A Bag-of-Words Speedometer for Single Camera SLAM

Tom Botterill and Richard Green
Department of Computer Science
University of Canterbury
Christchurch, NZ
Email: tom.botterill@grcnz.com

Steven Mills
Geospatial Research Centre
University of Canterbury
Christchurch, NZ

***Abstract*—This paper proposes a novel solution to the problem of scale drift in single camera SLAM based on recognising objects of known scale. 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. Our solution is to correct this scale estimate by recognising objects of known size. 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 scale drift by 75% while navigating a large indoor environment.**

I. Introduction

To work autonomously in an *a priori* unknown environment a mobile robot must be able to position itself using its sensors. A cheap, passive, compact and very common sensor is a single camera, and several practical schemes for navigation using monocular vision have recently been demonstrated [1]–[5]. Most [1], [3]–[5] operate within a Simultaneous Localisation and Mapping (SLAM) framework [6] in which a map is estimated as the robot explores. This map is used to correct errors that accumulate in any position estimate obtained through dead-reckoning, and is essential for long-term positioning [6], however for SLAM schemes to function reliably, and to allow the navigation of long paths away from previously mapped areas, these accumulated errors must be minimised [7].

A significant source of error unique to monocular vision is scale drift. A robot with a single camera can only work out the scale of the world, and hence its speed, by identifying and reconstructing the 3d structure of objects of known size (i.e. the calibration objects used to initialise some single camera SLAM schemes [1], [3], 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.

This paper describes a new algorithm exploiting a novel solution to this problem of scale drift: object classes are identified from the robot's internal SLAM map, the distribution of size within classes is measured, and this information is used to improve the robot's scale and speed estimates where more of these objects are observed. This algorithm is known as SCORE (Scale by Object Recognition), and it is demonstrated successfully reducing scale drift for a robot navigating a large indoor environment by sometimes 75%.

The following section describes the object recognition (OR) and machine learning techniques that SCORE is based on, and previous use of OR by mobile robots. Section III describes the problem in detail, and Section IV describes the SCORE algorithm. Section VI shows SCORE in operation, successfully reducing scale drift in single camera SLAM, and the final sections discuss our conclusions and SCORE's possible extensions.

II. Background

This section describes previous use of OR by mobile robots, and the OR schemes SCORE is derived from. Note that OR schemes either aim to recognise classes of objects (e.g. trees), or aim to identify particular objects (e.g. the small plane tree outside our office).

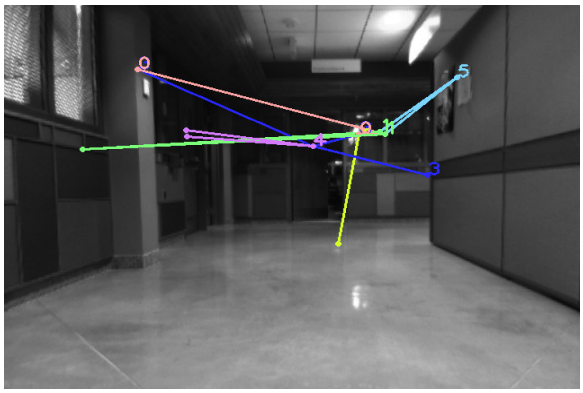


Fig. 1. Typically several objects are recognised and measured in each frame pair (University of Alberta dataset).

A. Real-time Object Recognition

For recognising a small database of distinctive objects local area descriptors such as SIFT [8] or SURF [9], or colour histograms [10] are commonly used [11]. These schemes do not scale to larger object databases however as individual descriptors are not sufficiently distinctive. Instead, combinations of descriptors are used, and the most popular model to represent these descriptors is the Bag-of-Words (BoW)

model [12]. This model describes an image by mapping each descriptor to its closest match from a 'dictionary'—a discrete set of representative features (also known as a 'codebook' or 'vocabulary'). This makes testing whether and image contains a particular set of 'image words' very fast, allowing large image databases to be searched for features from a particular object or location. This is often highly accurate despite ignoring the image geometry.

