Towards Improving SLAM Algorithm Development using Augmented Reality

Alex Kozlov
Department of Electrical and Computer Engineering
University of Auckland, Private Bag 92019, Auckland, New Zealand
akoz002@ec.auckland.ac.nz

Bruce MacDonald
Department of Electrical and Computer Engineering
University of Auckland, Private Bag 92019, Auckland, New Zealand
b.macdonald@auckland.ac.nz

Burkhard Wünsche
Department of Computer Science
University of Auckland, Private Bag 92019, Auckland, New Zealand
burkhard@cs.auckland.ac.nz

Abstract

Simultaneous Localisation and Mapping (SLAM) is a popular map building approach in autonomous mobile robotics. Because users demand faster and more effective algorithms, SLAM remains an active area of research. However, the increasing complexity of applications, such as the environments the algorithm is applied to, makes it difficult to debug, evaluate and optimise such algorithms. Our preliminary research indicates that the algorithm development can be improved by using Augmented Reality (AR) systems, which visualise the robot's internal program state and related information specifically in the context of testing and debugging SLAM algorithms. Using inherent SLAM uncertainties and error-sources identified in literature, we developed requirements which an AR system must fulfil in order to optimise the testing, debugging and design of SLAM algorithms.

1 Introduction

Current research in the field of Simultaneous Localisation and Mapping (SLAM) is bringing us ever closer to using autonomous mobile robot assistants [Bailey and Durrant-Whyte, 2006]. SLAM research will allow a mobile robot to be deployed in an unknown environment where it will autonomously map the vicinity. Having acquired a robust map through exploration the robot will be well capable of performing autonomous navigation tasks.

While core problems of SLAM have been intensively researched in terms of computational complexity, map representation and data association [Bailey and Durrant-Whyte, 2006], many challenges still remain. The most important remaining challenge is the development of algorithms for increasingly larger and more unstructured environments [Andreasson *et al.*, 2007;Blanco *et al.*, 2007;Miettinen *et al.*, 2007], e.g. for undersea applications. Concerns here include linearization errors and sensing difficulties due to unstructured environments.

Another current research area is *Multi-robot* SLAM [Neira *et al.*, 2003;Bryson and Sukkarieh, 2007b]. Using several robots to map different regions of a large environment simultaneously can save considerable time. The recent *Marginal-SLAM* algorithm [Martinez-Cantin *et al.*, 2007] aims to address the filter divergence problems associated with the popular Extended Kalman Filter (EKF) and Rao-Blackwellized Particle Filter (RBPF) SLAM solutions. *Robo-centric Map Joining* [Neira *et al.*, 2007] similarly aims to address inconsistency issues due to linearization of EKF SLAM. *MonoSLAM* [Davison *et al.*, 2007] is a real-time SLAM approach using 3D vision, designed to handle large, ambiguous environments and highly dynamic camera motion. A similar 3D vision SLAM system in [Tomono, 2007] aims to address large numbers of landmarks and robustly handle noise and outliers. The work in [Bryson and Sukkarieh, 2007a] concentrates on the issues of consistency and real-time execution for vision-based bearing-only SLAM. The graphical (trajectory-based) SLAM system in [Polkesson and Christensen, 2007] is currently being developed to work with sparse environments and topological map representations. Certain other newer systems, such as *Divide-and-Conquer SLAM* [Paz *et al.*, 2007], although offering improved consistency and complexity, still need to be further evaluated experimentally. Finally, new features are being proposed to enhance the conventional SLAM formula. *P-SLAM* [Chang *et al.*, 2007] aims to predict the mappings of unexplored regions based on knowledge already acquired, thereby saving exploration time. In [Ekvall *et al.*, 2007] while performing SLAM, the robot recognises task related objects and catalogues them in the map for future reference.

The previous paragraphs demonstrate the increasing complexity of research on SLAM algorithms. Consequently, support for SLAM development is essential, both for researchers to tackle the remaining SLAM challenges and for programmers to implement SLAM for commercial and industrial applications. A certain degree of support for SLAM implementation is offered by the popular RDEs (Robot Development Environments). *Player* [Gerkey *et al.*, 2001;Gerkey *et al.*, 2007] offers support for Monte-Carlo localisation and has been used to implement SLAM [Wolf and Sukhatme, 2005]. *CARMEN* (Carnegie Mellon robot navigation toolkit) [Montemerlo *et al.*, 2003a;Montemerlo *et al.*, 2007] and *Miro* (Middleware for robots) [Utz *et al.*, 2002;Utz, 2007] both offer high level support for localisation and map building.

What is missing is the support for testing and debugging that uses visualisation of the program state *imbedded in the real world*. An entirely virtual visualisation of SLAM state has been implemented in [Newman *et al.*, 2002]. As is shown in Figure 1, the vehicle pose and the obstacles are rendered to allow teleoperation when the operator can not physically see the robot.

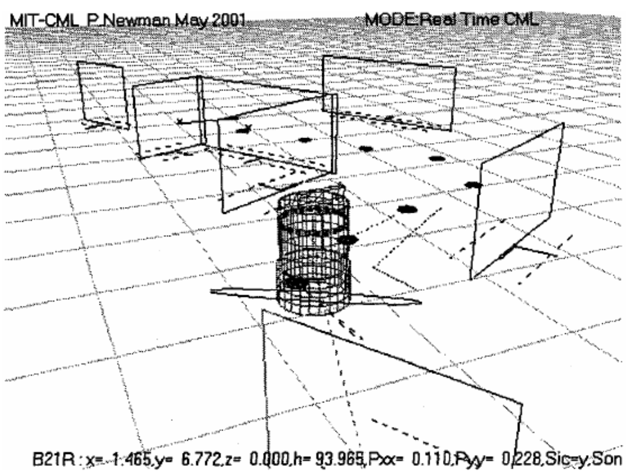


Figure 1: SLAM environment visualisation (with permission from [Newman *et al.*, 2002])

Vision-based SLAM systems from [Davison *et al.*, 2004;Davison *et al.*, 2007] also construct virtual models of the tracked point features for testing purposes. The main drawback with these types of SLAM visualisations is that they are solely virtual. This leaves a cognitive gap between the virtual space and the real world that must be bridged if developers are to understand the relationship between their software and the real world. To improve the perceptual and cognitive overlap between the robot and the human, a form of interaction imbedded in the real world is needed [Breazeal *et al.*, 2001]. *Augmented Reality (AR)* is an appropriate tool for such a system [Chong *et al.*, 2007].

