

Covariance Visualisations for Simultaneous Localisation and Mapping

Alex Kozlov
akoz002@aucklanduni.ac.nz
University of Auckland

Bruce MacDonald
b.macdonald@auckland.ac.nz
University of Auckland

Burkhard Wünsche
burkhard@cs.auckland.ac.nz
University of Auckland

Abstract

Simultaneous Localisation and Mapping (SLAM) is a method of environment mapping in mobile robotics. One of the most popular classes of this algorithm is the Extended-Kalman Filter (EKF) SLAM, which maps the environment by estimating similarities between currently registered scene objects and newly perceived ones. More advanced versions of this algorithm are necessary, e.g. for multiple robots or outdoor environments. However, development is difficult because of the complex interaction between the internal robot state, the perceived scene and the actual scene. New visualisation methods are hence required to enable developers to debug and evaluate EKF-SLAM algorithms. We present novel Augmented Reality based visualisation techniques which display the algorithm's progress by visualising feature and robot pose estimates, as well as correlations between features and clusters of features. The techniques allow a qualitative estimate of the algorithm's mapping compared with the ground truth and indicate the correctness and convergence properties of the SLAM system.

1 Introduction

Simultaneous Localisation and Mapping (SLAM) is a method of environment mapping in mobile robotics. The robot explores and maps an unknown environment using onboard sensors, while localising itself within its current map. The purpose is that, having acquired a robust map through exploration, the robot will be well capable of performing autonomous navigation tasks within the mapped environment. While core problems of SLAM have been intensively researched in terms of computational complexity, map representation and data association [Durrant-Whyte and Bailey, 2006], many

challenges still remain. The most important current challenges are the development of systems for increasingly larger and more unstructured environments [Andreasson *et al.*, 2007; Blanco *et al.*, 2007; Miettinen *et al.*, 2007], and achieving robustness and reliability necessary for real world SLAM applications [Bailey and Durrant-Whyte, 2006; Martinez-Cantin *et al.*, 2007; Polkesson and Christensen, 2007].

The most well established underlying algorithm for SLAM systems is the Extended-Kalman Filter (EKF) SLAM [Durrant-Whyte and Bailey, 2006; Maybeck, 1979], which uses a multivariate Gaussian distribution to model the state estimation. Despite the emergence of other algorithms, such as the application of Rao-Blackwellized Particle Filters to SLAM [Montemerlo *et al.*, 2002; 2003], EKF SLAM is still widely applied [Nieto *et al.*, 2005; Paz *et al.*, 2007; Neira *et al.*, 2007]. A number of important theorems for the behaviour of the map covariance in EKF SLAM have been proven in [Dissanayake *et al.*, 2001; Huang and Dissanayake, 2007]. These deal with increasing correlations between features and decreasing feature uncertainties over time. In general, regardless of the application, an EKF SLAM system must follow these behaviours for correct convergence of the SLAM solution (some exceptions do exist, such as systems with moving features [Wang *et al.*, 2003]). That is why, from the SLAM developer's point of view, it is necessary to *visualise* these behaviours to confirm the system's adherence. Only then can the developer compare the detailed behaviour of the SLAM software with the expected behaviour as the SLAM estimation unfolds. This will enable related algorithm errors to be identified and corrected.

Augmented Reality (AR) is a visualisation approach that involves rendering virtual objects spatially registered within the view of a real world scene [Bimber and Raskar, 2005; Azuma, 1997; Azuma *et al.*, 2001]. It is especially suited for applications where information must be rendered directly within the context of the real world. AR has been applied to a limited extent in mobile

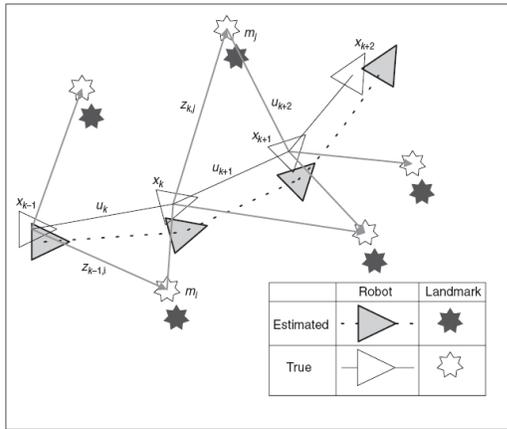


Figure 1: The SLAM problem (with permission from [Durrant-Whyte and Bailey, 2006]).

robotics; this includes range sensor visualisations [Collett and MacDonald, 2006], swarm control [Daily *et al.*, 2003] and way point designation [Nunez *et al.*, 2006]. The goals of these applications are to improve human robot interaction and to communicate the robot's *perception* of the world to the developer. One useful extension to visualising robot perceptive data in these works is to use AR to also visualise the internal workings of a robot's algorithms.

This paper presents an AR visualisation system for EKF SLAM algorithms. In particular, the contributions are: the development of novel visualisation methods of EKF SLAM covariance behaviours for feature correlations, and the implementation of these visualisations in AR. The developed visualisations are directly based on the established covariance behaviour theorems. At the basic level, presented AR visualisations of SLAM states provide a qualitative assessment of the estimates compared with the ground truth from the real image capture. More importantly however, visualising established covariance behaviours gives an indication of the correctness and convergence properties of the SLAM system. Moreover by using an AR environment, if the correct behaviours are not followed, bugs or potential real world causes may become apparent since AR shows the real world context.

The layout of the paper is as follows. Section 2 gives mathematical background on EKF SLAM and explains the covariance behaviours on which the visualisations are based. Section 3 outlines the developed AR system along with the specific SLAM visualisations presented. Experimental results for visualisation of EKF SLAM data are given in Section 4.