The BoW image representation also facilitates the automated learning of object classes or categories. These classes are usually defined by a weighted set of image words that tend to co-occur (and hence are assumed to arise as a result of an object in that class being visible). An image containing many of the words defining a class, particularly those that are most distinctive (those having the highest weights), is likely to contain the same objects as other images containing many of the same words. The original algorithm for automatically identifying these classes was Latent Semantic Analysis, or LSA [13] (popular variants include Probabilistic Latent Semantic Analysis [14] and Latent Dirichlet Analysis [15]). In LSA a sparse matrix is constructed where rows represent images, columns represent image words, and elements are the occurrence counts of each word in each image. These occurrence counts are weighted by the word's distinctiveness (measured using e.g. term-frequency-inverse document frequency, or TF-IDF [15], [16]). The principal components of this matrix are then found by by singular value decomposition (SVD). These principal components represent sets of co-occuring features that best partition images into those containing particular classes of objects; the three strongest might characterise for example 'cars', 'trees' and 'bikes', depending on the contents of the training images.

BoW-based OR is particularly suited for autonomous mobile robots as many SLAM schemes already maintain BoW databases in order to detect global loop-closure [4], [5], [17], and because many BoW schemes manage at least recognition is real-time [12], [16], [18], and sometimes training and recognition in real-time [19], [21]. Other more complex OR schemes, e.g. involving matching of 3d structure [22], non-feature-based approaches (e.g. texture analysis [23]), or hybrid approaches [24] are often highly accurate but are rarely fast enough for real-time OR (taking several minutes per object in these examples).

B. Object Recognition by Mobile Robots

Several mobile robots have been equipped with OR capability, but to our knowledge, all recognise particular object instances from an manually-trained database and none detect object classes for the purpose of scale estimation. Ahn et al. [25] and Castle et al. [26] manually construct databases of planar objects, then recognise these (using simple SIFT matching) and use them as robust landmarks for stereo EKF-SLAM and MonoSLAM [1] respectively. Castle et al. extended their scheme to incorporate the known size of particular objects into the SLAM solution [27], however the major limitation of these schemes is that they are only beneficial in environment with objects that have been mapped. This is much more restrictive than most SLAM schemes where emphasis is on exploring unknown environments.

Other applications of OR include spotting and tracking manually-tagged objects for Augmented Reality [28], and for planning task execution [11].

III. Analysis of problem

A robot observes classes of objects with particular size distributions as it explores an unknown environment. As it travels into previously unmapped areas its estimate of scale drifts and as a result its position and speed estimates deteriorate. However when it observes objects belonging to classes observed earlier, it can combine this information with its uncertain scale estimate to improve this scale estimate.

The major source of error in an object measurement from two frames is often the error in the estimated 'scale' between them (this equals the estimated speed of the robot between these two frames, which is also the baseline length from which points are reconstructed). This source of error applies equally to all objects measured in the same two frames. In addition as scales are accumulated sequentially, errors in each estimated scale are highly correlated with all other scale estimates from which this scale was calculated or will be used to calculate, and hence all object measurements are also correlated.

As with the SLAM problem [6], [7], these correlations are important for scale estimation. Kalman-filter-based SLAM algorithms [29] estimate matrices of correlations between landmark measurements. This is only feasible for moderately sized problems because blocks of this matrix are sparse, and has only been demonstrated on a larger scale when this map may be partitioned into local submaps of landmarks (e.g. [30]). Unlike landmarks however, object instances can not be assumed to easily partition into sets that are not observed at the same time, and due to the propagation of scale errors a matrix of object and scale covariances would be considerably more dense. This would makes a naive SLAM-like scale estimation computationally infeasible for real-time navigation. Instead, the following section describes our approximate solution.

IV. SCORE: a Bag-of-Words Speedometer scheme

The SCORE (Scale by Object Recognition) algorithm is outlined in Figure 2, and is detailed in the following subsections. In summary:

> As we re-train periodically to learn new environments, we also learn new object classes. The measured sizes of objects in these classes are used to improve existing and new scale estimates.

SCORE is designed to integrate with BoWSLAM, the monocular SLAM scheme described by [5]. BoWSLAM represents every frame as a 'Bag-of-Words' using the scheme by [21]. This high-level representation is used for active loop closure detection, fast stereo matching, and to select the sequence of frames used to position each new frame relative to. This representation is also ideal for object recognition. The scheme described here could easily be adapted to other single

For each frame:

When re-training:

1) Identify (learn) the B object classes
2) Measure each 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 objects reconstructed here
2) Combine the most likely scale given the observations with the measured scale

Fig. 2. Overview of the SCORE algorithm

camera SLAM schemes which index a subset of frames into a BoW database [4], [17], [31].

The effect of this algorithm is to propagate the reliable scale estimates from better mapped areas to areas where the scale is much less certain but where the same kinds of objects are visible. In practice there is often a dramatic distinction between scale estimates in different areas, for example scale estimates are reasonably accurate when the robot is moving is a straight line in a feature-rich environment, then deteriorate suddenly when the robot corners.