AR is the environment view resulting from generating virtual objects *spatially registered* in real time within a view of a real scene [Azuma, 1997;Azuma *et al.*, 2001;Bimber and Raskar, 2005]. AR has seen limited use in mobile robotics. *ARDev* [Collett and MacDonald, 2006] assists robot programming by visualising range sensor data using AR (Figure 2), which improves the perceptual overlap between the human and the robot. A recent AR system in [Nunez *et al.*, 2006] visualises the nodes of a topological map used by a mobile robot (Figure 3). The user could interact with the robot by creating nodes and assigning them as destination points. Another interesting AR system in [Daily *et al.*, 2003] allowed supervision of a robotic swarm (Figure 4). In a search and rescue application the arrows shown in the figure indicate individual swarm members and lead the rescue team to a survivor.

Research in human-computer interaction indicates that AR would be an ideal candidate for improving SLAM development tools because it provides a perceptual overlap between real and virtual worlds [Edward Swan II and Gabbard, 2005], which allows the user to better understand how the robot perceives and reacts to the sensed environment. In this paper we discuss the requirements for such an AR system, one that visualises the robot's internal program state and related information specifically in the context of testing and debugging SLAM algorithms. To the best of our knowledge no such system currently exists in the literature. The intent is that a tool such as this will ease, for researchers and implementers, the task of testing and debugging SLAM algorithms, and therefore indirectly assist in addressing the remaining SLAM challenges. The remainder of the paper details the functional visualisation requirements of such an AR system. In Section 2 we analyse the classic SLAM approach to deduct the fundamental requirements for visualising a SLAM system. In Section 3 we survey more recent SLAM procedures, which are then used to formulate further, more specialised SLAM visualisation requirements. Section 4 outlines the implementation and hardware arrangements of the proposed system. The paper is concluded in Section 5, summarising all of the requirements and giving directions for future work.

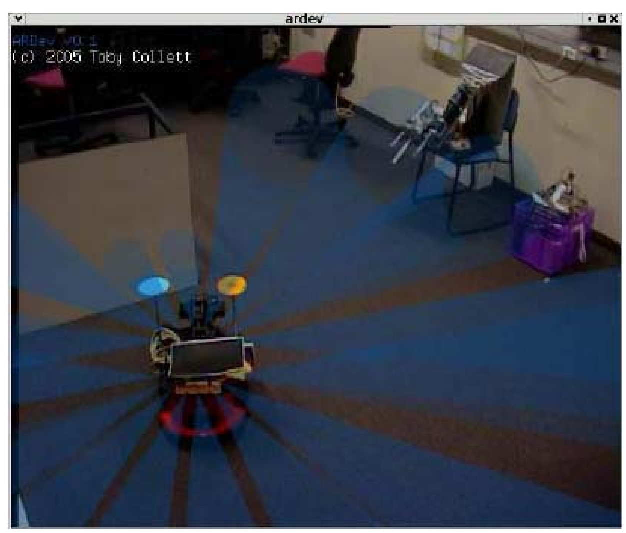


Figure 2: Sonar data visualisation with ARDev (with permission from [Collett and MacDonald, 2006])

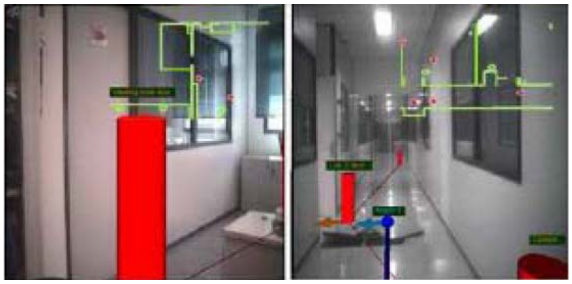


Figure 3:Topological map visualised with AR (with permission from [Nunez *et al.*, 2006])

Figure 5:Robot swarm supervision (with permission from [Daily *et al.*, 2003])

2 Fundamental Requirements

Figure 5 gives an overview of the SLAM process. Firstly, the state of a SLAM program may be visualised; however the state can be defined differently depending on the actual algorithm. For basic requirements, unless otherwise stated, we will consider the state of the popular and wide-spread EKF SLAM. The state of an EKF SLAM algorithm consists of the *stochastic map M:*

$$M = (x, P)$$

where x is the system state vector and P is the system covariance matrix (see Sections 2.1 and 2.2). Also fundamental to SLAM execution are *data association* and *map maintenance* (see Sections 2.3 and 2.4).

Note that this paper compiles a comprehensive list of requirements and not all of them should be visualised at the same time. This would overwhelm the user and lead to visual cluttering which makes it difficult to perceive and discern important information. Instead the user should be able to visualise only those aspects of a SLAM algorithm currently being analysed, with the irrelevant aspects being turned off.

When detailing the requirements, our aim is to list the various information the user will need, for the user to be able to clearly interpret the algorithm. The possible ways in which this information could be visualised will not be discussed here. As long as the information for a particular requirement is clearly perceived, the requirement can be considered satisfied. Methods of visualisation are important [Collett, 2007] but are not addressed here. The following subsections list fundamental requirements and give reasons for them based on previous research and our own observations.

