A Hybrid Approach to Finding Cycles in Hybrid Maps

Margaret E. Jefferies, Wenrong Weng, Jesse T. Baker,
Michael C. Cosgrove, Michael Mayo
Department of Computer Science
University of Waikato, New Zealand
{mjeff, ww19, jtb5, mcc2, mmayo}@cs.waikato.ac.nz

Abstract

One of the most difficult problems in Simultaneous Localisation and Mapping (SLAM) is that of identifying and closing cycles. While localisation methods exist that can provide local consistency in a map, residual errors can grow unbounded in global metric maps. Topological maps are favoured by some because they do not have the global consistency problems however they too have difficult correspondence problems due to perceptual aliasing. In this paper we discuss our approach to closing cycles in a topological map. We use both a global metric map and the topological map itself to identify cycles in the topological map.

1 Introduction

In this paper we discuss our approach to solving the correspondence problem in autonomous mobile robot mapping. In particular we are concerned with the problem as it arises in environments which contain cycles; to close the cycle in its map the robot needs to be able to determine that its current location corresponds to a part of the environment it has been to before. In robot mapping this is often termed *closing the loop*. Our mapping system is based on our previous work on cognitive mapping [Jefferies and Yeap 1998; Yeap and Jefferies 1999]. Cognitive mapping researchers have been interested in the correspondence problem for some time but it was not clear from their computer simulations that their algorithms would handle all the uncertainties that a robot faces in the real world [Kuipers and Byun 1988; Yeap 1988; Yeap and Jefferies 1999]. Recently solutions from roboticists have begun to appear [Gutmann and Konolige 1999; Thrun 2001; Tomatis et al. 2002; Bosse et al. 2003; Thrun et al. 2003] and cognitive mapping researchers have adapted their theories and algorithms for the real world problem robots encounter [Kuipers and Beeson 2002; Beeson et al. 2003; Jefferies et al. 2003].

Approaches used to represent the robot's map mainly fall into one of two categories - global metric where a single global coordinate system is used to represent the robot's total experience of its environment and topological where the connectivity between different parts of the environment are represented. In metric maps errors are reduced at the level of the local space but some residual error remains and the magnitude of this error can grow unbounded in the global space. This makes solving the correspondence problem for environments with large cycles difficult [Gutmann and Konolige 1999; Thrun 2001; Thrun et al. 2003]. Topological maps have found favour with some recently [Tomatis et al. 2002; Bosse et al. 2003] because they do not have the global consistency problem, i.e. errors are restricted to the local space. Topological maps have long been preferred in cognitive mapping [Kuipers and Byun 1988; Yeap 1988] for this reason. In addition Kuipers[Kuipers and Byun 1988] and Chown [Chown et al. 1995] argue that it is easier to compute a topological map than a global metric map directly from sensory information. In closing cycles in topological maps the problem is to match two nodes in the topological map if they represent the same physical space (the correspondence problem) and to distinguish two nodes that look the same if they represent different parts of the environment (the perceptual aliasing problem). Failure of the latter results in false positives and the former false negatives. Recently hybrid approaches have emerged [Thrun 1998; Tomatis et al. 2002; Bosse et al. 2003]and in [Bosse et al. 2003] the advantages of both the topological and metric mapping paradigms are exploited in closing large cycles. Hybrid approaches are popular in the cognitive mapping community [Kuipers and Byun 1988; Yeap 1988; Chown et al. 1995; Yeap and Jefferies 1999] however the metric and topological maps do not have equal status. The topological map is the dominant representation in their models.

Our approach to mapping the robot's environment extends the hybrid model of [Yeap and Jefferies 1999] and adheres to the dominant cognitive mapping tenet, that the prime representation is the topological map (see [Yeap and Jefferies 1999; Kuipers 2000] for a discussion on why this is so). Yeap and Jefferies' [Yeap and Jefferies 1999] topological map of metric local space descriptions has been implemented on a mobile robot with minor adaptations to handle input from a laser range sensor. Yeap and Jefferies [Yeap and Jefferies 1999] proposed a limited (in size) global metric map to close small cycles in the topological map. The restricted size of their global metric map accounts for the limitations in the human or animal path integration system with accumulating error [Gallistel and Cramer 1996]. Our approach uses both the topological map and the global metric map to detect cycles. In our implementation using a locally consistent global metric map we are able to detect significant cycles. The global metric map is discretised into the local space descriptions which correspond to the nodes in the topological map. This means that detecting cycles is a very cheap operation - we determine if the robot's location is inside one of the local space partitions in the global map. Provided there is some overlap between the current local space and the representation in the global map for the previous encounter with this space then the cycle will be detected. The method will fail as soon as the overlap with a neighbouring local space becomes more significant than the overlap with the correct local space.

