Representing Large Scale Space - A Computational Theory of Cognitive Maps

Margaret Jefferies and Wai-Kiang Yeap
Artificial Intelligence Laboratory
Department of Computer Science
University of Otago, Dunedin, New Zealand
e-mail: jefferies_me@cs.otago.ac.nz

Abstract

We present a computational theory of cognitive maps which proposes that the cognitive map, a representation for a viewer's experience of their spatial environment, comprises two loosely coupled modules, the raw cognitive map and the full cognitive map. The raw map is a representation for the spatial arrangement and physical characteristics of surfaces. The various interpretations (or conceptualisations) that are derived from the raw map form the full map. The cognitive map is built from the bottom up, starting with the sensory input. A representation termed an Absolute Space Representation (ASR) is computed for each local space visited and these ASRs connected in the way they are experienced are the raw map. We discuss the current status of our work on computing a cognitive map and in particular present a new algorithm for computing an ASR.

1 Introduction

The term "cognitive map" was first coined by Tolman (Tolman 1948) and there have been numerous theories proposed since then as to what should constitute such a map. Tolman was probably using the term to describe the spatial layout of the environment but after Lynch's (Lynch 1960) work it became apparent that the notion of a cognitive map is a complex one. For the roboticist, interested primarily in computational and navigational issues, the cognitive map is as envisaged by Tolman. But to most others (eg. geographers and psychologists (see (Downs & Stea 1973) (Siegel & White 1975)) a cognitive map is much more than this. The human's spatial behaviour governs what is represented and the map is thus affected by such things as mode of travel, past experiences, preferences and attitude.

Cognitive mapping is therefore a complex process which involves both one's perception and conception of the outside world. However we argue that perception and conception are two separate processes; that a cognitive mapping process must firstly compute a representation for what is actually perceived and then an abstraction process operates to find a mapping between the physical representation and one's conceptual knowledge. To the viewer these processes are indivisible in that one knows what one sees and one sees what one knows. But knowing about the existence of some object in space implies that the object has a spatiality of its own and a structure as well as an extension in space. This structure is an integral part of conception; one cannot conceive of what something is without firstly being aware of its form. Thus cognitive mapping is a multi-level process. Studies which have emphasised only one level, either the perceptual (eg. work on autonomous mobile robots) or the conceptual (eg. early models) are at best incomplete and have very often asked questions which are inappropriate to be solved at that level.

In developing a computational theory of cognitive maps we stress the need to understand what needs to be computed at each level of the cognitive mapping process and why it needs to be computed (Yeap 1988) (Yeap & Handley 1990). It is important that the process be studied as a whole from perception to conception and generally in that order. Our investigation of the process begins with Marr's (Marr 1982) computational theory of vision and, as in Marr's work, the notion of a representation is central to our study.

2 A Computational Theory of Cognitive Maps

The key to any successful computation is the correct processing of inputs. For a cognitive map the inputs are what is delivered by the senses, and in particular perception delivers the physical characteristics of surfaces (for example size, orientation and distance from the viewer). This suggests that one firstly computes a representation for the physical environment. Various interpretations of this initial representation give rise to the different conceptual views of places visited. Our theory therefore proposes that the cognitive map is divided into two loosely coupled modules. An early cognitive mapping process computes a representation for the spatial arrangement and physical properties of the surfaces in the sensory input. It is termed a "Raw" Cognitive Map to indicate that the information it contains is uninterpreted. A later cognitive mapping process computes a representation for the conceptual world - it is called a "Full" Cognitive Map to indicate the full richness of the map as a cognitive representation. Together the raw and full cognitive maps represent a viewer's entire experience of their spatial environment (see Figure 1). It is assumed at the input

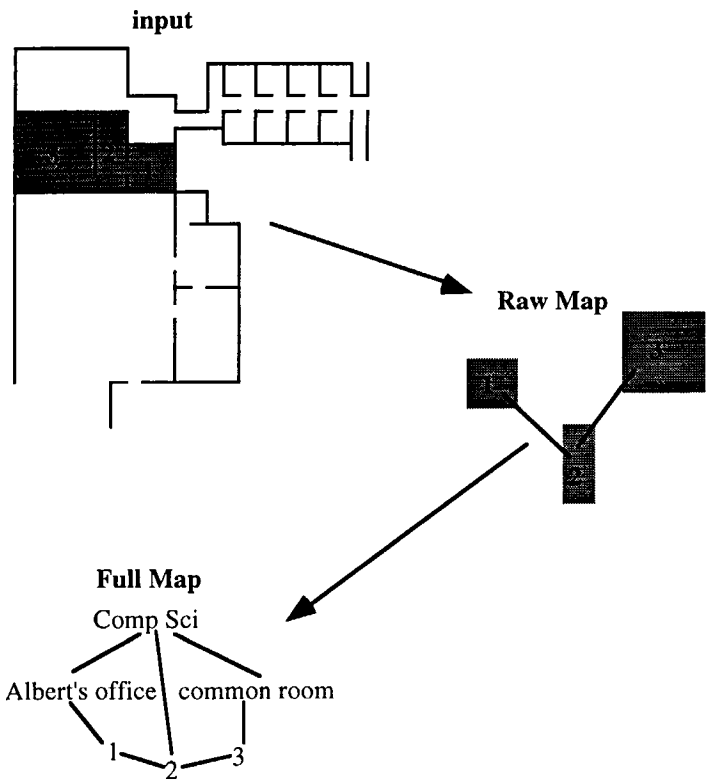

Figure 1. The Cognitive Mapping Process. The shaded areas in the input indicate the surfaces which have been input to the process so far. The actual inputs are egocentric views.

level there is some mechanism for processing sensory input and capturing the spatial arrangement and physical qualities of surfaces. For vision the resulting representation is akin to Marr's (Marr 1982) 2 1/2D sketch.

