Robot Mapping with a Topological Map of Local Space Representations

MARGARET E JEFFERIES[1], WAI-KIANG YEAP[2], JESSE BAKER[1]

[1]Department of Computer Science,
University of Waikato, Hamilton
NEW ZEALAND

[2]Artificial Intelligence Technology Centre
Auckland University of Technology, Auckland,
NEW ZEALAND

Abstract In this paper we show how a cognitive mapping theory [1,2] can be used to implement a navigational map for a robot. At the core of this theory is the notion that a representation is computed for each local space the robot visits. These representations are connected in the way they are experienced to form a topological network of local space descriptions. We show how the local space representation is computed using a laser scanner which gives a limited 180^{o} view of the robot's environment. We describe how the topological map grows as the robot moves through different local spaces. One of the most studied problems in robot mapping is that of tracking the robot's location when sensory and odometric errors are accumulating in the robot's map. We argue that the structure of the robot's map should simplify the process of localising the robot and improving the map as the robot becomes more familiar with its environment. Thus we show how our representation facilitates both updating the map and tracking the robot's location.

Key-words: Robot mapping, topological map, localisation, local map, metric map

1 Introduction

We have developed a theory of cognitive maps which has at its core the notion that an autonomous agent building a memory, i.e a "map in the head" termed a *cognitive map*, for the places it visits must first compute a representation for each individual space visited [1,2,3]. The space the agent occupies, termed the *local space*, is defined as the bounded region that appears to enclose the agent. In a cognitive map the representation for the local space would include what the agent saw, how things within it were arranged, what actions were carried out and perhaps the agent's impressions of that part of the environment. The cognitive map develops as the representation for each local space visited is added to a topological network of such representations (see Fig. 3). We initially tested our cognitive mapping theory in a simulation of a robot traversing a 2D representation of a complex hospital environment. The results of this simulation and a comprehensive description of the theory and the algorithms employed can be found in [2]. In this paper we show how our cognitive mapping theory can be applied to the problem of a robot equipped with a laser sensor building a map of its environment. The full expression of the "cognitive" local space as described above is outside the scope of the work we report here. We will therefore refer to the cognitive map as a topological map for the remainder of this paper.

Two themes which emerge from the cognitive mapping/robot mapping studies of artificial intelligence and robotics researchers are: (i) the notion of a representation for the local space [2, 4] versus (ii) a global representation [5, 6] in which conceivably the individual's total experience of their spatial environment could be represented using a single coordinate system. Related to these is the idea of a metric representation [2, 5], where properties such as distance, size and location are explicitly or implicitly represented, versus a topological representation [2, 4, 5, 7, 8] where relationships such as connectivity are represented. The local space could be represented topologically, as in for example, the relationships between some key landmarks [8], or metrically where the structure of the space would be identified within some reference frame. (see [9] for a discussion of topological versus metric representations of space). The individual's total memory for its environment could be stored in a topological representation, as a collection of metric local space representations, each with its own reference frame. The connections between pairs of local space representations would indicate that one could travel directly from one to the other. This idea of a topological network of metric local space representations is central to our cognitive mapping theory [2] and thus the robot mapping we describe in this paper.

Occupancy grids, where the environment is represented by a fine grained grid of cells which are marked as occupied or not, are a popular method for representing the environment metrically. They have been used in both global [5, 6] and topological maps [4]. A significant problem with occupancy grids is their computational complexity. Other methods for representing the environment, termed "feature-based", extract important geometric information (e.g. edges and corners) directly from the sensory data [10, 11, 12].

Our approach to computing a representation of the environment is feature-based; the description of the local space comprises the surfaces and exits on its boundary. It differs from other methods using this paradigm in that we firstly believe it is not necessary to compute an accurate representation from the robot's early experiences of its environment. Secondly we believe that the structure of the robot's map should simplify the process of tracking the robot's location and improving the map as the robot becomes more familiar with its environment. We have noted from studies of psychologists and geographers on cognitive maps for human and animal navigation [13, 14] that the most important piece of information that a navigating agent needs to compute is exits because they tell the agent how it can leave the space it is currently in. From a computational point of view it is much simpler to obtain surface information from exits than it is to obtain exits from surface information. Importantly, we do not require accurate and detailed surface information to obtain a useful description of the local space.