2.1 System State Vector

The system state vector x forms the core part of the

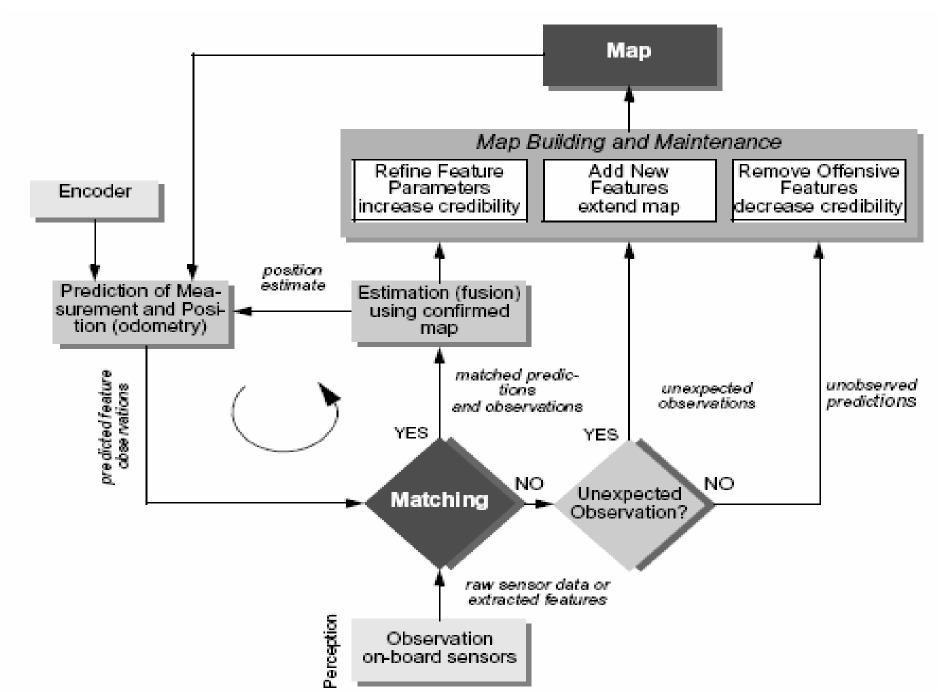


Figure 4: SLAM overview (with permission from [Siegwart and Nourbakhsh, 2004])

system state. It is defined as follows:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x_R} \\ \vdots \\ \mathbf{x_{Fn}} \end{bmatrix}$$

where x_R is the robot pose vector and x_{Fn} is a position vector for every obstacle on the map *Fn*. Essentially the idea here is to visualise the SLAM robot pose and map for comparison with the ground truth seen in the real image. The specifics of how the pose or the map is presented depend on the type of environment representation the robot is using.

Robot Pose

For a metric map representation it is necessary to clearly communicate the (x, y, θ) parameters of the robot pose. There is often a discrepancy between the estimated and the true robot pose, as the true pose is never known [Durrant-Whyte and Bailey, 2006]. This discrepancy needs to be visualised. AR appears to be the most appropriate method because it is not possible to see the true pose with completely virtual or textual visualisations. This does not mean the user must necessarily know the actual values of x, y and θ, but needs to be able to clearly perceive the position and orientation of the SLAM robot pose in relation to the real image of the robot. For any sort of tessellated map representation the current cell in which the robot resides must be shown. Similarly for a topological map, the node representing the current location must be rendered. For an EKF style algorithm the probabilistic mean of the pose should be shown. For a SLAM approach based on RBPFs, such as the popular *FastSLAM* [Montemerlo *et al.*, 2002;2003b], the mean pose or the mean robot *trajectory* of all particles should be visualised.

An implementation requirement is that a SLAM pose or map must be *registered* with the real world, before it can be virtually rendered in the scene. The AR system must track one or more real world objects that are also present in the SLAM model to enable this registration. A likely registration point is the robot pose. Although explicitly visualising the robot pose would be redundant, since the virtual pose would always be correctly superimposed over the real robot image, an important reason for rendering the pose is to visualise the uncertainty (i.e. the covariance) in the estimated robot location and orientation (see Section 2.2).

It may also be useful to visualise the latest *time-update* pose, i.e. the pose predicted by the robot motion model, but uncorrected by external sensors from the latest time-update cycle (Figure 5). This will assist in determining the correctness of the motion model, and the odometry hardware. A large discrepancy between the time-update pose and the observation-update pose may be a concern for the reliability of the motion model and associated hardware.

Map/Obstacles

The purpose of visualising the SLAM obstacle map is to test the mapping accuracy by comparison with the real obstacles seen in the AR image. The discrepancies between true and estimated obstacle positions, resulting from the fact that the true positions are not known [Durrant-Whyte and Bailey, 2006], need to be perceived.

Textual and fully virtual visualisations are not suitable here because they are not able to show the true obstacle positions. The specifics here again depend upon the style of environment representation used. For a metric map, point (x, y) and line (r, ά) features need to be visualised as they are present in the SLAM model. For a tessellated map, the occupied cells need to be visualised, along with the degree of occupancy. In a topological map, virtual markers should exemplify topological nodes, and the spatial boundaries between nodes may need to be conveyed. Alternatively, if nodes are strongly associated with prominent landmarks and spatial divisions are not of high importance, it may be adequate to only highlight each feature that constitutes a node. Secondly, the edges linking nodes must be depicted for topological maps. For recent hybrid approaches, such as metric-topological maps [Blanco *et al.*, 2007], characteristics of both representations will need to be visualised together as appropriate for the application. As for the pose, gaussian mean locations will be shown for the EKF, and particle means for RBPF methods.

Visualising the pose and the map is fundamental to subsequent visualisation requirements for state covariance, data association and map maintenance, which are associated with the pose and map visualisations. The aim of the following requirements is to graphically view some behavioural characteristic of the algorithm.

2.2 System Covariance Matrix

In EKF SLAM the system covariance matrix *P* is defined as follows:

$$\mathbf{P} = \begin{bmatrix} \mathbf{P_R} & \cdots & \mathbf{P_{RFn}} \\ \vdots & \ddots & \vdots \\ \mathbf{P_{FnR}} & \cdots & \mathbf{P_{Fn}} \end{bmatrix}$$

It represents two types of information. Firstly, the main-diagonal elements $\mathbf{P_R}...\mathbf{P_{Fn}}$ represent the absolute uncertainty of the robot pose and the landmark locations. Landmark uncertainties converge to a lower-bound dependent on the initial uncertainties [Durrant-Whyte and Bailey, 2006]. Therefore, these elements need to be graphically portrayed if the human programmer is to observe and test for correct behaviour. Similarly, for RBPF approaches, the uncertainty approximated by particles can be graphicaly displayed in some association with the pose and the landmarks.

Secondly, the off-diagonal elements $\mathbf{P_{RFn}}...\mathbf{P_{FnR}}$ represent the cross-correlations among the pose and various landmarks. As SLAM progresses successive observations should cause relative correlations between landmarks to monotonically increase [Durrant-Whyte and Bailey, 2006]. Consequently, these cross-correlations need to be visualised to confirm this behaviour and the algorithm's correctness.