2 SLAM

Figure 1 shows the essential SLAM problem. The robot explores and maps an unknown environment, while localising itself within its current map. Because the ground truth of the environment is never known, there will always be a discrepancy between the *true* and the *estimated* robot path and environment features, as shown in Figure 1. In this work we deal with two dimensional point-feature based EKF SLAM. A brief mathematical background is given below.

2.1 EKF SLAM

The EKF SLAM state is represented by a state vector and a state covariance matrix, comprising a multivariate Gaussian distribution [Dissanayake *et al.*, 2001]. The estimated state vector $\hat{x}(k|k)$ at time k is

$$\hat{x}(k|k) = \begin{bmatrix} \hat{x}_v(k|k) \\ \hat{x}_m(k|k) \end{bmatrix}$$

where \hat{x}_v is the vehicle pose estimate, and \hat{x}_m is the estimated map vector composed of n point features $f_1 \dots f_n$. The state covariance $P(k|k)$ is

$$P(k|k) = \begin{bmatrix} P_{vv}(k|k) & P_{vm}(k|k) \\ P_{vm}^T(k|k) & P_{mm}(k|k) \end{bmatrix}$$

where P_{vv} is the vehicle covariance matrix, P_{mm} is the map covariance matrix, and P_{vm} is the cross-covariance between the vehicle and the map. The state covariance represents the overall uncertainty associated with the state vector estimate. An EKF SLAM iteration involves three stages: prediction, observation and update.

Motion Prediction

The motion of the vehicle is modelled in the following way:

$$x_v(k+1) = f_v(x_v(k), u_v(k+1)) + v_v(k+1) \quad (1)$$

$$x_m(k+1) = x_m(k) \quad (2)$$

where $f_v(\cdot)$ is the kinematic prediction function, u_v is the vehicle control input and v_v is Gaussian motion noise. As features are assumed static, they are unaffected by the prediction step (as seen in (2)). Using the motion model in (1) and (2), the state at $k+1$ is predicted as follows:

$$\hat{x}(k+1|k) = f(\hat{x}(k|k), u(k)) \quad (3)$$

$$P(k+1|k) = F(k)P(k|k)F^T(k) + Q(k) \quad (4)$$

where $F(k)$ is the Jacobian linearization of $f(\cdot)$ at the estimate $\hat{x}(k|k)$. $Q(k)$ is the process noise covariance.

Observation

The observation model for features is:

$$z_i(k) = h_i(x(k)) + w_i(k) \quad (5)$$

where $z_i(k)$ is the sensor observation of landmark i . $h_i()$ is the observation function and $w_i(k)$ is observation noise. Using (5), the observations at $k+1$ are predicted:

$$\hat{z}_i(k+1|k) = h_i(\hat{x}(k+1|k)) \quad (6)$$

Following this prediction, an *actual* feature observation $z_i(k+1)$ is made. This enables the calculation of the innovation $v_i(k+1)$ and the innovation covariance $S_i(k+1)$ for every feature i :

$$v_i(k+1) = z_i(k+1) - \hat{z}_i(k+1|k) \quad (7)$$

$$S_i(k+1) = H_i(k+1)P(k+1|k)H_i^T(k+1) + R_i(k+1) \quad (8)$$

where $H_i(k+1)$ is the Jacobian linearization of $h_i()$ at the estimate $\hat{x}(k+1|k)$. $R_i(k+1)$ is the observation noise covariance.

Update

Finally, the innovations and their covariances are used to update the EKF state with the information gained from the sensor observations:

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + W_i(k+1)v_i(k+1) \quad (9)$$

$$P(k+1|k+1) = P(k+1|k) - W_i(k+1)S_i(k+1)W_i^T(k+1) \quad (10)$$

where $W_i(k+1)$ is the Kalman Filter Gain:

$$W_i(k+1) = P(k+1|k)H_i^T(k+1)S_i^{-1}(k+1) \quad (11)$$

2.2 Covariance Behaviour

In order to debug and improve an EKF SLAM algorithm the user must be able to detect unusual behaviour and must be able to verify that expected behaviour does occur. Our visualisations utilise properties of the EKF SLAM map covariance $P_{mm}(k|k)$ proven in [Dissanayake *et al.*, 2001; Huang and Dissanayake, 2007]. In particular, we will display feature correlations. This section outlines the mathematical properties of these variables during execution of the algorithm. The visualisations themselves are explained later in Section 3.2. The full proofs are found in the original papers.

Feature Correlations

The relevant theorem states: *In the limit the landmark estimates become strongly correlated.* Let d_{ij} be the relative position between any two feature estimates f_i and f_j . Let P_d be the covariance of d_{ij} :

$$d_{ij} = f_i - f_j \quad (12)$$

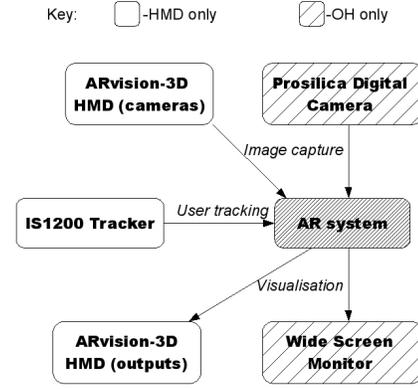


Figure 2: System Overview.

$$P_d = P_{ii} + P_{jj} - P_{ij} - P_{ij}^T \quad (13)$$

P_d is the *relative covariance* between the two features. It is an indication of how they vary relative to each other as the algorithm progresses. In [Huang and Dissanayake, 2007] it was shown that P_d will decrease to a certain lower bound as observations are made. As P_d decreases, the correlation between the features strengthens. In this work we visualise P_d between a pair of features to give an indication of their correlation, and to observe the decrease of P_d and thus the strengthening of the correlation between the features over time.