In addition we can detect cycles in the topological map using feature matching. However we can not match every feature in a local space because when it is approached from different view points different parts of the local space may be occluded. Therefore a backprop neural network is used to learn a signature for each ASR which is composed of the subset of features that are viewable from wherever the ASR is approached. However this is a chicken and egg problem. To learn a useful signature the robot needs to recognise the local space when viewed from different positions so that the feature set can be reduced to features common to both views. But the signature is needed to determine if they are the same local space. Therefore the signature is learned incrementally. A neural network algorithm is used to learn a signature for each unique local space. New local spaces are classified according to these signatures. If the classification process indicates a match then the neural network is retrained to account for the different views the robot will have of the same space when it is approached from different routes.

2 The Basic Mapping Approach

The topological map comprises a representation for each local space visited with connections to others which have been experienced as neighbours. The local space is defined as the space which "appears" to enclose the robot. The local space representation is referred to as an Absolute Space Representation (ASR) a term which emphasises the separateness and independence of each individual local space. Each ASR in the topological map has its own local coordinate frame.

The first step is to turn the raw laser data into lines representing surfaces in the robot's view of its environment. From this view the algorithm firstly works out where the exits occur. The occlusions are the lines marked *occ* in Fig 1(c). An occlusion map is constructed from the surfaces in view (see Fig 1 (c). The first occlusion map obtained for a local space is termed the *master occlusion map* as it is updated and used to recompute the ASR as the robot explores its local space.

Exits are then created from the occlusions and surfaces in the master occlusion map. For each occlusion in the master occlusion map the algorithm determines which part of the gap associated with it is the actual exit. The exit computed is the shortest "virtual surface" which "covers" the occlusion. We refer the reader to [Yeap and Jefferies 1999] for an in depth description of this part of the algorithm. Surfaces outside the exit are eliminated. Surfaces which lie between exits form the boundary of the ASR. The exit used to enter the ASR, E3 in Fig 1(b), and two unknowns, *U1* and *U2*, which mark the unexplored region of the room, are added to form a complete closure (see Fig 1(d)).

Exits computed as above have a dual role, in the traditional sense to indicate where the robot can leave the current space and to indicate parts of the environment which are yet to be uncovered. These two roles are distinguished by labelling the latter as *unknown* (see *U1 and U2* in Fig 1 (d)) and the former as *known (*see *E3-E5* in Fig 1(d)). As the robot moves about the local space parts of it that were once unknown are no longer so, and the exits covering these areas are updated. See [Yeap and Jefferies 1999; Jefferies et al. 2001] for a description of this updating process. In Fig 1 the robot first investigates the unknown *U1*. Fig 1(e) shows the updated master occlusion map and Fig 1(f) the resulting ASR. The unknown *U2* is similarly updated.

Rofer's [Rofer 2002] histogram correlation localisation method is used to provide consistency within ASRs. New ASRs are computed whenever the robot crosses an exit into an unexplored region and ASRs are linked, as they are experienced, via the exits which connect them to their neighbours in the topological map. ASRs are the nodes of the topological map and the exits are its edges. Fig 2 shows an example of a topological map constructed in this way.

The ASR construction mostly produces consistent ASRs from any view point once the local space is fully explored. However convex corners occasionally give rise to inconsistencies depending on how they are approached. Usually exits are formed at convex corners but if the convex corner is approached face on there is no occlusion and hence no exit. Fig 5 demonstrates this problem. Our answer to this problem is to maintain separate ASRs for the different approaches to a local space. This aspect of our method is outside the focus of this paper. It will be the subject of another paper.

