

Correcting Scale Drift by Object Recognition in Single Camera SLAM

Tom Botterill, Steven Mills, and Richard Green

Abstract—This paper proposes a novel solution to the problem of scale drift in single camera SLAM, based on recognising and measuring objects. When reconstructing the trajectory of a camera moving in an unknown environment, the scale of the environment, and equivalently the speed of the camera, is obtained by accumulating relative scale estimates over sequences of frames. This leads to scale drift: errors in scale accumulate over time. The proposed solution is to learn the classes of objects which appear throughout the environment, then to use measurements of the size of these objects to improve the scale estimate. A Bag-of-Words-based scheme to learn object classes, to recognise object instances, and to use these observations to correct scale drift is described, and is demonstrated reducing accumulated errors by 64% while navigating for 2.5km through a dynamic outdoor environment.

I. INTRODUCTION

SIMULTANEOUS Localisation and Mapping, or SLAM [1], enables a robot to position itself as it explores a previously unknown environment. The robot uses its sensors to map its environment, while simultaneously localising itself in this map. Errors which accumulate in the robot’s position estimate as it explores are corrected when previously mapped areas are re-visited (known as ‘loop closure’), allowing accurate long-term positioning.

One sensor often used for SLAM is a single camera: cameras are inexpensive, passive, compact, and non platform-specific, and many single-camera SLAM schemes have been demonstrated positioning robots on trajectories of hundreds of metres [2], [3], [4], [5]. A key strength of cameras is that the information-rich images they capture can be used both for measuring the robot’s incremental motion, by matching or tracking features between frames, and for localising the robot in the map, by recognising places which have been visited previously. On a larger scale however, errors still accumulate within large loops and in tracks without loop closure, and these errors can severely distort global maps.

A significant source of error unique to single camera SLAM is scale drift. A robot with a single camera can only resolve the scale of the world, and hence its speed, by identifying objects of known size, such as the calibration objects used to initialise some single camera SLAM schemes [6], [7], or previously mapped landmarks. As the robot explores away from previously mapped areas, small errors in the robot’s scale estimate accumulate, eventually rendering position estimates useless. Even when a loop is closed and the map is optimised,

substantial uncorrected errors in scale can remain, due to the trade-off between adjusting rotations, translations, and scales in order to reduce some global error function, therefore it is worthwhile to minimise this scale drift where possible [8].

This paper describes a new algorithm, SCORE2 (Scale Correction by Object Recognition), which addresses the problem of scale drift in a novel way: the robot first learns classes of objects which are present in its environment, then estimates the distribution of sizes of objects in each class. Later observations and measurements of these objects are used to correct scale drift. SCORE2 is based on earlier work on learning and measuring the objects present in a robot’s environment [9], although in this earlier work scale drift was not corrected successfully.

This paper is organised as follows: the following section describes previous uses of Object Recognition (OR) for robot positioning, and gives a brief overview of leading techniques for real-time OR. Section III describes the Bag-of-Words (BoW) algorithm and BoWSLAM single camera SLAM scheme with which SCORE2 is integrated. Section IV analyses the problem in detail, and describes the SCORE2 algorithm. Section V demonstrates SCORE2 successfully reducing scale drift in single camera SLAM, and the final sections discuss our conclusions and planned future developments.

II. BACKGROUND

In this paper we propose to integrate object measurements into a SLAM framework in order to correct scale drift. This work builds on a large body of prior research into the applications for Object Recognition (OR) for mobile robots, reviewed in Section II-B; and methods for OR, which are reviewed in Section II-C. Firstly, we examine alternative methods for reducing scale drift.

A. Scale drift correction in SLAM

Many Visual SLAM systems do not suffer scale drift because they incorporate measurements from other sensors. Sensors including IMUs (Inertial Measurement Units; [10], [11]), wheel encoders, depth cameras [12], or additional cameras, provide complementary measurements which can be used to estimate the true scale of the world. As the amount of extra information needed to resolve scale drift is small, even measurements of barometric pressure [13], or approximate depths inferred by depth-from-defocus [14] can be used to correct scale drift.

Alternatively, in some situations, properties of the environment can be used to correct scale drift. For a single camera attached to a wheeled vehicle, such as a car or

T. Botterill and R. Green are with the Department of Computer Science, University of Canterbury, Christchurch, NZ, e-mail: tom.botterill@canterbury.ac.nz.

S. Mills is with the Department of Computer Science, University of Otago, Dunedin, NZ.

bicycle, the distance of the camera above the ground plane is approximately constant, so the scale of the world can be measured by identifying points on the ground plane. This assumption is used to eliminate scale drift in the single camera SLAM schemes by Scaramuzza et al. [15] and Kitt et al. [16]. For SLAM using a helmet camera, the periodic motion of the camera, corresponding to the wearer’s strides, can be used to estimate the camera’s speed and hence correct scale drift [17]. In earlier work [18], the authors used an assumption that reconstructed 3D points in the world have depths from a particular distribution, which allows scale drift to be eliminated. The assumption improves the accuracy of maps, despite distortions where it does not hold. Such assumptions are not always true however, and alternative sensors are not always available. In this paper, we propose a scheme which can eliminate scale drift without making such assumptions, and without any additional sensors.

B. Object recognition for mobile robots

Autonomous mobile robots generally need the ability to recognise objects so that they can perceive and interact with their world. Often this is limited to identifying a particular object in order to perform a specific task, however many robots are equipped with more general purpose object recognition capabilities, and incorporate observations of objects into their internal world map [19], [20], [21], [22]. This section reviews some of these schemes, with focus on those which use OR to improve robot positioning or localisation.

Anati et al. [23] developed a robot which can localise itself by recognising objects as it explores a train station. A human marks objects (bins, clocks, ticket machines) on a simple map of the station. As the robot explores, it recognises these objects. A particle filter fuses object observations, and observation uncertainties, with position hypotheses. The correct position is resolved after multiple object observations.

A challenge facing many SLAM systems is that of data association. Data association is the process of matching measurements from the robot’s sensors to landmarks in a map, however this can be challenging when multiple landmarks have similar appearance. Incorrectly matching measurements to landmarks can corrupt the state of the robot’s map, and cause positioning to fail. Objects detected when multiple features are present in a scene make ideal landmarks for SLAM, as the feature combinations are more distinctive than features alone. OR is used to improve the performance of visual SLAM by Ahn et al. [24] and Castle et al. [25]. These schemes both recognise planar objects (from a database of posters and photographs) and use them as robust landmarks for stereo and monocular Extended Kalman Filter (EKF) based SLAM respectively. Castle et al. extended their scheme to incorporate the known size of particular objects into the SLAM solution [26]. Further work by Castle and Murray [27] incorporates similar planar objects into a modern PTAM-based SLAM scheme (Parallel Tracking and Mapping; where the scene and trajectory are reconstructed by bundle-adjustment-like optimisation over recent frames). Objects are localised accurately with respect to the camera, enabling Augmented Reality annotations to be added to the

objects. Civera et al. [28] also extend an EKF-SLAM scheme to incorporate 3D models of objects. The robot can download details of objects which are likely to occur in a particular environment as needed. When these objects are observed, they are reconstructed from multiple views, then are incorporated into the robot’s SLAM map.

Objects are also valuable as sources of additional information for more general 3D reconstruction tasks: Bao et al. [29] incorporate the structure of instances of known objects into a 3D reconstruction of the scene. Optimising object and camera poses jointly gives more accurate results than a reconstruction based on 3D points alone.

The SLAM system by Decrouez et al. [30] detects objects present in the environment in order to mitigate a different problem faced in Single Camera SLAM: the problem that mapped objects which later move can corrupt the internal SLAM map and cause positioning failure. Moving objects are detected by identifying groups of landmarks which have moved with respect to other landmarks in the SLAM map since they were first mapped. The SLAM map includes both static landmarks, and landmarks on objects which may move (e.g. on a book or mouse on a desk). Hsiao et al. [31] identify individual SLAM landmarks which move with respect to the world. These objects’ motion is modelled in an EKF-SLAM framework.

Alternatively, the positions of landmarks in the SLAM map can be used to help segment the robot’s environment into objects. Angeli and Davison [32] detect individual objects by clustering landmarks based on their location and appearance (so nearby and similar-looking landmarks are assumed to fall on the same object). The system partitions the SLAM map into a set of distinct objects, including books, a clock and a keyboard.

The systems reviewed in this section have demonstrated the many uses of OR for mobile robots, which include augmenting the capabilities of SLAM systems. Detected objects are shown to make effective SLAM landmarks, can be used to categorise the location being mapped, and single camera SLAM schemes which identify moving objects can avoid positioning failures in dynamic environments. A limitation of many of these previous schemes is that only a small number of known objects from a pre-defined database are recognised, so these schemes are only useful when the environment is likely to contain those particular objects. Ideally, SLAM schemes should enable robots to explore environments about which they have little or no prior knowledge. In this paper we propose that OR is used to address a different problem faced by single camera SLAM, that of scale drift. We also propose that the robot should be able to operate with minimal prior knowledge of its environment, and for this reason SCORE2 is designed to learn the classes of objects which are present in the environment being explored. Many proposed OR methods have the capability to learn the classes of objects present in the environment. The following section reviews contemporary OR methods, including those which have this required capability.