2.1 Computing a Raw Cognitive Map

Based on the notion that a person can experience only a part of the environment at any moment, Yeap (Yeap 1988) argued that it is important that each local space visited have an explicit representation in the raw map. By local space we mean the space the viewer currently occupies. Such a representation is termed an Absolute Space Representation (ASR), a term which emphasises the local nature and the independence of each ASR representation. This independence is achieved by way of a local coordinate system for each ASR and is motivated by the fact that humans do not build a global map of their environment. From a practical point of view any mechanical or computational error will be confined to a single ASR, thus minimising the accumulation of error which is inherent in large global representations.

Our theory argues that one does not immediately compute a representation for the spatial extent of individual objects in the perceptual array. Firstly one needs to know the extent of the space one is currently in and from this one can then determine where specific objects are. Intuitively we might argue that what we see tends to be described as what objects we see rather than where. It was not until Marr's (Marr 1982) seminal work on computer vision that any attention was paid to the simple fact that our eyes deliver descriptions of surfaces seen at various distances in front of us. That objects appear to us with a spatiality derived from where they exist in space is evidenced by Schulman (Schulman 1983) who discussed the importance of the whereness of things in remembering them in one's memory. Nelson's (Nelson 1986) work, which emphasised event knowledge as the basic mental representation from which one's concept's and categories are derived, also provides support in that (object) concepts and categories are not learned without a context. Thus in this sense the ASR behaves as a container for objects and actions in space.

The individual ASRs are connected together in the way in which they are experienced to form the raw map. However, defining what constitutes a local environment (and hence an ASR) is a problem. This is particularly the case when one has to explore the environment in a piecemeal fashion. We refer to this as the boundary problem for computing an ASR. One needs to be clear as to the role of ASRs in the raw map. While they are the building blocks for constructing the raw cognitive map they do not by themselves define the map. Equally important is the way in which individual ASRs are related to one another. Until one is able to organise one's ASRs in such a way that the map contains enough information to tell us

how the different parts are related one has but a blurred map. To achieve a useful cognitive map one has to be familiar with the environment and being familiar with the environment implies that one has to be able to recognise parts which have already been visited. However recognition depends to a large extent on the conceptual knowledge acquired about places and objects and is thus more appropriately tackled at the level of the full map. ASRs are computed to identify the current space in which one is bounded and to which one can then relate one's current actions.

ASR Computations The result of ASR construction is to partition the environment in such a way as to derive a natural boundary to describe one's local space, ie. the space one currently occupies. We refer to such an approach, which describes each local environment as a space of its own, as "space-based". This is in contrast to representation schemes which are object-based in that they focus on the relationship between individual objects (or landmarks) in the environment. Robotics researchers' attempts to compute a representation of the empty space in the environment for path planning came close to tackling the problem of computing a space-based representation. Early examples of such systems (the SHAKEY (Nilsson 1969) and JPL (Thompson 1977) robots) used a global coordinate system to describe the environment which they later explored to build a representation for path planning. These early works foreshadowed the development of more complex representations of the environment as used in current autonomous mobile robots.

These robots mostly partition their free space into convex polygons (Chatila 1982), (Crowley 1985), (Iyengar et al. 1985) whereupon they can plan efficient paths which move from one convex polygon to another. Our main criticism of these methods is that they partition the environment in an unnecessary way and ignore why and how humans perceive their surroundings. Robotics engineers see the problem of representing the robot's environment as one of finding the most suitable partition of free space rather than one of finding a representation which will contain the robot. Many of the later problems they have encountered (such as the need to recognise revisited ASRs) have to be dealt with in an ad hoc manner (such as providing precise measurements for their sensors). Meng and Kak (Meng & Kak 1993) recently remarked that the robotics' approach is "non-human-like". We agree but also believe that their problem lies not in their failure to observe human navigational behaviour but in a failure to understand what information is made explicit and at what level in a cognitive mapping process. See (Yeap, Naylor & Jefferies 1990) for a more in-depth review of the above works.

Computing an ASR - our algorithm. Our early attempts at computing the local space (Yeap, Naylor & Jefferies 1990) were primarily concerned with working out which surfaces in the viewer's immediate vicinity appeared to form a closure surrounding the viewer. Such surfaces would form a natural boundary for describing the ASR. The lack of an adequate notion of how such surfaces should be connected meant the results were often less than satisfactory. We have recently developed a new algorithm which is based on the observation that when humans were wanderers, before they needed to compute a raw cognitive map, it was still essential to have a clear description of the current space. Probably the single most important piece of information in terms of one's survival, was a description of the exits so that one would know where to run from danger and where to go to next to look for food. Thus one would firstly look for exits and one would quickly form an impression of the local space. It follows that in this case the exit information must be obtained prior to developing the ASRs and the raw map. Note that identifying which surfaces form the boundary of an ASR indirectly provides us with the exits in the boundary and conversely identifying the exits should also tell us where the boundary surfaces are. However the latter approach is much easier.