Note that it may be suitable to visualise this information without AR, as text or graphics on screen. But as we aim for an *immersive* testing environment, we hypothesise this information should be incorporated into the AR view using a consistent visualisation approach to improve usability, possibly as text-based AR [Starner *et* *al.*, 1997].

2.3 Data Association

The aim of the data association stage is to formulate a hypothesis *h*:

$$h = \left[j_1, j_2, \ldots, j_m \right]$$

Each element of the hypothesis j_i signifies which mapped feature F_j the observation $z_{,i}$ corresponds to, where *m* is the number of observations made. The first step is to make some *predictions* of the features F_j the system expects to observe, using the observation model. Then *actual* observations $z_{,i}$ are obtained with external sensors. The data association algorithm then matches predicted to actual observations to construct the hypothesis *h* (also see Figure 5). Classic SLAM formulates a single hypothesis during the observation-update step.

It is desirable to visualise the hypothesis *h*. In other words the AR system should graphically depict for each real observation which feature in the SLAM map the observation corresponds to. To achieve this is it necessary for the visualisation of every mapped feature to be uniquely identifiable. The purpose of this is to keep track of the *consistency* of the data association hypothesis throughout successive observation-updates. For example, say in time-step *k*, physical obstacle A is rendered in red, i.e. colour is used to uniquely identify mapped features. If in the subsequent time-step *k+1*, obstacle A is now blue, then there is obviously a problem as the hypothesis at *k+1* is inconsistent with the hypothesis at *k*. The system thinks the observation of the physical obstacle A now refers to a different mapped feature. Traditionally, data association has been an error-prone and problematic area of SLAM [Neira and Tardos, 2001]. Although recent research brought considerable improvements [Bailey and Durrant-Whyte, 2006], specialised data association algorithms are currently being developed for certain applications, such as *Divide-and-Conquer SLAM* [Paz *et al.*, 2007]. This ongoing research suggests that the visualisation of data association algorithm information is indeed a useful tool for SLAM development.

Data associations could be shown as text. Each real observation and each mapped feature would be uniquely labelled; then the associations would simply be printed on screen. Research has shown that immersive programming environments can be quite effective compared to conventional programming environments [Osawa *et al.*, 2002]. We therefore suggest a purely AR mode or alternatively combining text and AR modes, resulting in a fully immersive environment.

As explained earlier, the hypothesis *h* is constructed from predicted and actual observations. We suggest that both predicted and actual observations should be distinctly visualised. Then the programmer can explicitly see the discrepancies between predicted and actual observations (i.e. the *innovation*). This makes it easier to identify any erratic behaviour, and easier to assess the degree of robustness of the data association method. Moreover, if the actual observation visualisation does not precisely align with the real view obstacle, it may be an indication of a sensor malfunction[1].

Multi-Hypothesis Tracking (MHT) [Davey, 2007] is a recent data association technique where multiple hypotheses are tracked to produce multiple localisation and mapping possibilities. It may be desirable to visualise multiple hypotheses, but care must be taken not to overwhelm the user with information. Each hypothesis represents its own map, and so the AR view may become cluttered and overloaded in rendering more than a handful of hypotheses.

2.4 Observation-Update and Map Maintenance

There are three elemental outcomes of the data association step: *matched* predictions, *unmatched* predictions and *unexpected* observations (see Figure 5). Each case has slightly different visualisation requirements. Since this data is closely related to the data association step, it should be visualised in the same way, that is with AR as opposed to purely graphical or textual methods.

Matched predictions are elements j_i of the hypothesis *h*, i.e. those predictions successfully matched to real observations. SLAM subsequently uses the hypothesis and the real observations to update the parameters of the state vector and the covariance matrix. Earlier we stated that it is necessary to virtually render predicted and actual observations. In addition, it is also important to visualise the *updated* features, i.e. the new feature positions updated using the real observations. Visualising all three (predicted, actual, updated) feature locations will provide the programmer with near complete knowledge of the observation-update process, thus bridging the cognitive gap between the programmer and the robot. In particular, the programmer will be able to see how the updated location was produced from the predicted and actual feature locations, easily noting any inconsistencies.

The second possible outcome for a mapped feature is that its prediction is not matched to any real observation. This could be due to the predicted feature being created by spurious measurements or being removed from the map (however, often it is assumed that the latter can not happen). The usual response in this situation is to remove the feature from the SLAM map. The AR system needs to ensure the user in fact observed the change. In other words it needs to make clear that a previously existing feature, if any, was removed from the map a moment ago. The reason for this is to communicate to the programmer whether the SLAM system appropriately handles unmatched predictions.

The final possible outcome is the presence of observations that do not match any predictions. These could be spurious measurements or, more importantly, new features that should be added to the map. If the SLAM algorithm decides an observation is in fact a new feature, this addition needs to be visualised with the AR tool. In order to allow the programmer to ensure the SLAM algorithm correctly adds new features to the map, the AR tool must ensure the user observes any changes.

There may be an additional *credibility indicator* attached to each map feature. These denote how likely a given obstacle is to exist in the environment. Hence unmatched predictions may reduce the credibility value instead of removing the obstacle. On the other hand matched predictions would increase credibility values. If credibility values are implemented in the algorithm, it would be helpful to visualise them with AR, to verify their correct operation.

[1] Work in [Collett and MacDonald, 2006] deals with visualising sensor data with AR.

3 Application-specific Requirements

Many recently presented SLAM methods have been developed for specialised applications and hence require individually tailored visualisations which we outline in this section. Visualisation requirements for these systems are typically variations of the fundamental requirements for SLAM described in Section 2. These are broadly divided into algorithms using different map representations and addressing computational complexity.

3.1 Map Representation

A number of recent SLAM procedures use map representations somewhat different from conventional ones. Although the representations are different, the aim of visualising them with AR remains the same: comparison with the ground truth in the real image, in order to reveal the discrepancy between true and estimated parameters [Durrant-Whyte and Bailey, 2006]. Textual or fully graphical methods are generally not appropriate here as the real environment is not directly perceived. Certain pieces of information may sufficiently be represented with text, i.e. differentiating between predicted, temporary, permanent obstacles and generally labelling them with auxiliary information (see below). However, we instead propose to incorporate them into the AR view for an immersive and consistent environment.