C. Real-time OR using computer vision, and the BoW model

With so many applications, OR has been an active area of computer vision research for decades, and a wide range of methods has been developed. Many of these methods are based on descriptors of local image features, such as SIFT (Scale Invariant Feature Transform; [33]) or SURF (Speeded-Up Robust Features; [34]), or descriptors of colour [20]. These descriptors allow the similarity in appearance of features in two images to be measured. For recognising objects from a small set of distinctive items, simple approaches based on matching a descriptor observed in an image with descriptors from training images are often sufficient [23], [19], [35].

For more general-purpose OR, for example to recognise instances of objects from a class, or to recognise objects against background clutter, these simple approaches are unsuitable, as single descriptors are not sufficiently distinctive. Objects are detected more reliably when each is represented by a set of features instead [28], [27], [24], however searching images for many different combinations of features can be computationally expensive. Approaches based on the Bag-of-Words (BoW) model address these limitations, and enable object instances from large databases to be identified effectively and efficiently [36], [37], [38].

The BoW model is an efficient representation of the set of feature descriptors an image contains, which works by quantising each descriptor to the most similar-looking from a set of ‘image words’. Image words are a set of representative descriptors chosen by clustering a training set of descriptors, with each cluster centre corresponding to one image word. When the BoW model is used for OR, an object, or object class, is represented by a set of co-occurring image words. Testing whether the BoW representation of an image contains a particular set of image words associated with an object is very fast, allowing large image databases to be searched for large numbers of objects. A strength of the BoW model is that the large numbers of image words used enables reliable object detection even in the presence of background clutter, and when occlusion and variations in object pose cause varying sets of features on the object to be detected.

BoW-based OR has often been deployed on mobile robots, for example by Ramisa et al. [39], who use a BoW model to recognise instances of various household objects observed while a robot explores; and Jebari et al. [40], who use SURF descriptors plus colour histograms in a BoW model to recognise multiple instances of objects in scenes observed by a robot.

BoW-based OR is also widely used for the related problem of identifying when two images show the same place, by identifying those with a large number of descriptors in common [41], [42], [43]. This makes BoW-based OR particularly well suited for autonomous mobile robots as many already maintain BoW databases in order to detect loop closure for SLAM [4], [44], [45], [46], or for localisation in a previously-mapped environment [47].

Most OR schemes use a supervised learning procedure to train a classifier on labelled training images of the objects to be recognised. Classifiers such as Support Vector Machines [48]

or Random Forests [21] can be applied to the BoW word frequency vectors to identify whether an object occurs in an image. These classifiers work because they identify the set of image words which co-occur when that object is present.

For some applications, for example when a robot is exploring an environment about which there is little prior knowledge, it is useful to be able to identify the objects which are present without prior training data. When images are represented with the BoW model, object classes can be learned automatically by finding sets of image words which co-occur, for example by finding principal components of a matrix of the BoW word frequency vectors. These approaches include Latent Semantic Analysis (LSA [49]), and its variants, Probabilistic LSA [50] and Latent Dirichlet Analysis [37], [51]. Similarly, groups of images showing the same object classes can be found by clustering the BoW word frequency vectors representing each image: Zhang et al. [52] apply an Expectation-Maximisation-based clustering framework to a automatically selected subset of features from the word frequency vectors. Each cluster corresponds to a different object class, and unlabelled query images are given the same annotations as training images in the same cluster. The feature combinations found by these approaches often correspond to different parts of an object, but also include features which tend to co-occur with the object, e.g. features associated with ‘cars’ might include the cars’ shadows on the road. For this reason, the phrase ‘BoW object’ is used in this paper for these sets of co-occurring image words.

Many other schemes for OR have been developed for the case when training data is available. Brown and Lowe [53] match the 3D structure of objects from a training set to images; Serre et al. [54] use descriptors of texture computed from training images; and Grauman and Darrell [55] learn spatial relationships between image features. Rothganger et al. use a hybrid approach to improve the accuracy of recognition [56]. Hoover et al. [57] decompose simulated images of objects into frequency components, then perform an approximate principal component analysis on these frequency components. Each object’s corresponding principal components form a concise description of that object, which can then be used for object classification. These approaches are often highly accurate but either are often not fast enough for real-time OR, or cannot identify objects amongst background clutter.

A recent innovation which has been successfully applied in OR competitions, such as the PASCAL Visual Object Class challenges [58], is to compute large descriptors describing edge orientations in images of object from each class. Edge orientations throughout each image are computed, and for each object class, a classifier (e.g. a Support Vector Machine) is trained on images of objects from that class. To detect objects, images are searched systematically for regions where the edge responses are classified as that object class. These descriptors include the Histogram of Oriented Gradient descriptor [59], and the Enhanced Biologically Inspired Model descriptor [60]. Of course OR is not limited to using 2D images alone—when available, other sensors such as depth cameras can also be used [61], [21].

In summary, for integrating OR with Single Camera SLAM,

BoW-based methods are ideal, as they can identify object class instances in single images, and in the presence of background clutter; they can be computationally efficient; they integrate easily with the BoW databases already maintained for loop closure detection; and they enable the objects present in to be learned automatically.

III. BOWSLAM AND THE BOW MODEL

This section provides an overview of BoWSLAM, the single camera SLAM scheme with which SCORE2 is integrated. SCORE2 is designed to integrate with BoWSLAM, however other single camera SLAM schemes also maintain BoW databases for loop closure detection, e.g. [4]. SCORE2 could easily be adapted to work with any such schemes.

BoWSLAM is designed to demonstrate that robust and large scale single camera SLAM is possible while making minimal assumptions about the motion of the camera or contents of the environment. Previous work by the authors [18], [62] describes the development of BoWSLAM in detail, and demonstrates that BoWSLAM can map long trajectories through visually challenging environments which contain many moving objects, featureless regions, and erratic motion. An important feature of BoWSLAM is that a high-level BoW representation is built for every frame, and these BoW representations are used for both loop closure detection, and for finding feature matches between pairs of frames, from which the camera’s incremental motion is computed.

In most BoW schemes, the dictionary of image words is built offline from training data, however SLAM is most useful in environments of which the robot has limited prior knowledge, in which case appropriate training data may be unavailable. Some BoW schemes address this problem by building dictionaries online from the descriptors which have actually been observed, hence choosing image words appropriate for describing whatever environment the robot is exploring; these include the schemes by Angeli et al. [63] and Eade and Drummond [4], and this is also the approach taken by BoWSLAM. In BoWSLAM, a hierarchical dictionary is built by recursively clustering small random subsets of the descriptors which have been observed, using k-medoids clustering, as described in [64]. Every time the number of descriptors increases by some fraction (e.g. 25%), a new dictionary is built in a separate thread, which takes typically a few seconds. This scheme is relatively simple, but is shown to have lower complexity than other schemes for building dictionaries online, and can scale to large environments [62, Chapter 5].

The descriptors used by BoWSLAM in this paper are simple 11×11 image patches centred on FAST corner features (Features from Accelerated Segment Test; [65]), which are compared by the sum-of-squared difference between pixel values. Other feature detectors and descriptors can be used, however this combination provides consistently good performance for both location recognition and for estimating relative poses.

BoWSLAM is based on the pose graph model of SLAM, where the robot’s trajectory is represented as a graph of relative pose estimates, and observed landmarks are reconstructed

relative to nodes in this graph. Relative pose estimates, and their uncertainty, are computed by 5-point Random Sample Consensus (RANSAC; [66]) and two-frame nonlinear refinement (two-frame bundle adjustment) [67], and relative scale is computed by a robustified least-squares alignment between sets of reconstructed 3D points.

BoWSLAM models relative scale estimates as lognormally-distributed random variables. A variable is lognormally distributed if its log is normally distributed (conventionally, natural logarithms are used). As well as being a good fit to observations, the lognormal distribution has the useful property that the product of lognormally distributed variables is lognormal, i.e. the relative scale estimate computed from a sequence of relative pose estimates is lognormal. Around a loop in the pose graph, relative scale estimates should accumulate to zero, and this constraint is used to correct scale drift when loop closure occurs. Within large loops, and when positioning away from previously mapped areas, it is errors in these relative scale estimates which accumulate and cause scale drift.

