Large-scale mapping and navigation in virtual worlds.

Katrina Samperi
k.samperi@cs.bham.ac.uk

September 1, 2010

1 The Problem

The goal of this project is to enable an intelligent mobile virtual agent to autonomously generate maps in large scale, dynamic, unstructured worlds, and subsequently to use these maps to plan and navigate routes from one locale to another. While this problem has been tackled before for small-scale, mostly static environments, there are no solutions for the more complex problem posed by large-scale dynamic environments. Kuipers and Levitt (1988) define large scale space as follows:

> "A space whose structure is at a significantly larger scale than the observations available at an instant"

This indicates that "large scale" is defined in relation to the agent perceiving the environment. Rather than using this definition, we want to look at vast-scale spaces. In the real world this would, to a human sized observer, be closer to the size and complexity of a city, rather than a single building. A good comparison is used by Sibley et al. (2010a) who show routes between London and Oxford, 121 km long, for their work. This scale of environment can pose problems for mapping as the space requirements are equally vast. Static environments are used in the vast majority of mapping research, as problems with sensor noise and pose uncertainty are hard enough without the addition of moving objects. Likewise, structured environments are often used, as these are simpler to segment and generate maps with. Most static environments used consist of office like environments with rooms leading from a central corridor (Fox et al., 2005). We want agents to be able to move away from this structure and be capable of mapping any style of environment, indoors and out. We can investigate these issues by looking at mapping large scale, dynamic *virtual* worlds. In a virtual world the agent is always able to request its current location, thus removing the pose uncertainty (also known as the localisation problem), and the world is presented as a set of geometric primitives, without the need for using unreliable sensors. The range an agent is able to perceive the environment is still limited, but there are less issues from having a sparsely populated environment to map and navigate within.

The problem of generating maps in large-scale, dynamic worlds can be tackled in two stages. The first stage is to determine how to represent the map, the second stage is then how to generate this. We can test how successful the map representation and generation has been by using it to plan and navigate routes. Various questions exist at each stage about what makes a good or bad representation and how we can evaluate this. Evaluating the map representation requires looking at how the map will be used. The map is being created so that the agent can navigate from one place to another. In order to ensure the agent is able to plan routes and navigate we will need to look at methods for route planning and how these can be applied in the new map representation.

2 Literature Review

The process of addressing how to enable an intelligent mobile agent to generate maps of large scale, dynamic and unstructured environments can be split into sub-problems. The central issue is how to represent a map of this type of environment. Current literature focuses on small scale, static, office like environments (Thrun, 1998, 2002; Shi et al., 2010; Thrun and Bücken, 1996; Simmons et al., 2000; Birk and Carpin, 2006; Dijk et al., 2003; Doyle et al., 1997; Jan et al., 2009; Filliat, 2003; Stoffel et al., 2008; Zender et al., 2008; Choset and Nagatani, 2001; Simmons and Koenig, 1995; Tapus and Vasudevan, 2005). While some attempts have been made to change one of these aspects, nothing has been done to solve the full problem (Wolf and Sukhatme, 2003; Biswas et al., 2002; Pronobis et al., 2009; Kuipers et al., 2004; Maurette, 2003; Schmidt and Azarm, 1992; Zilberstein and Russell, 1993; Yuan et al., To Appear; Bi et al., 2009; Elfes, 1989; Kuipers and Levitt, 1988; Hhnel et al., 2003; Sibley et al., 2010a; Huang et al., 2008; Konolige et al., 2003; Sibley et al., 2010b). Once a map representation has been selected then the agent needs a method of generating this map by exploring the environment. The final problem is how this can be used for route planning and navigation. We will look at route planning algorithms designed to work in dynamic, large scale or any type of environment.

The virtual world that we will use is Second Life (Linden Research Inc.). Second Life is a continuous, persistent, virtual world which thousands of people use every day. The world is split into $255m^3$ areas of space called regions. Each region has a unique name and can either be owned privately or by Linden Lab. A region is surrounded by other regions or empty space. Movement between and within regions can be restricted with only certain people given access to private regions, or modes of transport being disallowed.

Using virtual worlds as a test bed for AI is not a new idea and there are many advantages associated with it (Kaminka et al., 2002). In a virtual world we are able to record everything the agent does and how decisions are made much more easily than in the real world. In experimental terms, we can set up the exact same environment to reliably compare test results. Since most of the current techniques for mapping and route planning were designed for agents working in the real world, the differences between the virtual world and the real are important. Some of these differences work to the agents advantage. Localisation in the real world can be a hard problem; this is made trivial in Second Life as global coordinates and pose information is always available from the server. Sensor noise is also not a problem in virtual worlds as the environment is presented as a set of geometric primitive objects with their location and dimensions described exactly. Mechanical issues that may arise with the sensors and motors are not a problem in virtual worlds. These are replaced with issues concerning network latency and disconnection. Virtual worlds are very dynamic environments. The real world can change while an agent is mapping the environment, but it is less likely to have major structural changes during that period. Second Life is a constantly changing environment and this will need to be taken into account.

2.1 Map Representation

Traditionally map representations have been split into three categories: metric, topological and hybrid maps (Thrun, 1998, 2002; Wolf and Sukhatme, 2003; Thrun and Bücken, 1996; Huang et al., 2008). We will describe how each of these maps are generated and what information they contain, as well as generic advantages and disadvantages. We will also look at map representations designed to function in unstructured, large scale or dynamic environments.

2.1.1 Metric Maps

Metric maps are a simple form of map directly generated from sensor information. They represent the geometric information about the world and where free space exists (Kuipers and Levitt, 1988; Filliat, 2003; Thrun, 1998, 2002; Zender et al., 2008). A general advantage of this type of map is that they translate well to the real world and between systems, allowing easier validation of a representations (Filliat, 2003). They are also considered easier to build, robust and highly accurate (Thrun, 2002; Wolf and Sukhatme, 2003; Thrun and Bücken, 1996). They are, however, less efficient to plan routes in than higher level abstractions due to the greater number of potential paths (Filliat, 2003).

The information stored at the metric level is invaluable for generating such abstractions and so will always be required in a mapping system. The memory space required is a disadvantage to metric maps, they tend towards large sparse data representations with large time and space complexities(Zender et al., 2008; Elfes, 1989; Thrun and Bücken, 1996). The problem with space requirements can be reduced using different data storage methods such as the octree (Lefebvre et al., 2009).