The exits are identified as follows: whenever one surface is viewed as occluded by another surface, a gap exists which we label as an occluded edge (for example, see edges BC, FG, and IJ in Figure 2). An exit is the shortest gap covering the occluded edge. By covering, it is meant that the viewer must cross that gap in order to reach the occluded edge. For example, in Figure 2, the exit JF is said to cover the occluded edge FG (and IJ). Computing such an exit turns out to be straightforward - find the nearest point on a surface which is not on the same side of the occluded edge containing the surface with the occluding vertex. Thus, the points to consider when calculating the exit at point F (B) are H and J (C, E, G, and J) and the exit becomes JF (BC).

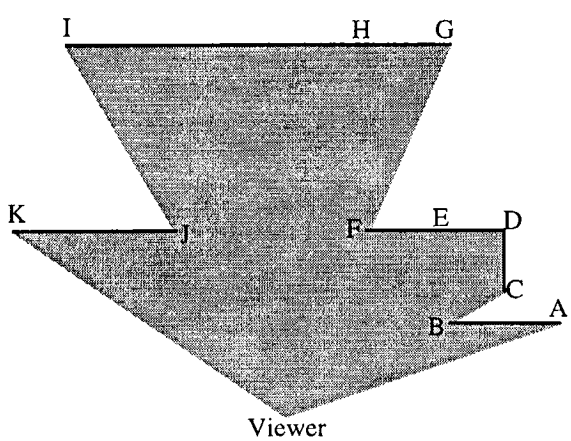

Figure 2. A view of the environment (in 2D).

Once the exits are computed, the boundary is obtained by removing exits and surfaces which are perceived as outside of the current ASR, a process we have dubbed "trimming" (see Figure 3).

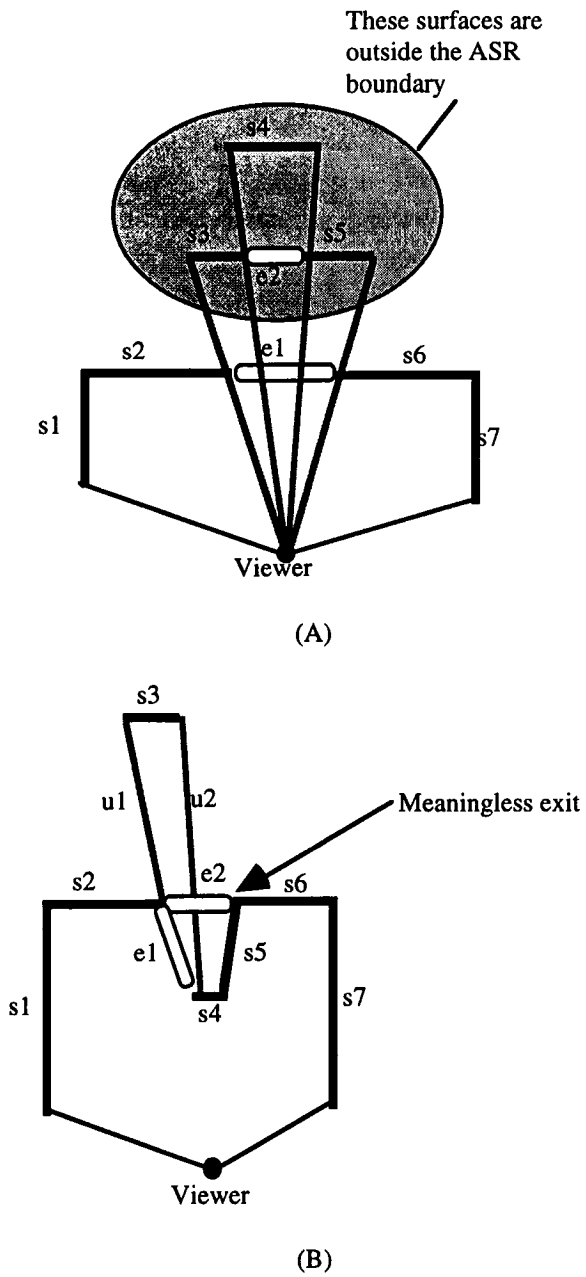

Figure 3 (A) The surfaces in the shaded area beyond exit e2 are trimmed from the ASR. (B) A situation where an exit which is not in view is an inappropriate selection for the boundary. In this case exit e2, and the surfaces beyond it would be excluded.

The algorithm, for computing the boundary is given below:

CALC_ASR

Input: The list of ordered surfaces in the current view. Label this list POTENTIAL_ASR

1. With each occluding vertex in POTENTIAL_ASR do
   Calculate an exit
   1.1 Label this vertex EXIT_vertex1
   1.2 Create an edge between EXIT_vertex1 and the vertex it occludes. Call this edge OCCLUDED_EDGE.
   1.3 Split the list of surfaces in Potential_ASR into two lists, separated by OCCLUDED_EDGE. Call the list of surfaces containing the EXIT_vertex1 and the surfaces extending to the periphery, SET1.
   Call the list of surfaces containing the occluded vertex and the surfaces extending to the opposite periphery SET2.
   1.4 Find the closest point to EXIT_vertex1 from the list of surfaces in SET2. Label this point EXIT_vertex2.
   1.5 Insert the exit comprising EXIT_vertex1 and EXIT_vertex2 in its appropriate position in the ordered list POTENTIAL_ASR.
   1.6 Label the exit as doubtful if EXIT_vertex2 is the occluded vertex.
   ;;required by EXTEND_ASR
   Trim surfaces outside the boundary
   1.7 Remove all the surfaces which lie between the exit and the surfaces which contributed its vertices.
2. Add side exits to POTENTIAL_ASR to account for the peripheries of the current view.
3. Return POTENTIAL_ASR as the ASR boundary

Note that there are two kinds of exits computed, which we dub DL (doubtless) and DF (doubtful) exits. The former has no occluded vertices ie. both vertices must be clearly visible. The point is, a DF exit needs to be updated when more information becomes available. This allows the shape of an ASR to be modified later and is important since not all parts of the environment are perceived with equal clarity. Consequently, the shape of an ASR keeps changing as one walks down, say a long corridor.