A. Classes of Measurable Objects

This subsection describes how SCORE defines a class of objects, and how the distribution of sizes within that class are measured.

For this application, two important requirements are that object classes can be learnt and identified in real-time, and that identified objects can be measured. The object classes recognised in real-time by contemporary OR schemes (Section II-A) 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. We do not consider measurements of more than two features as this would add to the complexity (there are $\binom{n}{2}$ possible measurable distances between n points), and would introduces difficulties in coping with partially-observed and partially-reconstructed objects. As a result SCORE's object classes are each defined by the co-occurrence of two image words. Multiple instances of the same object are likely to be visible in many scenes; however by assuming objects are separated by more than the separation of the features within them we only need to observe the least separation from all possible pairs of two features visible in a scene.

To identify co-occuring image words we use the same term-document matrix as LSA. Elements of this matrix represent the frequency each word occurs in each image, multiplied by their TF-IDF weight to favour distinctive words. We only count features that are successfully reconstructed to avoid learning object classes which cannot often be measured. As computing the SVD of this matrix (which can realistically have 10 000 rows and 50 000 columns) is infeasible for a real-time system, and because we are only interested in simple two-word objects, we use this matrix to select the B pairs of reconstructed words that are most likely to co-occur.To learn these two-word object classes we multiply every pair of co-occuring word's matrix elements, and sum these products for each co-occurrences. The B pairs of words with the highest total are selected as defining the object classes. This method is derived from LSA: in the simplified case where two-word objects occur independently and each word arises only as a result of one of these objects, the objects we select are exactly the same as those that would be selected by LSA.

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

B. Estimating Object Class Size Distribution Parameters

This subsection describes how a distribution is fitted to these noisy measurements, that incorporates both errors and variability in object sizes within each class. There are five sources of variability in these observed feature 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 occuring in multiple contexts.
4) Errors in reconstructing 3d point positions.
5) Errors from measurements of multiple partially-visible objects, or features occuring in multiple objects.

The combination of these variables is assumed Normal; a heavy-tailed distribution is likely to provide a more accurate model but makes the combination of distributions intractable.All we assume about measurements of an object is they are drawn from the same distribution (this is unlike SLAM observations which should be distributed about their actual values). Examination of measurements (Figure 5) suggests these assumptions are not unreasonable.

In BoWSLAM the estimated scales have certainty measured by a condition number [5], which measures the additional error in a subsequent scale estimates calculated by adding this relative scale estimate to others (a scale estimate with condition $c \in (0, 1)$ will give a standard deviation estimate $s_{later} = s_{earlier}/c$ where was the previous estimated s.d. in scale). This condition number is a heuristic, but is a useful

simplification when measuring uncertainties that propagate by multiplication (a very similar heuristic is used in other SLAM schemes [4], [17] based on the ATLAS framework [20]).

We then assume measurement errors (including errors due to scale uncertainty) are independent; for each object class this allows the parameters of a Normal distribution to be estimated by taking the weighted sample mean μ and weighted sample variance σ^2 of the N measurements x_i, weighted by the (normalised) condition numbers w_i.

$$\mu = \sum_{i=1}^{n} w_i x_i \tag{1}$$

$$\sigma^2 = \frac{1}{1-\sum_{i=1}^{n} w_i^2} \sum_{i=1}^{N} w_i \left(x_i - \mu\right)^2 \tag{2}$$

Variance is scaled to avoid any single object measurement, even one much bigger or smaller than any previously observed values, from overly distorting scale estimates (these observations are not generally outliers, and are the reason a heavy-tailed distribution would be more appropriate).

Fig. 3. Objects from the same class observed on several similar-looking doors, University of Alberta dataset. (Each 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.)

C. Object Observations and Improving Scale Estimates

Following re-training [21], every scale estimate (between pairs of frames) is updated with information from measurements of objects reconstructed between these two frames. The same method is used to update new scale estimates as new edges are added.