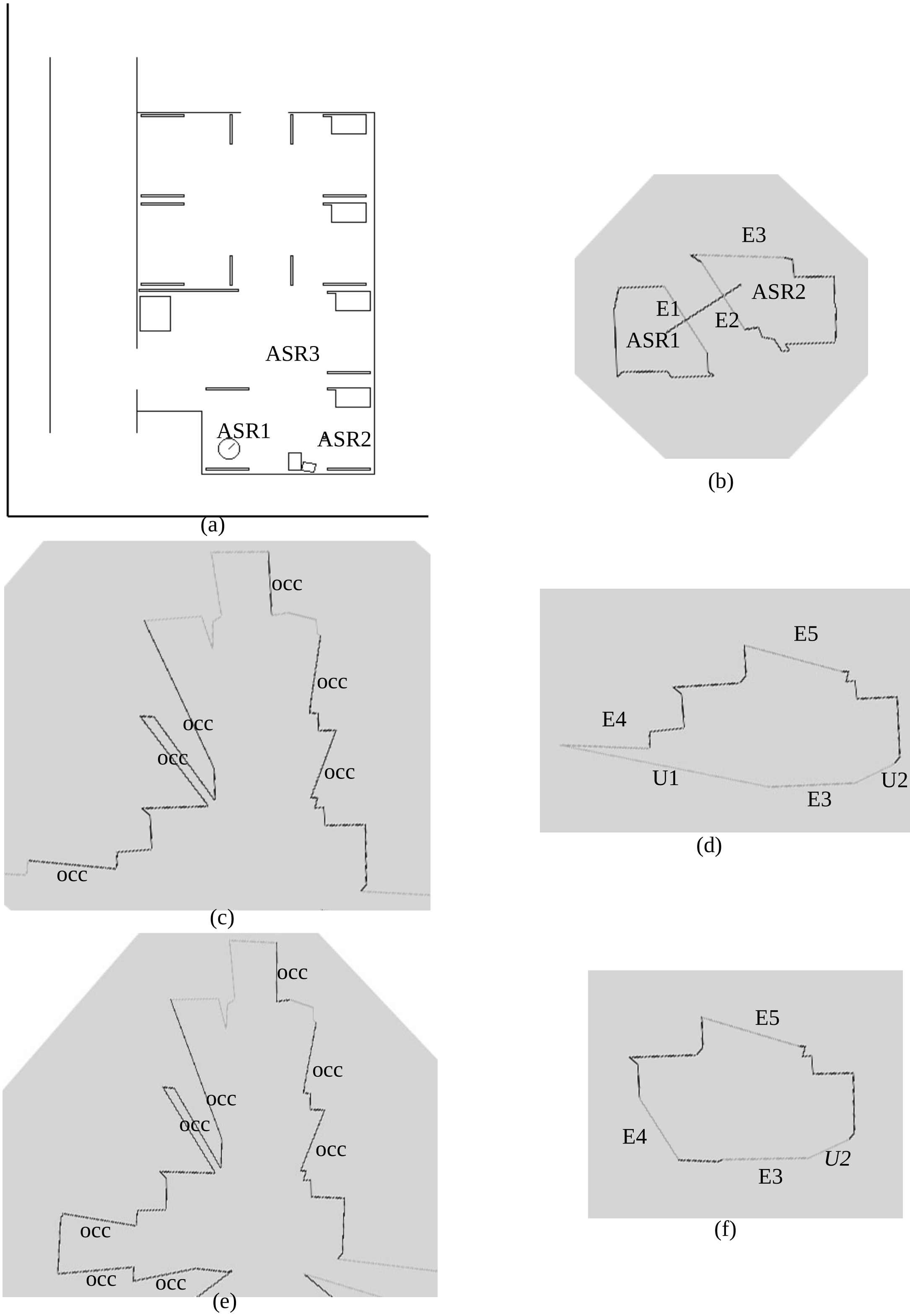

Fig 1 Constructing the ASR (a) A floor plan of the robot's environment (b) The initial ASRs in the robot's cognitive map. The robot leaves via exit *E3* and begins to construct a new ASR. (c) The occlusions are made explicit in the master occlusion map. (d) The resulting ASR. *U*1 and *U2* mark the unknown regions to be explored. *E3* is the exit used to enter the local space. (e) The master occlusion map when the robot investigates the unknown *U1*. (f) The resulting ASR.