In [Ekvall *et al.*, 2007] SLAM has been combined with an object detection/recognition system. This allows the system to detect predefined objects and integrate them into the map for future reference, usually for household service tasks. When testing such a system with an AR tool it is desirable to appropriately label the predefined task-related objects so the programmer can see the reliability of the system in operation.

ScanSLAM [Nieto *et al.*, 2005] uses dense raw scan data to generate features of arbitrary shapes in the SLAM map (Figure 6). Systems in [Gutmann and Konolige, 1999] and [Newman *et al.*, 2006] use similar ideas of map representations generated by arbitrary sensor scan patterns. The AR visualisation tool should reflect these representations and render the scan patterns as obstacles. *Trajectory-based* SLAM, where the whole history of robot poses is maintained, normally associates sensor scan patterns to robot poses [Eustice *et al.*, 2005b;Dellaert and Kaess, 2006]. In this instance the AR tool should visualise the trajectory together with the scan patterns.

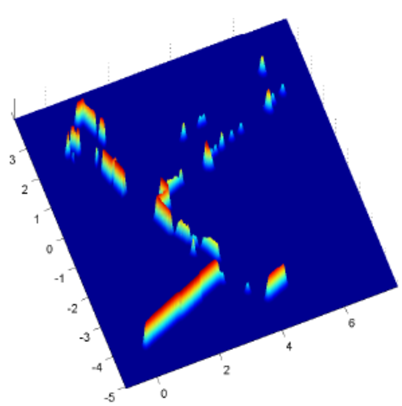


Figure 6: Range-laser data for mapping with ScanSLAM (with permission from [Nieto *et al.*, 2005])

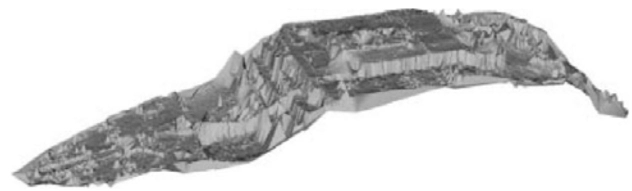
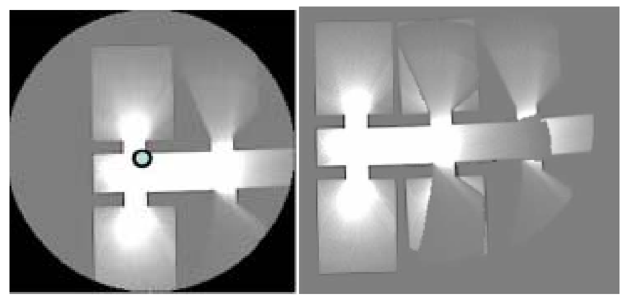


Figure 7: Undersea mapping of RMS Titanic in 3D (with permission from [Eustice *et al.*, 2005a])



(a) (b)

Figure 8: Predicting unexplored areas in P-SLAM. (a) shows the explored region. (b) shows the explored region combined with the prediction hypothesis. (with permisson from [Chang *et al.*, 2007])

3D environment models have become prominent as SLAM applications move towards unstructured, outdoor environments [Kim and Sukkarieh, 2004;Eustice *et al.*, 2005a;Zhang *et al.*, 2007]. As SLAM becomes implemented for airborne and underwater vehicles, the ability to move in 3D entails a 3D environment model (Figure 7). Vision based SLAM typically uses some kind of a 3D environment model of salient feature points [Chen *et al.*, 2007]. Evidently, an AR visualisation system must address these models by visualising the obstacles and the robot pose in 3D. However, this could be complex to implement, and registration may be more difficult because of the unstructured environment.

The real world is dynamic in the sense of having transient and moving obstacles. SLAM algorithms need to be able to deal with that. This could be achieved by discriminating between temporary and permanent obstacles [Bailey, 2002;Hahnel *et al.*, 2003]. In this case the AR visualisation must also depict permanent and temporary obstacles differently. An alternative way of addressing dynamic environments is by tracking moving obstacles [Chieh-Chih *et al.*, 2003]. In this case the AR system must track obstacles as they move; moving obstacles can not be sufficiently represented with text or solely virtual graphics.

Finally, the approach in [Chang *et al.*, 2007] attempts to predict unexplored areas with the knowledge of what has already been explored (Figure 8). If a repeating pattern in a region is detected, the system hypothesises that an unexplored region has the same shape. When testing such a system with an AR tool it will be necessary to distinguish between explored and hypothesised regions.

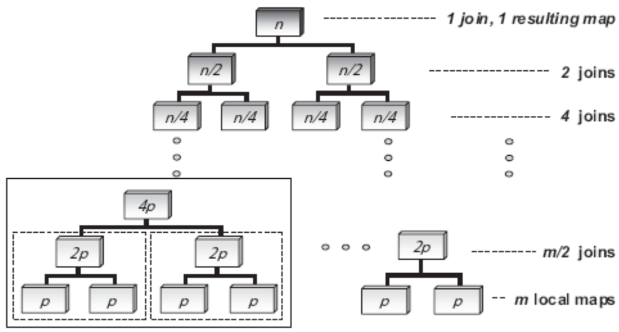


Figure 9: Map hierarchy in Divide-and-Conquer SLAM. The leaf nodes are local maps and the parent nodes are joined maps (with permission from [Paz *et al.*, 2007])

3.2 Computational Complexity and Submapping

Much work has been done to improve the computational complexity of the classic SLAM problem. In this section we discuss which concepts of algorithm efficiency would benefit visualisation within an AR environment.

State-augmentation [Williams, 2001] has been proposed to improve the computational complexity of the time-update step. Instead of updating the whole covariance matrix during the time-update step, only the elements that directly involve the robot pose are updated, i.e. the covariance and cross-covariance matrices of landmarks only, are unchanged. This method suggests the AR tool clearly show which covariance elements are updated during the time-update step, in order to ensure the algorithm is operating correctly.