To compute a globally accurate map, a minimal spanning subgraph of relative poses which is unlikely to contain errors is first selected. Scale estimates are optimised around cycles in the subgraph, then the subgraph is refined using the TORO ‘Tree-based network Optimizer’ framework by Grisetti et al. [68]. TORO optimises the orientation and position estimates separately, and performs well at producing accurate maps, although a more modern refinement algorithm such as g2o (‘a General framework for Graph Optimisation’; [69]) or iSAM2 (‘incremental Smoothing and Mapping’; [70]), which jointly optimise rotations and translations (and scale estimates in the case of g2o), would improve BoWSLAM’s accuracy.

Other single camera SLAM and odometry schemes which model relative scale estimates use different approaches which could potentially be more accurate. Strasdat et al. [5] jointly optimise relative pose and scale estimates to reduce scale drift, however substantial (three-fold) scale drift still accumulates before a 200m loop is closed. Esteban et al. [71] propose an alternative least-squares method for estimating relative scales by aligning 2D image features to 3D structure. The method is employed in a Visual Odometry scheme which shows very low levels of scale drift (0.5%) around a small loop. Regardless of how accurate relative scale estimates are however, scale drift will inevitably still accumulate over longer tracks.

IV. SCALE CORRECTION BY OBJECT RECOGNITION

This section describes SCORE2, our new scheme to learn classes of BoW objects, to measure instances of these BoW objects, to estimate the distribution of sizes of each class of BoW objects, and to use later object size measurements to correct accumulated errors in scale. This section is organised as follows: firstly, Section IV-A analyses the problem of correcting scale drift from BoW object observations. Secondly, Section IV-B gives an overview of the proposed solution. Thirdly, Section IV-C describes how BoW objects are measured and how BoW object classes are learnt. Fourthly, Section IV-D describes how these uncertain object size measurements are

TABLE I
SELECTED NOTATION USED IN THIS PAPER. NATURAL LOGS (\log_e) ARE USED THROUGHOUT.

B	Number of BoW object classes
w_1, w_2	Two co-occurring words defining a BoW object
M	Term-document matrix, with elements m_{ij}
m_{ij}	The number of times word j occurred in image i , weighted by TF-IDF
λ_b	Measurement of BoW object b (distance between features defining b), assuming baseline 1
γ_b	Underlying measurement of BoW object b if scale was known
r_b, u_b	Scaled and logged measurement of a BoW object, with variance from uncertainty in scale
μ_b, σ_b	Parameters of the lognormal distn. of sizes of BoW objects in class b
$\{(r_{bi}, u_{bi}), i = 1, \dots, N\}$	N measurements of BoW object b
D_{SLAM}, G_{SLAM}^2	Parameters of a scale estimate from SLAM; $s \sim \text{Log-}\mathcal{N}(D_{SLAM}, G_{SLAM}^2)$
D_{OR}, G_{OR}^2	Parameters of a scale estimate from SCORE2; $s \sim \text{Log-}\mathcal{N}(D_{OR}, G_{OR}^2)$
$D_{combined}, G_{combined}^2$	Parameters of combined scale estimate; $s \sim \text{Log-}\mathcal{N}(D_{combined}, G_{combined}^2)$
$M_i, 1/t_i^2$	Parameters of conjugate distn. for μ ; $\mu \sim \mathcal{N}(M_i, 1/t_i^2)$ after i observations
α_i, β_i	Parameters of conjugate distn. for $\tau = 1/\sigma_b^2$; $\tau \sim \Gamma(\alpha_i, \beta_i)$ after i observations

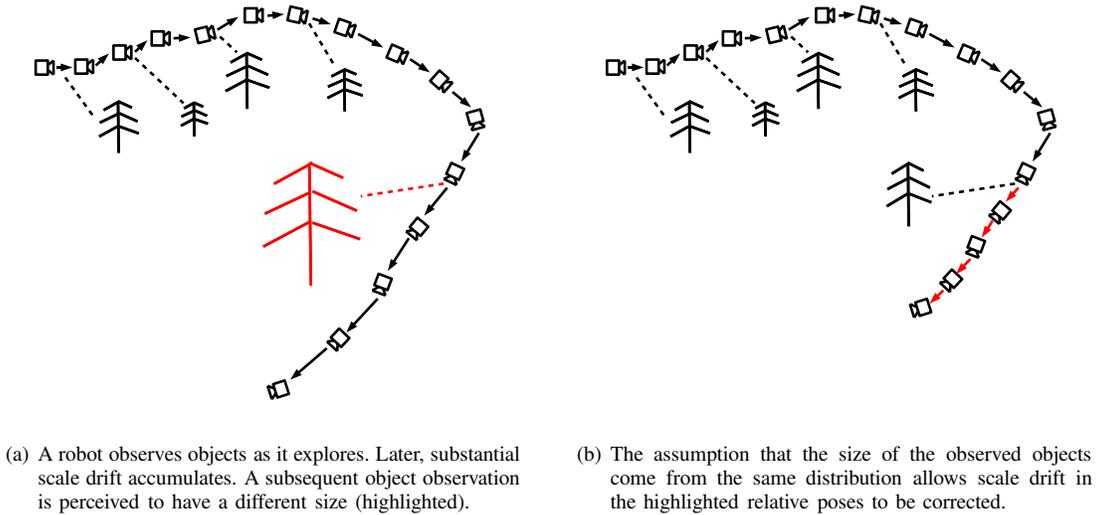


Fig. 1. Figure showing how SCORE2 corrects scale drift. (a) As the robot explores it observes and measures objects. Later, scale drift degrades its position estimate. (b) A subsequent observation of the object allows scale drift to be corrected.

used to parameterise a size distribution for each class. Finally, Section IV-E describes how measurements of BoW objects from these classes are used to correct accumulated errors in scale.

A. Analysis of scale correction problem

A robot identifies classes of objects as it explores an unknown environment. The distribution of sizes in each class is measured. As the robot travels into a previously unmapped area, its estimate of scale drifts, and as a result its position and speed estimates deteriorate. When the robot observes and measures objects belonging to classes observed earlier, it can use these measurements to improve its scale estimate. This idea is illustrated in Figure 1.

A BoW object is measured by reconstructing features on the object which are viewed from two frames, then measuring the distance between the features. The major source of error in a BoW object measurement is usually the error in the estimated baseline between the pair of frames (which is equivalent to the estimated speed of the robot). This source of error applies equally to all objects measured in the same two frames. In

addition, as scales are accumulated sequentially, errors in an estimated scale are correlated with errors in other scale estimates from which this scale was calculated (or will be used to calculate), and hence errors in all object measurements made along a trajectory are correlated (Figure 2).

As with the SLAM problem, where correlations between landmark positions and the robot pose should be modelled [1], [8], modeling the correlations between object measurements would be useful for estimating object class size distributions. Maintaining these correlations would be challenging however. Unlike SLAM measurements, which can be partitioned into local submaps, the most suitable object classes for scale drift reduction are those that are encountered throughout the robot's environment. Currently measurements are assumed independent, to simplify the problem. This approximation is most appropriate when scale drift is low, which is when the most reliable measurements of objects (those contributing most to their estimated size distribution) are made.

B. Overview of solution

The SCORE2 (Scale by Object Recognition) algorithm is outlined in Figure 3, and is detailed in the following

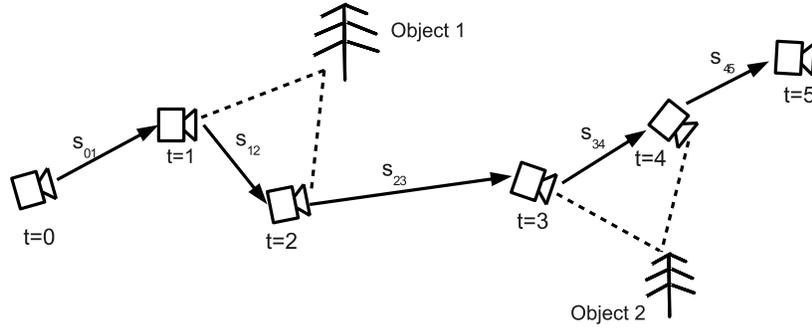


Fig. 2. An object observed at times $t = i$ and $t = j$ is measured by reconstructing features on it, then measuring the distances between these features. The size of this measurement is proportional to the baseline length, s_{ij} . As baseline length estimates are correlated, different object measurements are also correlated.

For each frame:

When re-training:

- 1) Identify (learn) the B BoW object classes
- 2) Measure each BoW object (subject to scale estimate from SLAM)
- 3) Estimate distribution parameters
- 4) For each edge combine the most likely scale given the observations with the measured scale

When a new edge (relative pose estimate) is added:

- 1) Observe any BoW objects reconstructed here
- 2) Update BoW object size distributions
- 3) Combine the scale estimated from the observations with the measured scale

Fig. 3. Overview of the SCORE2 algorithm

sections. In summary, when retraining occurs (when a new BoW dictionary is created), a set of BoW object classes is identified. The measured sizes of objects in these classes are used to improve existing and new scale estimates.