A common type of metric map is the occupancy grid (Elfes, 1989). Occupancy grids are capable of representing any type of environment, they are very simple to create, and robust. Most topological maps and hybrid systems use an occupancy grid as the initial or low level representation (Thrun, 1998; Wolf and Sukhatme, 2003; Biswas et al., 2002). An occupancy grid is built up by splitting the world into a grid typically in two or three dimensions. The probability of occupancy for each cell in the grid is computed over time. Most implementations of occupancy grids are in two dimensions and are small enough for the map to be held in memory. For the large scale virtual world we will need information stored in three dimensions, leading to the space required to store an occupancy grid being very large. Due to the occupancy of each cell being calculated over time, this type of map is also unsuitable for fast changing dynamic environments. Any changes will either be regarded as noise and so cancelled out by the other readings, or they will need to be in place for a significant amount of time to effect a change in the grid (Wolf and Sukhatme, 2003; Biswas et al., 2002).

Object maps (Thrun, 2002) work in a different way. They reduce the world to geometric primitives and then hold information about the relative dimensions of shapes and where they are positioned. This is an unpopular approach in robotics as the real world can be hard to reduce to primitive shapes. Virtual worlds, however, are already presented in this form meaning that an object map is an ideal representation. The object map will also be capable of representing unstructured space as it is concerned with where the objects lie and the free space between them. Thrun (Thrun, 2002) also believes that it would not be possible to look at dynamic objects without an object map representation.

2.1.2 Topological Maps

Topological maps are an abstract representation of the environment, consisting of nodes and arcs. The node is a distinct place or region in the world and the arc holds information about the connectivity of the node and how to travel between them (Huang et al., 2008). A topological map is a more compact representation than the metric map as it only holds key information about the world (Filliat, 2003; Kuipers and Levitt, 1988; Kuipers et al., 2004; Thrun and Bücken, 1996). A topological map is designed to hold structural information about large scale spaces (Kuipers et al., 2004). They are usually, but not always, generated from a metric representation of the world (Barbosa and Rodrigues, 2006; Thrun, 1998; Shi et al., 2010). A general advantage to topological maps is that the more compact representation allows for faster route planning (Sibley et al., 2010b). Accurate localisation is not needed in a topological map, just knowledge of which node or arc the agent is currently travelling through (Thrun, 1998; Sibley et al., 2010b). A disadvantage is that topological maps are considered more difficult to generate and keep up to date in large scale worlds (Thrun, 1998; Filliat, 2003). Generating a topological map will need more computational power than simply using a metric map, but the advantages in route planning time should compensate for this. The method used to generate a topological map is an important factor in whether the map will need regenerating as the environment changes. Good decompositions will cause only one node to need changing at a time, so the overall structure of a topological map is not affected.

There are two approaches to generating a topological map. The first generates it from a metric representation by decomposing the space into individual nodes. The other method is to generate a topological map directly from sensor readings. We will look at several specific methods for generating a topological map. Two of these require an occupancy grid: voronoi decomposition (Thrun, 1998) and image skeletonisation (de Oliveira and Romero, 2009). The third is a method of direct generation called Topological Map Navigation (TMN) (Shi et al., 2010).

Both voronoi decomposition and image skeletonisation use an occupancy grid as the metric input to a topological generation process and are very similar throughout. Voronoi decomposition is a method of segmenting space based on a set of seed points and the distance of a location from these. Thrun (1998) uses the boundaries of the world as seed points and generates voronoi lines equidistant from these. Critical points are then assigned along the line where the boundaries narrow (usually a door). The points are extended to critical lines from one boundary to another, splitting the map into regions. At this point only the boundaries and critical lines are important as each region forms a single node in the topological map. Image skeletonisation was proposed as an alternative to voronoi decomposition. Originally it was an image processing technique used to understand image composition. An image is reduced to single lines by finding the points equidistant between boundaries. Nodes are then defined by selecting the intersections and points where boundaries narrow. This is very similar to how voronoi decomposition is performed. The difference between the two techniques, according to Oliveira, is that image skeletonisation does not create unnecessary arcs and nodes in the environment, but little evidence is given to back up this claim.

An entirely different approach is proposed by Shi et al. (2010) referred to as Topological Map Navigation (TMN). No metric map is ever created, instead a topological map is formed directly from sensor information provided by a laser scanner. Nodes then fall into three groups: dead ends, corners and intersections. Whereas in the previous techniques the arcs held information about the connectivity of nodes, now they simply inform the agent the direction and distance between nodes.

As can be seen there are many similarities between the voronoi and image skeletonisation method. Both reduce the world to lines, equidistant from boundaries, then select points on these to form nodes. A disadvantage to voronoi is that superfluous nodes can be created, which then require pruning to give a more compact representation (Thrun and Bücken, 1996). Image skeletonisation claims to never generate these unnecessary lines and points and so requires less computation. Oliveira and Romero state that both the voronoi decomposition and the image skeletonisation method are robust in different environments.

Since the environment that we are interested in generating maps of is large scale, dynamic and unstructured it is also useful to look at how the different methods will perform in these conditions. Voronoi decomposition can easily cater to large scale worlds as it only requires adding more nodes. TMN however may not scale well as it is difficult to generate topological maps directly and even more so in large scale spaces (Thrun, 1998). The larger world makes it harder for the agent to know which node it is in, and which set of features it is currently observing. Dynamism in the world is seen through the underlying metric representation. As the occupancy grid changes a topological map generated by image skeletonisation and voronoi decomposition is updated. TMN has a different approach for dealing with changing environments. If the type of node does not change then there is no need to update the map. None of the methods described deal well with unstructured worlds. TMN is based around the sensors receiving information from close objects. While it works well in an office or indoor environment, unstructured or empty spaces will cause issues for classifying nodes and travelling to locations away from physical barriers. Image skeletonisation will reduce an unstructured empty space to a single line down the middle of the area. Voronoi as described in (Thrun, 1998) will have the same problem as image skeletonisation, but this is due to the way voronoi decomposition is used by Thrun. If a different set of seed points were chosen, based around the destination points rather than the boundaries, then voronoi decomposition would function well in unstructured environments. Since the environment that we are interested in generating maps of is large scale, dynamic and unstructured it is also useful to look at how the different methods will perform in these conditions. Voronoi decomposition can easily cater to large scale worlds as it only requires adding more nodes. TMN however may not scale well as it is difficult to generate topological maps directly and even more so in large scale spaces (Thrun, 1998). The larger world makes it harder for the agent to know which node it is in, and which set of features it is currently observing. Dynamism in the world is seen through the underlying metric representation. As the occupancy grid changes a topological map generated by image skeletonisation and voronoi decomposition is updated. TMN has a different approach for dealing with changing environments. If the type of node does not change then there is no need to update the map. None of the methods described deal well with unstructured worlds. TMN is based around the sensors receiving information from close objects. While it works well in an office or indoor environment, unstructured or empty spaces will cause issues for classifying nodes and travelling to locations away from physical barriers. Image skeletonisation will reduce an unstructured empty space to a single line down the middle of the area. Voronoi as described in (Thrun, 1998) will have the same problem as image skeletonisation, but this is due to the way voronoi decomposition is used by Thrun. If a different set of seed points were chosen, based around the destination points rather than the boundaries, then voronoi decomposition would function well in unstructured environments.