Given two frames where 3-D points have been reconstructed, a set of image words corresponding to these reconstructed points are found. This list is searched for each of the objects. These objects are measured, assuming a unit baseline, giving measurements $\mathbf{x}$. Our assumptions imply $s\mathbf{x} \sim N(\boldsymbol{\mu}, \mathrm{diag}(\boldsymbol{\sigma}^2))$, where s is the scale. The MLE of s given object measurements, s_{OR} is then given by:

$$s_{OR} = \frac{\sum_{j=1}^{M} \frac{x_j \mu_j}{\sigma_j^2}}{\sum_{j=1}^{M} \left(\frac{x_j}{\sigma_j}\right)^2} \tag{3}$$

This estimate is derived by differentiating the likelihood of s given $\mathbf{x}$ and has variance:

$$v_{OR} = \frac{1}{\sum_{j=1}^{M} \left(\frac{x_j}{\sigma_j}\right)^2} \tag{4}$$

We now have two scale measurements: the estimated scale and its variance from SLAM (s_{SLAM} and v_{SLAM}, Section IV-B), and the estimated scale and its variance from object measurements (s_{OR} and v_{OR}, Equations 3 and 4). The MLE of the scale between these two frames under our assumptions is then given by Equation 5 (this is equivalent to two Kalman Filter updates):

$$s = \frac{s_{OR} v_{SLAM} + s_{SLAM} v_{OR}}{v_{OR} + v_{SLAM}} \tag{5}$$

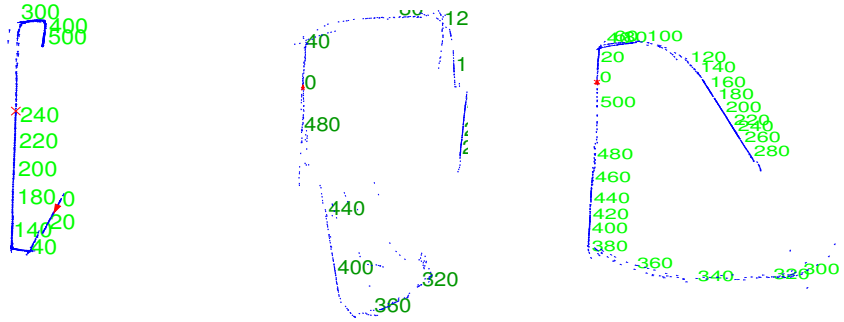
D. Analysis of SCORE

SCORE measures the size distributions of object classes, weighted towards the observations with the least uncertainty in scale. When updating scales, those that have low initial certainty are often significantly adjusted, whereas the most reliable scale estimates are barely adjusted at all. This propagates the certainty of the best position estimates to those that are less good. It does not matter that in many frames few or no objects are observed—scales are calculated by accumulating sequences of scale estimates, so just the occasional observation can improve local estimates.

Note that prior scale estimates from the SLAM algorithm alone are used to measure object sizes every time we re-train; iteratively updating scale estimates from previous object sets would cause the measured object sizes to converge incorrectly to a tighter distribution (not reflecting true variations in object sizes) when only a small number, of objects are observed in a pair of frames.

V. Complexity

For long-term SLAM constant time complexity in the number of frames n is essential, however no SLAM schemes to-date has quite achieved this, and many [1], [30] have $O(n^2)$ complexity. BoWSLAM has $O(n \log(n))$ complexity (updating the map and retraining the BoW dictionary). Identifying the B best object classes gives the SCORE algorithm



(a) Significant scale-drift is observed with no scale-drift correction

(b) SCORE reduces scale drift by 75% in this example

(c) The performance of SCORE is comparable with the heuristic used in BoWSLAM (used here)

Fig. 4. Maps of robot poses from University of Alberta dataset: a robot travels 75m around a rectangular coridoor at constant speed. With no scale correction large scale-drift is seen, stretching the map (a). SCORE successfully reduces this scale drift (b). (Maps' scales are arbitrary due to global scale ambiguity.)

complexity $O(n \log(n) + Bn)$. As B may be fixed then SCORE does not add to the complexity of BoWSLAM. In practice with a database of 100 objects only about 10% (20ms per frame) to BoWSLAM's running time; this is still fast enough for real-time performance. Other schemes [4], [17] representing images as bags of words index images at a similar or slower rate, which would make this a small additional cost.

If an object database good enough for indefinite use is found (or if a manually-trained database with known scales was used) the complexity would be reduced to just $O(B)$ per frame, giving the SCORE algorithm potential to be used for long-term navigation.