The effect of SCORE2 is to propagate reliable scale estimates from better mapped areas to areas where the scale is much less certain, but where the same BoW objects are visible. In practice there is often a large difference between the uncertainty in scale estimates in different areas, for example scale estimates are accurate when the robot is moving in a straight line in a feature-rich environment, then deteriorate when the robot corners. SCORE2 is also able to initialise a scale estimate when a new map component is started, after the robot has become lost.

C. Classes of measurable objects

This section describes how SCORE2 defines a class of objects, and how objects in that class are measured. Notation used is summarised in Table I.

SCORE2 requires that BoW object classes can be learnt and identified in real-time, and that identified BoW objects can be measured. The object classes recognised in real-time by contemporary OR schemes (Section II-C) consist of occurrences of one or more of a set of features defining that class. The most obvious measure of an object in one of these classes is the distance between two features on the object. Measurements of more than two features are not considered, as this would add to the complexity (there are $\binom{n}{2}$ possible measurable

distances between n points), and would introduce difficulties in coping with partially-observed and partially-reconstructed BoW objects. As a result, SCORE2's BoW objects are each defined by the co-occurrence of two image words. Multiple instances of the same BoW object are likely to be visible in many scenes; however by assuming BoW objects are separated by more than the separation of the features within them, only the least separation between all possible pairs of two features visible in a scene must be measured.

To identify co-occurring image words, the same term-document matrix as LSA is used. This sparse matrix, M , has elements m_{ij} representing the number of times word j occurred in image i . Columns are weighted by the Term Frequency-Inverse Document Frequency (TF-IDF) score [41]; a heuristic measure of distinctiveness. Each row of this matrix, \mathbf{r}_i , is the BoW representation of image i .

Contemporary LSA systems find co-occurring features by computing the principal components of the matrix M by Singular Value Decomposition (SVD; [72]). M could realistically have 10,000 rows, 50,000 columns, and 5 million non-zero entries however, making the computation challenging for a real-time application. In addition, only co-occurring pairs of features are of interest, rather than co-occurring sets of features.

The principal component \mathbf{p} of M is a unit vector maximising $\|M'\mathbf{p}\|$, where M' is given by subtracting the column's mean from each column of M . As co-occurring pairs of features are required, vectors \mathbf{p}_2 with two equal nonzero elements are found, which each correspond to one co-occurrence. For

each co-occurrence of two words w_1 and w_2 , we calculate the sum:

$$\sum_{\text{Images } i \text{ containing } w_1, w_2} m_{i w_1}^2 + m_{i w_2}^2 \quad (1)$$

If co-occurring words only occurred in these pairs, then the pair of words maximising $\|M' \mathbf{p}_2\|$ would be the pair of words for which this sum is greatest. The B pairs of words with the highest value for Equation 1 are chosen as the B BoW object classes. Only features that are successfully reconstructed are used, which limits the number of co-occurrences that must be considered, and avoids learning BoW objects which cannot often be measured.

Equation 1 is a heuristic, as is LSA itself, however the experiments in Section V show it to work reasonably well at identifying pairs of features corresponding to real objects. Many alternative schemes for choosing the word pairs could be used, for example choosing the pairs of features with the highest mutual information. The mutual information of two words is a measure of the amount of information provided about the occurrence or non-occurrence of one word, given that the other has been observed, and the co-occurrences with the highest mutual information are selected by [73] in order to compute the probability that pairs of images show the same scene. Computing mutual information in this scheme requires heuristic estimates of word co-occurrence probability however.

Once a set of BoW object classes has been identified, the instances of the corresponding BoW objects are identified (by searching each of the sets of 3D points reconstructed between pairs of frames) and measured.

D. Estimating object class size distribution parameters

This section describes how a distribution is fitted to the noisy measurements of BoW object sizes. The distribution incorporates both measurement error, and variability in BoW object sizes within each class. There are five sources of variability in the observed object sizes:

- 1) Uncertainty in the baseline length (scale) from which objects are reconstructed.
- 2) Variation in true size of objects (e.g. cars are 1.5 to 2.5m high).
- 3) The same two-word combination occurring in multiple contexts.
- 4) Errors in reconstructing 3D point positions.
- 5) Errors from measurements of multiple partially-visible objects, or features occurring in multiple objects.

The combination of these sources of variability is assumed lognormal, as this fits observations well. The analysis in [62, Chapter 9], shows that these size measurements are considerably better approximated by a lognormal distribution than a normal distribution. The lognormal assumption makes integration with the lognormally-distributed scales estimated by BoWSLAM simple, and is underpinned by evidence that the lognormal distribution is very often a good model for size measurements including the heights of plants or people, or the length of words [74]. BoW object size measurements are also assumed to be independent.

A BoW object b is observed in two frames, and is measured to have size λ_b when points are reconstructed with baseline 1. The object's true size, γ_b , is an unknown random variable drawn from the underlying BoW object class size distribution $\text{Log-}\mathcal{N}(\mu_b, \sigma_b^2)$, which includes variation in object sizes. For the true baseline length s , $\gamma_b = s\lambda_b$. From SLAM, s is assumed to be lognormally distributed, with $s \sim \text{Log-}\mathcal{N}(D_{SLAM}, G_{SLAM}^2)$, therefore the logged BoW object size measurement $r_b = \log(\lambda_b) + D_{SLAM}$ is normally distributed about $\log \gamma_b$ with variance $u_b^2 = G_{SLAM}^2$.

From measurements of multiple BoW objects from a class, a Bayesian approach is used to estimate the parameters of the BoW object class size distribution, μ and σ^2 (dropping the subscript bs for clarity). Each time a measurement λ_i of a BoW object instance is made, the corresponding logged measurement r_i and its variance u_i^2 are used to update the estimate of the BoW object class size distribution parameters, μ , σ^2 . Appropriate conjugate prior distributions for each parameter are the normal distribution for the mean size of BoW objects, μ , and the gamma distribution for the precision (inverse of the variance) of the BoW object size, $\tau = \frac{1}{\sigma^2}$ [75]. When i measurements have been made:

$$\mu \sim \mathcal{N}(M_i, 1/t_i^2) \quad (2)$$

$$\tau \sim \Gamma(\alpha_i, \beta_i) \quad (3)$$

and parameters of these conjugate distributions are given by:

$$M_i = \frac{t_{i-1}M_{i-1} + \frac{r_i}{u_i^2}}{t_i + \frac{1}{u_i^2}} \quad (4)$$

$$t_i = t_{i-1} + \frac{1}{u_i^2} \quad (5)$$

$$\alpha_i = \alpha_{i-1} + \frac{1}{2} \quad (6)$$

$$\beta_i = \beta_{i-1} + \frac{1}{2}(M_i - r_i)^2 \quad (7)$$

Initially, uninformative parameters $M_0 = 0, t_0 = \epsilon, \alpha_0 = 1, \beta_0 = \epsilon$, for some small $\epsilon > 0$, are used. After i observations, the parameters for the distribution of BoW object sizes are taken to be the mean of these distributions:

$$\mu = M_i, \quad \sigma^2 = \frac{\alpha_i}{\beta_i}. \quad (8)$$

σ^2 is initially very large, but typically after four or five observations is low enough that subsequent BoW object observations can substantially affect scale estimates. By Equation 4, the most accurate measurements (those where u_i^2 is low) have the greatest effect on the estimated parameters.

E. Object observation and scale updates

When the BoW dictionary is re-created, every scale estimate (between pairs of frames) is updated with information from measurements of BoW objects reconstructed between the two frames. The same method is used to update new scale estimates as new relative poses are added to the map.

The relative pose of two cameras has a baseline length (scale) estimate modeled by a lognormal distribution with

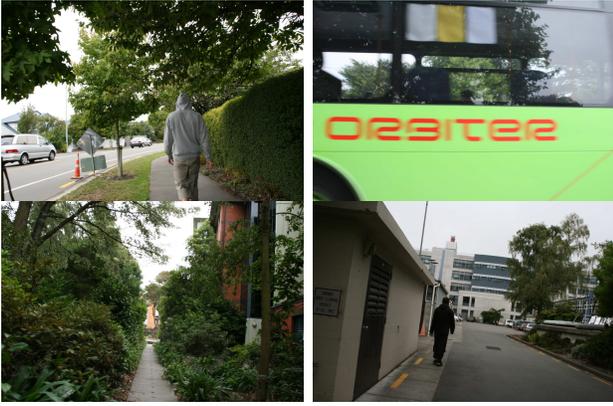


Fig. 4. Frames from the Suburban dataset, captured at 2.5Hz using a handheld Canon 400D SLR camera. The trajectory is 2.5km long, takes 25 minutes, and includes multiple loops. Dozens of pedestrians and cars pass by, and one sequence of three frames is completely occluded by a passing bus.

