by using visual appearance to reliably detect intersections.

7 Conclusion and Future Work

A novel method for detecting and aligning similar intersections of local maps built by multiple robots has been demonstrated to work well in an indoor environment covering a distance of over 180m. A more extensive experiment mapping a larger area is underway. To further enhance the robustness of the similarity matrix, we can incorporate spatial descriptors as well as visual descriptors to describe the local environment as was done in [9]. Furthermore, we can improve upon our algorithm to find best "gapped" alignments. A gap in a subsequence of images may be possible due to poor measurements of one image (For example in the extreme case when a human walks very close to the camera and covers the whole field of view). We envision using this algorithm to combine local maps built by a single robot from different missions over many days. This will allow a large map to be incrementally enlarged from each map building session in an offline fashion.

Acknowledgements

The authors would like to thank Andrew Zisserman, Krystian Mikolajczyk and Josef Sivic for their comments and discussions.

References

[1] J. Fenwick, P. Newman, J. Leonard. Cooperative Concurrent Mapping and Localization. Proceedings of the 2002 IEEE International Conference on Robotics and Automation, pp. 1810–1817, May, 2002.

[2] D. Fox, W. Burgard, H. Kruppa, and S. Thrun. A probabilistic approach to collaborative multi-robot localization. Autonomous Robots, vol. 8, no. 3, 2000.

[3] Konolige, K. and Fox, D. and Limketkai, B. and Ko, J. and Stewart,B. Map Merging for Distributed Robot Navigation Proceedings of International Conference on Intelligent Robots and Systems, 2003

[4] S. Thrun. A Probabilistic Online Mapping Algorithm for Teams of Mobile Robots. International Journal of Robotics Research, vol. 20 no. 5, pp. 335–363, 2001

[5] G. Dedeoglu and G. Sukhatme. Landmark-based matching algorithm for cooperative mapping mapping by autonomous robots. Proceedings of the Fifth International Symposium on Distribution Autonomous Robotics Systems, 2000

[6] S. Thrun, Y. Liu. Multi-robot SLAM with sparse extended information filers. Proceedings of the 11th International Symposium of Robotics Research, 2003.

[7] H. Hajjdiab and R. Laganiere. Vision-base Multi-Robot Simultaneous Localization and Mapping. Canadian Conference on Computer and Robot Vision. pp. 155-162, 2004

[8] P. Newman, K. Ho. SLAM - Loop Closing With Visually Salient Features. Proceedings of the 2005 IEEE International Conference on Robotics and Automation, April 2005

[9] K. Ho, P. Newman. Combining Visual and Spatial Appearance for Loop Closure Detection. To be published in European Conference on Mobile Robotics, September 2005

[10] M. Waterman Introduction to Computational Biology: Maps, sequences and genomes. Chapman and Hall, 1995

[11] J.Matas, O. Chum, M. Urban and T. Pajdla. Robust Wide Baseline Stereo from Maximally Stable Extremal Regions. Proceedings of British Machine Vision Conference, 2002

[12] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision, vol. 60, no. 2, pp. 91–110 2004

[13] Lowe, D. G. and Se, S. and Little, J. Mobile Robot Localization and Mapping with uncertainty using Scale-Invariant visual landmarks. International Journal of Robotics Research, vol. 21, no. 8, pp. 735-758, 2002

[14] Timor Kadir and Michael Brady Saliency, Scale and Image Description International Journal Computer Vision, pp. 83-105, 2001

[15] Sivic, J. and Zisserman, A. Visual Google: A Text Retrieval Approach to Object Matching in Videos. Proceedings of the International Conference on Computer Vision, Oct 2003

[16] Karen Sparck Jones. Exhaustivity and Specificity. Journal of Documentation, vol. 28, no. 1, pp 11-21, 1972

[17] Smith, T.F. and Waterman, M.S. Identification of common molecular subsequences. Journal of Molecular Biology, vol. 147, pp. 195-197, 1981

[18] Gumbel, E. J. Statistics of Extremes. Columbia Univeristy Press, New York, NY, 1958

[19] Altschul and B. Erickson. Significance of Nucleotide Sequence Alignments: A Method for Random Sequence Permutation That Preserves Dinuclotide and Codon Usage. Mol. Biol. Evol. vol. 2, pp. 526-538, 1985

[20] Konolige, K. Large-Scale Map-Making. Proceedings of the National Conference on AI (AAAI), San Jose, CA, 2004

[21] J. Leonard, P. Newman, and R. Rikoski. Towards Robust Data Association and Feature Modeling for Concurrent Mapping and Localization. Proceedings of the Tenth International Symposium on Robotics Research, 2001