VI. Results

The first dataset demonstrates object recognition and scale estimation in a typical indoor environment. The dataset, from the University of Alberta [32], consists of a 75m rectangular loop around corridors and landings, traversed at an approximately constant speed. Figure 4 shows three maps of this loop; one has no scale correction, and exhibits large scale-drift (speeding up to several times its original speed). The next map used the SCORE algorithm to improve scale estimates. SCORE successfully reduces the scale drift: on the final re-training, after 300 frames, SCORE improves the mean speed estimate between all pairs of frames from 0.17 of the original speed to 0.79 (the true value is approximately 1.0). Previously, BoWSLAM has used a heuristic that the measured depth of points cannot fall outside an allowed range [5]. The assumption this heuristic is based on is not true in general, and it can never eliminate scale drift within the allowed range, however it does work reasonably well in practice, as shown by the third map. This dataset shows that SCORE can match the performance of this heuristic.

SCORE selects 100 objects each time it is re-trained. Typically several of these are found in each frame (Figure 1). The repeating structures indoors help SCORE find reliable objects—one of these occuring on several similar doors is shown in Figure 3.

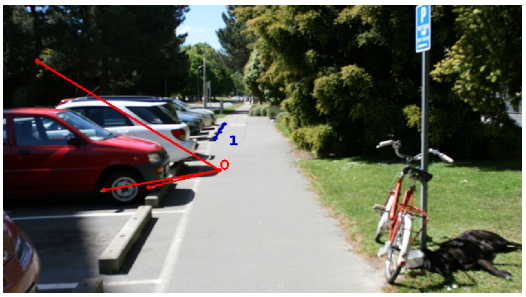
The second dataset was captured outdoors in a suburban environment. Figure 5 shows some of the measurements made of two of the objects that were detected, and a histogram of these measurements (15 and 19 measurements were made). One measures 'round edged shadows' and another 'car wheels'.The distributions of these objects are significantly different, with weighted means 1.0 and 5.9, however large standard deviation estimates (4.2 and 6.2) limit their impact when updating scales. While SCORE is successfully detecting and measuring objects outdoors, the scale estimates from these observations do not significantly improve the estimates from SLAM. This is probably due to the large distributions of object sizes outdoors, and the high depth disparities of distant reconstructed points. Future work will investigate these issues in detail.

VII. Conclusions

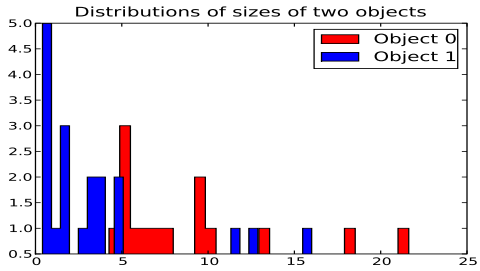
This paper has demonstrated that BoW Object Recognition may be used to learn to recognise objects, measure their distribution of sizes, and use this information to correct for scale drift in single-camera SLAM. Our SCORE algorithm is the first solution to this problem which has the potential to correct scale drift over indefinitely long tracks, and results in only a small increase in the computational cost. Experiments on indoor and outdoor datasets demonstrate that object classes with a variety of different size distributions are found. Indoors, this can be used to greatly reduce scale drift, reducing errors by an average of about 75% in one example. Outdoors, the SCORE algorithm does not perform well, however this is only the first version and analysis of the problem has has suggested possibilities by which a future version of SCORE may be extended to work well in all environments.

VIII. Future work

There are certainly many small improvements that would increase the reliability of our scheme (in particular investigating methods for choosing and robustly measuring more complex objects, maybe even manually defining object classes), however the obvious major extension would be to improve the



(a) Distances measured of two objects, 'round shadows', and 'car wheels'.



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

Fig. 5. Object classes measured in the outdoor dataset have a variety of significantly different size distributions.

model so that scales and object size distributions are measured simultaneously, taking advantage of the correlation between them. This problem has high complexity, does not partition easily, and has no well-defined choice of objective function, however future research may solve some of these issues in similar ways as difficulties in SLAM are being overcome [30]. Viewing the correlations between scales and objects as a graph to be optimised may provide insight [33].

ACKNOWLEDGMENT

The University of Alberta CSC data set was obtained from the Robotics Data Set Repository (Radish) [32]. Thanks go to Jonathan Klippenstein for providing this data.

REFERENCES

[1] A. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *Proc. Int. Conf. Computer Vision*, Oct. 2003.

[2] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry for ground vehicle applications," *Journal of Field Robotics*, vol. 23, no. 1, 2006.

[3] E. Eade and T. Drummond, "Scalable monocular SLAM," in *Proc. CVPR*, vol. 1. Los Alamitos, CA, USA, 2006, pp. 469–476.