Submapping has become highly popular as a way of tackling large-environment SLAM. The idea is to divide the global environment map into submaps of fixed or varying sizes, with the aim being to reduce the complexity by operating within a smaller submap. The submaps could possibly be merged into the global map after a period of time [Guivant and Nebot, 2001;Knight *et al.*, 2001;Paz *et al.*, 2007] (Figure 9), or alternatively the submaps could remain separated [Guivant and Nebot, 2002;Frese, 2006]. For these sorts of approaches, an AR tool will need to visualise the submap region boundaries, and clarify which features belong to which region, to communicate the SLAM map to the programmer. The reason is that linearization errors can be significant in large environments for some approaches, such as global submapping [Bailey and Durrant-Whyte, 2006]; AR visualisations will help detect such cases. The argument for using AR here instead of text/graphics is the same as for other map representations. Multi-robot SLAM [Neira *et al.*, 2003;Bryson and Sukkarieh, 2007b] utilises submapping; different robots explore different regions to build independent submaps which are then joined together. The AR requirement here is slightly different. Different robots' poses will need to be rendered uniquely; furthermore each robot's submap and its boundaries will need to be distinguishable.

4 AR system for SLAM visualisation

In this section the proposed implementation of the AR system for SLAM visualisation is briefly outlined. Additionally, the AR hardware arrangement to be used for testing the system is described.

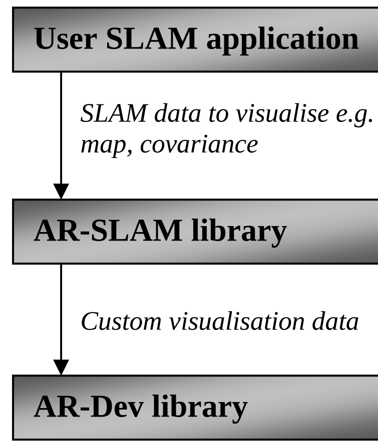


Figure 10: Implementation outline of the proposed AR system

4.1 Implementation

It is intended to implement the AR system for SLAM using the functionality provided by an existing AR system, ARDev [Collett and MacDonald, 2006;Collett, 2007]. ARDev is primarily used for visualising robot sensor data in mobile robot applications. However it provides additional functionality for custom visualisations in the context of mobile robotics. Figure 10 shows the implementation outline and intended usage. A user's SLAM application will pass the data to be visualised to the AR-SLAM library; for example SLAM pose and map information. The AR-SLAM system, using the custom visualisation functionality of AR-Dev, will pass the required information to AR-Dev to generate the AR images. This approach reuses existing functionality of AR-Dev, avoiding the need to reimplement it.

4.2 Hardware

The system will be tested in a laboratory setup which uses a wall-mounted camera, providing video-based see-through AR. This can be seen in Figure 11. The AR system will use a fiducial attached to the robot to track it. The camera image with virtual augmentations will then be displayed on a wall-mounted screen. An important consideration will be registration of the virtual SLAM imagery with the real camera image. This will be done by aligning the SLAM pose with the robot's fiducial. This will ensure the SLAM map and other data are correctly aligned with the real world when rendered. The system may be extended to wearable AR displays in the future.

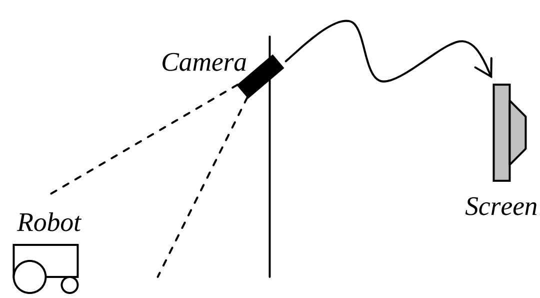


Figure 11: Hardware arrangement intended for testing

5 Conclusions

Our research in SLAM development and AR applications showed that an AR system that visualises the robot's internal program state and related information can improve the testing and debugging of SLAM algorithms. In this paper the requirements for such a system have been detailed, and these are summarised in Figure 12. Fundamental requirements include visualising the system state vector, system covariance matrix, plus related information on data association and map maintenance. Newer SLAM approaches elicit their own particular requirements. Non-conventional map representations will necessitate visualisations of 3D environments, sensor scan-patterns, path trajectories and dynamic obstacles. New methods of addressing computational complexity, such as submapping, require visualisations of submaps and their boundaries.

The next step is to implement a prototype of such an AR system for SLAM development. Initially we will concentrate on satisfying fundamental requirements, and eventually expand to add more system specific functions. An important consideration will be the choice of visualisations for each different type of information.

Acknowledgement

Alex Kozlov is supported by a University of Auckland Doctoral Scholarship.

References

[Andreasson, H. *et al.*, 2007] Andreasson, H. *et al.* Mini-SLAM: Minimalistic Visual SLAM in Large-Scale Environments Based on a New Interpretation of Image Similarity. *IEEE International Conference on Robotics and Automation*, pages 4096-4101, 2007.

[Azuma, R. *et al.*, 2001] Azuma, R. *et al.* Recent advances in augmented reality. *Computer Graphics and Applications, IEEE,* 21(6): 34-47, 2001.

[Azuma, R. T., 1997] Azuma, R. T. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments(1054-7460),* 6(4): 355-385, 1997.

[Bailey, T., 2002] Bailey, T. *Mobile robot localisation and mapping in extensive outdoor environments*. PhD Thesis. Australian Center of Field Robotics. University of Sydney 2002.

[Bailey, T. and Durrant-Whyte, H., 2006] Bailey, T. and Durrant-Whyte, H. Simultaneous localization and mapping (SLAM): part 2. *IEEE Robotics and Automation Magazine,* 13(3): 108-117, 2006.

[Bimber, O. and Raskar, R., 2005] Bimber, O. and Raskar, R. *Spatial Augmented Reality: Merging Real and Virtual Worlds* A K Peters, Limited, 2005.

[Blanco, J.-L. *et al.*, 2007] Blanco, J.-L. *et al.* A New Approach for Large-Scale Localization and Mapping: Hybrid Metric-Topological SLAM. *IEEE International Conference on Robotics and Automation*, pages 2061-2067, 2007.

[Breazeal, C. *et al.*, 2001] Breazeal, C. *et al.* Active vision for sociable robots. *Systems, Man and Cybernetics, Part A, IEEE Transactions on,* 31(5): 443-453, 2001.

[Bryson, M. and Sukkarieh, S., 2007a] Bryson, M. and Sukkarieh, S. Building a robust implementation of bearing-only inertial SLAM for a UAV. *Journal of Field Robotics,* 24(1-2): 113-43, 2007a.