In the next section we describe our exit-based approach to computing a robot's local space. In Section 3 we describe a straightforward method for localising the robot in a local space. We present our conclusions in Section 4.

2 Building a topological map from partial laser scans of the robot's environment.

The topological map comprises a representation for each local space visited with connections to others which have been experienced as neighbours. We refer to local space representation as an Absolute Space Representation (ASR) a term which emphasises the separateness and independence of each individual local space. This term will be used to signify the local space representation throughout the remainder of the paper.

2.1 Constructing the representation for the local space (ASR)

From the "view" the robot has of its environment the ASR algorithm firstly works out where the exits are. It does this by looking for surfaces which occlude other surfaces as it is here that the gaps in the boundary of the local space, i.e. the exits occur. The first step constructs an occlusion map from the readings in the initial 180^o laser scan the robot takes of its environment (see Fig. 1). An occlusion is detected when the disparity between adjacent readings in the laser scan is greater than half a metre. The occlusions are the lines labelled *occ.* 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 environment. The ASR depicted in Fig. 1 is the very first ASR the robot computed at startup. From its initial 180^o view of its environment the robot has no notion of what is behind it. However, one can safely add a point directly behind the robot to the occlusion map, so that the ASR algorithm will form a complete closure around the robot. As the robot enters subsequent ASRs the robot will have the exit just traversed directly behind it.

The next step in the algorithm, calculating exits, requires surface information. Coarse surface information is created by connecting the points which lie between occluding points. These rough surfaces are the dark lines *not* marked *occ* in Fig. 1 (c). 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 [2] for an in depth description of this part of the algorithm. Surfaces outside the exit are eliminated. The point behind the robot ensures that