2.1.3 Hybrid Maps

Hybrid maps, also known as multi-level, are a relatively new concept being used in an attempt to solve problems that arise with dynamic and large scale worlds (Thrun, 2002; Wolf and Sukhatme, 2003; Thrun and Bücken, 1996; Tapus and Vasudevan, 2005). A hybrid map contains several representations of the world in different formats. Each representation can then be used for a different task in the environment. Changes made to one level should propagate through the rest of the representation. The advantages to this type of map is the ability to pick and combine advantages that other types of maps contain and when there is a disadvantage in one level, move to different representation to solve this.

Hierarchical maps are a type of multi-level representation that take advantage of how humans intuitively partition space (Tapus and Vasudevan, 2005; Zender et al., 2008). Kuipers proposed the Spatial Semantic Hierarchy (SSH) to combine metric and topological representations of space to give an overall more useful representation to resolve ambiguity in the environment (Kuipers et al., 2004). Space is split up into small metric representations of the world (usually occupancy grid based) linked by a topological graph over the top. This allows route planning on two levels, the higher level which states which nodes to visit in order and the low level where more detailed route planning can occur. Hierarchical maps have also been used to explore pedestrian indoor navigation (Stoffel et al., 2008). The map described by Stoffel et al. (2008) holds different topological maps containing different levels of detail. When planning a route it is possible to move from one level to a more appropriate one. The structure is held as a tree with each leaf being a small scale map of the environment with links to other small parts.

2.1.4 Mapping Unstructured worlds

The vast majority of robot mapping is performed in indoor, office environments made up of long corridors and several rooms. By contrast, an unstructured environment is an open space with few well defined boundaries and limits on movement. A structured environment can help navigation as there are very limited configurations of the space and semantic information can be inferred. The range of sensing equipment can also be a limiting factor in robotics as large, open spaces will not allow laser or sonar scanners to receive any information (Shi et al., 2010).

Of the two metric representations mentioned earlier, the occupancy grid (Elfes, 1989) and the object map (Thrun, 2002), only the first has been used to generate maps of unstructured worlds. The advantage the occupancy grid has when describing this type of environment is that structure makes very little difference in how the map is generated. The world can still be split into a grid system and the occupancy of each square still calculated independently of one another. Object maps will also be capable of representing this type of environment, assuming that objects can still be split into geometric primitives. Neither representation needs to use any underlying structure to represent the world, unlike the topological map. Node definition in topological maps requires splitting the metric map into sections, this is usually done by using the structure of the world and declaring boundaries at doors or other landmarks (Thrun, 1998). Without the structure of the world nodes become harder to define and so the map harder to create.

2.1.5 Mapping Large Scale Worlds

We previously described large scale worlds as an environment the size of a city, rather than the standard description used. Metric maps, such as the occupancy grid can scale to any size with any number of dimensions. The problem lies with the space and time required to process a map of this size without giving the agent unlimited resources. The level of detail required for an occupancy grid to give a truly accurate representation of large scale worlds comes at the cost of very large memory and storage requirements. In limiting the space available, we also limit the accuracy. Topological maps are a more compact representation that only hold what is considered relevant, everything else is removed (Pronobis et al., 2009; Thrun and Bücken, 1996). This leads to another question to be answered, how do we know what data is relevant and should be maintained?

Topological maps have been used by various people to try and solve the problem of mapping large scale spaces (Thrun, 1998; Ramloll and Mowat, 2001; Choset and Nagatani, 2001; Kuipers and Levitt, 1988; Stoffel et al., 2008; Kuipers et al., 2004). For large scale worlds topological maps have considerably advantages over metric representations. This level of abstraction reduces the space required to store the map and allows route planning to take less time. The main advantage of a topological representation is that it allows faster route planning than the metric map (Thrun, 1998). This is due to the smaller number of potential places the agent can move to and the compactness of the representation

2.1.6 Mapping Dynamic Worlds

Second Life is a dynamic environment with user created content being added every second of the day. This rate of change must be taken into account when generating maps of the environment. Existing maps must have the capability to adapt and change and the agent must be able to recognise when the structure of the space is different. There are two types of change that occur: structural changes and the movement of objects and people. Structural changes are rarer in established regions but can happen rapidly with no warning.

We will look at two groups that have attempted to solve this problem. Both these groups use multiple occupancy grid maps and compare the differences between them in an attempt to track changes which are not observed. The Robot Object Mapping Algorithm (ROMA) (Biswas et al., 2002) is based on the Dynamic Occupancy Grid Mapping Algorithm (DOGMA) (Thrun, 2002) and designed to track changes in dynamic environments. Several occupancy grids are kept in memory and these are compared, giving a time line to the objects in the world and their locations. Another approach very similar to ROMA was proposed by Wolf and Sukhatme (Wolf and Sukhatme, 2003). This was designed to keep track of mobile objects in a static environment again, by combining information held on two occupancy grid representations. On one map the static parts of the environment were displayed and on the other, the moving objects. The grids are once again compared and objects can be tracked from one location to another. The disadvantage to these methods is that it relies on objects moving so slowly that they appeared to be static in separate snapshots of the world. Faster moving objects, such as people, are reduced to noise in the environment.

There has been little focus on using topological maps in dynamic worlds. As a topological map is an abstraction then it is less affected by change, except in the case of major structural differences (Shi et al., 2010). In this case a topological map would have to be at least partially regenerated from the metric map or sensor readings.

2.2 Map Generation

The agent needs be able to generate maps of previously unknown environments. To do this it will need some type of exploration strategy that will allow the agent to know when and how it will explore an entire region (Simmons et al., 2000). The simplest form of exploration is to exhaustively search an environment in a static pattern. A lawn mower approach covering the entire environment, or a spiral starting from the outside edge are two examples of exhaustive search. Using these methods will lead to a full map, but will take a very long time to complete. A better alternative is to use greedy exploration (Thrun, 1998). In greedy exploration the agent always moves along the lowest cost path towards an unexplored area. This has a disadvantage in large scale and continuous worlds as the agent has the opportunity to walk in one direction for a very long time and never return to explore the rest of a region. Knowing when the entire environment has been covered is a different problem. For environments with boundaries the exploration process can calculate how much new information there is to learn (Simmons et al., 2000) and declare the end point when the information gain does not exceed a given threshold. Other approaches require the world to be annotated (Doyle et al., 1997) and while this will make exploration faster, it will also have a detrimental effect as we want our agent to be able to function in any environment.

