Navigating, Recognizing and Describing Urban Spaces With Vision and Lasers

Paul Newman
Gabe Sibley
Mike Smith
Mark Cummins
Alastair Harrison

Oxford Mobile Robotics Group,
Department of Engineering Science,
University of Oxford, Parks Road, Oxford, UK
pnewman@robots.ox.ac.uk

Chris Mei

Active Vision Lab,
Department of Engineering Science,
University of Oxford, Parks Road, Oxford, UK

Ingmar Posner
Robbie Shade
Derik Schroeter
Liz Murphy
Winston Churchill
Dave Cole

Oxford Mobile Robotics Group,
Department of Engineering Science,
University of Oxford, Parks Road, Oxford, UK

Ian Reid

Active Vision Lab,
Department of Engineering Science,
University of Oxford, Parks Road, Oxford, UK
ian@robots.ox.ac.uk

Abstract

In this paper we describe a body of work aimed at extending the reach of mobile navigation and mapping. We describe how running topological and metric mapping and pose estimation processes concurrently, using vision and laser ranging, has produced a full six-degree-of-freedom outdoor navigation system. It is capable of producing intricate three-dimensional maps over many kilometers and in real time. We consider issues concerning the intrinsic quality of the built maps and describe our progress towards adding semantic labels to maps via scene de-construction and labeling. We show how our choices of representation, inference methods and use of both topological and metric techniques naturally allow us to fuse maps built from multiple sessions with no need for manual frame alignment or data association.

KEY WORDS—Mobile robotics, navigation, mapping, SLAM, laser, vision, stereo, visual odometry, semantic labeling, systems, topological navigation, loop closure, FABMAP

The International Journal of Robotics Research
Vol. 28, No. 11–12, November/December 2009, pp. 1406–1433
DOI: 10.1177/0278364909341483

Figures 1–11, 15, 16, 19–21, 23, 25–31 appear in color online:
http://ijr.sagepub.com

1. Introduction

In this paper we describe the techniques that we are employing to build an end-to-end and infrastructure-free urban navigation system. We wish to build an embedded system capable of repetitively and progressively (i.e. over multiple sessions) mapping large urban areas time and time again in six degrees of freedom (DOFs). Our concerns range from the low-level control of sensors and filtering their output through to perception, estimation and inference, longevity, introspection, loop closing, data management, software architectures and up to high-level semantic labeling of maps. In the spirit of the *International Symposium of Robotics Research* (ISRR), we aim to provide the reader with a technical panorama of how these components work together and, while doing so, direct the reader to more detailed technical accounts, to discuss their strengths and weaknesses and, where applicable, any open questions.

Recent years have seen wholesome progress in building robotic systems that can navigate in outdoor settings. The recent literature on the DARPA Grand Challenges (Iagnemma and Buehler 2006a,b; Thrun et al. 2006; Urmson et al. 2008) is a testament to the complexity of the problems involved@ problems that necessitate both a systems point of view and a deep understanding of the perception and inference tasks involved. This paper, submitted to the special issue on ISRR07, describes our progress towards building a combination of hardware and software which will enable a robot to operate in typical urban environments (with or without *a priori* information) over extended periods of time with no reliance on the Global Positioning System (GPS). For any particular session, in real time, our software infrastructure is able to process stereo images (collected at 20 Hz), infer 6-DOF pose and dense disparity maps, detect and apply loop closures using images from a panoramic camera, generate hi-fidelity three-dimensional (3D) laser maps and shade them with reflectance and/or color image data. Following this, we can annotate these maps with textual semantic labels.

While this functionality is a good substrate for single-session mobile autonomy, we have the additional goal of supporting a "life long learning" paradigm. We learn, in an unsupervised fashion, models of the appearance of typical workspaces from large amounts of training data (thousands of images). By logging all data (at around 60 Mb s^{-1}) and considering the totality of all datasets offline, this model, via the Fast Appearance Based topological mapping framework ("FAB-MAP") described in Section 3, allows us to stitch together intersecting vehicle trajectories from sessions taken days apart with no user intervention. Our loop closure apparatus browses the union of recorded images and discovers intersections and overlaps between sessions. With these topological constraints in hand, we are able to fuse chunks of maps together, building ever larger metric and topological representations of the workspace. We now outline the structure of this paper by walking through the key components of our system.