parameters D_{SLAM} and G_{SLAM}^2 . A set of M BoW objects, with size distributions parametrised by $\{(\mu_j, \sigma_j), j = 1, \dots, M\}$, is observed in a frame, with logged measurements $\{\log \lambda_j\}$. The scale of the edge $s \sim \text{Log-}\mathcal{N}(D, G^2)$, therefore $s\lambda_j$ is a random variable from $\text{Log-}\mathcal{N}(\mu_j, \sigma_j)$, and hence $\log s \sim \mathcal{N}(\mu_j - \log \lambda_j, \sigma_j)$. The maximum likelihood estimate of the scale of these BoW object observations is given by differentiating the log of this likelihood function and setting equal to zero:

$$D_{OR} = \sum_{j=1}^M \frac{\mu_j - \log \lambda_j}{\sigma_j^2} \quad (9)$$

$$G_{OR}^2 = \frac{1}{\sum_{j=1}^M \frac{1}{\sigma_j^2}} \quad (10)$$

Combining these parameters with the parameters from SLAM gives:

$$D_{combined} = \left(\frac{D_{OR}}{G_{OR}^2} + \frac{D_{SLAM}}{G_{SLAM}^2} \right) / \left(\frac{1}{G_{OR}^2} + \frac{1}{G_{SLAM}^2} \right) \quad (11)$$

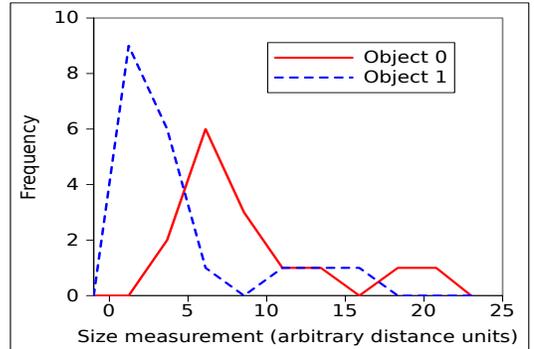
where $D_{combined}$ is the scale estimate combining the scale estimate from SLAM, and the scale estimate from observing the BoW objects.

In summary, scale parameter estimates from SLAM alone are used to estimate the size distributions of BoW objects. These size distributions are used to compute the most likely scale parameter for each edge, given the BoW objects that have been observed. Observations of BoW objects only substantially affect scale estimates when scale estimates from SLAM have high uncertainty compared to the scale estimates when object sizes were measured.

The computational cost of SCORE2 is dominated by identifying the B best BoW object classes following retraining. This takes time $O(BT)$ after time T , as a fixed number of feature co-occurrences occur in each image. When processing the NZi3 dataset (described in Section V), 4% of the total computational cost is related to SCORE2. As BoWSLAM has $O(T \log T)$ complexity per frame, SCORE2 does not add significantly to the total cost.



(a) Distances measured of two BoW object classes, ‘round shadows’, and ‘car wheels’.



(b) Distributions of the sizes of these BoW object classes (15 and 19 measurements are made respectively).

Fig. 5. Two BoW object classes measured in an outdoor dataset have significantly different size distributions. The lognormal distribution is a good model, as it is unimodal, heavy-tailed, and non-negative.

TABLE II
RMS ERRORS IN GLOBAL MAPS OPTIMISED BY TORO.

Motion model	RMS error	Error relative to distance travelled
No motion model	198m	7.9%
SCORE2	71m	2.8%
Constrained acceleration	83m	3.3%
Constrained depths	56m	2.2%

V. RESULTS

This section describes results obtained from running BoWSLAM, with SCORE2, on outdoor and indoor datasets. First SCORE2 is tested on the the 2.5km, 3662 frame, outdoor ‘Suburban’ dataset, as shown in Figure 4. Maps of the trajectories reconstructed using SCORE2, and using alternative motion models, are shown in Figure 6. About 2000 BoW objects (pairs of co-occurring features) are identified and measured; of these, 455 have distributions with $\sigma_b < 1$ (while all BoW object measurements are used, higher values are too uncertain to have a significant effect on scale estimates). The 455 BoW objects are each measured between 5 and 31 times (typically one or two per frame are measured); the BoW object with the tightest distribution has $\sigma_b = 0.47$, whereas the least accurate scales from SLAM have distributions with $G_{SLAM}^2 \approx 1$. Examples of BoW objects found in a similar outdoor environment are shown in Figure 5.

Without using SCORE2, RMS errors of 198m remain in BoWSLAM’s optimised map (Table II). SCORE2 reduces the RMS errors to 71m; a 64% reduction in error.

In [18], two motion models (assumptions about the camera’s motion relative to its environment) were proposed in order to reduce scale drift. The first, “Scale MM”, makes the assumption that the mean depth of reconstructed points is lognormally distributed. The most likely camera speed and scale, given the depth of reconstructed points and the scale estimate from SLAM, is used. The second motion model, “Acceleration MM” is roughly equivalent to a constant velocity motion model, and makes the assumption that the relative acceleration between frames is lognormally distributed. Relative scale estimates from SLAM are combined with the estimate from the motion model. Both motion models reduce scale drift; Scale MM reduces RMS errors to 56m in the suburban dataset (although variation of tens of metres between different runs and parametrisations are observed).

Two indoor datasets are also used to evaluate SCORE2. The first dataset is captured at 2.8Hz with a Canon 400D SLR camera in the NZi3 office building at the University of Canterbury. Examples of frames from this dataset, and BoW objects detected, are shown in Figure 7. The dataset consists of a 20m straight line, two sharp right-angle corners, then another 20m straight line. Scale drift often accumulates when cornering rapidly, and is measured by comparing the estimated lengths of the two 20m sections: the longer length, as a fraction of the shorter length, provides a measure of scale drift (Figure 8).

On this dataset, SCORE2 is compared to BOWSLAM alone, and the two motion models proposed in [18] (outlined in Figure 9). As levels of scale drift vary between runs (due to random BoW clustering), thirty runs are made in each case. Figure 10 shows the distribution of scale drift observed for each motion model. SCORE2 substantially outperforms a motion model constraining acceleration between frames (“Acceleration MM”), and is close to matching the performance of the model constraining point depths (“Scale MM”). SCORE2 reduces scale drift by an average of 75% compared with BOWSLAM without a motion model.

It is possible that the objects chosen by SCORE2 are irrelevant; i.e. constraining any measurements of the world would be equally effective. To verify that this is not the case, SCORE2 is run where each object measurement is replaced with a measurement of a random pair of reconstructed points. This scheme (“Random measurements”) only reduces scale drift by a small amount; considerably less than the reduction when using SCORE2. This verifies that SCORE2 works because BoW object instances are being measured, rather than simply that the scale of the world is constrained. Secondly, we verify that BoW objects detected do have significantly different size distributions. Figure 11 shows the distribution of the measured sizes of the first five BoW object classes detected in the NZi3 dataset (out of 19 in total). Several BoW object classes detected have significantly different distributions of sizes. Note also that the range of measured object sizes is large (with often a 10-fold difference in size between the smallest and largest measurement); an effect of this is that each observation makes only a small difference to the estimated scale. The correction to the scale is due to the accumulated effect of observing many objects in many frames, and a single outlier observation is

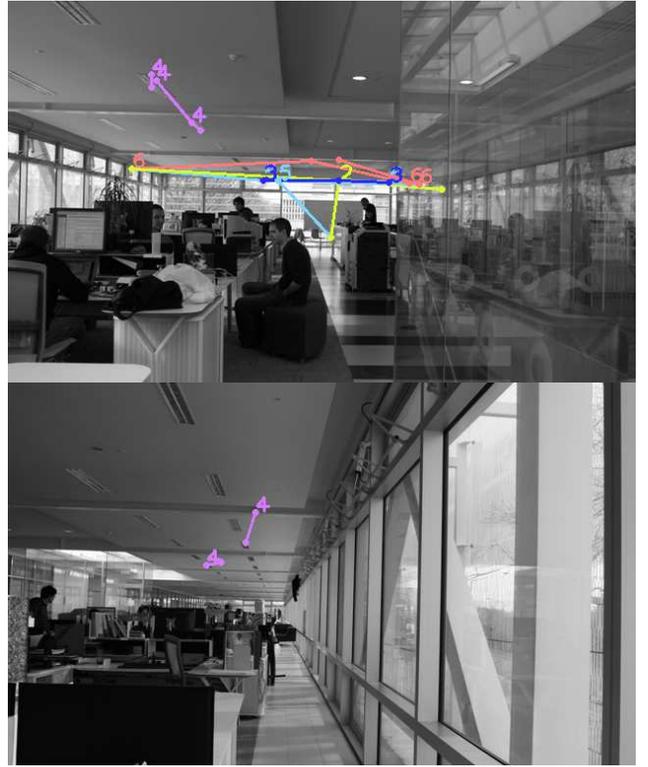


Fig. 7. Frames from the NZi3 dataset. This dataset is captured in a modern office environment, including many repeated features, large amounts of reflective glass, and people. The distances between ceiling lights and sprinklers (BoW object 4), and the distances between ceiling beams and top windows (object 6) are measured throughout. Some other detected BoW objects do not appear to correspond to any particular objects; measurements of these may have a similar effect to constraining the depth of reconstructed points.