Exploration done by single agents is still a slow process, even with techniques like greedy exploration. Using multiple agents to explore a region of space and combining the maps produced is one method that has been used to try and speed up this process (Nair and Tambe; Ramloll, 1997; Birk and Carpin, 2006; Konolige et al., 2003; Fox et al., 2005). These exploration agents can be simplified versions of the main agent, only capable of exploring and generating the small maps which are then passed onto a centralised controller to combine and process though this is not always the case.

While this may speed up generating the map for the environment, the overall cost of creating these agents, combining and validating the map must not exceed the time it would take a single agent to do the same job. Overheads for the agents are a valid concern, as are methods enable the agents to work together in exploring the entire environment, rather than just one corner of it several times (Nair and Tambe; Dixon and Henlich, 1997; Simmons et al., 2000). Other concerns with using the sub-map approach are mentioned by Sibley et al. (2010b). Map overlap, data duplication, map merging alignment and the optimal size for sub-maps all have to be considered when deciding if a multi-agent approach is viable.

Merging the individual maps becomes a new challenge for the agents. Previous work in this field has focused on merging occupancy grid maps of static environments by translating and rotating images until they match (Birk and Carpin, 2006; Simmons et al., 2000; Konolige et al., 2003). The main problem in merging metric representations is that the exact pose information is not always available for other agents (Simmons et al., 2000; Fox et al., 2005). This leads to uncertainty as to whether one region corresponds to another. Individual maps are able to be validated if more than one agent is responsible for mapping an area, reducing the problem of sensor noise (Birk and Carpin, 2006; Simmons et al., 2000). This can also be used to ensure that an agent is not reporting malicious information of any type. In both (Birk and Carpin, 2006) and (Simmons et al., 2000) the maps are generated by individual agents and then merged at a central point. New instructions are given to the agent by the central controller. The agents then attempt to find as much new information as possible.

Topological maps can also be merged by adding arcs connecting between the two. While being simple to achieve, unless a topological maps are in the same scale and built in the same way, it is not as useful.

2.3 Route planning

To evaluate how successfully the map has been generated we must look at how it will be used. A map is designed to allow the virtual agent to identify key destination points in the environment and navigate from one place to another. As the focus of this project is on the map representation and generation process, rather than new methods of route planning, we will only look briefly at methods of route planning and navigation. Route planning in large scale, dynamic, unstructured worlds can have problems. As the size of the world gets larger the number of potential routes grows exponentially. As the world is dynamic there is no guarantee that a route will be navigated successfully so the agent must be capable of re-planning if the environment changes.

A basic approach to route planning is to use A* search. This is a best first search which finds the least-cost path from the start point to the end by combining both the cost of movement and an estimate of how close to the end that step has taken the agent (Russell and Norvig, 2003).

Approaches have been suggested for dynamic environments such as using Unsteady Diffusion Equations (Schmidt and Azarm, 1992), Ant colony algorithms (Bi et al., 2009), trail based methods (Yuan et al., To Appear) and anytime route planning (Zilberstein and Russell, 1993). The unsteady diffusion method uses the idea that a goal point emits a gaseous substance through the environment. The agent then chooses a path based on the concentration of gas, with a higher concentration indicating that it is moving close to the goal (Schmidt and Azarm, 1992). This approach requires full knowledge of the environment to generate, and in large scale spaces there will be a significant cost in pre-processing the map to enable the agent to plan a route. The ant colony approach simulates the foraging process used by ants (Bi et al., 2009). Numerous trails are generated in the environment, each one taking into account where the previous 'ants' have travelled. The greater number of times a path has been chosen the more likely it is to be used again, strengthening that trail as an optimal one. A final route is then generated from this network of ant trails. This method is similar to generating the trail maps from user trails, but the agents will generate several thousand routes, rather than observing the movements of other users. This method has also been explored by Yuan et al. (To Appear). The robot here observes a human moving through an environment and uses this to generate a road map of the 'safe' spaces.

Anytime route planning (Zilberstein and Russell, 1993) proposes the idea that planning can be done at different levels of detail. There is a trade off between the quality of the plan made and the time spent sensing the environment. The agent can then act on a partial plan, rather than waiting for every step to have been completed. The anytime planner creates a series of plans over time, each one improving on the last and being able to take into account the world changing as the agent moves. To follow incomplete plans then the agent will need obstacle avoidance procedures and the planning process must be robust enough cope with these changes.

2.4 Summary

We have looked at current methods for map representation, map generation and route planning. We found that there is no single map representation that takes into account all the aspects of a large scale, dynamic, unstructured world. Thrun (2002) also states in his review of mapping that there is no current solution for this problem.

We looked at the differences between metric, topological and hybrid maps, summarised in the following table.

Table 1: Differences between map types

	Metric	Topological	Hybrid
Built from	Sensor Information	Sensor Information or Metric Map	Sensor Information and other maps
Stores	Geometric Information	Abstraction of the metric map	All information at different levels
Advantages	Simple Easy to construct Robust Translates well to real world Easy validation Used to generate other maps	Better representation for planning than the metric Compact representation Only holds key information	All the advantages of other maps
Disadvantages	Very large size maps Can be difficult to process Less efficient for route planning than topological maps	Loss of information Harder to validate than metric representations Harder to generate and keep up to date	More space required Need to link the maps together More overheads

We then looked at how to generate maps of unstructured, large scale, and dynamic worlds.

Since metric maps are generated directly from the sensor information with no additional requirements, metric maps are the better option for open unstructured worlds. We compared two: the occupancy grid, a very common metric map, and object maps, a less popular one. The occupancy grid is a simple and robust map with the ability to scale to any size and dimension. The problem with this representation is the space and computational requirements to generate and update the map. In contrast the object map still represents the free space and objects in the world; it is a more compact representation and reflects the nature of the virtual world ideally. The main drawback of using object maps in robotics is that it requires the real world to be reduced to geometric primitives, a trivial task in virtual worlds. Topological maps were not considered useful for this type of environment due to the difficulty in generating the nodes and arcs without any inherent structure to go from.

When mapping large scale worlds, the preferred style to use is a topological map. The main issue then becomes how to generate topological maps. We looked at three methods for this. The first two, voronoi decomposition and image skeletonisation required a metric map as input and they split the world in very similar ways. The third method, TMN, took the sensor readings and generated topological nodes directly from this. While this third method required less computational time and space, it does not fully describe the environment. It concerns itself only with certain node types (dead ends, corners and intersections of paths). TMN was also stated to be unsuitable for unstructured environments due to the sensor range and inherent structural requirements. Image skeletonisation also requires that the world has set boundaries and a structured approach to generate the skeleton used as the arcs. In an unstructured environment this would fail as there are no boundaries to generate the lines from. Voronoi, as used by Thrun (1998), also has this problem. Voronoi decomposition can also be performed using individual seed points rather than the boundaries of free space. These seed points can be the key destination points recognised in the map. The advantages and disadvantages of each method can be seen below.