3 AR Visualisation System

3.1 System Overview

Figure 2 presents an overview of the AR visualisation system. The system has been implemented with two different hardware arrangements: a fixed overhead (OH) camera and a head-mounted-display (HMD) with a vision-based tracker. The necessary modules within a video-see-through AR system are: image capture (input), camera tracking (when the camera is mobile), processing and image augmentation, and visualisation display (output). The HMD setup used a Trivisio ARvision-3D HMD for image capture and visualisation display, together with a Matrox DualHead2Go device to obtain stereo output. For the OH system, a fixed, ceiling mounted Prosilica Digital Camera captures images, with the visualisation showing on a wide screen monitor. Camera (user) tracking for the HMD was done with an InterSense IS1200 vision tracker. As the IS1200 driver for user tracking is only available for Windows, VMWare Server 2 was running guest Windows XP and the IS1200 driver. No camera tracking is required for the OH setup because the camera is fixed, and this is the main difference between the two setups.

With both setups of the presented AR system, it is necessary to correctly align the virtual SLAM visualisations with the real world in the captured images. This is known as the *registration* problem. Here, registration is achieved by using ARToolKitPlus (ARTKP) [Wagner and Schmalstieg, 2007] to visually track the robot within the captured images. A Pioneer 3-DX mobile robot is used with an ARTKP marker mounted on top. When the robot starts SLAM, its initial position is assumed to be the origin of the SLAM map. This position is computed in world coordinates with ARTKP, stored in memory and subsequently used for registration of all SLAM visualisations. Once the initial position is obtained, the robot is no longer tracked with ARTKP. The initial position is all that is required for registration. It should be emphasized that this is purely a *testing environment* for SLAM systems. Obviously in real SLAM applications, no camera would be available to localise the robot. The ARTKP localisation is used purely for registration of AR images to visualise the SLAM estimates computed by the robot.

There are two stages in the operation of the AR system: *initialisation* and *execution*. During initialisation, the robot’s initial position is computed with ARTKP for registration as outlined above. Execution visualises time-varying SLAM data as SLAM runs. An execution render cycle involves the following three steps:

1. Update the user’s pose using the tracker software (only for the HMD setup)
2. Query the SLAM system for the latest data to be visualised
3. Update AR visualisations accordingly

The system was implemented in C++ using the Observer software pattern [Gamma *et al.*, 1995]. The Observer pattern is well suited for this application; it is designed for an *observer* entity to monitor a *subject* entity while keeping the two as loosely coupled as possible. In the present application the SLAM system is the subject and the AR visualisation system is the observer monitoring the subject. The SLAM subject notifies the observer every SLAM update; this triggers the AR observer’s render cycle, querying the subject for the new data and visualising it.

3.2 SLAM Visualisations

The work in this paper centers on using AR to visualise certain behaviours of the EKF SLAM map covariance $P_{mm}(k|k)$ as explained earlier in Section 2.2. Adherence to these behaviours is a good indicator that the SLAM system has been implemented correctly and is likely to lead to a convergent solution. The visualisations show the developer whether these established behaviours are being followed.

State Estimate Visualisation

This is the fundamental visualisation of the SLAM state vector. It is the conventional way to visualise SLAM uncertainty, and is necessary for subsequent visualisations of the correlations. The visualisation consists of the following:

- *Robot path*: The robot position is shown as a green conical marker and the orientation is shown as a flat yellow arrow. A green line indicates the path of the robot. The conical shape was chosen as it tapers to a point, which denotes the 2D SLAM position estimate.
- *Feature locations*: Feature locations are shown as cyan conical markers. Similarly to the robot pose, their points denote the estimated 2D SLAM feature positions.

Using AR with non-simulation SLAM systems, the user can see the *estimated* SLAM features and robot pose (virtual objects) in direct comparison with the *actual* robot pose and environment features (within the captured image). This allows a good qualitative assessment of the SLAM error and performance. A problem can be present if the *registration error* of the AR system is high. In that situation it is unclear whether the visual discrepancy between virtual and actual objects is due to SLAM or AR registration error. This is slightly problematic for the HMD AR setup, but not for the fixed overhead camera AR setup where the registration error is negligible.

Correlation Visualisations

The purpose here is to visualise the relative covariance P_d between a pair of features, as explained in Section 2.2. By doing so we are visualising the strength of the correlation between the pair, and likewise the change of this correlation over time. To do this we apply the tensor ellipsoid technique from scientific visualisation. For each pair of features F_i and F_j the visualisation contains:

- A yellow line linking the features F_i and F_j in question
- A red tensor “correlation” ellipsoid for $P_d(i, j)$ rendered in the center of the line

Tensor ellipsoids are a standard visualisation technique for second-order tensor-valued scientific data [Chen *et al.*, 2009; Cisternas *et al.*, 2008; Slavin *et al.*, 2004]. Firstly, eigendecomposition is performed on the tensor in question, to represent it in its canonical form. Then the tensor is visualised as an n -dimensional ellipsoid, where: the directions of the principal axes are given by the eigenvectors, the lengths of the axes are given by the corresponding eigenvalues and n is the dimension of the tensor. In our application, P_d is treated as a 2-dimensional

second-order tensor, thus resulting in a 2-dimensional ellipsoid glyph.