- *Pose and trajectory estimation* is a fundamental requirement for our work and we currently have two alternatives. The first, discussed by Newman et al. (2006), Cole and Newman (2006) and Ho and Newman (2007), is a simultaneous localization and mapping (SLAM) system driven by scan matching between 3D laser point clouds, which is based on the Exactly Sparse Delayed State formulation proposed by Eustice et al. (2005). The second, which we focus on in this paper, is more suited to the vehicle shown in Figure 2. It is based on the Sliding Window Filter of Sibley et al. (2007) and is driven by robust inter-frame feature tracking across sequential stereo image pairs. This vision system is described in Section 2. Our motivation for pushing the vision-based system over our 3D laser-based system is threefold: first, stereo cameras are cheap; second, they capture the geometry of the local scene orders of magnitude faster than scanning lasers; finally, in contrast to many scan matching techniques, the registration between sequential stereo views (modulo correct feature tracking) uses the same real-world artifacts rather than two different clouds of laser points sampled from the workspace's surfaces.

- *Topology inference.* However good the online pose estimation engine is, without global information loop closure detection and prosecution (acting on the loop closure detection and altering trajectory and map estimates) will always be a concern. Our loop closure detection component, "FAB-MAP" (Cummins and Newman 2007, 2008b,a) is probabilistic and solely appearance based. Crucially for our needs, it is exceptionally fast and has an extremely low false-positive rate; it is discussed further in Section 3.

- *Global optimization.* Between them, the trajectory estimation and loop closing (FAB-MAP) processes produce a graph of poses where edges represent the metric proximity between poses. The pose estimation system directly provides high-quality interpose constraints. The metric parameterizations of the loop closures are however very uncertain: all we know is that we are close to a place we have been before. In Section 4 we describe how this topological information is upgraded to a metric constraint. We do so either using an iterative closest point (ICP) match of local-region point clouds or using two pairs of stereo images. Following that we perform pose relaxation over the graph of poses and we discuss the formulation of the optimization in Section 5.

- *3D map creation.* In this paper, and in contrast to our earlier work, we do not use lasers for pose estimation; instead, given a high-quality 6-DOF vehicle trajectory, we can capture the far-field 3D structure, color

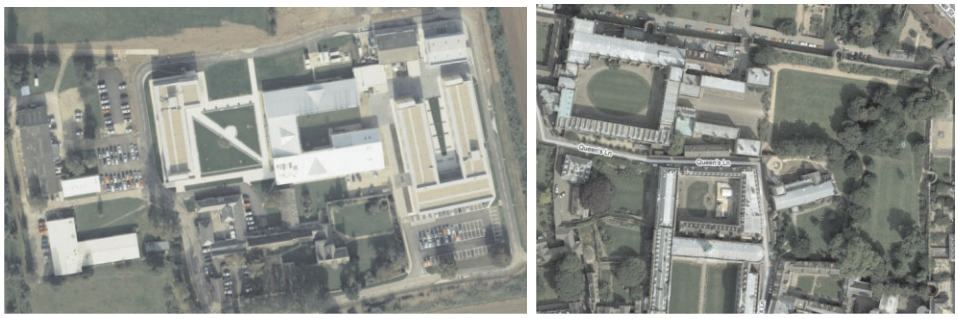


Fig. 1. Aerial photos of the data collection sites, "Begbroke" (left) and "New College" (right). The crisscross pattern in the figures of the Begbroke dataset was executed in the vertical (green) wedge-shaped patch in the east and the large loop around the "C" shaped building. The "quad" of New College, around which many small (circa 100 m) loops were made, can be seen in the North West of the right image. The large "dog leg" shaped loop in the New College datasets runs East out of the quad and around the perimeter of the gardens.

and surface reflectance properties of the workspace by "trawling" a pair of vertically oriented lasers through the workspace while taking a great deal of care regarding time-stamping and system delay estimation. In Section 6 we present some of the maps we are able to produce and go on to analyze their detail and quality.