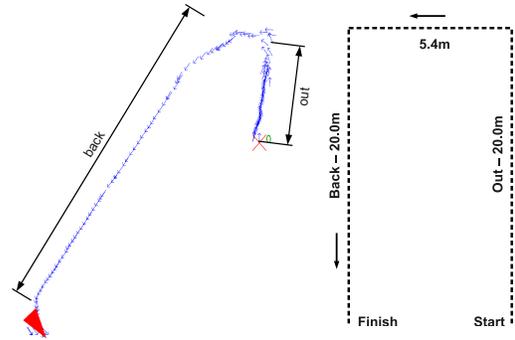


Fig. 8. Map of robot poses, NZi3 dataset, and ground truth (measured) trajectory. Significant scale drift (and also errors in orientation) accumulate when cornering rapidly. Scale drift is measured as the ratio of the estimated lengths of two sections of the trajectory, which have equal length (marked ‘out’ and ‘back’).

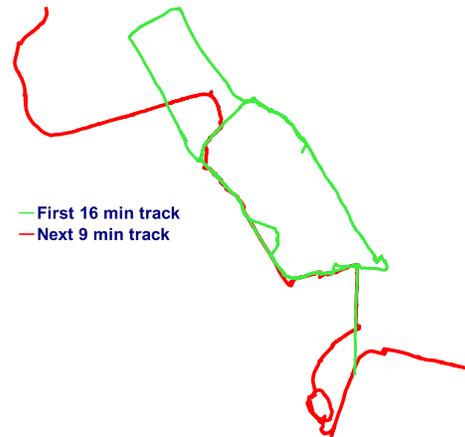
Suburban dataset
No motion model



(a) Global map, no motion model. Scale drift accumulates at either end of the second trajectory, and within loops.

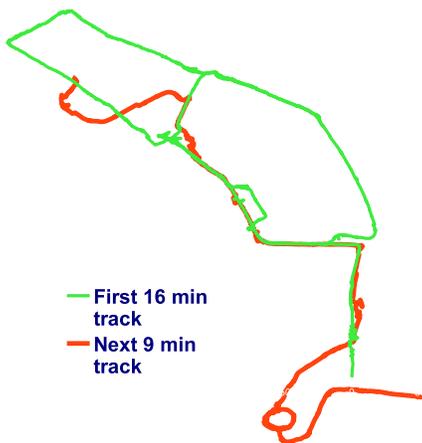
Suburban dataset

Scale-drift reduced using Object Recognition



(b) SCORE2: scale drift is reduced, and accuracy is comparable with an alternative motion model constraining the scale of the world.

Suburban Dataset
MM constraining the scale of the world



(c) Motion model constraining allowed scale of the world. Scale drift is reduced, although the map is distorted where the true scale of the environment does not reflect the depth distribution assumed by the motion model.



(d) Ground truth data from GPS

Fig. 6. Maps of robot poses compared with ground truth from GPS, from the Suburban dataset (Figure 4). The path starts at **S** and traverses the large loop twice, with various diversions. After 16 minutes the camera is stopped, then restarted at **R**. Later the original path is re-joined. Large errors in scale and orientation occur at the ends of the trajectory, where loop-closure is not detected.

SCORE2 BoWSLAM run with SCORE2 to recognise BoW objects and correct scale drift.

Scale MM Reconstructed 3D points are assumed to have depths from a lognormal distribution (as described in [18]).

Acceleration MM Constant velocity motion model; acceleration is assumed to have a lognormal distribution with zero median (as described in [18]).

Random measurements SCORE2 modified to make a measurement between two random points, rather than points on a BoW object. Verifies that SCORE2 works because measurements of instances from object classes are made, rather than because it constrains the scale of the world.

No MM or OR BoWSLAM run without SCORE2 or any motion model.

Fig. 9. Motion models used for comparison with SCORE2.

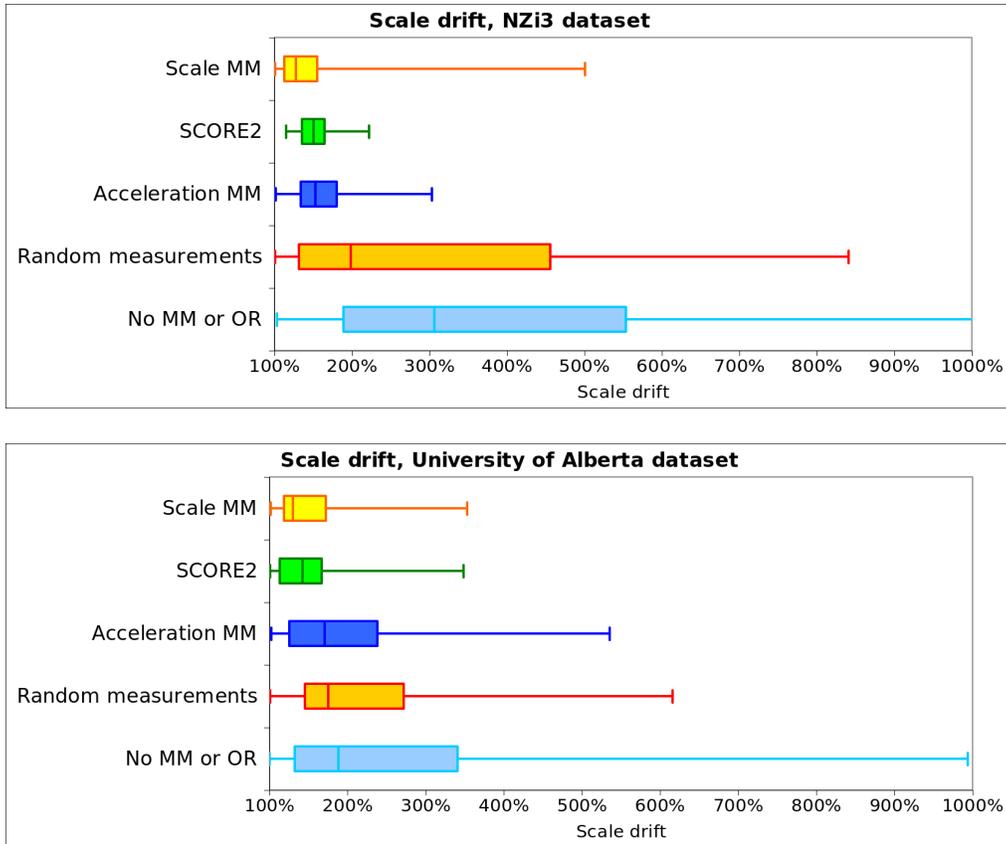


Fig. 10. Box plots showing the distribution of scale estimates from 30 runs for each model on each dataset. The range, quantiles and median scale drift is shown. A score of 100% would indicate no scale drift. SCORE2 greatly reduces scale drift in both the NZi3 dataset and the University of Alberta dataset. SCORE2 outperforms a motion model constraining acceleration, and matches the performance of motion model constraining the scale of the world.

unlikely to substantially degrade the scale estimate.

A similar experiment is conducted with the University of Alberta dataset. This dataset is captured from a wheeled robot following a rectangular loop along corridors. Frames from this dataset, and examples of BoW objects detected, are shown in Figure 12. Again SCORE2 is compared to other motion models (Figure 10), and again, SCORE2 substantially reduces scale drift, and matches the performance of a constraint on the depths of observed points. Frames are rearranged in the test dataset, so that loop closure never occurs, however in the original dataset, loop closure substantially reduces scale drift, which can be partially corrected around the loop.

VI. CONCLUSIONS

This paper has demonstrated a novel solution to the problem of scale drift in single camera SLAM. Bag-of-Words based Object Recognition is used to learn sets of co-occurring features which correspond to object classes, and to recognise instances of the BoW objects from these classes. The distribution of BoW object sizes in each class is estimated, and this information is used to correct scale drift. Scale drift can be corrected even over long tracks where loop closure does not occur, and results in only a small increase in the computational cost.