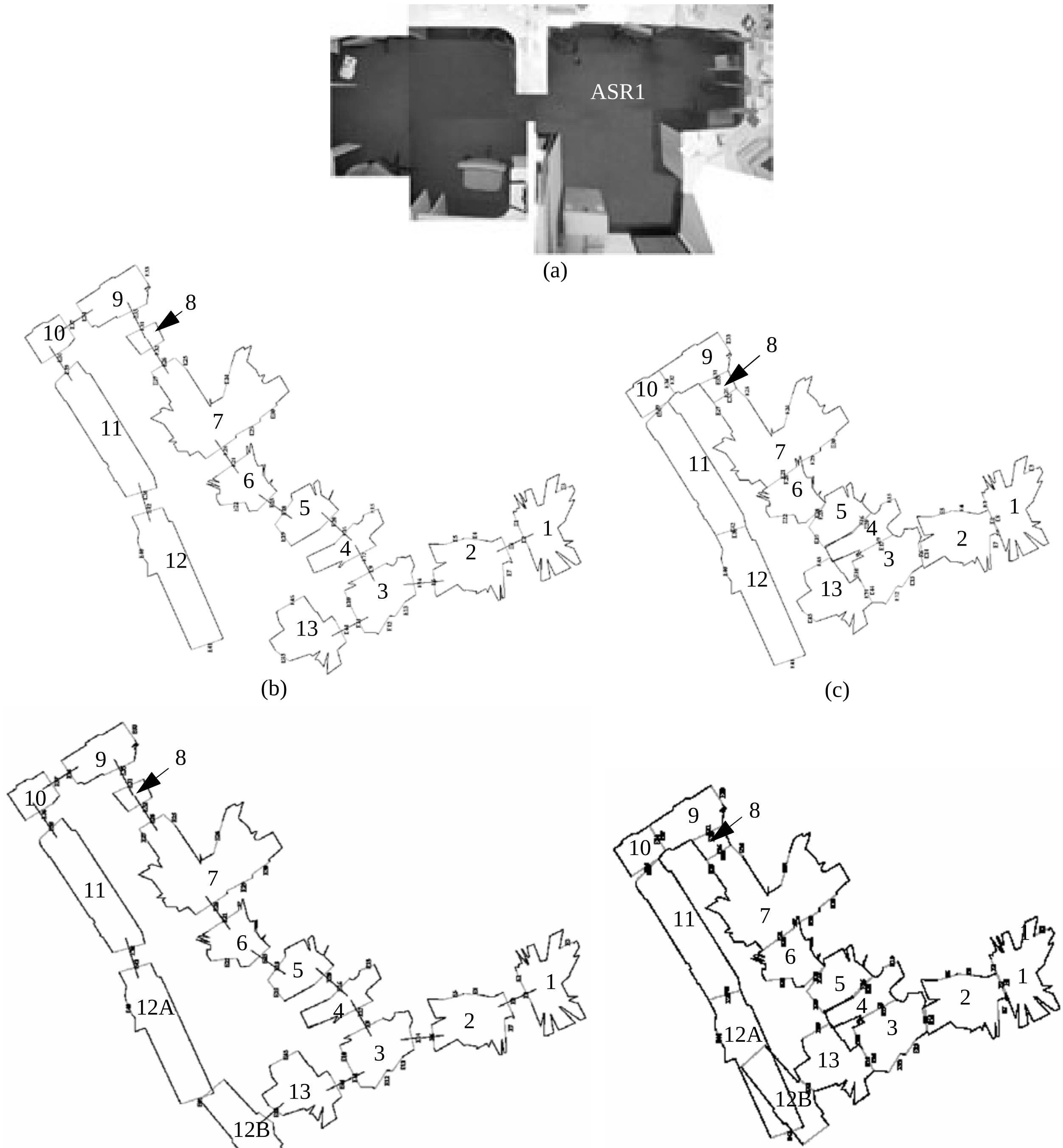

Fig 2 The topological and metric maps. The ASRs are labelled according to the order in which they are encountered. The green, or lighter coloured lines in grey scale are exits. (a) A section of the environment the robot explored. The label *ASR1* shows where the robot starting mapping from and corresponds to the ASR labelled 1 in (b) -(d). (b) The topological map just before the robot encounters a cycle. The robot's path through the environment follows ASR1 - ASR12 in sequence, whereupon it retraces its steps back to ASR3. It then computes ASR13. (c) The global metric map just before the robot encounters a cycle. (d) The robot enters and computes ASR12B (e) A cycle can be detected from the overlap of ASR12A and ASR12B.

3 Closing Cycles with a Global Metric Map

The function of the global metric map is to provide global position information that can be used to close cycles in the topological map. It is constructed alongside the topological map and comprises the ASRs represented in a global coordinate frame. A straightforward approach is used to close cycles. When a seemingly new ASR is encountered the robot looks for an ASR which is in roughly the same location in the global metric map. Because the locations of ASRs in the global map coordinates are being matched, rather than robot poses, the location of the ASRs do not need to match exactly. It suffices that there be significant overlap. Thus depending on the size of the ASRs this method can accommodate a significant amount of accumulated error.

To match the locations of the ASRs we take the centre of the newly encountered ASR and check to see if it is contained within another ASR in the global metric map. However, as error accumulates the ASR will also overlap neighbours of the bona fide matching ASR. These false positives are reduced (but not eliminated) by cross matching the centres of the ASRs. When a match is detected we check that the centre of this ASR is inside the newly encountered ASR.

In Fig 2 the robot traverses our laboratory and the adjoining corridor following a route which takes it though ASRs 1-12 in sequence. It then follows the same sequence of ASRs back to ASR3 and enters ASR13 which completes a cycle. The cycle should be detected when the robot enters ASR12*(see figs 2 (d) and (e) from ASR13. In Fig 2 (e) the overlap of ASR12 and ASR12* is obvious and even though they are misaligned it is possible to detect that they cover the same space.

Once a match is indicated, exits in the corresponding ASRs are used to snap the two representations together. This will give rise to a discrepancy value which can be used to propagate a correction factor through the map to realign it.

4 Using Feature Matching to Close Cycles

In this section we present an alternative method for closing cycles in the topological map. It is presented as a separate method but we envisage a symbiotic relationship between the global map matching of ASRs described above and feature matching of ASRs described below.

The feature based matching approach operates as follows. As the robot enters a local space and constructs an ASR for it, the set of features for the ASR are classified by the neural network. The neural network returns its prediction, a score for each ASR in the topological map, which indicates its degree of similarity with the ASR the robot currently occupies. If all the values are below a chosen threshold then it is treated as a new ASR. The neural network is then trained on the new ASR's feature set to find a signature that will be used to recognise it when it is revisited. If a match is indicated from the classification process the neural network is retrained. Because the matching ASRs are computed from different view points not all the features in one ASR will be common to the other. Thus this process refines the signature so that better predictions are possible for future classifications.