The expected correlation behaviour discussed in Section 2.2 dictates that P_d will decrease thus indicating a strengthening correlation between the pair of features. In the visualisation this behaviour is embodied as the reduction of the area of the tensor ellipsoid for P_d (termed ‘‘correlation ellipsoid’’). Hence, the correct behaviour can be confirmed from the visualisation if the correlation ellipsoids become visibly smaller over time. This is intended as a testing tool for SLAM systems. If the ellipsoids instead become *larger*, this indicates divergent behaviour and a problem with the SLAM implementation. The orientation of the ellipsoid shows the correlation between the two elements of the $d_{i,j}$ vector.

The problem of visual cluttering may arise when too many ellipsoids are rendered close together on the screen, making it difficult to discern individual ellipsoids. Several steps have been taken to address this problem in the visualisation. Firstly, the 2D ellipsoid glyph is expanded into 3D, using the minor eigenvalue as the length of the axis into the third dimension. This method gives better distinction to overlapping ellipsoids. Secondly, the features are divided into spatial clusters, and only the correlations between features in different clusters are shown. This reduces cluttering by reducing the total number of correlations in view. The clustering was performed using the basic single-linkage hierarchical clustering method [Legendre and Legendre, 1998], where the metric used is the Euclidean distance between the features.

To further reduce the number of ellipsoids in view, an extension to the clustering method was implemented. Instead of visualising all of the inter-cluster correlations, only the minimum, mean, and maximum correlations are shown for every pair of clusters. The visualisation consists of a line joining the centroids of the clusters, the mean correlation in the center of the line, with the minimum underneath and the maximum above it. The drawback of this approach is the information loss from removal of individual correlations.

4 Experimental Results

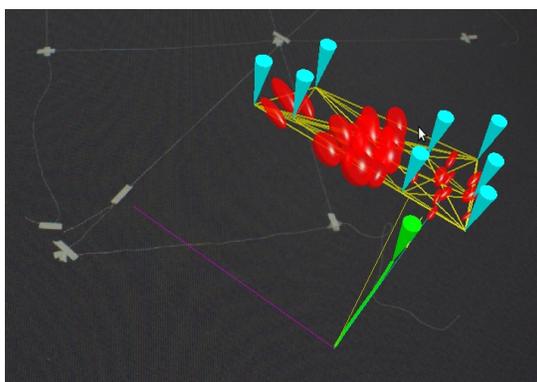
The experiments were performed on an Intel Pentium D Dual Core 3.0 GHz processor with 2.0 GB of memory running Ubuntu 8.10, using the overhead fixed camera setup only. Three types of experiments were performed: simulation with correct data, simulation with faulty data, and real SLAM data. All data was from point-feature based EKF SLAM systems. The state vector and covariance data at each time step of the experiment were sent to the AR system. The AR system then visualised in real time the feature correlations, along with the mean estimates of the features and the robot path.

For the simulation experiments, a map of 15 features was used, with the features arranged in three spatial clusters. The features were enclosed in a 3 by 3 meter area, and the robot travelled in a 1 by 1 meter loop within that area. The feature area was kept small to ensure it would fit within the field of view of the camera used for image capture. Figure 3 shows the correlations between all of the features for correct simulated data. Observing the time progression of the experiment (Fig. 3 (a) to (b)), the correlation ellipsoids between the features initialised in (a) visibly decrease. This follows the established behaviour shown in [Dissanayake *et al.*, 2001; Huang and Dissanayake, 2007], stating that relative covariances P_d decrease, which in turn means that the correlations between the features *increase* as observations are made. Although the overall behaviour can be observed, one difficulty is seeing *individual* ellipsoids between pairs of features. This is true when ellipsoids are bunched close together in the view, such as ones positioned between the clusters of features.

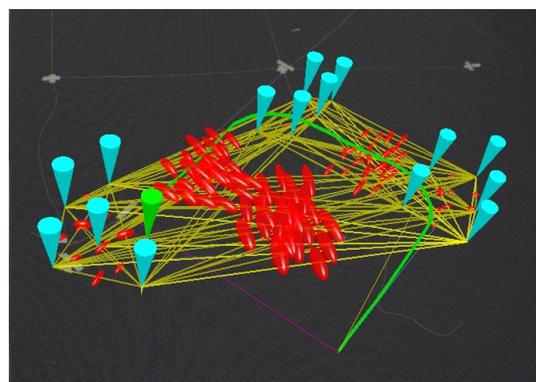
Figure 4 shows the correlations between the features in different spatial clusters for correct simulated data. This method removes from view the correlations inside clusters, in order to remedy the visual cluttering problem. As above, it can be observed that the correlations between clusters increase, as the ellipsoids decrease during the time progression from Fig. 4 (a) to (b). However, the visual cluttering problem is still present to a lesser extent, as individual ellipses between clusters can be hard to see.

Figure 5 shows the min/mean/max correlations between the different spatial clusters for correct simulated data. This addresses the visual cluttering problem quite well. Now the number of visualised correlations between clusters is reduced to three, representing the given metrics of inter-cluster correlations. However, a consequence of this is a small degree of information loss, as it is not possible to see the correlation between a specific pair of features. The expected behaviour is seen as the min, mean and max ellipsoids between the first two clusters decrease in the transition from Fig. 5 (a) to (b). The minimum ellipsoid must be non-increasing, as was the case in the experiment. The maximum ellipsoid was found to increase when a new feature is initialised in a cluster, as that feature is initially poorly correlated to the existing SLAM map. The mean ellipsoid showed a general decreasing trend over time.