[4] E. Eade and T. Drummond, "Unified loop closing and recovery for real time monocular SLAM," in *British Machine Vision Conference*, 2008.

[5] T. Botterill, S. Mills, and R. Green, "Bag-of-Words-driven single camera SLAM," Geospatial Research Centre, Tech. Rep., 2009. [Online]: http://open.grcnz.com/papers/Botterill-etal-2009b-preprint.pdf

[6] R. Smith, M. Self, and P. Cheeseman, *Autonomous Robot Vehicles*. Amsterdam: Springer Verlag, 1990, ch. Estimating Uncertain Spatial Relationships in Robotics, pp. 435–461.

[7] 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.

[8] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. Int. Conf. Computer Vision*, 1999, pp. 1150–1157.

[9] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, 2008.

[10] P. Jensfelt, S. Ekvall, D. Kragic, and D. Aarno, "Integrating slam and object detection for service robot tasks," in *IROS 2005 Workshop on Mobile Manipulators: Basic Techniques, New Trends and Applications*. Edmonton, Canada, 2005.

[11] A. Ramisa, S. Vasudevan, D. Scharamuzza, R. L. de Mántaras, and R. Siegwart, "A tale of two object recognition methods for mobile robots," in *Proc. Int. Conf. Computer Visions Systems*, vol. 5008, Springer Verlag. Santorini, Greece: Springer Verlag, 2008, pp. 353–362.

[12] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Proc. Int. Conf. Computer Vision*, Oct. 2003, pp. 1470–1477.

[13] J. Dvorsky, P. Praks, and V. Snasel, "Latent semantic indexing for image retrieval systems," in *Linear Algebra Proceedings of the Society for Industrial and Applied Mathematics*, 2003, pp. 1–8.

[14] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman, "Discovering object categories in image collections," in *Proc. Int. Conf. Computer Vision*, 2005.

[15] L. Fei-Fei and P. Pietro, "A bayesian hierarchical model for learning natural scene categories," in *Proc. CVPR*, 2005.

[16] D. Nistér and H. Stewénius, "Scalable recognition with a vocabulary tree," in *Proc. CVPR*, Jun. 2006, pp. 2161–2168.

[17] 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," *To appear in Int. Journal of Robotics Research*, 2009.

[18] M. Cummins and P. Newman, "Accelerated appearance-only SLAM," in *ICRA*, May 2008, pp. 1828–1833.

[19] A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer, "Real-time visual loop-closure detection," in *ICRA*, 2008.

[20] M. Bosse, P. Newman, J. Leonard, and S. Teller. "Simultaneous localization and map building in large-scale cyclic environments using the atlas framework," in *Int. Journal of Robotics Research*, 23, 2004.

[21] 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.

[22] M. Brown and D. Lowe, "Unsupervised 3d object recognition and reconstruction in unordered datasets," in *Proc. Int. Workshop on 3-D Digital Imaging and Modeling*, 2005, pp. 1–8.

[23] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio, "Robust object recognition with cortex-like mechanisms," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 3, 2007.

[24] 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," *Int. Journal of Computer Vision*, vol. 66, pp. 231–259, 2006.

[25] 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. Intelligent Robots and Systems*, Oct. 2006, pp. 2588–2594.

[26] R. O. Castle, D. J. Gawley, G. Klein, and D. W. Murray, "Video-rate recognition and localization for wearable cameras," in *Proc. British Machine Vision Conference*, 2007, pp. 1100–1109.

[27] R. O. Castle, G. Klein, and D. W. Murray, "Combining localization with recognition for scene augmentation using a wearable camera," 2009, preprint.

[28] G. Reitmayr, E. Eade, and T. Drummond, "Semi-automatic annotations in unknown environments," in *Proc. ISMAR*, Nara, Japan, 2007.

[29] G. Dissanayake, H. Durrant-Whyte, and T. Bailey, "A computationally efficient solution to the simultaneous localisation and map building (SLAM) problem," in *ICRA*, 2000, pp. 1009–1014 vol.2.

[30] J. Guivant, "Efficient simultaneous localisation and mapping in large environments," Ph.D. dissertation, The University of Sydney, 2002.

[31] P. Newman, D. Cole, and K. Ho, "Outdoor slam using visual appearance and laser ranging," in *ICRA*, Florida, 2006.

[32] A. Howard and N. Roy, "The robotics data set repository (radish)," 2003. [Online]. Available: http://radish.sourceforge.net/

[33] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *Proc. AAAI National Conf. Artificial Intelligence*, Edmonton, 2002.