SCORE2 reduces scale drift by 75% in one indoor dataset,

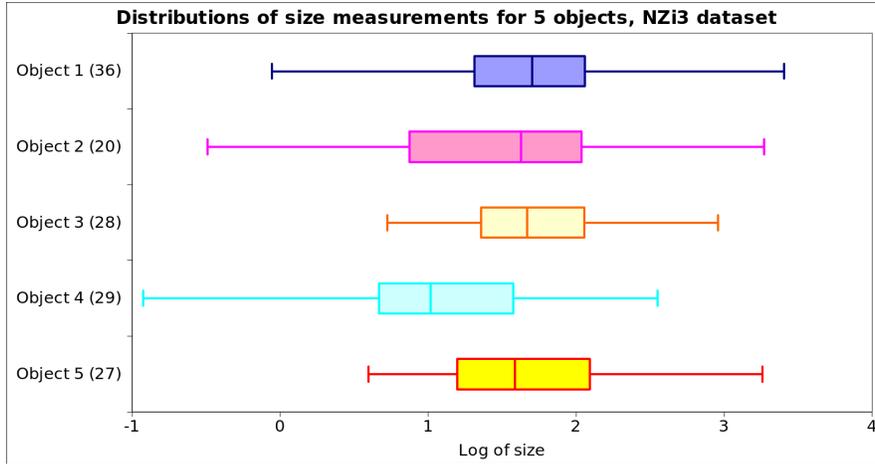


Fig. 11. Distributions of sizes of the first five of the 19 BoW object classes detected in the NZi3 dataset. Some have significantly different size distributions, indicating that the assumption that co-occurring features correspond to object classes with distinct size distributions is valid. The numbers in brackets indicate the number of measurements of each object.



Fig. 12. BoW objects from the same class observed on several similar-looking doors, University of Alberta dataset. (Each BoW object is measured in one place per pair of frames, however in BoWSLAM each frame is registered to many others, so the same type of object may be measured in several places in the same frame.)

and reduces the total accumulated error by 64% in a large outdoor dataset. SCORE2 has similar performance to a heuristic model of the scale of the world at reducing scale drift.

Experiments on indoor and outdoor datasets demonstrate that BoW object classes with a variety of different size distributions are found, and SCORE2 is verified to use the different sizes of these different objects to correct scale drift.

VII. FUTURE WORK

There are many small improvements that would improve the performance of SCORE2, in particular more sophisticated methods for choosing and robustly measuring more complex objects. In this section some planned improvements and extensions are detailed.

SCORE2 currently assumes that measurements of objects are independent, however this is not true in general, as errors in absolute scale estimates are correlated with each other. While SCORE2 works well despite this assumption, the assumption could be avoided if distributions of relative object sizes were maintained instead. An object measured in two locations would then introduce a constraint to the pose graph between scale estimates in the two different locations. The constraint on scale would then be applied when the pose graph was optimised, although at the expense of increasing the cost of the pose graph optimisation, as different sections of the map would no longer partition so easily. A modern optimisation framework such as g2o [69] could incorporate constraints of this form.

The main limitation of SCORE2 results from the requirement that BoWSLAM should operate with minimal prior knowledge of the environment to be explored, however for many practical applications, prior knowledge about the environment and the objects it contains is available. In this case, both the BoW dictionary, and classes of objects which are likely to be encountered, could be learned in advance from training data. Examples of object classes might include cars,

bicycles, people, and household objects; these are amongst the 20 categories which are recognised by schemes competing in the PASCAL Visual Object Class challenges [58]; many of these schemes are BoW based and some achieve recognition rates of around 60% of object occurrences, in real-time [36]. Observations of these objects would provide absolute scale estimates wherever they were observed.

BoWSLAM uses each image's BoW representation for both recognising locations, and for feature matching for relative pose estimation. Simple image patch descriptors centred on FAST corners perform well for these tasks, however for OR, when objects should be recognised regardless of their pose, scale, and location in the image, descriptors which are more invariant to changes in scale, orientation and lighting levels, such as SIFT or SURF, are often used. While BoW objects are still recognised at a range of scales when using FAST corners (as seen in Figures 11 and 12), OR performance could potentially be improved by using a different descriptor and detector combination.

This paper has demonstrated that BoW-based object recognition can be used to correct scale drift in single camera SLAM, however there are many other potential applications of OR in visual navigation, and the authors believe that SLAM schemes building maps based on high-level objects, rather than low-level features, will enable many of the difficulties still faced by SLAM schemes to be addressed.

REFERENCES

- [1] R. Smith, M. Self, and P. Cheeseman, *Autonomous Robot Vehicles*. Amsterdam: Springer Verlag, 1990, ch. Estimating Uncertain Spatial Relationships in Robotics, pp. 435–461.
- [2] L. A. Clemente, A. J. Davison, I. Reid, J. Neira, and J. D. Tardos, "Mapping large loops with a single hand-held camera," in *Proc. Robotics: Science and Systems*, 2007.
- [3] T. Lemaire and S. Lacroix, "SLAM with panoramic vision," *Journal of Field Robotics*, vol. 24, pp. 91 – 111, 2007.
- [4] E. Eade and T. Drummond, "Unified loop closing and recovery for real time monocular SLAM," in *Proc. British Machine Vision Conference*, 2008.
- [5] H. Strasdat, J. M. M. Montiel, and A. J. Davison, "Scale drift-aware large scale monocular SLAM," in *Proc. Robotics: Science and Systems Conference*, 2010.
- [6] A. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *Proc. Int. Conf. Comput. Vis.*, Oct. 2003.
- [7] E. Eade and T. Drummond, "Scalable monocular SLAM," in *Proc. Int. Conf. Comput. Vis. and Pattern Recognition*, Los Alamitos, CA, 2006, pp. 469–476.
- [8] H. Durrant-Whyte and T. Bailey, "Simultaneous localisation and mapping (SLAM): Part I the essential algorithms." *IEEE Robotics and Automation Magazine*, vol. June, pp. 1–9, 2006.
- [9] T. Botterill, R. Green, and S. Mills, "A Bag-of-Words Speedometer for Single Camera SLAM," in *Proc. Image and Vision Computing New Zealand*, Wellington, NZ, November 2009, pp. 1–6.
- [10] C. Hide, T. Botterill, and M. Andreotti, "Vision-aided IMU for handheld pedestrian navigation," in *Proc. Institute of Navigation GNSS Conference*, Fort Worth, Texas, 2010, pp. 1–9.
- [11] G. Nützi, S. Weiss, D. Scaramuzza, and R. Siegwart, "Fusion of IMU and vision for absolute scale estimation in monocular slam," *Journal of Intelligent & Robotic Systems*, vol. 61, no. 1, pp. 287–299, 2011.
- [12] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard, "An evaluation of the RGB-D SLAM system," in *Proc. IEEE Int. Conf. Robot. Autom.*, St. Paul, MA, USA, May 2012.
- [13] M. Achtelik, M. Achtelik, S. Weiss, and R. Siegwart, "Onboard IMU and monocular vision based control for MAVs in unknown in- and outdoor environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 3056–3063.
- [14] K. Č. Pucihar and P. Coulton, "Estimating scale using depth from focus for mobile augmented reality," in *Proc. 3rd ACM SIGCHI symposium on Engineering interactive computing systems*, 2011, pp. 253–258.
- [15] D. Scaramuzza, F. Fraundorfer, M. Pollefeys, and R. Siegwart, "Absolute scale in structure from motion from a single vehicle mounted camera by exploiting nonholonomic constraints," in *Proc. Int. Conf. Comput. Vis.*, 2009.
- [16] B. Kitt, J. Rehder, A. Chambers, M. Schönbein, H. Lategahn, and S. Singh, "Monocular visual odometry using a planar road model to solve scale ambiguity," in *Proc. European Conference on Mobile Robots*, 2011.
- [17] D. Gutiérrez-Gómez, L. Puig, and J. Guerrero, "Full scaled 3d visual odometry from a single wearable omnidirectional camera," Technical report, University of Zaragoza. 4, 5, Tech. Rep., 2012.
- [18] T. Botterill, S. Mills, and R. Green, "Bag-of-words-driven single camera simultaneous localisation and mapping," *Journal of Field Robotics*, vol. 28, pp. 204–226, 2011.
- [19] S. Vasudevan, S. Gächter, V. Nguyen, and R. Siegwart, "Cognitive maps for mobile robotsan object based approach," *Robotics and Autonomous Systems*, vol. 55, no. 5, pp. 359–371, 2007.
- [20] P. Jensfelt, S. Ekvall, D. Kragic, and D. Aarno, "Integrating SLAM and object detection for service robot tasks," in *Proc. IROS Workshop on Mobile Manipulators: Basic Techniques, New Trends and Applications*, Edmonton, Canada, 2005.
- [21] J. Stückler, N. Biresev, and S. Behnke, "Semantic mapping using object-class segmentation of RGB-D images," in *Proc. Int. Conf. Intell. Robots Syst.*, 2012.
- [22] M. Beetz, M. Saito, H. Azuma, K. Okada, and M. Inaba, "Searching objects in large-scale indoor environments: A decision-theoretic approach," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2012, pp. 4385–4390.
- [23] R. Anati, D. Scaramuzza, K. Derpanis, and K. Daniilidis, "Robot localization using soft object detection," in *Proc. IEEE Int. Conf. Robot. Autom.*, may 2012, pp. 4992–4999.
- [24] S. Ahn, M. Choi, J. Choi, and W. K. Chung, "Data association using visual object recognition for EKF-SLAM in home environment," in *Proc. Int. Conf. Intell. Robots Syst.*, Oct. 2006, pp. 2588–2594.
- [25] R. Castle, D. J. Gawley, G. Klein, and D. Murray, "Video-rate recognition and localization for wearable cameras," in *Proc. British Machine Vision Conference*, 2007, pp. 1100–1109.
- [26] R. Castle, G. Klein, and D. Murray, "Combining monoSLAM with object recognition for scene augmentation using a wearable camera," *Image and Vision Computing*, vol. 28, no. 11, pp. 1548 – 1556, 2010.
- [27] R. Castle and D. Murray, "Keyframe-based recognition and localization during video-rate parallel tracking and mapping," *Image and Vision Computing*, vol. 29, no. 8, p. 524, 2011.
- [28] J. Civera, D. Gálvez-López, L. Riazuelo, J. Tardós, and J. Montiel, "Towards semantic SLAM using a monocular camera," in *Proc. Int. Conf. Intell. Robots Syst. IEEE*, 2011, pp. 1277–1284.
- [29] S. Bao, M. Bagra, and S. Savarese, "Semantic structure from motion with object and point interactions," in *ICCV Computer Vision Workshop*. IEEE, 2011, pp. 982–989.
- [30] M. Decrouez, R. Dupont, F. Gaspard, F. Devernay, J. Crowley *et al.*, "Modélisation explicite des objets et de l'environnement en combinant les approches topologique et métrique pour la localisation," 2011.
- [31] C. Hsiao and C. Wang, "Achieving undelayed initialization in monocular slam with generalized objects using velocity estimate-based classification," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 4060–4066.
- [32] A. Angeli and A. Davison, "Live feature clustering in video using appearance and 3d geometry," in *Proc. British Machine Vision Conference*, 2010, pp. 41–51.
- [33] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. Int. Conf. Comput. Vis.*, 1999, pp. 1150–1157.
- [34] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, pp. 346–359, 2008.
- [35] Y. Yu, G. Mann, and R. Gosine, "An object-based visual attention model for robotic applications," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 40, no. 5, pp. 1398–1412, 2010.
- [36] J. Uijlings, A. Smeulders, and R. Scha, "Real-time bag-of-words, approximately," in *Proc. Conference On Image And Video Retrieval*, 2009.
- [37] F.-F. Li and P. Perona, "A Bayesian hierarchical model for learning natural scene categories," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005.