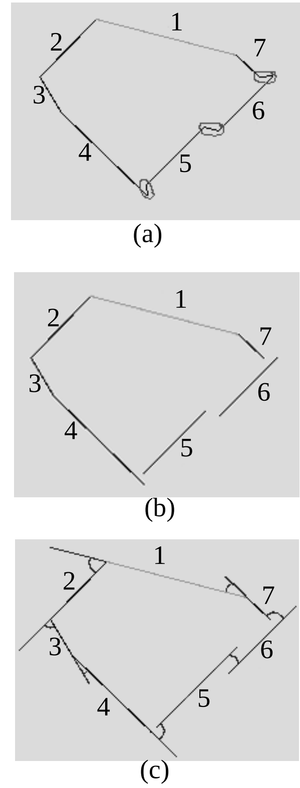

Fig 3. The Features extracted from an ASR. (a) The ASR with minor segments encircled and major segments labelled 1-7. (b) Minor segments are removed. (c) The segments and angles which comprise the initial feature set.

4.1 Feature selection

The feature set needs to accommodate sensing errors and be able to handle partial matches resulting from occlusions. We divide the ASR into segments, where each segment is a region of the ASR boundary which has a consistent gradient. The segments are divided into minor (short) segments and major (long) segments. Minor segments often result from spurious effects therefore they are not included in the feature set. The remaining segments are used to form the initial set of features given to the neural network. In addition to the segment, a feature comprises the angles corresponding to the change in gradient between adjacent segments, traversing the ASR in a clockwise direction. (see Fig 3). Fig 3 shows the segments extracted for the ASR depicted. There are 7 major segments labelled 1-7, and 3 minor segments. Segment 1 denotes an exit. Segment 3 represents a gap in the boundary but is turned into an a surface because it is too small for the robot to pass through. The features extracted are listed in Table 1.

Table 1: The initial features extracted for the ASR in Fig 3

Segment No	Length (mm)	Angle 1 (degrees)	Angle 2 (degrees)
1	1800	32.86	58.81
2	1008.9	58.81	-281.17
3	522.3	-281.17	10.6
4	1506.9	10.6	90.38
5	1014.9	90.38	0.07
6	991.3	0.03	88.44
7	392.8	88.44	32.86

4.2 Signature Learning and ASR Classification

The requirements of the learning algorithm were as follows. The learning algorithm needed to be incremental and be able to add new classes (ASR signatures) online as new ASRs are encountered. There could be no restriction on either the number of boundary segments or the number of distinct ASRs in the environment. The algorithm needed to be able to decrement the effect of features common to many ASRs while strengthening the effect of those that distinguish ASRs. While the learning process could run in the background a fast prediction process was essential if it was to run in real time. Therefore, a back-propagation neural network was chosen to learn the ASR signatures and predict matches of newly computed ASRs with previously visited ASRs. Nguyen-Widrow Initialisation, Momentum and Batch updating of weights are used along with a bipolar sigmoid activation function.

The ranges of the input values (10m for length, and 360^{o} for angles) are discretised into intervals. This is a practical requirement for a neural network but also accommodates sensor error. In the current implementation, a length interval of 200mm and angle interval of 45^{o} are used. Each input neuron represents a particular length, angle, angle combination. When classifying an ASR, the output neuron associated with each ASR, outputs a value between 0 and 1 indicating the similarity of the new ASR with the visited ASR.

An example of a cycle is shown in Fig 4. The robot has traversed the environment depicted in Fig 4 (a) constructing the ASRs in the topological map (Fig 4 (b)) in the order they are numbered. The robot re-enters ASR2 via ASR7. The newly computed ASR2* is shown in Fig 4 (c). The similarity predictions for ASRs 1 - 11 are shown in Table 2. Five values stand out, .78, .94, .89, .71, and .72 for ASRs 1, 2, 3, 4, and 6 respectively. If the threshold value were set at 0.7, say, then