Table 2: Comparison of decomposition methods

	Voronoi Decomposition	Image Skeletonisation	TMN
Advantages	Well known Simple Options for different seed points	Does not generate superfluous lines and nodes Better computational efficiency then Voronoi	No metric map No exact localisation needed Better computational efficiency without the metric map
Disadvantages	If there is no structure in the environment then a new method for generating seed points must be found Superfluous nodes must be pruned	Very little evidence given in the paper to back up claims Method appears to be voronoi decomposition with a different name	Metric map can still be useful for localisation and finding objects in the world No ability to navigate from one corner to another without the exact localisation

The dynamic nature of the environment comes in two parts. Dynamic objects and structural changes will need to be taken into account when generating and updating the map. As objects move they can be identified using a global unique identifier with current and previous locations being linked to it. Structural changes will require changes in the map at a metric level and occasionally the topological. If the method of decomposition chosen was an appropriate one the changes should be within a single node, ensuring that only a small part of either map ever needs changing. This plasticity of representation is important if the map is to be used, rather than eternally updated. The movement of users around the world is also available. This can be used to generate cluster points and a trail map. The cluster points can then be used as key destinations in the environment. Trails can be merged together to give a road map of the environment, which can be used for navigation. When combined with probabilistic road maps generated from the object map we will have a full set of potential routes throughout an environment which can be used for navigation.

Hybrid maps have been investigated, but not extensively outside small, structured, static environments. The table below gives a comparison of the different types of maps described and which environments they are suitable for.

Table 3: Comparison of map representations

	Occ Grid	Object Map	Topological Map	TMN	ROMA	Prob. Road maps
Type	Metric	Metric	Topological	Topological	Metric	Topological
Large Scale	No	Yes	Yes	Unknown	No	Yes
Dynamic	No	No	No	Some	Yes	Some
Unstructured	Yes	Yes	No	No	Yes	No
3D	Yes	Yes	Yes	Unknown	Yes	Unknown
Incremental Update	Yes	Yes	Yes	Yes	Yes	No
Route Planning	Slow	Slow	Yes	Yes	Yes	Yes

As can be seen, there is no single map representation which will allow us to generate maps of large scale, dynamic and unstructured environments. All of them are able to map three dimensional environments and most update incrementally, but route planning is a problem in wholly metric representations.

We also took a brief look at how to generate a map of this type of environment. Four methods of exploration were described. Naive exploration picks a random point and navigates to there, generating the map as it moves. This method is not very useful for ensuring the entire environment is explored in a timely manner. Exhaustive search will ensure that the entire environment is explored, but can waste a lot of time in doing so. Greedy exploration plans routes to the nearest unmapped point, on the least cost path. While a good solution, greedy exploration will still take a long time to generate a full map of the environment. We looked at the potential for multiple agents to generate maps of different parts of the world, which are merged at a central point. Each agent using greedy search based around a different point in the world has the potential to generate maps much faster and any overlap will allow for validation of each individual section.

Route planning was looked at in the context of evaluation. The simplest approach is to use A* to plan routes, evaluating and updating that plan as the agent moves. This method will work, in an environment of any size, but it does not take into account dynamic aspects. Of the dynamic methods the simplest appears to be anytime route planning. This can be used to generate routes at different levels of detail. An overarching route can be planned before movement, and it is gradually made more detailed as each part of the route is visited. Testing can then be done on routes planned and undertaken to measure if the new map representation is an improvement over existing methods.

3 Proposed Solution

To reiterate, the environment that the agent is attempting to generate maps of are large scale, dynamic and unstructured. This poses problems to current mapping techniques described in the literature review, and while some attempts have been made to solve individual parts, there is no solution to the entire problem. We propose to investigate new methods of representing maps along with methods to allow a mobile intelligent agent to generate and use these. The representation and generation methods will be tested through a series of experiments in which the agent is given a set of tasks to achieve. These tasks will involve the agent being placed in a new environment and asked first to generate a map of this space before navigating between two given points. We can measure the time taken and processing required, as well as the success of the agent when completing the task and then compare this to current methods used to represent and generate the resulting map. If the agent is able to perform faster than standard methods and able to cope well with the type of environment we are interested in this test will be successful.

To represent and generate the map we must first look at the different information that the Second Life servers provide. Sensor information is provided as a list of geometric primitives, including the location and exact dimensions of each object. Information is also always available about the agents current pose and the movement of all other users in this region. From this other pieces of information can be derived, such as where free space exists and the structure of the world. The paths that users take through the world can be observed using the coordinate information given for each person. Both the observations and free space can be used to infer where potential routes and destinations may be found. As the environment grows in size, more information is available and the agent must be capable of storing only what is required.

3.1 Map Representation

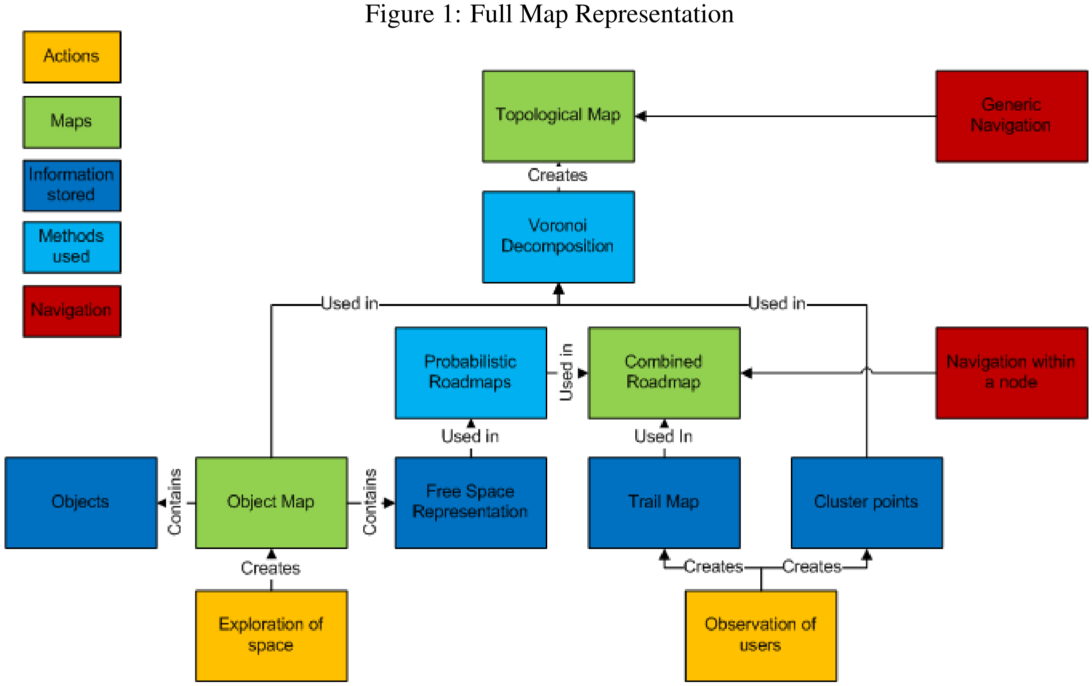
The issue with current map representations is that no single style is able to represent every aspect of a large scale, dynamic or unstructured environments. To map large scale places a topological map is often used. Unstructured environments require the use of a metric representation. Dynamism in objects and people is often shown by comparing the differences between pre-existing maps.