To update a DF exit, we use a simple algorithm: replace the previous description with the latest. That is, if we can identify those surfaces incident on each vertex of a DF exit in the current input, we replace the old description with the new one. For example in Figure 4, EDF10 from ASR2.1 is replaced because both EDL2 and S17 match with surfaces in the new view (view 2). The result is ASR2.2. However, one consequence is that some information may be lost in the process but the amount is insignificant. More importantly, the information lost is rarely significant to humans. The algorithm for updating an ASR in this way is outlined in detail below. Note that because updating DF

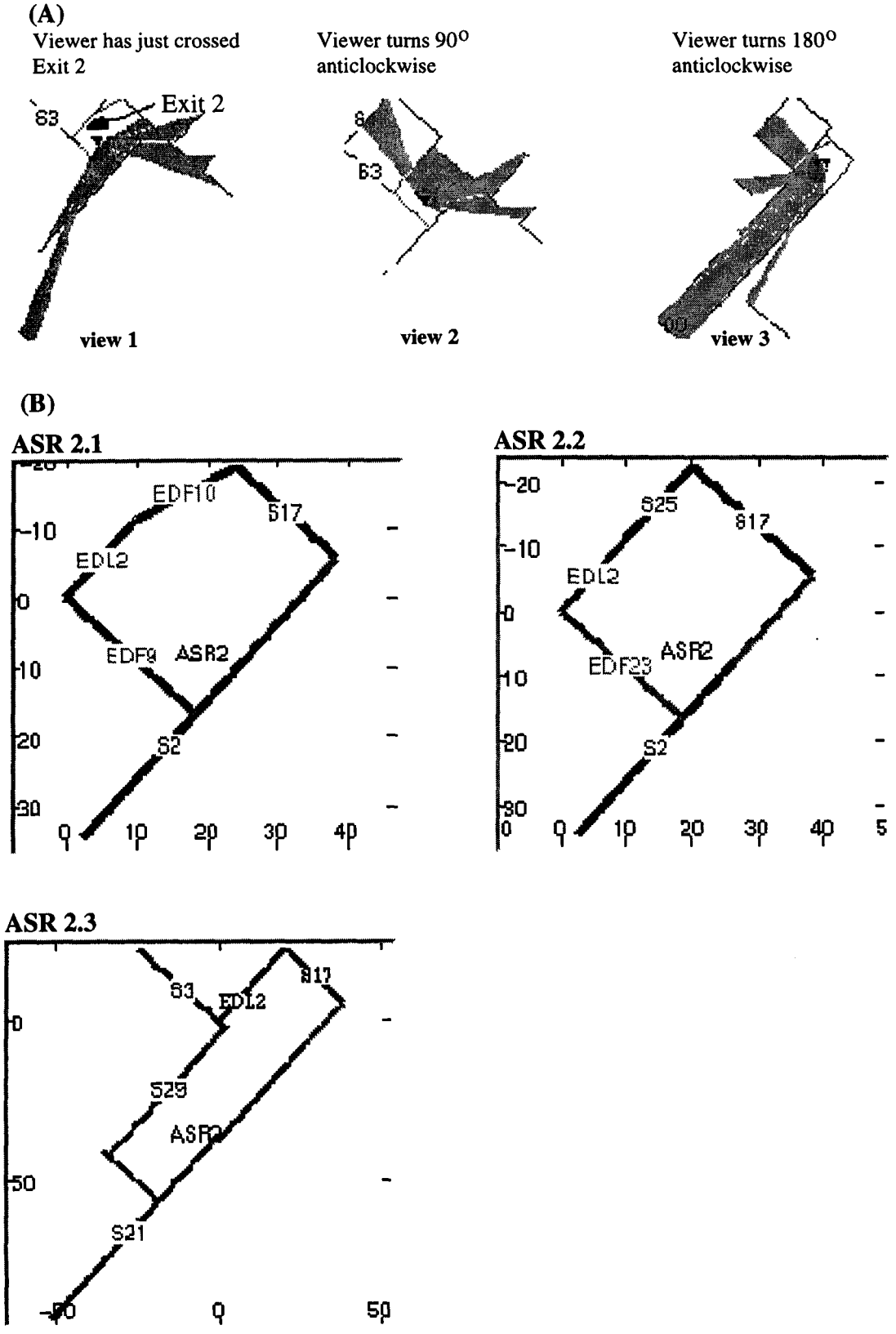

Figure 4. Extending an ASR

exits actually extends the ASR we call this algorithm extend_asr.

EXTEND_ASR

Input: The list of surfaces in current view, call this CURRENT_VIEW.
The list of surfaces in the current ASR description, call this CURRENT_ASR.