- *Dense stereo.* We have a facility to compute dense disparity maps from our stereo rig in real time. This can be used for obstacle avoidance tasks but here it is used to fill in the 3D structure of the workspace which is not sampled by our laser scanners, thus producing total scene coverage. In Section 6.2 we describe the approach we use for disparity calculation and present statistics regarding its performance.
- *Scene labeling.* After map building comes our final step, which is the addition of semantic labels to the maps. Section 8 describes how by learning a generative model of visual and geometric appearance we are able to classify regions of the point clouds into one of (currently) seven classes using a support vector machine.

Table 1. Summary of the Salient Properties of the Two Datasets Used in This Paper

Name	Measure	Value
Begbroke	Size	9.3 GB
	Laser	
	Stereo	20 Hz at 512 × 384 mono
	Omnicam	2 Hz, five images color
	Distance driven	1.08 km
	Sessions	single shot
New College	Size	Laser: 2.9 GB, Images 53 GB
	Laser	2 × 75 Hz over 90° at 0.5° resolution
	Stereo	20 Hz at 512 × 384 mono
	Omnicam	2 Hz, five images color
	Distance driven	5.13 km
	Sessions	Multiple over three days

1.1. Data Sets

For reasons of clarity, figures and tables of the results will be presented close to the text that describes the techniques that generate them, rather than in a monolithic results section. We therefore need to describe the datasets up front so they can be referred to in individual sections. We collected data from two principal sites in Oxfordshire, UK. We refer to them as "Begbroke" and "New College" and their characteristics are summarized in Table 1. Aerial photos of both data collection sites are shown in Figure 1, and the caption describes how to locate the trajectories of the vehicle shown in this paper within these aerial images. In all 67.2 GB of data was logged, all of which has been processed and presented in this paper. Much of the New College data has been published as part of an IJRR Data Paper and can be downloaded and used by interested readers (Smith et al. 2009).

1.2. Platform

All of the algorithms, systems and results in this paper have been applied to data gathered by the vehicle shown in Figure 2. While there is nothing vehicle-specific in our work, it

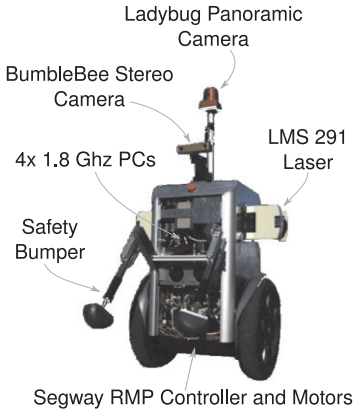


Fig. 2. The results in this paper correspond to data gathered from the modified Segway platform shown above. The vehicle has a sensor payload of two SICK lasers, an XSens inertial sensor, a GARMIN GPS, a Point Grey stereo "Bumblebee" camera and a "Ladybug 2" panoramic camera. It carries four small form factor PCs linked with a Gigabit internal network. Total onboard storage is of the order of 1 TB.

is worth swiftly summarizing the vehicle's characteristics. The vehicle is actuated by a RMP200 base from Segway. It has four internal PCs at 1.6 GHz with around 1 TB of total storage. Images streamed at 2 Hz from a Point Grey Ladybug camera (five panoramic images) are used in our appearance-based loop closure (FAB-MAP) algorithm. Stereo pairs read at 20 Hz from a Point Grey Bumblebee camera are used for the online pose estimation and dense stereo. Two vertically mounted LMS 291 lasers are used in 75 Hz mode to capture the far-field geometry. The vehicle can run for approximately 90 minutes on a single battery charge with all systems powered.