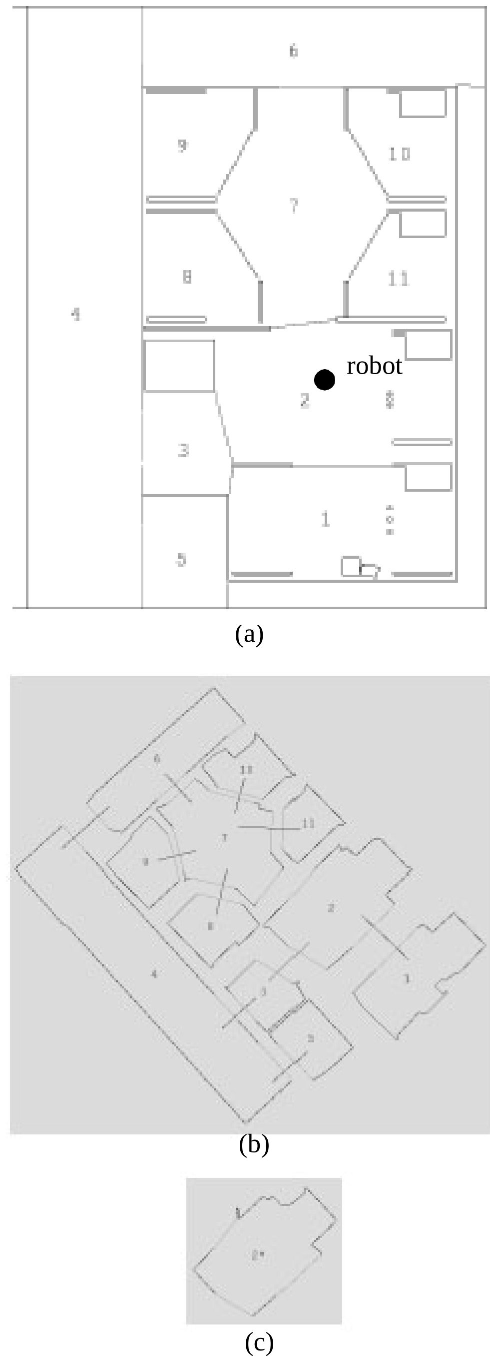

Fig 4 A positive match. (a) The environment. (b) the topological map constructed in the order the ASRs are numbered. (c) The robot has re-entered ASR2 via ASR7. ASR2* depicts the newly computed ASR to be matched.

these would all be candidate matches. One cannot simply choose the best match because in many environments the ASRs for different local spaces will look similar (the perceptual aliasing problem). More evidence is needed to choose between them if indeed any of them should be chosen. In this

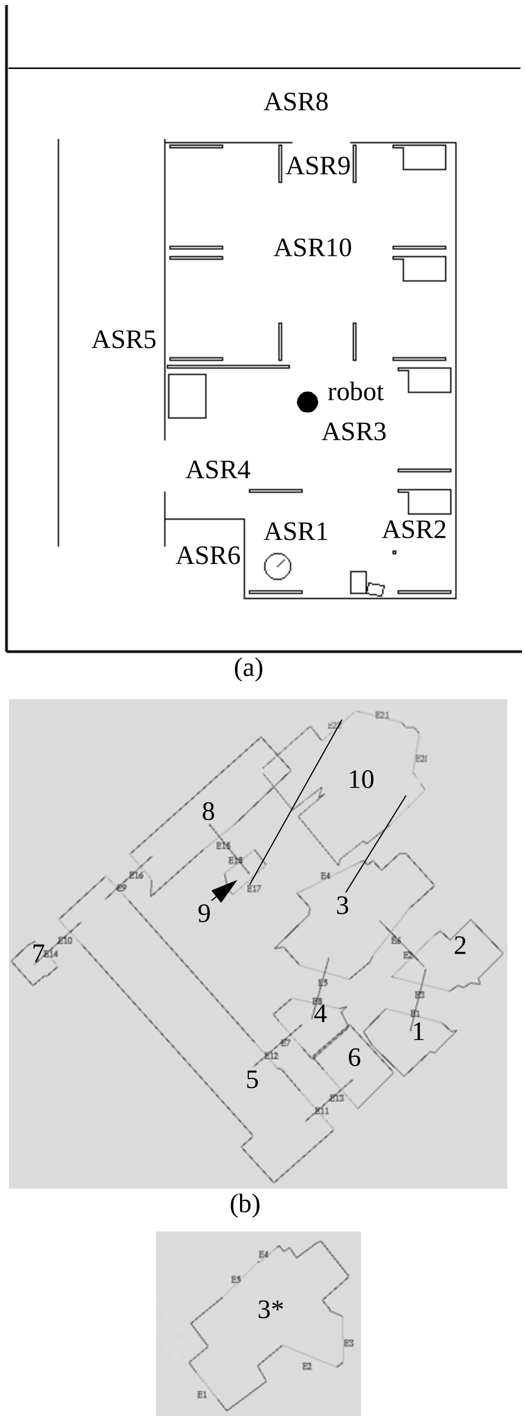

Fig 5 An example of a false positive prediction (a) the environment (b) the topological map (c) the robot re-enters ASR3 and computes the ASR as depicted. It covers both ASR3 and ASR4 and extends into ASR2. The highest similarity value is for ASR2

Table 2: The similarity values for ASR 2* in Fig 4

ASR	prediction
1	.78
2	.94
3	.89
4	.71
5	-.11
6	.72
7	.18
8	.51
9	.34
10	.36
11	.04

case it is appropriate to choose the largest value. However this is not always so as can be seen in the next example. While we are gathering empirical evidence as to what is a good threshold value, currently we take a conservative approach and reject similarity values below 0.9.

In the example in Fig 5 the robot re-enters ASR3 via ASR10. The similarity values for ASRs 1-10 are shown in Table 3. Four values stand out, .97, .91, .88, and .77 for ASRs 2, 3, 8, and 10 respectively. With a threshold value of 0.9 we need to choose between 0.97 for ASR2 and 0.91 for ASR3. The highest value, for ASR2, is an example of a false positive.