1. Construct a new ASR description using CURRENT _VIEW, call this NEW_ASR.
2. For each doubtful exit in CURRENT_ASR do
   2 .1 Call the surfaces which contribute vertices to the exit CONTRIBUTES1 and CONTRIBUTES2.
   2.2 If CONTRIBUTES1 and CONTRIBUTES2 occur in NEW_ASR then replace the exit with any surfaces which lie between them.
3. Return CURRENT_ASR as the updated ASR.

That the ASR works lies in the dynamic nature of the process of computing exits and subsequently that of computing the boundary itself. Figure 5 shows the environment and a path we have used to test our algorithms. Figure 6 shows the ASRs constructed by a simulated robot traversing this path. Note that by projecting the 2 1/2 D representation for a surface on to the ground we can represent the surface in 1 1/2D. Because ASR computations are only concerned with surfaces which obstruct the viewer's line of sight, 1 1/2 dimensions are sufficient at this level.

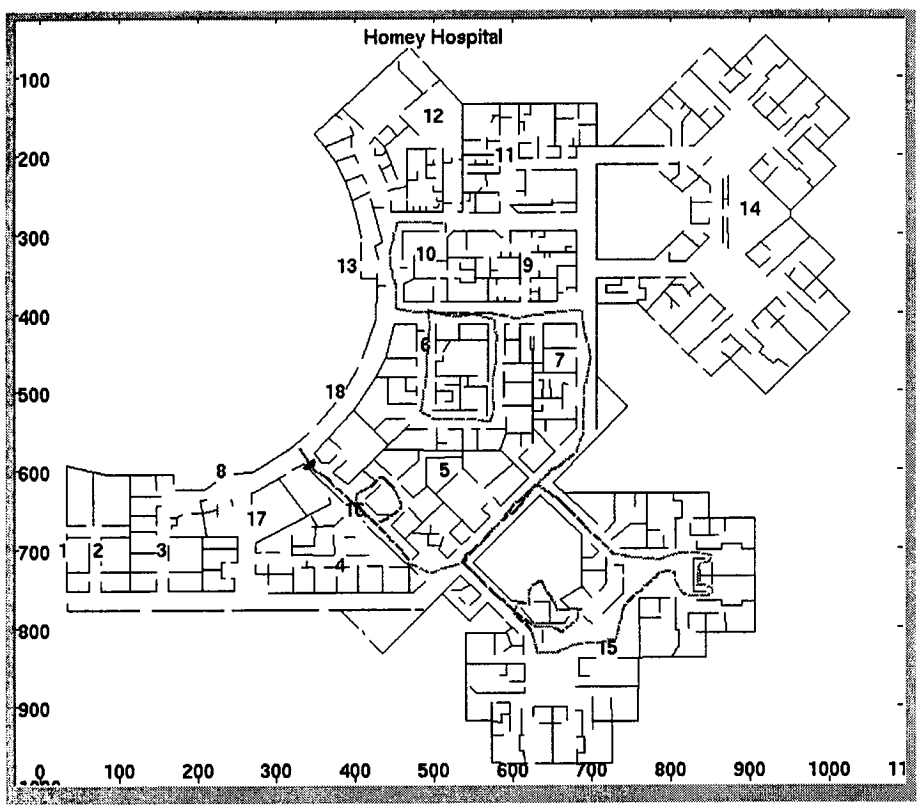

Figure 5. The environment used to test our algorithms. Reprinted with permission from (Peponis, Zimring & Choi 1990) The "eye" represents a simulated robot/person. The ASRs computed for the path through this environment are shown in Figure 6.

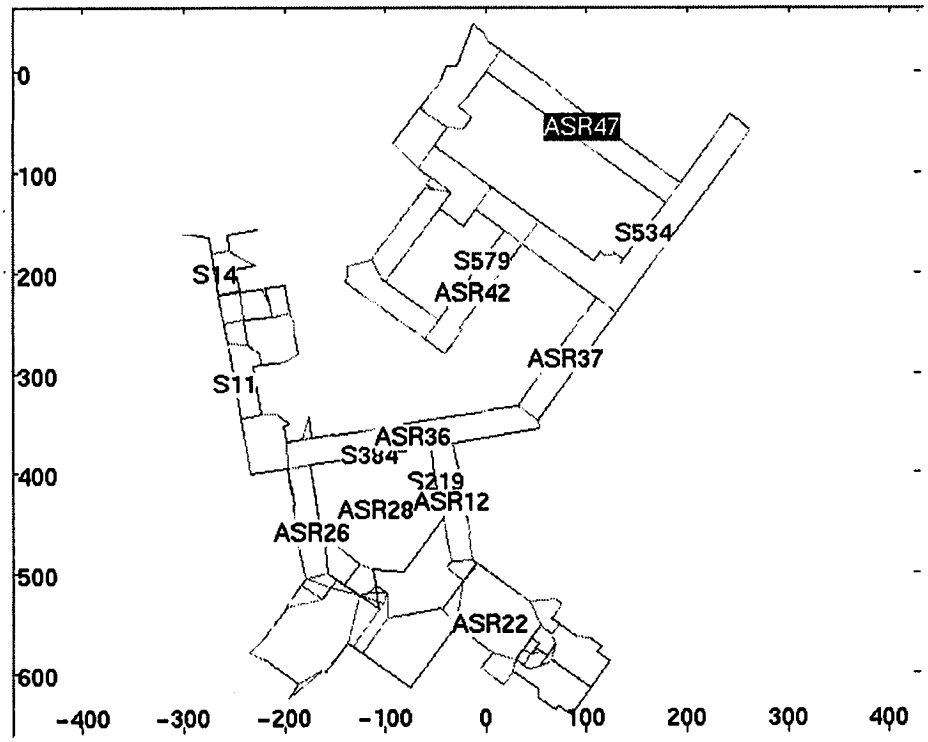

Figure 6. The ASRs computed from the path traversed in Figure 5. Note that the ASRs are laid out in this way for display purposes. Each ASR has its own local coordinate system and is connected to adjacent ASRs only via traversed exits.