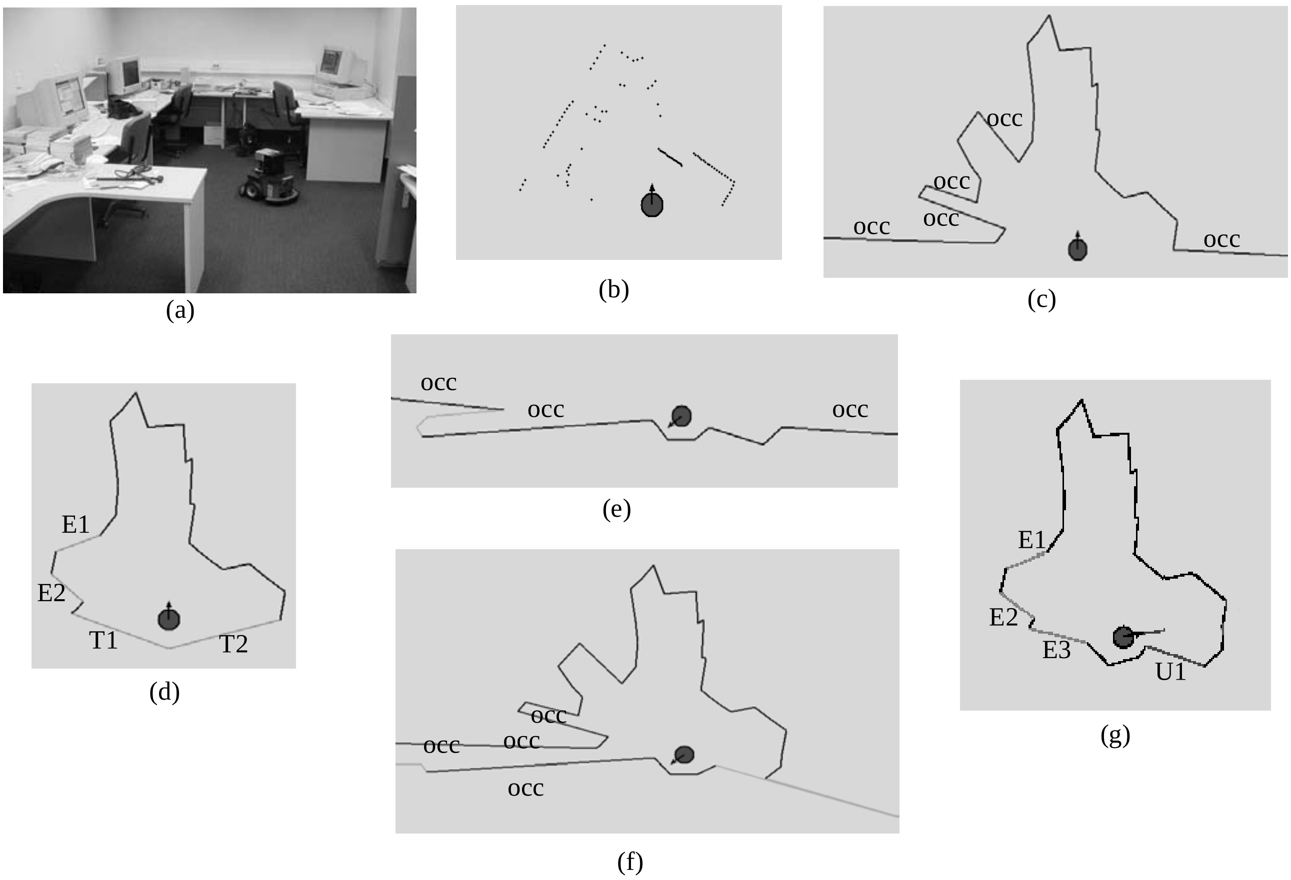


Fig. 1. Computing the first ASR. (a) The environment (b) The laser scan. (c) The first occlusion map constructed from the points in (b). The occlusions are marked *occ*. The black lines are surfaces formed by connecting points between the occluding points. (d) The ASR constructed from the master occlusion map in (c). E1 and E2 are *known* exits, T1 and T2 temporary exits. (e) the occlusion map obtained when the robot turns towards the temporary exits behind the robot. (f) The master occlusion map after (e) has been incorporated into (c). (g) The ASR generated from the master occlusion map in (f). U1 is an *unknown* exit. Note different scales have been used to make the figures fit the space available.

two temporary exits are added to form a complete enclosure (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* in Fig. 1 (g)) and the former as *known (*see *E1-E3* in Fig. 1 (g)). 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. We describe the updating process in the next section. A more complete discussion on the role of exits in the updating process can be found in [2]. Currently we limit the range of the laser scan to 8 metres. Gaps in the boundary which result from surfaces that are outside this range are marked as unknown exits as they perform a similar function as the unknown exits described above.

Note the ASR computed here is a robot-centred representation, i.e. ASRs do not look very “room like”. Clutter such as desks and chairs are significant objects to a small robot and appear to enclose it. A typical cluttered laboratory could give rise to several ASRs.

2.2 Extending the initial local space representation

The initial ASR roughly describes the space surrounding the robot. It may define exits which the robot could use to leave the local space and it identifies regions that require further exploration. At this point the robot could choose to leave via one of these exits or it could decide to stay and explore the unknown regions. In this section we describe this exploration process - the filling in of the unknown regions of the ASR with a “better” description as the robot moves around the local space. For the ASR in