We propose to use a hybrid multi-level map to represent the world, combining the advantages of each individual map and negate some of the disadvantages. A hybrid multi-level map can hold both a metric and topological map of the environment, with links between the two allowing the agent to use whichever representation is best for the task at hand. The metric level map is best represented using an object map, rather than an occupancy grid, due to it being compact while still retaining enough information about free space and objects present. In parallel to this is the road map, the combination of observing user movements in the environment and Probabilistic Roadmaps (Kavraki et al., 1998). This combination will contain the full list of routes the available to the agent. Cluster points are generated by observing users and where groups of people gather together. These are our key destination points, which can also be used as seed points for a voronoi graph, later used to generate a topological map.

The purpose of each type of map must also be made clear. The metric map defines where free space is in the environment. It also holds exact information about the location and dimensions of objects of interest. These objects are used by the agent to complete set tasks. The combination road map is used by the agent to plan routes at a higher level of detail, whereas the topological map is used to link places and regions together and plan more general routes. The agent plans routes at three different levels of detail. First the agent plans a route from one place to another using the topological map, giving a list of nodes to visit in order. On reaching a node the agent will need to plan how to traverse the space, this is done using the road maps generated. The final level of route planning is reactive, as the agent moves from one point to another, avoiding objects and recording if any changes to the environment that may have occurred.

Figure 1 below shows how the map will be represented, what information is held in each type of map and how this information is used.

Figure 1: Full Map Representation



3.2 Generating the map

Once the agent has a method of representing the space we must focus on methods of generating that map. The simplest method is to use an exhaustive search, but in a large scale space this method will take up more time than is considered reasonable. One method for speeding up this process is to use multiple agents exploring in parallel. Another method is to use a different exploration strategy. We will use greedy exploration as it is still simple, therefore errors are less likely. The problem then moves away from exploration to combining maps, provided by independent agents, into a full representation of the world.

4 Research Questions and Methods of Evaluation

The problem, as stated previously, is: “How can we enable an intelligent mobile virtual agent to represent and generate maps of large scale, dynamic and unstructured environments?” This question can be broken down into two parts, “How do we represent the map?” and once we have that, “How should the agent generate it?” Answering these questions will provide the main contribution of this work, a new map representation for this type of environment and a method for generating it in a timely fashion.

A good solution to the problem will allow an agent to use the map generated to navigate from one point to another in increasingly large environments, without allowing the storage and processing requirements to grow too large. The map should be generated autonomously and online, rather than requiring constant feedback from a supervisor, or lengthy pre-processing of the environment. While achieving this, we aim to generate the map in the smallest amount of time possible and to keep the storage compact. To take into account the dynamic nature of the environment, the map must have a certain level of plasticity, and so be able to change small areas without requiring the entire map to be regenerated.

We also need a method of evaluating whether or not the proposed solution is a good one. Methods for comparing and evaluating map representations have been discussed and a set of standards have yet to be formalised (Amigoni et al., 2007, 2009). A brief survey of evaluation methods indicate, unsurprisingly, that accuracy is important for any map and the best way to evaluate one is by looking at how the map will be used. Accuracy is usually tested via visual inspection, or compared to a ground truth map of the environment being looked at (Fox et al., 2005). Other methods include the time taken to generate the map and percentage of the environment explored (Konolige et al., 2003).

As it will be difficult to achieve comparable results with current published papers, we will compare the proposed system with a basic agent. The basic agent will generate, through exhaustive search, an occupancy grid map of the environment. The size and time taken to generate this, as well as how successfully the agent can plan routes will be compared. The proposed representation will be an improvement if the map produced allows the agent to navigate the environment faster and more accurately than using the basic system, while still ensuring the size of the representation and time taken for generation are as small as possible. A second basic system can also be tested against, this one generating a topological map from the occupancy grid representation and using this to aid with navigation. The basic agent and the proposed solution will be tested first in small, static, two dimensional environments before introducing the added complexities of large scale, dynamic and unstructured worlds.

As well as the main question and contribution there are many smaller questions which must be answered along the path to a final solution. The first of these is, “How should the agent group objects together?” There are three main types of objects the agent will be interested in: landmarks, interactive and static objects. Landmarks should be recorded in the map and used when navigating. Interactive and dynamic objects must be noted, including any interesting properties that may be used (doors that are able to be opened, for example). Static objects make up the bulk of the world and these are the type that can be grouped together for ease of storage. These objects make up the walls of buildings, the paths which can be walked on, and platforms which exist as a single entity to a human observer. Grouping objects together successfully will allow the agent to reduce the amount of space required in the object map, as well as being able to reduce the free space available and cut route planning times. With the environment becoming increasingly large, the amount of information possessed by the agent will also increase. We will need to investigate what is considered ‘useful’ information and separate this from trivial information that can be safely removed, without compromising the accuracy of the map. Knowing how to merge objects together will go some way towards helping this problem, but other solutions may be required.

We elected to use an object map for the metric level representation in the proposed solution due to its more compact representation, and ease of application in virtual worlds. This, however, throws up many new challenges for us to solve.

The first challenge is, "How do we generate probabilistic roadmaps from an object map?" The agent is relying on using probabilistic roadmaps to discover unknown routes through the environment. The roadmaps have only been generated using occupancy grids, and so we must find a method of applying the same technique to the object map. Another challenge is, "How do we decompose the space into discrete segments for a topological map to be built?" Once again decomposition of space has traditionally used occupancy grids to achieve this, or even direct sensor readings, rather than an object map. Comparisons between occupancy grid maps have been used to identify dynamic objects. As everything in the object map will have a unique identifier, supplied by the virtual world, discovering when objects have moved should be far easier. The final challenge is, "How do we merge object maps together?" Much work has been done on merging occupancy grids, but again, none on the object map. All of these challenges question whether or not an object map really is an appropriate representation. If it turns out that the overheads created by using an object map outstrip the benefits, then the proposed solution will need revising and a new metric representation selected.