[Bryson, M. and Sukkarieh, S., 2007b] Bryson, M. and Sukkarieh, S. Co-operative Localisation and Mapping for Multiple UAVs in Unknown Environments. *Aerospace Conference, IEEE* pages 1-12, 2007b.

[Chang, H. J. *et al.*, 2007] Chang, H. J. *et al.* P-SLAM: Simultaneous Localization and Mapping With Environmental-Structure Prediction. *IEEE Transactions on Robotics,* 23(2): 281-293, 2007.

Requirement	Details of Visualisations	Purpose
Robot Pose	Metric parameters in 2D or 3D, currently occupied topological node or grid cell, mean path trajectory	To compare with the ground truth seen in the real image
State Map	Point or line feature, grid cells, topological nodes with connections, sensor scan-patterns , 3D environment models, submaps, dynamic /temporary /predicted obstacles	To compare with the ground truth seen in the real image, and ensure correctness of the mapping method
State Covariance	Location uncertainties of pose and landmarks, correlations between pose and landmarks, state augmentation updates	To test for algorithm correctness and expected convergence behaviours
Data Association	The data association hypothesis, predicted and actual observations	To ensure consistency and correctness of the hypothesis, compare real observations with the real scene and view the innovation
Map Maintenance	Map maintenance behaviours for matched and unmatched predictions, and unexpected observations	To ensure correct behaviour in each of the possible outcomes of the data association step

Figure 12: Summary of AR visualisation requirements for SLAM

[Chen, Z. *et al.*, 2007] Chen, Z. *et al.* Recent advances in simultaneous localization and map-building using computer vision. *Advanced Robotics,* 21(3-4): 233-65, 2007.

[Chieh-Chih, W. *et al.*, 2003] Chieh-Chih, W. *et al.* Online simultaneous localization and mapping with detection and tracking of moving objects: theory and results from a ground vehicle in crowded urban areas. *Proceedings. ICRA '03. IEEE International Conference on Robotics and Automation*, pages 842-849 vol.1, 2003.

[Chong, J. W. S. *et al.*, 2007] Chong, J. W. S. *et al.* Methodologies for Immersive Robot Programming in an Augmented Reality Environment. *The International Journal of Virtual Reality,* 6(1): 69-79, 2007.

[Collett, T. H. J., 2007] Collett, T. H. J. *Augmented reality visualisation for mobile robot developers*. PhD Thesis. Electrical and Computer Engineering. University of Auckland, Auckland, 2007.

[Collett, T. H. J. and MacDonald, B. A., 2006] Collett, T. H. J. and MacDonald, B. A. Developer oriented visualisation of a robot program. *ACM SIGCHI/SIGART Human-Robot Interaction*: 49-56, 2006.

[Daily, M. *et al.*, 2003] Daily, M. *et al.* World embedded interfaces for human-robot interaction. *System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on*, 2003.

[Davey, S. J., 2007] Davey, S. J. Simultaneous Localization and Map Building Using the Probabilistic Multi-Hypothesis Tracker. *IEEE Transactions on Robotics,* 23(2): 271-280, 2007.

[Davison, A. J. *et al.*, 2004] Davison, A. J. *et al.* Real-time 3D SLAM with wide-angle vision. *Proc. IFAC Symposium on Intelligent Autonomous Vehicles, Lisbon*, 2004.

[Davison, A. J. *et al.*, 2007] Davison, A. J. *et al.* MonoSLAM: Real-Time Single Camera SLAM. *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* 29(6): 1052-1067, 2007.

[Dellaert, F. and Kaess, M., 2006] Dellaert, F. and Kaess, M. Square Root SAM: Simultaneous Localization and Mapping via Square Root Information Smoothing. *The International Journal of Robotics Research,* 25(12): 1181, 2006.

[Durrant-Whyte, H. and Bailey, T., 2006] Durrant-Whyte, H. and Bailey, T. Simultaneous Localisation and Mapping: Part 1. *IEEE Robotics and Automation Magazine,* 13(2): 99-108, 2006.

[Edward Swan II, J. and Gabbard, J. L., 2005] Edward Swan II, J. and Gabbard, J. L. Survey of User-Based Experimentation in Augmented Reality. *1st International Conference on Virtual Reality*, Las Vegas, Nevada, 2005.

[Ekvall, S. *et al.*, 2007] Ekvall, S. *et al.* Object detection and mapping for service robot tasks. *Robotica,* 25: 175-87, 2007.

[Eustice, R. *et al.*, 2005a] Eustice, R. *et al.* Visually navigating the RMS Titanic with SLAM information filters. *Proceedings of Robotics: Science and Systems*, 2005a.

[Eustice, R. M. *et al.*, 2005b] Eustice, R. M. *et al.* Exactly Sparse Delayed-State Filters. *ICRA. Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2417-2424, 2005b.

[Frese, U., 2006] Frese, U. Treemap: An O(log n) algorithm for indoor simultaneous localization and mapping. *Autonomous Robots,* 21(2): 103-122, 2006.

[Gerkey, B. *et al.*, 2007] Gerkey, B. *et al. The Player Project*. Retrieved 26 June 2007, from http://playerstage.sourceforge.net/.

[Gerkey, B. P. *et al.*, 2001] Gerkey, B. P. *et al.* Most valuable player: a robot device server for distributed control. *Proceedings. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1226-1231 vol.3, 2001.

[Guivant, J. and Nebot, E., 2002] Guivant, J. and Nebot, E. Improving computational and memory requirements of simultaneous localization and map building algorithms. *Proceedings. ICRA '02. IEEE International Conference on Robotics and Automation*, pages 2731-2736, 2002.

[Guivant, J. E. and Nebot, E. M., 2001] Guivant, J. E. and Nebot, E. M. Optimization of the simultaneous localization and map-building algorithm for real-time implementation. *IEEE Transactions on Robotics and Automation,* 17(3): 242 - 257, 2001.

[Gutmann, J. S. and Konolige, K., 1999] Gutmann, J. S. and Konolige, K. Incremental mapping of large cyclic environments. *CIRA '99. Proceedings. 1999 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 318-325, 1999.