Table 3: The similarity values for ASR 3* in Fig 5

ASR	prediction
1	.46
2	.97
3	.91
4	.48
5	.64
6	.26
7	.57
8	.88
9	.15
10	.77

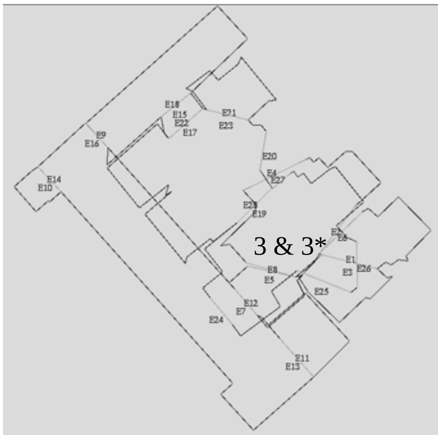

Fig 6 Overlapping ASRs in the global metric map associated with the topological map in Fig 5.

Clearly in this case, the new ASR, ASR3* overlaps both ASR3 and ASR2 in the global metric map (see Fig 6), the ASRs with the highest predictions. Using the global metric map's cycle detection method, ASR3 would match the new ASR3* but ASR2 would not. Further evidence comes in the form of the ASRs' neighbours in the topological map. If the new ASR does match a previously visited ASR then one would expect that its neighbours would match neighbours of the matched ASR. We currently gather evidence in this way for sequences of three ASRs, combining their predictions (see Section 4.3). However ASR3* in Fig 5 is not a good example to demonstrate this. None of its exits matches an exit in ASR3. The exit it would match is in ASR2.

There is evidence to suggest that the new ASR is a combination of both ASR3 and ASR2. This evidence comes in the form of the high predictions for ASR2 and ASR3 which are linked in the topological map and the overlap which occurs in the global metric map. However we need to do further testing to determine if there is any gain in matching under these circumstances. It may be that taking the conservative approach of rejecting the match would be less problematic.

4.3 Topological matching

The idea behind topological matching is to delay committing to a match in the topological map until it can be verified that a sequence of ASRs in the topological map containing the new ASR matches a sequence containing the previously computed ASR. We have found that sequences of order 3 give good results in the environments our robot navigates. A simple environment is used in Fig 7 to demonstrate the process. The robot traverses the environment computing ASRs which are numbered in the order they are encountered. In Fig 7 (b) the robot has re-entered ASR1 via ASR5. A new ASR is constructed and labelled ASR6. The ASR similarities for ASR6 are listed in Table 4. The robot continues to explore, obtaining the sequence of order 3, ASRs 6, 7 and 8 in Fig 7

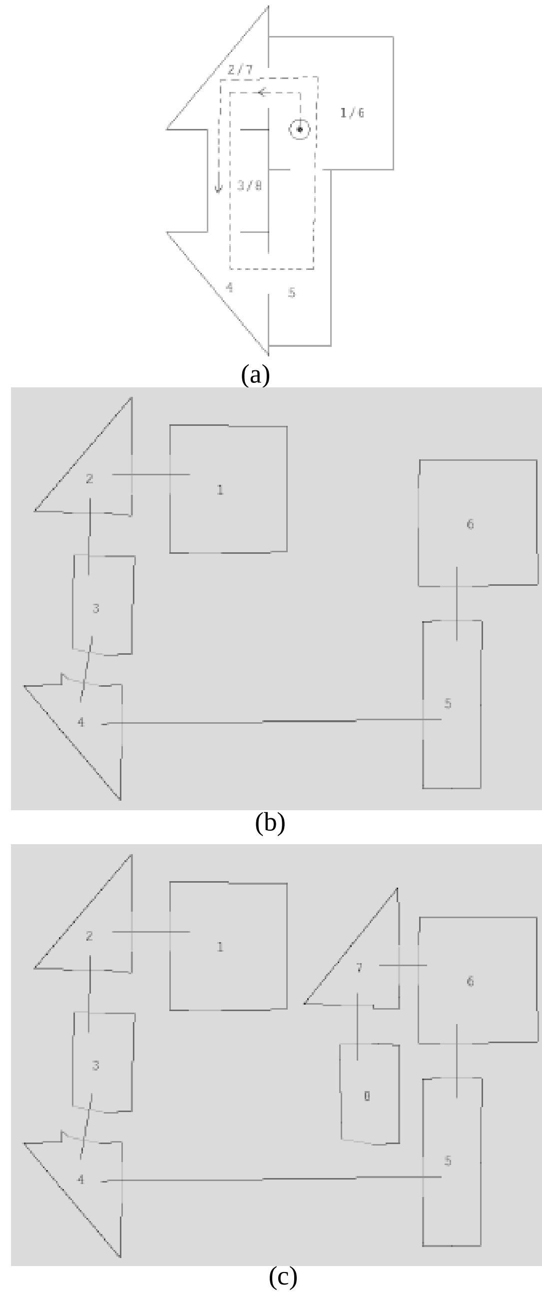

Fig 7 Topological matching. (a) A simple environment (b) The topological map after the robot has re-entered ASR 1 via ASR5. A new ASR5 for the same space, ASR5 is linked to ASR5. (c) The sequence ASR6, ASR7, ASR8 match the sequence ASR1, ASR2, ASR3 confirming the match of ASRs 1 and 6.