2.2 Computing a Full Cognitive Map

The raw map contains much useful information but apart from its network of ASRs it lacks any organisation. Its only task is to capture the essential characteristics of surfaces and the way they relate to each other. Any further structuring of its raw information is left for the full cognitive map. Therefore the full map is proposed as a collection of representations which grows out of the raw map, each of which imposes a particular way of looking at (some of) the information in the raw map. One important notion is the concept of a "place" and in particular, the concept of "activity places" (Steinitz 1968). What is needed at this level is an abstraction process which makes a place explicit from what is known in the raw map. A straightforward approach is to form places from groups of ASRs. Many experiments (eg. (Steinitz 1968) (Hirtle & Jonides 1985)) have demonstrated the hierarchical nature of a place representation. Humans reason about their environment at a variety of levels as in for example, "the university is in North Dunedin and you will find Albert's office in the attic of the Archway West building." Places are grouped together at many different levels and our full map reflects this. A full cognitive map representation of the above example would see the place "Albert's office" at the lowest level mapping directly to an actual ASR in the raw map. At the next level "Albert's office" along with other places are grouped to form a higher-order place, "attic", and so on (see Figure 7).

A program has been written which simulates a person moving through the environment building up a place representation hierarchy by being told (by a human operator) whereabouts she is at each step. The hierarchical relations are computed automatically. Each place in the full map has its own descriptor with four entries: NAME, IN, HAVE and EXIT. NAME is required so that the place may be referenced; IN captures the place in the level above in the hierarchy to which it belongs; EXIT indicates the ways in which a place may be left for other places. The entry HAVE captures the lower level places and ASRs which comprise a place, ie. the places below a node in the hierarchy.

We have investigated the problem of path planning to demonstrate the uses of a cognitive map. Along with environmental psychologists, urban planners and designers we argue that both modules of the cognitive map are important in way-finding; however, such a representation used in its entirety is not as popular with AI and robotics researchers. It is possible to plan paths using only the information made explicit in the raw map and this is the approach adopted by many roboticists. Psychological studies of humans' judgements of direction and distance information to various places/landmarks in the environment (Hirtle & Jonides 1985) and humans' subjective organisation of the environment in terms of a hierarchy of places suggest that a more efficient approach is initially to plan in a hierarchy of abstraction spaces. The

planning approach which we have adopted uses information from both the raw and full maps. Our aim is not to design a new planning algorithm as such but to investigate how information in a cognitive map can be used for this task. For example, one such study (Yeap & Robertson 1990) investigated how direction information between two known places might be obtained in ways commonly observed by humans, using a simulated walk through the environment. We achieved this in the program by first formulating a crude plan using information in the full map where higher level places in the hierarchy form intermediate goals. A simulated walk follows this plan to determine the direction of the goal from the start, which is then used to constrain a search through the raw map for a