[Hahnel, D. *et al.*, 2003] Hahnel, D. *et al.* Map building with mobile robots in dynamic environments. *Proceedings. ICRA '03. IEEE International Conference on Robotics and Automation*, pages 1557-1563 vol.2, 2003.

[Kim, J. and Sukkarieh, S., 2004] Kim, J. and Sukkarieh, S. Autonomous airborne navigation in unknown terrain environments. *Aerospace and Electronic Systems, IEEE Transactions on,* 40(3): 1031-1045, 2004.

[Knight, J. *et al.*, 2001] Knight, J. *et al.* Towards constant time SLAM using postponement. *Proceedings. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 405-413 vol.1, 2001.

[Martinez-Cantin, R. *et al.*, 2007] Martinez-Cantin, R. *et al.* Analysis of Particle Methods for Simultaneous Robot Localization and Mapping and a New Algorithm: Marginal-SLAM. *IEEE International Conference on Robotics and Automation*, pages 2415-2420, 2007.

[Miettinen, M. *et al.*, 2007] Miettinen, M. *et al.* Simultaneous Localization and Mapping for Forest Harvesters. *IEEE International Conference on Robotics and Automation*, pages 517-522, 2007.

[Montemerlo, M. *et al.*, 2003a] Montemerlo, M. *et al.* Perspectives on standardization in mobile robot programming: the Carnegie Mellon Navigation (CARMEN) Toolkit. *(IROS 2003). Proceedings.*

IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 2436-2441 vol.3, 2003a.

[Montemerlo, M. *et al.*, 2007] Montemerlo, M. *et al.* *Carmen, Robot Navigation Toolkit*. Retrieved 27 June 2007, from http://carmen.sourceforge.net/.

[Montemerlo, M. *et al.*, 2002] Montemerlo, M. *et al.* Fast-SLAM: A factored solution to the simultaneous localization and mapping problem. *Proc. AAAI Nat. Conf. Artif. Intell.*, 2002.

[Montemerlo, M. *et al.*, 2003b] Montemerlo, M. *et al.* Fast-SLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. *Proc. Int. Joint Conf. Artif. Intell.*, 2003b.

[Neira, J. *et al.*, 2007] Neira, J. *et al.* Robocentric map joining: Improving the consistency of EKF-SLAM. *Robotics and Autonomous Systems,* 55(1): 21-9, 2007.

[Neira, J. and Tardos, J. D., 2001] Neira, J. and Tardos, J. D. Data association in stochastic mapping using the joint compatibility test. *Robotics and Automation, IEEE Transactions on,* 17(6): 890-897, 2001.

[Neira, J. *et al.*, 2003] Neira, J. *et al.* Linear time vehicle relocation in SLAM. *Proceedings. ICRA '03. IEEE International Conference on Robotics and Automation*, pages 427-433, 2003.

[Newman, P. *et al.*, 2006] Newman, P. *et al.* Outdoor SLAM using visual appearance and laser ranging. *ICRA 2006. Proceedings IEEE International Conference on Robotics and Automation*, pages 1180-1187, 2006.

[Newman, P. *et al.*, 2002] Newman, P. *et al.* Explore and return: experimental validation of real-time concurrent mapping and localization. *Proceedings. ICRA '02. IEEE International Conference on Robotics and Automation*, 2002.

[Nieto, J. *et al.*, 2005] Nieto, J. *et al.* Scan-SLAM: Combining EKF-SLAM and Scan Correlation. *International Conference on Field and Service Robotics (FSR)*, 2005.

[Nunez, R. *et al.*, 2006] Nunez, R. *et al.* A human-robot interaction system for navigation supervision based on augmented reality. *MELECON 2006. 2006 IEEE Mediterranean Electrotechnical Conference . Malaga, Spain. IEEE Region 8. IEEE Spain Sect. 16-19 May* 2006.

[Osawa, N. *et al.*, 2002] Osawa, N. *et al.* An Immersive Programming System: Ougi. *The 12th International Conference on Artificial Reality and Telexistence (ICAT 2002)*, pages 36-43, 2002.

[Paz, L. M. *et al.*, 2007] Paz, L. M. *et al.* EKF SLAM updates in O(n) with Divide and Conquer SLAM. *IEEE International Conference on Robotics and Automation*, pages 1657-1663, 2007.

[Polkesson, J. and Christensen, H. I., 2007] Polkesson, J. and Christensen, H. I. Graphical SLAM for outdoor applications. *Journal of Field Robotics,* 24(1-2): 51-70, 2007.

[Siegwart, R. and Nourbakhsh, I. R., 2004] Siegwart, R. and Nourbakhsh, I. R. *Introduction to Autonomous Mobile Robots*. The MIT Press, Cambridge, Massachusetts, 2004.

[Starner, T. *et al.*, 1997] Starner, T. *et al.* Augmented reality through wearable computing. *Presence: Teleoperators and Virtual Environments,* 6(4): 386-398, 1997.

[Tomono, M., 2007] Tomono, M. Monocular SLAM Using a Rao-Blackwellised Particle Filter with Exhaustive Pose Space Search. *IEEE International Conference on Robotics and Automation*, pages 2421-2426, 2007.

[Utz, H., 2007] Utz, H. *Miro (Middleware for Robots)*. Retrieved 6 July 2007, from http://smart.informatik.uni-ulm.de/MIRO/idl_051205/index.html.

[Utz, H. *et al.*, 2002] Utz, H. *et al.* Miro - middleware for mobile robot applications. *IEEE Transactions on Robotics and Automation,* 18(4): 493-497, 2002.

[Williams, S. B., 2001] Williams, S. B. *Efficient solutions to autonomous mapping and navigation problems*. PhD Thesis. Australian Center of Field Robotics. University of Sydney, 2001.

[Wolf, D. F. and Sukhatme, G. S., 2005] Wolf, D. F. and Sukhatme, G. S. Mobile robot simultaneous localization and mapping in dynamic environments. *Autonomous Robots,* 19(1): 53-65, 2005.

[Zhang, Z. *et al.*, 2007] Zhang, Z. *et al.* Finding Disaster Victims: A Sensory System for Robot-Assisted 3D Mapping of Urban Search and Rescue Environments. *IEEE International Conference on Robotics and Automation*, pages 3889-3894, 2007.