- [38] A. Ramanan and M. Niranjan, "A review of codebook models in patch-based visual object recognition," *Journal of Signal Processing Systems*, pp. 1–20, 2011.
- [39] A. Ramisa, S. Vasudevan, D. Scaramuzza, R. L. de Mántaras, and R. Siegwart, "A tale of two object recognition methods for mobile robots," in *Proc. Int. Conf. Comput. Vis.s Systems*, vol. 5008, Springer Verlag, Santorini, Greece: Springer Verlag, 2008, pp. 353–362.
- [40] I. Jebari, S. Bazeille, E. Battesti, H. Tekaya, M. Klein, A. Tapus, D. Filliat, C. Meyer, S. Ieng, R. Benosman, R. Benosman, E. Cizeron, J.-C. Mamanna, and B. Pothier, "Multi-sensor semantic mapping and exploration of indoor environments," in *IEEE Conference on Technologies for Practical Robot Applications (TePRA)*, 2011, pp. 151–156.
- [41] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Proc. Int. Conf. Comput. Vis.*, Oct. 2003, pp. 1470–1477.
- [42] D. Nistér and H. Stewénius, "Scalable recognition with a vocabulary tree," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2006, pp. 2161–2168.
- [43] M. Cummins and P. Newman, "Accelerated appearance-only SLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2008, pp. 1828–1833.
- [44] P. Newman, G. Sibley, M. Smith, M. Cummins, A. Harrison, C. Mei, I. Posner, R. Shade, D. Schrter, L. Murphy, W. Churchill, D. Cole, and I. Reid, "Navigating, recognising and describing urban spaces with vision and laser," *International Journal of Robotics Research*, vol. 28, pp. 1406–1433, 2009.
- [45] K. Konolige, J. Bowman, J. D. Chen, P. Mihelich, M. Calonder, V. Lepetit, and P. Fua, "View-based maps," in *Proc. Robotics: Science and Systems*, Seattle, USA, June 2009.
- [46] G. Sibley, C. Mei, I. Reid, and P. Newman, "Vast-scale outdoor navigation using adaptive relative bundle adjustment," *International Journal of Robotics Research*, vol. 29, pp. 958–980, June 2010.
- [47] J. Wang, H. Zha, and R. Cipolla, "Coarse-to-fine vision-based localization by indexing scale-invariant features," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 36, no. 2, pp. 413–422, 2006.
- [48] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Proc. ECCV International Workshop on Statistical Learning in Computer Vision*, 2004.
- [49] J. Dvorsky, P. Praks, and V. Snasel, "Latent semantic indexing for image retrieval systems," in *Linear Algebra Proc. Society for Industrial and Applied Mathematics*, 2003, pp. 1–8.
- [50] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman, "Discovering object categories in image collections," in *Proc. Int. Conf. Comput. Vis.*, 2005.
- [51] J. Sivic, B. Russell, A. Zisserman, W. Freeman, and A. Efros, "Unsupervised discovery of visual object class hierarchies," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, 2008, pp. 1–8.
- [52] S. Zhang, J. Huang, H. Li, and D. Metaxas, "Automatic image annotation and retrieval using group sparsity," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 3, pp. 838–849, 2012.
- [53] M. Brown and D. Lowe, "Unsupervised 3d object recognition and reconstruction in unordered datasets," in *Proc. IEEE International Workshop on 3-D Digital Imaging and Modeling*, 2005, pp. 1–8.
- [54] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio, "Robust object recognition with cortex-like mechanisms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 3, pp. 411–426, 2007.
- [55] K. Grauman and T. Darrell, "The pyramid match kernel: Efficient learning with sets of features," *The Journal of Machine Learning Research*, vol. 8, pp. 725–760, 2007.
- [56] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce, "3d object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints," *International Journal of Computer Vision*, vol. 66, pp. 231–259, 2006.
- [57] R. Hoover, A. Maciejewski, and R. Roberts, "Fast eigenspace decomposition of images of objects with variation in illumination and pose," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 41, no. 2, pp. 318–329, 2011.
- [58] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 3, pp. 3–338, 2010.
- [59] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. Int. Conf. Comput. Vis. and Pattern Recognition*, 2005.
- [60] Y. Huang, K. Huang, D. Tao, T. Tan, and X. Li, "Enhanced biologically inspired model for object recognition," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 41, no. 6, pp. 1668–1680, 2011.
- [61] K. Lai, L. Bo, X. Ren, and D. Fox, "Detection-based object labeling in 3d scenes," in *Proc. IEEE Int. Conf. Robot. Autom.*, may 2012, pp. 1330–1337.
- [62] T. Botterill, "Visual navigation for mobile robots using the bag-of-words algorithm," Ph.D. dissertation, University of Canterbury, 2010. [Online]. Available: <http://hilandtom.com/tombotterill/thesis.pdf>
- [63] A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer, "A fast and incremental method for loop-closure detection using bags of visual words," *IEEE Transactions on Robotics*, vol. Special issue on Visual SLAM, pp. 1–11, 2008.
- [64] T. Botterill, S. Mills, and R. Green, "Speeded-up bag-of-words algorithm for robot localisation through scene recognition," in *Proc. Image and Vision Computing New Zealand*, Nov. 2008, pp. 1–6.
- [65] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proc. European Conference on Computer Vision*, 2006, pp. 430–443.
- [66] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [67] T. Botterill, S. Mills, and R. Green, "Fast RANSAC hypothesis generation for essential matrix estimation," in *Proc. International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 2011.
- [68] G. Grisetti, S. Grzonka, C. Stachniss, P. Pfaff, and W. Burgard, "Efficient estimation of accurate maximum likelihood maps in 3d," in *Proc. Int. Conf. Intell. Robots Syst.*, 2007.
- [69] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *International Conference on Robotics and Automation*. IEEE, 2011, pp. 3607–3613.
- [70] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering," in *International Conference on Robotics and Automation*, 2011, pp. 3281–3288.
- [71] I. Esteban, L. Dorst, and J. Dijk, "Closed form solution for the scale ambiguity problem in monocular visual odometry," *Intelligent Robotics and Applications*, vol. 6424, pp. 665–679, 2010.
- [72] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge, UK: Cambridge University Press, 2003.
- [73] M. Cummins and P. Newman, "FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance," *International Journal of Robotics Research*, vol. 27, no. 6, pp. 647–665, 2008.
- [74] E. Limpert, W. A. Stahel, and M. Abbt, "Log-normal distributions across the sciences: Keys and clues," *BioScience*, vol. 51, no. 5, pp. 341–352, 2001.
- [75] S. Weerahandi, *Exact Statistical Methods for Data Analysis*. Springer, 1995.