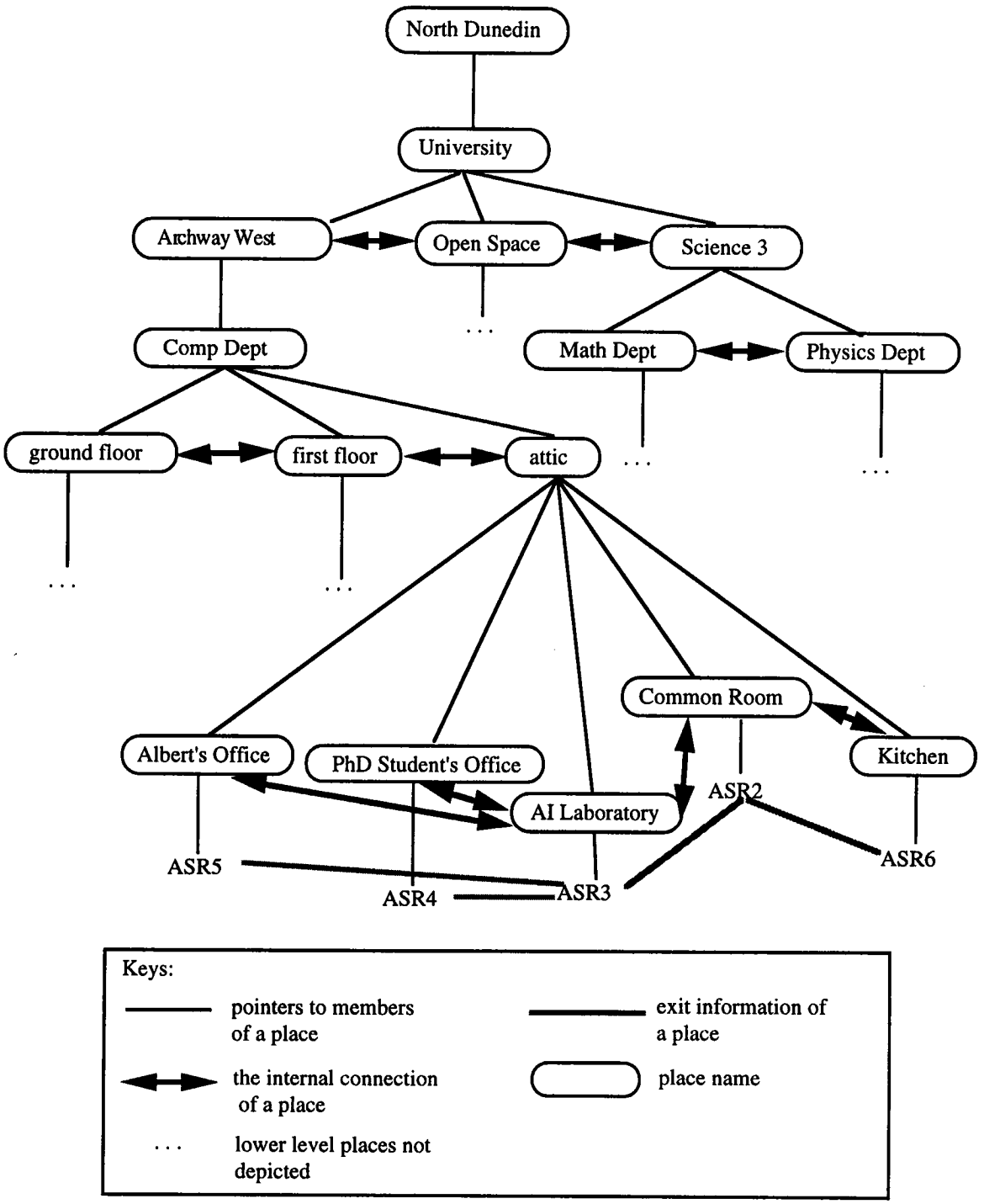

Figure 7. A hierarchy of place representations.

more "direct" route. Next, we asked what happens when the plan fails in the execution phase due to an unforeseen change in the environment. In analysing this problem (Yeap & Finnie 1992) we discovered that it is possible to avoid excessive searching of the raw map by using various strategies which make use of conceptual knowledge one already has of the current place. For example, if the plan has failed because of a locked door which separates two different places one should first find a way out of the current place rather than searching for a path heading towards the goal from the current position.

3 On Cognitive Maps

We have approached the problem of representing a viewer's experience of their spatial environment from an Artificial Intelligence perspective. We argue that much can be learnt about how to solve the computational problems of cognitive mapping by studying how humans do the same thing. In so doing one needs to address the cognitive issues at all levels of the cognitive mapping process. These issues have been addressed in a variety of forms by environmental psychologists, geographers and urban planners. Their approaches to representing the environment include the use of landmarks in routes and as organising features (Lynch 1960) (Presson 1988) (Couclelis et al. 1987) (Gopal, Klatzky & Smith 1989), are often characterised as either route or survey maps (Gopal, Klatzky &Smith 1989) (Kirasic, Allen& Siegel 1984) and are hierarchical in nature (Hirtle & Jonides 1985) (Couclelis et al. 1987) Do any of these relate to our work? To some extent they all do. Landmarks are but one interpretation of the information stored in the raw map and would play an important role in organising the full map. We are currently investigating the role of landmarks in the full map. Does our cognitive map contain route or survey knowledge? It does to a certain extent. Currently we don't store knowledge of actual routes per se, but route knowledge is implicit in both the raw and full maps. We can work out from the full map which places one has to go through to get from one place to another but how we actually do it, ie. the knowledge of the physical connections between ASRs is stored in the raw map. Our raw map somewhat resembles a survey map, how much depends on how familiar one is with the environment. It is integrated in that each new ASR is absorbed into the overall structure as it is encountered for the first time. But from within an ASR one does not know "where" it is in relation to ASRs other than to its immediate neighbours. The global overview which is typical of survey maps is a characteristic of the full map. By searching its hierarchy (up or down) one can always determine where one is in relation to other places in the environment.

Qualitative representations have been preferred by many researchers, particularly those in robotics and Artificial Intelligence (Kuipers & Byun 1988) (Zimmer & Puttkamer 1994) who have tackled the problems inherent in brittle geometric representations. One of the major problems we have encountered with qualitative spatial representations is the lack of a clear notion as to what information is appropriately represented in this way and at what level in the cognitive mapping process. The only unifying factor is that they do not use precise geometry. Kuipers and Byun's (Kuipers & Byun 1988) model comprises three layers: procedural knowledge for movement, a topological model for the structure of the environment and metrical information for geometric accuracy. The first layer in the actual representation of the environment is the topological model which comprises locally distinctive places connected by travel edges (the sequence of moves followed between adjacent distinctive places). We argue that it is inappropriate to be determining which parts of the environment are the most distinctive at this level. The very notion of a distinctive place requires a level of conceptualisation which goes far beyond representing the raw information provided by the senses. To call this a first level representation is to pretend the crude information which underpins the "distinctive places" did not exist (at least not initially). The cognitive map of Zimmer and Puttkamer's (Zimmer & Puttkamer 1994) robot consists of a topological representation of the environment which is qualitative in the sense that sensor information itself is represented rather than any interpreted objects and their underlying geometry. However if the robot is to exhibit behaviour which rises above the level of an idiot savant then it needs to comprehend the "where" and "what" of its environment. Geometry is important for interpreting sensor information. The problem one has to solve at this level is what form the geometric representation should take so that in spite of its imprecise and incomplete nature reasonable decisions can be made with it.