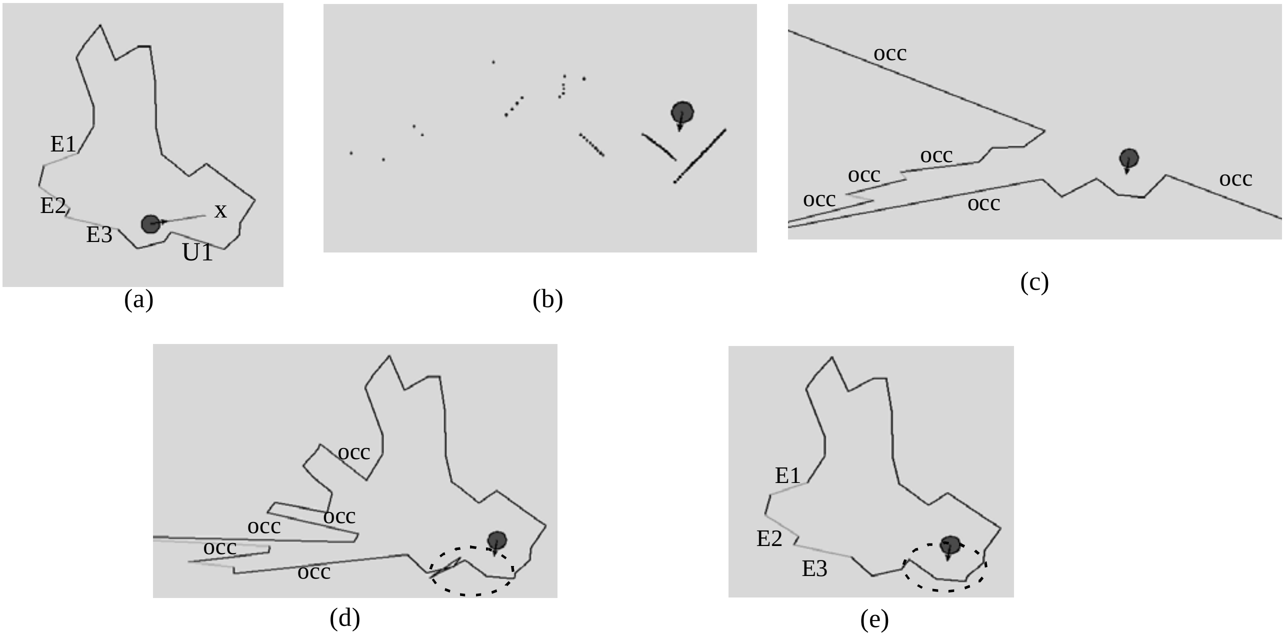


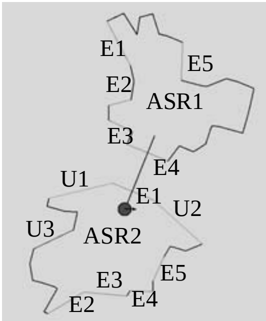
Fig. 2. Investigating an *unknown* exit. (a) A path has been plotted to the *unknown* exit the robot is about to investigate. (b) The laser scan for the position marked x in (a) with robot facing the *unknown* exit. (c) the occlusion map generated from the laser scan in (b) We only show the relevant parts as the map is too large to be include in its entirety. (d) the master occlusion map after the occlusion map in (c) has been integrated into the master occlusion map Fig. 1 (e). (e) the ASR constructed from the master occlusion map in (d). In (d) and (e) the parts which have been modified are encircled with dots.

Fig. 1 (d) the robot would turn towards the temporary exits T1 and T2. Fig. 1 (g) shows the updated ASR. The updating process results in a further unknown exit, U1 – the robot moves towards U1 and orients itself for a better view which it can use to expand the unknown region. As the updating process is similar for both cases we will only describe the updating of U1 below.

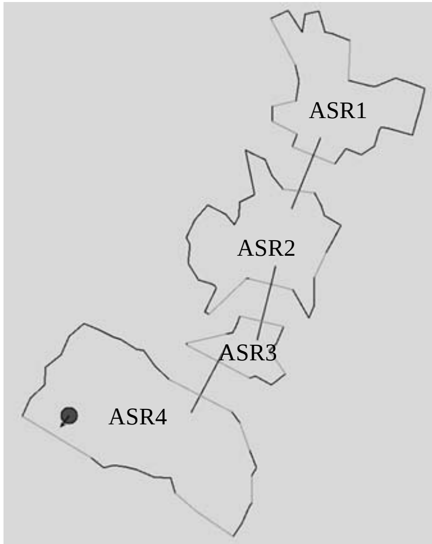
Thus the robot visits each of the unknown exits in turn moving to a position where it can obtain a better view of the region of the ASR they encompass. It plots a path from its current position to a point in front of the unknown exit (see Fig. 2 (a)) and moves to this position (x in Fig. 2 (a)). It then takes a new laser scan and from the resulting set of readings shown in Fig. 2 (b) creates a new occlusion map. Fig. 2 (c) shows the new occlusion map for the point x in Fig. 2 (a). Points from this map in the region of the unknown exit U1 are incorporated into the master occlusion map, i.e. the one depicted in Fig. 1 (f). The updated master occlusion map is shown in Fig. 2 (d) and contains some spurious occlusions. This is due to accumulating error. We describe how we address this problem in Section 4. Finally the ASR is reconstructed from the master occlusion map (Fig. 2 (e)). In this way the master occlusion map serves to collect the data needed to construct an ever better ASR. However it is the ASR itself which is used to decide its own completeness. While it may seem that it would be more efficient to update the unknown exit in the ASR directly using information from the new occlusion map this is not the case. Occasionally a *known* exit is formed along side an *unknown* exit. Once the unknown region is exposed it is clear that a very different exit is called for. Recomputing the ASR ensures that the newly exposed region is appropriately blended into the ASR, i.e. the ASR that results is the same as that which would have been computed if the unknown region had been exposed at the outset.