For the next set of experiments, a programming fault was introduced into the simulation. The fault caused incorrect update of the covariance of certain mapped features, causing the feature correlations to worsen over time and violate the established behaviour. The map used was identical to the previous experiment. Figure 6 shows the correlations between all of the features for

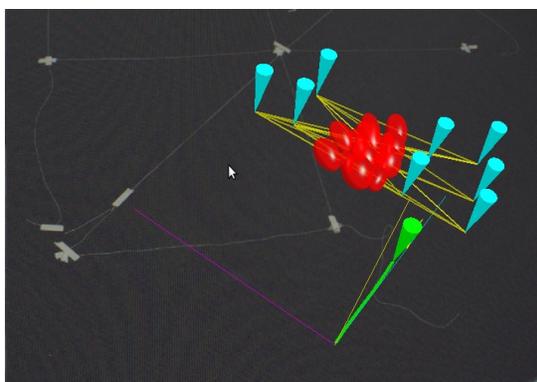


(a) Start of simulation

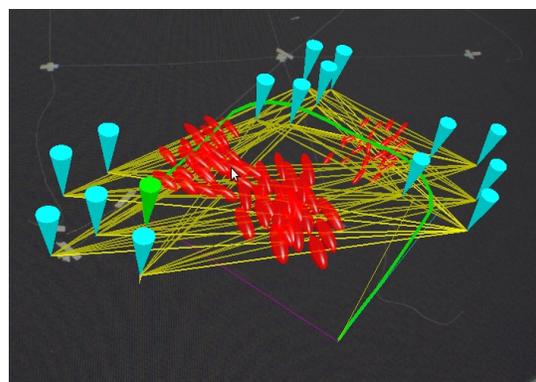


(b) End of simulation

Figure 3: Simulation with correct data, visualising all of the correlations. The ellipses decrease in size during the progression from (a) to (b), which shows the expected behaviour of strengthening correlations between features.

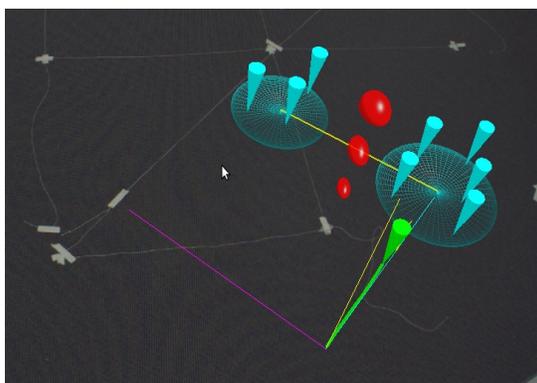


(a) Start of simulation

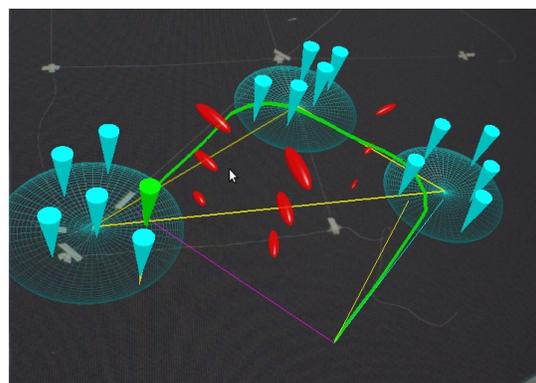


(b) End of simulation

Figure 4: Simulation with correct data, visualising only the correlations between spatial clusters. The ellipses decrease in size during the progression from (a) to (b), which shows the expected behaviour of strengthening correlations between clusters.



(a) Start of simulation



(b) End of simulation

Figure 5: Simulation with correct data, visualising the min/mean/max values of the inter-cluster correlations. The ellipses decrease in size during the progression from (a) to (b), which shows the expected behaviour of strengthening correlations between clusters.

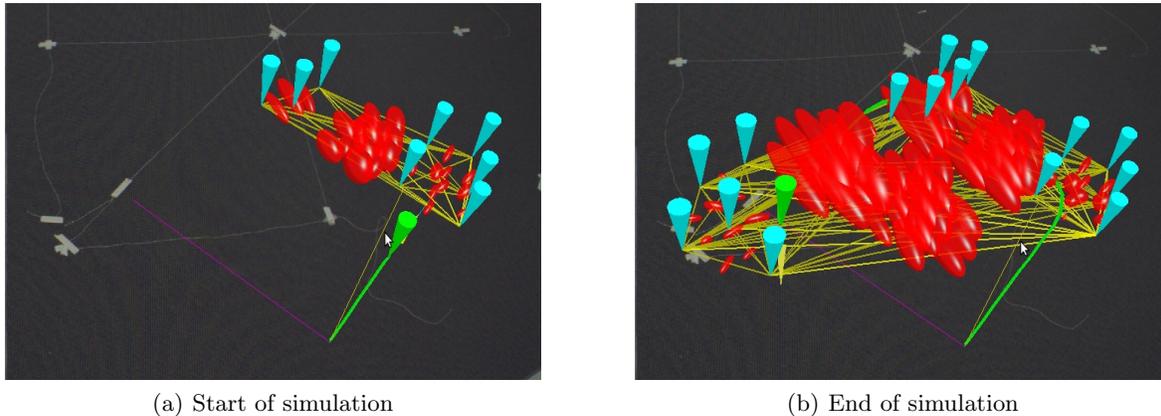


Figure 6: Simulation with faulty data, visualising all of the correlations. The ellipses increase from (a) to (b) and compared to Fig. 3, which shows the violation of the expected behaviour of strengthening correlations.

the faulty simulated data. The visualisation makes the problem immediately clear, as the ellipses grow from Fig. 6 (a) to (b), showing worsening correlations. However, the biggest problem is that it is difficult to pin point the fault. From the visualisation it appears that all of the correlations are affected, which is not the case.

Figure 7 shows the correlations between the features in different spatial clusters for faulty simulated data. As above, the error is obvious as the ellipses become larger from Fig. 7 (a) to (b). Visual clutter is problematic here; as the ellipses grow they become more difficult to distinguish individually. As in the previous case, it is hard to specify exactly which correlations are affected by the fault.