4 Conclusion

A computational theory of cognitive maps is being developed which draws on our observations of human and animal behaviour for its ideas. We stress that in developing such a theory the primary concern should be to understand what information needs to made explicit and why at each level of the cognitive mapping process. It is not adequate to say that the information is qualitative or quantitative, metric or nonmetric; what is important is that the appropriate information is made available at each step. We don't yet fully understand the variety of forms the representations at each level may take; but whatever form is presented, provided the information is appropriate for that level there should be mechanisms in place to utilise it at this or higher levels of the process.

We also stress the importance of understanding what needs to be computed from the bottom up (ie. starting with sensory information). This provides a clear idea of what is available and hence what can be done. Ultimately the measure of performance of a system is its behaviour; if the simulated behaviour doesn't match the observed behaviour then the model needs to be improved.

References

Chatila, R. 1982. Path Planning and Environment Learning in a Mobile Robot System. Proceedings of European Conference on Artificial Intelligence, 211-215.

Couclelis, H.; Golledge, R. G.; Gale, N. and Tobler, W.. 1987. Exploring the Anchor-Point Hypothesis of Spatial Cognition. *Journal of Environmental Psychology (1987) 7* : 99-122.

Crowley, J. L. 1985. Dynamic World Modelling for an Intelligent Mobile Robot Using a Rotating Ultra-Sonic Ranging Device. Proceedings of the 1985 IEEE International Conference on Robotics and Automation, 128-135.

Downs, R. M. and Stea, D. 1973. *Image & Environment: Cognitive Mapping & Spatial Behavior.* Aldine Publishing Coy., Chicago.

Gopal, S.; Klatzky, R. L. and Smith, T. R. 1989. Navigator: A Psychologically Based Model of Environmental Learning Through Navigation. *Journal of Environment Psychology* 9: 309-331.

Hirtle, S. C. and Jonides, J. 1985. Evidence of Hierarchies in Cognitive Maps. *Memory & Cognition* 13(3): 208-217.

Iyengar, S. S.; Jorgensen, C. C.; Rao, S. V. N. and Weisbin, C. R. 1985. Learned Navigation Paths for a Robot in Unexplored Terrain. *AI Applications: The Engineering of Knowledge-based Systems* Ed. C. R. Weisbin. 148-155.

Kirasic, K. C.; Allen, G. L. and Siegel, A. W. 1984. Expression of Configurational Knowledge of Large-Scale Environments - Students' Performance of Cognitive Tasks. *Environment & Behavior* 16(6): 687-712.

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), 774-779.

Lynch, K. 1960. *The Image of the City.* Cambridge, MA, MIT Press.

Marr, D. 1982. *Vision: A Computational Investigation into the Human Representation and Processing of Visula Information.* New York, W.H. Freeman Coy.

Meng, M. and Kak, A. C. 1993. Mobile Robot Navigation Using Neural Networks and Nonmetrical Environment Models. *IEEE Control Systems* October, 1993: 30-39.

Nelson, K. 1986. *Event Knowledge: Structure and Function in Development.* Lawrence Erlbaum Associates.

Nilsson, N. J. 1969. A Mobile Automaton: An Application of Artificial Intelligence Techniques. Proceedings of the 1st International Joint Conference on Artificial Intelligence, 509-519.

Peponis, J.; Zimring, C. and Choi, Y. K. 1990. Finding The Building In Wayfinding. *Environment and Behavior* 22(5): 555-590.

Presson, C. C. 1988. Special section: Landmarks in Spatial Cognition and Spatial Development. *British Journal of Developmental Psychology* 6: 369-393.

Schulman, A. I. 1983. Maps and Memorability. *The Acquisition of Symbolic Skills* : 359-367.

Siegel, A. W. and White, S. H. 1975. The Development of Spatial Representations of Large-Scale Environments. *Advances in Child Development & Behavior by H.W. Reese* Vol 10.

Steinitz, C. 1968. Meaning and the Congruence of Urban Form and Activity. *Journal of the American Institute of Planners* 34: 233-248.

Thompson, A. M. 1977. The Navigation System of the JPL Robot. Proceedings of the 5th International Joint Conference on Artificial Intelligence, 749-757. Cambridge MA.

Tolman, E. C. 1948. Cognitive Maps in Rats and Men. *Psychological Review* 55(4): 189-208.

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

Yeap, W. K. and Finnie, G. A. 1992. Re-planning from a Cognitive Map. Proceedings of the 2nd Pacific Rim International Conference on Artificial Intelligence, 96-102. Korea. Korea Information Science Society.

Yeap, W. K. and Handley, C. C. 1990. Four Important Issues in Cognitive Mapping. *From Animals to Animats.* 176-183.

Yeap, W. K.; Naylor, P. S. and Jefferies, M. 1990. Computing a Representation of the Physical Environment. Proceedings of the 1st Pacific Rim International Conference on Artificial Intelligence., 847-852. Nagoya, Japan.

Yeap, W. K. and Robertson, C. J. 1990. Planning a Route from a Cognitive Map. *Cybernetics & Systems Research ed by R. Trappl* 895-902.

Zimmer, U. R. and Puttkamer, E. V. 1994. Comparing World-Modelling Strategies for Autonomous Mobile Robots. IWK'94, 1-6. Ilmenau, Germany.