A question which arose when looking at object maps is, "What is the best method for decomposing space?" We looked at three methods for doing this and, without alteration, none of them were found to be suitable for our needs. A good decomposition will segment the environment into just enough distinct parts that when a change happens in the world, only a very small number of segments will need updating. We can test the various methods of decomposition with the object map and compare the resulting topological map generated. Other factors to consider are whether the key destination, or cluster points, should be in the centre of a node or on the edge. The topological map created will change based on which of these is considered better as will the type of routes generated. Konolige et al. (2003) evaluate topological maps based on its accuracy, size, whether all the points are reachable by the robots and number of points generated. There needs to be sufficient points for a good solution, but not so many that searching becomes computationally infeasible.

Another important area to investigate is "How can user movements be used to generate road maps?" Generating cluster information and merging trails to give a coherent picture of where users travel will be important for our agents ability to navigate the world. While there are some methods in intelligent data analysis which may be useful here, it is likely that using a simpler method, such as a line of best fit over a small set of points, will be feasible for merging routes together. We also want to merge this information with probabilistic roadmaps generated by the agent. As different modes of transport are available for the agent and users of the virtual world it will become important to identify the type used for each route. The road map will need to label each path with the mode of transport used to enable the agent to successfully use these.

Finally, there is a question about whether multiple agents will be useful for generating maps of large scale environments. We proposed to limit the time required to generate the map by running several agents in parallel. This creates extra overheads which need to be taken into account by the system. Each additional agent adds to the space and computational power needed to generate the map, as well as additional overheads in merging the maps together. If these extra overheads exceed the time required for a single agent to explore the world then it would appear that this method is not useful for generating maps of large scale worlds. An alternative is that, by using the object map, the merging process takes too long and so it would be better to use an occupancy grid for the metric level representation.

5 Timetable

The following is a approximate timetable of work to be done and the time required for each stage. The first priority is to ensure that the framework between Second Life and Java is working, as everything relies on this framework. Before the next RSMG meeting I hope to have this completed and much of the work regarding trails completed.

Testing and analysing the success of the system will be on going after every stage, with extra time taken at the end to ensure that everything is done properly and comparable results are achieved.

Table 4: Timetable for completion

July - October 2010	Interface between Second Life and Java
August - October 2011	Potentially write a paper on using trails for autonomous systems
August - September 2011	Add the trail map and generate cluster points
September 2010	Year 3, Prep for teaching
October 2010	RSMG 4
November - February 2010	Implement baseline system and generate test data
February - March 2011	Add the object map
March 2011	Include the trail and probabilistic roadmaps
April - May 2011	Add the topological map
June 2011	RSMG 5
June - July 2011	Test Map representation against current standards
August - November 2011	Add the multi-agent systems for map generation
September 2011	Year 4, Prep for teaching
November - December 2011	Testing everything and comparing the outputs
January - February 2012	Write a paper on the entire mapping system
February 2012	RSMG 6
January - October 2012	Writing up, (potentially until January 2014)

References

F. Amigoni, S. Gasparini, and M. Gini. Good experimental methodologies for robotic mapping: A proposal. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 4176 –4181, 10-14 2007. doi: 10.1109/ROBOT.2007.364121.

Francesco Amigoni, Monica Reggiani, and Viola Schiaffonati. An insightful comparison between experiments in mobile robotics and in science. *Autonomous Robots*, 2009. doi: 10.1007/s10514-009-9137-8. URL http://dx.doi.org/10.1007/s10514-009-9137-8.

Rafael Garcia Barbosa and Maria Andréia Formico Rodrigues. Supporting guided navigation in mobile virtual environments. In *VRST '06: Proceedings of the ACM symposium on Virtual reality software and technology*, pages 220–226, New York, NY, USA, 2006. ACM. ISBN 1-59593-321-2. doi: http://doi.acm.org/10.1145/1180495.1180541.

Zeng Bi, Yang Yimin, and Xu Yisan. Mobile robot navigation in unknown dynamic environment based on ant colony algorithm. In *GCIS '09: Proceedings of the 2009 WRI Global Congress on Intelligent Systems*, pages 98–102, Washington, DC, USA, 2009. IEEE Computer Society. ISBN 978-0-7695-3571-5. doi: http://dx.doi.org/10.1109/GCIS.2009.274.

Andreas Birk and Stefano Carpin. Merging occupancy grid maps from multiple robots. *IEEE Proceedings, special issue on Multi-Robot Systems*, 94(7):1384–1397, 2006.

Rahul Biswas, Benson Limketkai, Scott Sanner, and Sebastian Thrun. Towards object mapping in non-stationary environments with mobile robots, 2002.

Howie Choset and Keiji Nagatani. Topological simultaneous localization and mapping (slam): Toward exact localization without explicit localization. *IEEE Transactions on Robotics and Automation*, 17:125–137, 2001.

J.R. de Oliveira and R.A.F. Romero. Image skeletonization method applied to generation of topological maps. In *Robotics Symposium (LARS), 2009 6th Latin American*, pages 1 –6, 29-30 2009. doi: 10.1109/LARS.2009.5418319.

Betsy Van Dijk, Rieks Op Den Akker, Anton Nijholt, and Job Zwiers. Navigation assistance in virtual worlds, 2003.

J. Dixon and O Henlich. Mobile robot navigation, final report. http://www.doc.ic.ac.uk/~nd/surprise_97/journal/vol4/jmd, 1997.

Patrick Doyle, Patrick Doyle, Barbara Hayes-roth, and Barbara Hayes-roth. Guided exploration of virtual worlds. In *Network and Netplay: Virtual Groups on the Internet*, pages 243–264. MIT Press, 1997.

A. Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57, 1989. ISSN 0018-9162. doi: http://dx.doi.org/10.1109/2.30720.

D. Filliat. Map-based navigation in mobile robots: I. a review of localization strategies. *Cognitive Systems Research*, 4 (4):243–282, December 2003. ISSN 13890417. doi: 10.1016/S1389-0417(03)00008-1. URL http://dx.doi.org/10.1016/S1389-0417(03)00008-1.

Dieter Fox, Jonathan Ko, Kurt Konolige, and Benjamin Stewart. A hierarchical bayesian approach to the revisiting problem in mobile robot map building. In Paolo Dario and Raja Chatila, editors, *Robotics Research*, volume 15 of *Springer Tracts in Advanced Robotics*, pages 60–69. Springer Berlin / Heidelberg, 2005. URL http://dx.doi.org/10.1007/11008941_7. 10.1007/11008941$_7$.

Dirk Hhnel, Dirk Schulz, and Wolfram Burgard. Mobile robot mapping in populated environments. *Advanced Robotics*, 17:2003, 2003.