(c). The sequence ASR 1-3 is the only sequence of order 3 containing ASR1. Classifying ASRs 7 and 8 give the predictions 0.92 and 0.93 respectively, that they match ASRs 3 and 2. All three predictions are above the 0.9 threshold indicating

Table 4: The similarity values for ASR 6 in Fig 7

ASR	prediction
1	.9
2	.73
3	.74
4	.58
5	.55

a positive match of ASR1 and ASR6 and the topological map can be adjusted to reflect this.

In this example there was only one prediction to be validated. In more complex environments multiple hypotheses would be carried. We are currently investigating how best to converge to a winning hypothesis particularly in environments with a high similarity. Sequences of higher order may be needed in these environments.

5 Conclusion

In this paper we have shown that significant cycles can be detected in a global metric map with some accumulated error. The locations of corresponding ASRs in the global metric map do not need to match exactly therefore some error can be accommodated.

We have also shown how ASRs in a topological map can be recognised from a characteristic subset of their features. Future work will investigate how the two matching approaches can support each other, i.e topological matching can be used to detect cycles in the global metric map and vice versa.

References

[Beeson et al. 2003] Beeson, P., MacMahon, M., Modayil, J., Provost, J., Savelli, F. and Kuipers, B. (2003). *Exploiting local perceptual models for topological map building.* IJCAI-2003 Workshop on Reasoning with Uncertainty in Robotics.

[Bosse et al. 2003] Bosse, M., Newman, P., Leonard, J., Soika, M., Feiten, W. and Teller, S. (2003). *An Atlas framework for scalable mapping*. International Conference on Robotics and Automation.

[Chown et al. 1995] Chown, E., Kaplan, S. and Kortenkamp, D. (1995). Prototypes, Location, and Associative Networks (PLAN): Towards a Unified Theory of Cognitive Mapping. *Cognitive Science* 19: 1-51.

[Gallistel and Cramer 1996] Gallistel, C. R. and Cramer, A. E. (1996). Computations on metric maps in mammals: getting oriented and choosing a multi-destination route. *The Journal of Experimental Biology* 199: 211-217.

[Gutmann and Konolige 1999] Gutmann, J.-S. and Konolige, K. (1999). *Incremental mapping of large cyclic environments*. International Symposium on Computational Intelligence in Robotics and Automation.

[Jefferies et al. 2003] Jefferies, M. E., Baker, J. and Weng, W. (2003). *Robot cognitive mapping: A role for a global metric map in a cognitive mapping process*. Workshop on Robot and Cognitive Approaches to Spatial Mapping.

[Jefferies and Yeap 1998] Jefferies, M. E. and Yeap, W. K. (1998). *Representing the local space qualitatively in a cognitive map*. Twentieth Annual Conference of the Cognitive Society.

[Jefferies et al. 2001] Jefferies, M. E., Yeap, W. K., Smith, L. and Ferguson, D. (2001). *Building a map for robot navigation using a theory of cognitive maps*. IASTED International Conference on Artificial Intelligence Applications.

[Kuipers 2000] Kuipers, B. (2000). The spatial semantic hierarchy. *Artificial Intelligence* 119: 191-233.

[Kuipers and Beeson 2002] Kuipers, B. and Beeson, P. (2002). *Bootstrap learning for place recognition*. 18th International Conference on Artificial Intelligence.

[Kuipers and Byun 1988] Kuipers, B. J. and Byun, Y.-T. (1988). *A Robust, Qualitative method for Robot Spatial learning.* Proceedings of the National Conference on Artificial Intelligence (AAAI-88).

[Rofer 2002] Rofer, T. (2002). *Using histogram correlation to create consistent laser scan maps*. IEEE International Conference on Intelligent Robotics Systems.

[Thrun 1998] Thrun, S. (1998). Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence* 99(1): 21-71.

[Thrun 2001] Thrun, S. (2001). A probabilistic online mapping algorithm for teams of mobile robots. *International journal of robotics research* 20(5): 335-363.

[Thrun et al. 2003] Thrun, S., Haehnel, D., Ferguson, D., Montemerlo, M., Triebel, R., Burgard, W., Baker, C., Omohundro, Z., Thayer, S. and Whittaker, W. (2003). *A system for volumetric robotic mapping of abandoned mines*. International Conference on Robotics and Automation.

[Tomatis et al. 2002] Tomatis, N., Nourbakhsh, I. and Siegwart, R. (2002). *Hybrid simultaneous localization and map building: Closing the loop with multi-hypotheses tracking*. International Conference on Robotics and Automation.

[Yeap 1988] Yeap, W. K. (1988). Towards a Computational Theory of Cognitive Maps. *Artificial Intelligence* 34: 297-360.

[Yeap and Jefferies 1999] Yeap, W. K. and Jefferies, M. E. (1999). Computing a representation of the local environment. *Artificial Intelligence* 107: 265-301.