2.3 The robot moves out of the initial ASR

Immediately the robot moves out of the initial ASR, i.e. crosses one of its exits, it constructs an ASR for the new local space it has entered. The construction of the ASR proceeds mostly as before with one exception. On entering the new local space the robot knows something of the space behind it, i.e. the whereabouts of the exit it used to enter the local space. This exit must be part of the boundary of the new ASR from the outset as it is the connection to the ASR the robot has just left. However by the time the robot comes to compute the new ASR the exit is behind the robot and out of sight. Thus in all ASRs subsequent to the startup ASR the exit used to enter



(a)



(b)

Fig. 3. The topological map. (a) The beginnings of a topological map. The robot has just left ASR1 and computed its initial representation for ASR2. The *unknown* exits U1 and U2 indicate the unexplored region behind the robot. Similarly U3 indicates a region in the ASR yet to be explored. (b) A topological map formed from the robot's experience of our laboratory.

the ASR is added to the master occlusion map. This forces two *unknown exits* to be constructed behind the robot, adjoining the exit just crossed (see U1 and U2 in Fig. 3(a)). Immediately the ASR depicted in Fig. 2 (e) is left and a new ASR constructed for the newly entered space, a connection between them is established forming the beginnings of a topological map (see Fig. 3 (a)). Fig. 3 (b) shows the topological map created by the robot as it explored our laboratory. Note that ASR 1 is somewhat different to the one constructed for the same space and depicted in Figs 2 and 3. A computer laboratory is a very dynamic environment. People come and go, chairs move, bags are placed on the floor; all influence the shape of the boundary of the ASR. We are currently investigating how the ASR can be updated on the fly as these changes occur.

3 Localisation in the ASR

One of the advantages of a cognitive map comprising a topological network of local spaces is that error is constrained to the local space. As the robot leaves one ASR for another it starts afresh. However error does accumulate as the robot moves around the local space. This results in small localised distortions on the boundary of the ASR where unknown exits are updated (see Fig. 4). Over time the robot's represented location will drift significantly from its actual location.

In our system, error correction is a by product of integrating the occlusion map for the robot's current view with the master occlusion map. To correct for rotation error we capture an occlusion map before the robot turns and immediately after, ensuring that whenever the robot turns there is sufficient overlap in consecutive occlusion maps. Occluding points in each of the maps are matched and the average discrepancy used to align the later occlusion map with the earlier one and correct the error in the robot's location. Large outliers in error caused by moving objects are ignored. A similar process is invoked to account for translational error. Fig. 4 shows the effect of this localisation technique on an ASR. Obviously this simple technique is well suited to cluttered environments which have an abundance of occlusions. It would not be well suited to a robot which found itself in a spartan local environment with the only door closed. In such environments other reference points from which to localise would need to be found. We are currently investigating extensions to our method which would account for a wider range of environments.