In the final simulation experiment, Figure 8 shows the min/mean/max correlations between the spatial clusters for faulty simulated data. The error can be observed although it is less apparent. The best indicator is that the maximum ellipse increases even during update steps when no new features are found. However, the minimum diminishes as per correct behaviour, likely due to the correlations not affected by the fault. The fault is also easily seen when Fig. 8 is compared to the case when the correct simulation data is used in Fig. 5.

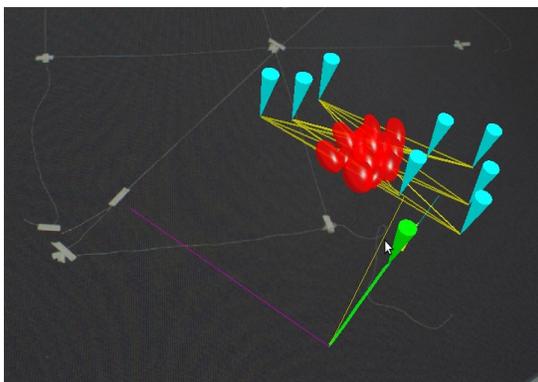
For the experiments with the real SLAM data, a Pioneer robot performed SLAM in a 1 by 1 meter loop using laser-rangefinder sensors. Five features were positioned within the 3 by 3 meter field of view of the camera. No more features could be placed within that area due to a minimum separating distance between features required for data association. For the simulated data this was not a concern. For this reason, it is difficult to compose visible clusters for the real data experiment, and cluster visualisations were not used. Instead all of the correlations were visualised. Registration was achieved with an ARTKP marker mounted on top of the robot.

The robot's initial position is taken as the origin of the SLAM map.

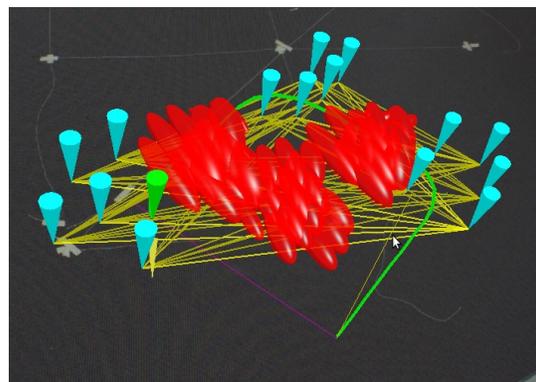
Figure 9 shows the early stage in the experiment with real SLAM data. The robot has only observed the first three features. As the experiment continues in Fig. 10, all of the features have been found. Also note the presence of a spurious feature to the far right in Fig. 10 (b). This is a result of the laser readings reflected by walls and the surrounding environment. The expected behaviour can be seen in Fig. 11 after several loops. Comparing Fig. 10 (b) and Fig. 11 (b), the ellipsoids have shrunk thus indicating that the correlations between the features have strengthened. In addition, Figures 9 - 11 give a qualitative assessment of the overall performance of the system. The user can see the ground truth (the robot and the physical features) compared with the SLAM estimates (the conical markers).

5 Conclusions and Future Work

This paper presented novel AR visualisation methods for EKF SLAM state and covariance behaviour pertaining to feature correlations. Visualising this behaviour helps the developer assess the correctness and convergence properties of their SLAM system. The results exhibit that the correct behaviour can be confirmed from the visualisation, as well as incorrect behaviour resulting from programmer error. To address visual cluttering, features are divided into spatial clusters and inter-cluster correlations are visualised; alternatively only the minimum, mean and maximum inter-cluster correlations can be shown. Future work will look at dynamically manipulating the visualisation, and explicitly flagging behaviour violations to the user.

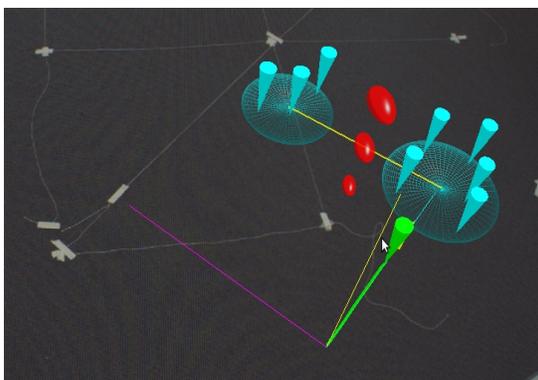


(a) Start of simulation

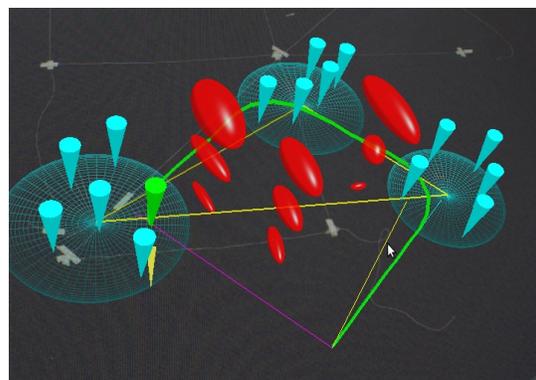


(b) End of simulation

Figure 7: Simulation with faulty data, visualising only the correlations between spatial clusters. The ellipses increase from (a) to (b) and compared to Fig. 4, which shows the violation of the expected behaviour of strengthening correlations.