G.Q. Huang, A.B. Rad, and Y.K. Wong. A new solution to map dynamic indoor environments. *International Journal of Advanced Robotic Systems*, 2008. URL http://sciyo.com/articles/show/title/a_new_solution_to_map_dynamic_indoor_environments.

Dusan Jan, Antonio Roque, Anton Leuski, Jacki Morie, and David Traum. A virtual tour guide for virtual worlds. In *IVA '09: Proceedings of the 9th International Conference on Intelligent Virtual Agents*, pages 372–378, Berlin, Heidelberg, 2009. Springer-Verlag. ISBN 978-3-642-04379-6. doi: http://dx.doi.org/10.1007/978-3-642-04380-2_40.

Gal A. Kaminka, Manuela M. Veloso, Steve Schaffer, Chris Sollitto, Rogelio Adobbati, Andrew N. Marshall, Andrew Scholer, and Sheila Tejada. Gamebots: a flexible test bed for multiagent team research. *Commun. ACM*, 45(1):43–45, 2002. ISSN 0001-0782. doi: http://doi.acm.org/10.1145/502269.502293.

Lydia E. Kavraki, Jean-Claude Latombe, and E. Latombe. Probabilistic roadmaps for robot path planning, 1998.

Kurt Konolige, Charles Ortiz, Regis Vincent, Andrew Agno, Benson Limketkai, Mark Lewis, Linda Briesemeister, Dieter Fox, Jonathan Ko, Benjamin Stewart, and Leonidas Guibas. Centibots: Large scale robot teams. In *In AAMAS*, 2003.

Benjamin Kuipers, Joseph Modayil, Patrick Beeson, Matt Macmahon, and Francesco Savelli. Local metrical and global topological maps in the hybrid spatial semantic hierarchy. In *in IEEE Int. Conf. on Robotics and Automation (ICRA-04)*, pages 4845–4851, 2004.

Benjamin J. Kuipers and Tod S. Levitt. Navigation and mapping in large-scale space. *AI MAGAZINE*, 9:25–43, 1988.

Sylvain Lefebvre, Samuel Hornus, and Fabrice Neyret. Gpu gems - chapter 37, octree textures on the gpu. `http://http.developer.nvidia.com/GPUGems2/gpugems2_chapter37.html`, 2009.

Linden Research Inc. Second life official site. `http://secondlife.com`, 2010.

M. Maurette. Mars rover autonomous navigation. *Auton. Robots*, 14(2-3):199–208, 2003. ISSN 0929-5593. doi: http://dx.doi.org/10.1023/A:1022283719900.

Ranjit Nair and Milind Tambe. Coordinating teams in uncertain environments: A hybrid bdi-pomdp approach.

A. Pronobis, K. Sjoo, A. Aydemir, A.N. Bishop, and P. Jensfelt. A framework for robust cognitive spatial mapping. pages 1 –8, june 2009.

R Ramloll. Navigational agents for large scale virtual worlds. ECSCW'97 Conference Supplement, ECSCW'97 pp.55-56, 1997.

Rameshsharma Ramloll and Darren Mowat. Wayfinding in virtual environments using an interactive spatial cognitive map. In *IV2001 Proceedings (London), IEEE*, pages 2–0. Press, 2001.

Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2003. ISBN 0137903952. URL `http://portal.acm.org/citation.cfm?id=773294`.

G.K. Schmidt and K. Azarm. Mobile robot navigation in a dynamic world using an unsteady diffusion equation strategy. volume 1, pages 642 –647, jul 1992.

Chaoxia Shi, Yanqing Wang, and Jingyu Yang. Online topological map building and qualitative localization in large-scale environment. *Robotics and Autonomous Systems*, 58(5):488 – 496, 2010. ISSN 0921-8890. doi: DOI:10.1016/j.robot.2010.01.009. URL `http://www.sciencedirect.com/science/article/B6V16-4Y95RN6-1/2/3c9e817d78c23aa6ba850c75933aea87`.

Gabe Sibley, Christopher Mei, Ian Reid, and Paul Newman. Vast-scale Outdoor Navigation Using Adaptive Relative Bundle Adjustment. *The International Journal of Robotics Research*, 29(8):958–980, 2010a. doi: 10.1177/0278364910369268. URL `http://online.sagepub.comhttp://ijr.sagepub.com/content/29/8/958.abstract`.

Gabe Sibley, Christopher Mei, Ian Reid, and Paul Newman. Planes, trains and automobiles autonomy for the modern robot. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 285 –292, 3-7 2010b. doi: 10.1109/ROBOT.2010.5509527.

R. Simmons, D. Apfelbaum, W. Burgard, M. Fox, D. an Moors, S. Thrun, and H. Younes. Coordination for multi-robot exploration and mapping. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, Austin, TX, 2000. AAAI.

Reid Simmons and Sven Koenig. Probabilistic robot navigation in partially observable environments. In *IJCAI'95: Proceedings of the 14th international joint conference on Artificial intelligence*, pages 1080–1087, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. ISBN 1-55860-363-8.

Edgar-Philipp Stoffel, Korbinian Schoder, and Hans Jürgen Ohlbach. Applying hierarchical graphs to pedestrian indoor navigation. In *GIS '08: Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems*, pages 1–4, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-323-5. doi: http://doi.acm.org/10.1145/1463434.1463499.

Adriana Tapus and Shrihari Vasudevan. Towards a multilevel cognitive probabilistic representation of space. In *In Proc. of the International Conference on Human Vision and Electronic Imaging X, part of the IST-SPIE Symposium on Electronic Imaging*, 2005.

S. Thrun. Robotic mapping: A survey. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, 2002. to appear.

S. Thrun and A. Bücken. Integrating grid-based and topological maps for mobile robot navigation. In *Proceedings of the AAAI Thirteenth National Conference on Artificial Intelligence*, Portland, Oregon, 1996.

Sebastian Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artif. Intell.*, 99(1):21–71, 1998. ISSN 0004-3702. doi: http://dx.doi.org/10.1016/S0004-3702(97)00078-7.

Denis Wolf and Gaurav Sukhatme. Towards mapping dynamic environments, 2003.

Fang Yuan, Lukas Twardon, and Marc Hanheide. Dynamic path planning adopting human navigation strategies for a domestic mobile robot. *Submitted to the International Conference on Intelligent Robots and Systems*, To Appear. To appear.

H. Zender, O. Martínez Mozos, P. Jensfelt, G. J. M. Kruijff, and W. Burgard. Conceptual spatial representations for indoor mobile robots. *Robot. Auton. Syst.*, 56(6):493–502, 2008. ISSN 0921-8890. doi: http://dx.doi.org/10.1016/j.robot.2008.03.007.

Shlomo Zilberstein and Stuart J. Russell. Anytime sensing, planning and action: A practical model for robot control. In *In Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, pages 1402–1407, 1993.