4 Conclusion

We have shown how a useful topological map for a robot's experience of its environment can be constructed using a theory of cognitive maps. The robot is able to use the map to navigate from the moment

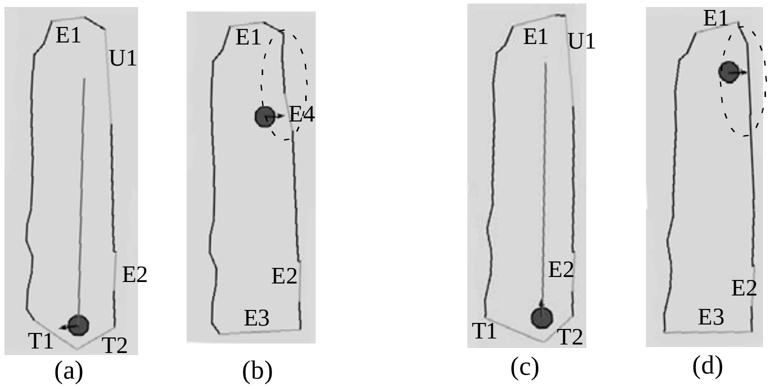


Fig. 4. Localisation. (a) The initial ASR computed without localisation. (b) The distortion in the boundary from odometric errors as the unknown U1 is updated is encircled. (c) A new run with localisation invoked (d) the unknown U1 updated with localisation.

the first ASR is created. The ASR defines the local space the robot can move around in. *Known* exits tell the robot where it can go to next. The initial ASR computed is a rough representation which evolves into a "better" representation as the robot moves around the local space. While we have shown the robot updating the ASR as it explores its unknown parts the robot need not do this. Even if the robot left the ASR without completing its exploration the ASR would still form a useful link in the robot's map of its environment. In a dynamic environment the space could be quite different when the robot returned. But nevertheless it too provides an important link in the robot's map. We have shown how the ASR paradigm for mapping the robot's environment assists in localising the robot in its environment. However we are currently not exploiting the information gathered in the occlusion maps during localisation, in the ASR updating process. We also ignore the information the robot collects while moving itself into position to update an unknown exit. The updating could have been completed long before the robot reached the unknown region. Likewise we are currently not using the occlusion maps collected during localisation to update dynamic information in the ASR. We are currently investigating these problems.

References

1. Yeap, W. K.: Towards a Computational Theory of Cognitive Maps. *Artificial Intelligence* 34 (1988) 297-360.
2. Yeap, W. K. and Jefferies, M. E.: Computing a representation of the local environment. *Artificial Intelligence* 107 (1999) 265-301.
3. Jefferies, M. E., Yeap, W. K., Smith, L. and Ferguson, D.: Computing individual local spaces for a mobile robot: An initial algorithm. *In: International Conference on Cognitive Science* (2001).
4. Langley, P., Pfleger, K., and Sahami, M. Lazy acquisition of place knowledge. *Artificial Intelligence Review*, 11 (1997) 315-342.
5. Thrun, S.: Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence* 99 (1) (1998) 21-71.
6. Elfes, A.: Using occupancy grids for mobile robot perception and navigation. *IEEE Computer* June (1989) 46-57.
7. Kuipers, B. The Spatial Semantic Hierarchy. *Artificial Intelligence* 119 (2000) 191-233.
8. Levitt, T.S. and Lawton, D.T. Qualitative Navigation for Mobile Robots, *Artificial Intelligence*, 44 (1990) 305-360.
9. Yeap, W.K. and Jefferies, M.E. On early cognitive mapping. *Spatial Cognition and Computation* 2 (2000) 85-116.
10. Tomatis, N., Nourbakhsh, I. and Siegwart, R.: Simultaneous localization and map building: A global topological model with local metric maps. *In: International Conference on Intelligent Robots and Systems* (2001) 421-426.
11. Crowley, J. L.: Navigation of an intelligent mobile robot. *IEEE Journal of Robotics Research* 1 (1) (1985) 31-41.
12. MacKenzie, P. and Dudek, G.: Precise positioning using model-based maps. *In: IEEE International Conference on Robotics and Automation* (1994) 1615-1621.
13. Gallistel, C. R.: Animal cognition: The representation of space, time and number. *Annual Review of Psychology* 40 (1989) 155-189.
14. Orians, G. H. and Heerwagen, J. H.: Evolved responses to landscape. In: Barkow, J. H., Cosmides, L. and Tooby, J. (eds.): *The Adapted Mind: Evolutionary Psychology and the Generation of Culture*, Oxford University Press: New York (1992) 556-577.