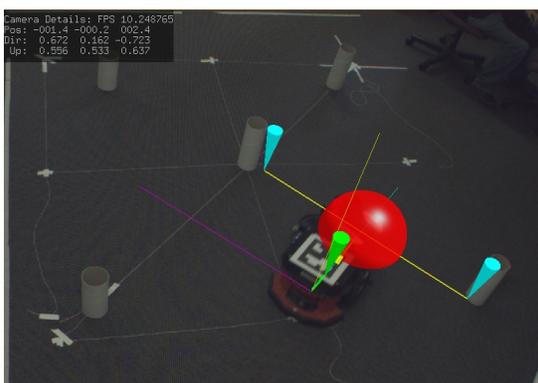


(a) Start of simulation

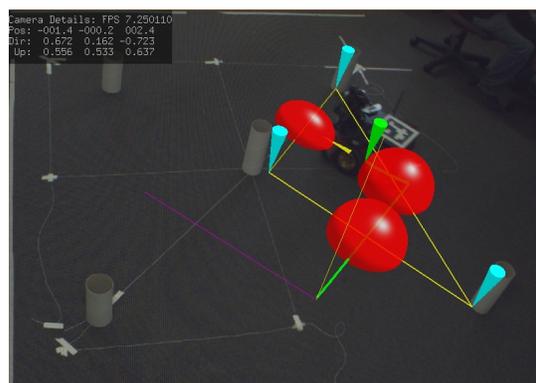


(b) End of simulation

Figure 8: Simulation with faulty data, visualising the min/mean/max values of the inter-cluster correlations. The ellipses increase from (a) to (b) and compared to Fig. 5, which shows the violation of the expected behaviour of strengthening correlations.



(a)



(b)

Figure 9: Early stage in the experiment with real SLAM data. The robot has only observed the first three features.

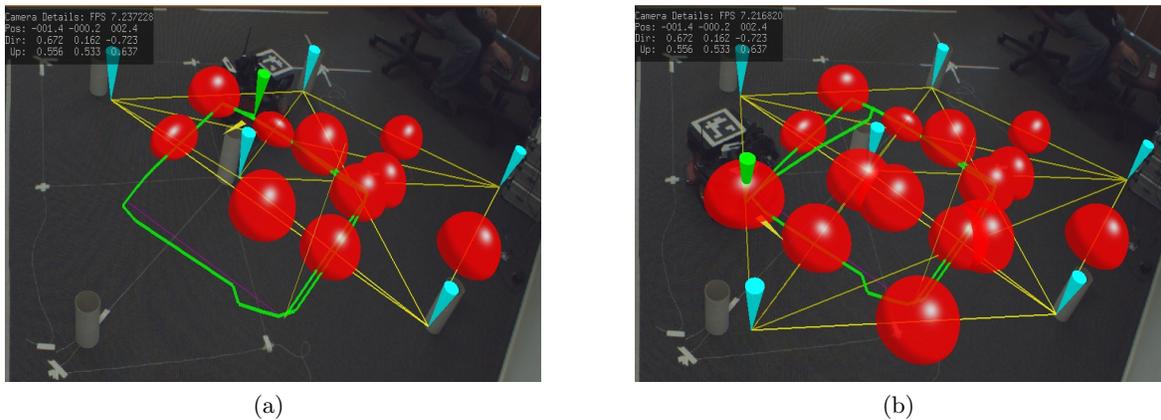


Figure 10: Middle stage in the experiment with real SLAM data. In (b) all of the features have been found, including one erroneous feature to the far right.

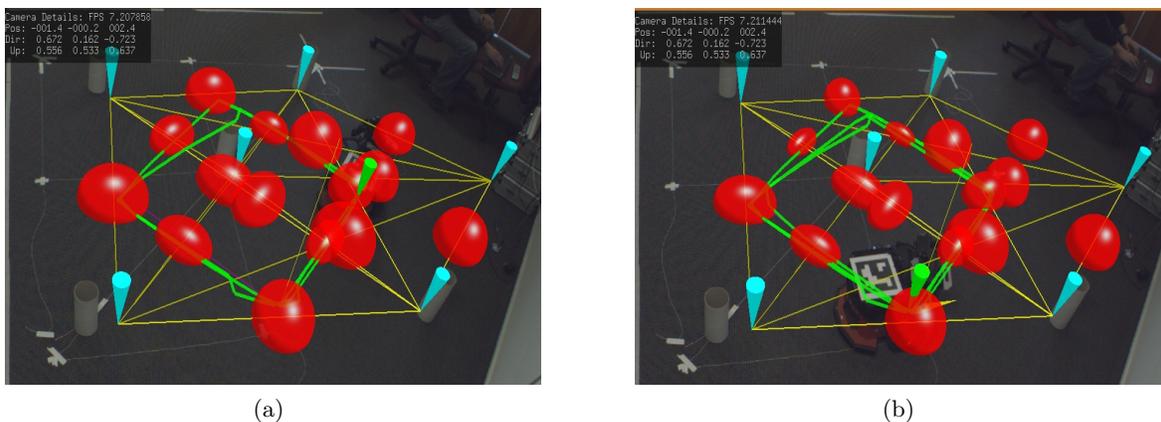


Figure 11: Late stage in the experiment with real SLAM data. The ellipses decrease in size during the experiment, (compare Fig. 10 (b) and Fig. 11 (b)), which shows the expected behaviour of strengthening correlations between features.

References

- [Andreasson *et al.*, 2007] H. Andreasson, T. Duckett, and A. Lilienthal. Mini-SLAM: Minimalistic Visual SLAM in Large-Scale Environments Based on a New Interpretation of Image Similarity. In *IEEE International Conference on Robotics and Automation*, pages 4096–4101, 2007.
- [Azuma *et al.*, 2001] R. Azuma, Y. Baillet, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre. Recent advances in augmented reality. *IEEE Computer Graphics and Applications*, 21(6):34–47, 2001.
- [Azuma, 1997] R.T. Azuma. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385, 1997.
- [Bailey and Durrant-Whyte, 2006] T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (slam): part 2. *IEEE Robotics and Automation Magazine*, 13(3):108–117, 2006.
- [Bimber and Raskar, 2005] O. Bimber and R. Raskar. *Spatial Augmented Reality : Merging Real and Virtual Worlds*. A K Peters, Limited, 2005.
- [Blanco *et al.*, 2007] J.L. Blanco, J.A. Fernandez-Madrigal, and J. Gonzalez. A New Approach for Large-Scale Localization and Mapping: Hybrid Metric-Topological SLAM. In *IEEE International Conference on Robotics and Automation*, pages 2061–2067, 2007.
- [Chen *et al.*, 2009] Wei Chen, Song Zhang, S. Correia, and D.F. Tate. Visualizing diffusion tensor imaging data with merging ellipsoids. In *Visualization Sympos-*

- sium, 2009. PacificVis '09. IEEE Pacific*, pages 145–151, 2009.
- [Cisternas *et al.*, 2008] J. Cisternas, T. Asahi, M. Galvez, and G. Rojas. Regularization of diffusion tensor images. In *Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008. 5th IEEE International Symposium on*, pages 935–938, 2008.
- [Collett and MacDonald, 2006] T.H.J. Collett and B.A. MacDonald. Developer oriented visualisation of a robot program. In *ACM SIGCHI/SIGART Human-Robot Interaction*, pages 49–56, 2006.
- [Daily *et al.*, 2003] M. Daily, Y. Cho, K. Martin, and D. Payton. World embedded interfaces for human-robot interaction. In *Annual Hawaii International Conference on System Sciences*, pages 6–12, 2003.
- [Dissanayake *et al.*, 2001] M. W. M. Gamini Dissanayake, Paul Newman, Steven Clark, Hugh F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (slam) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, 2001.
- [Durrant-Whyte and Bailey, 2006] H. Durrant-Whyte and T. Bailey. Simultaneous localisation and mapping: Part 1. *IEEE Robotics and Automation Magazine*, 13(2):99–108, 2006.
- [Gamma *et al.*, 1995] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design patterns: elements of reusable object-oriented software*. Addison-Wesley, 1995.
- [Huang and Dissanayake, 2007] Shoudong Huang and Gamini Dissanayake. Convergence and consistency analysis for extended kalman filter based slam. *Robotics, IEEE Transactions on*, 23(5):1036–1049, 2007.
- [Legendre and Legendre, 1998] Pierre Legendre and Louis Legendre. *Numerical ecology*. New York : Elsevier, 1998.
- [Martinez-Cantin *et al.*, 2007] R. Martinez-Cantin, N. De Freitas, and J.A. Castellanos. Analysis of particle methods for simultaneous robot localization and mapping and a new algorithm: Marginal-slam. In *IEEE International Conference on Robotics and Automation*, pages 2415–2420, 2007.
- [Maybeck, 1979] P.S. Maybeck. *Stochastic Models, Estimation and Control, vol. I*. New York: Academic, 1979.
- [Miettinen *et al.*, 2007] M. Miettinen, M. Ohman, A. Visala, and P. Forsman. Simultaneous Localization and Mapping for Forest Harvesters. In *IEEE International Conference on Robotics and Automation*, pages 517–522, 2007.
- [Montemerlo *et al.*, 2002] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. Fast-slam: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the National Conference on Artificial Intelligence*, pages 593–598, 2002.
- [Montemerlo *et al.*, 2003] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. Fast-slam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1151–1156, 2003.
- [Neira *et al.*, 2007] J. Neira, J.A. Castellanos, R. Martinez-Cantin, and J.D. Tardos. Robocentric map joining: Improving the consistency of EKF-SLAM. *Robotics and Autonomous Systems*, 55(1):21–29, 2007.
- [Nieto *et al.*, 2005] J. Nieto, T. Bailey, and E. Nebot. Scan-slam: Combining ekf-slam and scan correlation. In *International Conference on Field and Service Robotics (FSR)*, pages 129–140, 2005.
- [Nunez *et al.*, 2006] R. Nunez, J.R. Bandera, J.M. Perez-Lorenzo, and F. Sandoval. A human-robot interaction system for navigation supervision based on augmented reality. In *IEEE Mediterranean Electrotechnical Conference*, pages 441–444, 2006.
- [Paz *et al.*, 2007] L.M. Paz, P. Jensfelt, J.D. Tardos, and J. Neira. EKF SLAM updates in $O(n)$ with Divide and Conquer SLAM. In *IEEE International Conference on Robotics and Automation*, pages 1657–1663, 2007.
- [Polkesson and Christensen, 2007] J. Polkesson and H.I. Christensen. Graphical SLAM for outdoor applications. *Journal of Field Robotics*, 24(2):51–70, 2007.
- [Slavin *et al.*, 2004] V.A. Slavin, D.H. Laidlaw, R. Pelcovits, Song Zhang, G. Loriot, and A. Callan-Jones. Visualization of Topological Defects in Nematic Liquid Crystals Using Streamtubes, Streamsurfaces and Ellipsoids. In *Visualization, 2004. IEEE*, pages 21–22, 2004.
- [Wagner and Schmalstieg, 2007] D. Wagner and D. Schmalstieg. Artoolkitplus for pose tracking on mobile devices. In *Proceedings of 12th Computer Vision Winter Workshop (CVWW'07)*, pages 139–146, 2007.
- [Wang *et al.*, 2003] C.C. Wang, C. Thorpe, and S. Thrun. On-line simultaneous localisation and mapping with detection and tracking of moving objects. In *IEEE International Conference on Robotics and Automation*, pages 2918–2924, 2003.