2. Real-time Pose Estimation from Stereo

To reveal the underlying structure of the pose estimation in unknown environments problem, it is useful to approach it from the non-linear least-squares optimization perspective. This point of view is much more in line with traditional statistical point estimation than state space filtering. This perspective is useful for a number of reasons. First, it highlights the fundamental minimization principle at work in least squares, which is sometimes harder to see from the state-space filtering perspective. Second, starting with the underlying probability density functions (PDFs) that describe our problem, it clearly shows the probabilistic nature of the task, that is, tracking a joint distribution through a large state space; a state space that changes dimension as we undertake the fundamental probabilistic operations of removing parameters via marginalization, and adding parameters via error propagation and conditioning. A third reason to use statistical point estimation is because it

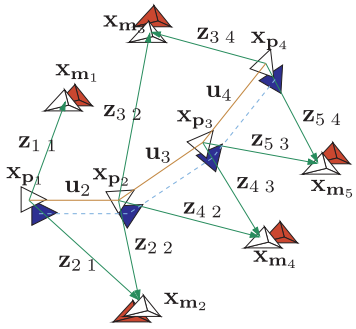


Fig. 3. SLAM notation.

exposes a rich body of theory about the convergence of least-squares estimators. Further, starting from least squares one can easily see the connection to many important concepts such as Newton's method, Fisher information, and the Cramer Rao lower bound, all of which have intuitive derivations starting from traditional statistical point estimation.

2.1. Notation

We adopt the following notation as illustrated by Figure 3:

- the 6D robot poses will be denoted by $\mathbf{x_p} = [\mathbf{x}_{\mathbf{p}_1}^{\mathrm{T}}, \ldots, \mathbf{x}_{\mathbf{p}_m}^{\mathrm{T}}]^{\mathrm{T}}$;
- the 3D landmarks will be written as $\mathbf{x_m} = [\mathbf{x}_{\mathbf{m}_1}^{\mathrm{T}}, \ldots, \mathbf{x}_{\mathbf{m}_n}^{\mathrm{T}}]^{\mathrm{T}}$;
- $\mathbf{z}_{ij}$ will indicate a measurement of the ith landmark observed from the jth pose;
- an input command to the robot (or a motion model) from a jth pose will be written $\mathbf{u}_j$.

The state vector, comprising the map and poses, is $\mathbf{x} = [\mathbf{x}_{\mathbf{m}}^{\mathrm{T}}, \mathbf{x}_{\mathbf{p}}^{\mathrm{T}}]^{\mathrm{T}}$ and has dimension $\dim(\mathbf{x}) = 6m + 3n$. The aim is to estimate the state vector from the input commands and measurements. The effect of the input command on the pose is modeled by the *process model* and the effect of the measurement appears through the *sensor model*.

- *Process model.* The process model describes how the current pose can be estimated from the previous pose using the input command $f_j : \mathbb{R}^6 \rightarrow \mathbb{R}^6$, $\mathbf{x}_{\mathbf{p}_j} = f_j(\mathbf{x}_{\mathbf{p}_j}, \mathbf{u}_{j+1}) + \mathbf{w}_{j+1}$, where $\mathbf{w}_{j+1}$ is the process noise that we assume to be Gaussian (this is a common assumption). The noise vector $\mathbf{w}_{j+1}$ is additive and we assume it follows a normal distribution $\mathbf{w}_{j+1} \sim \mathcal{N}(0, \mathbf{Q}_{j+1})$, so that $\mathbf{x}_{\mathbf{p}_{j+1}} \sim \mathcal{N}(f_j(\mathbf{x}_{\mathbf{p}_j}, \mathbf{u}_{j+1}), \mathbf{Q}_{j+1})$.

- *Sensor model.* The sensor model, $h_{ij} : \mathbb{R}^{\dim(\mathbf{x})} \rightarrow \mathbb{R}^{\dim(\mathbf{z}_{ij})}$, returns the expected value the sensor will give when the ith landmark is observed from the jth pose: $\mathbf{z}_{ij} = h_{ij}(\mathbf{x}_{\mathbf{m}_i}, \mathbf{x}_{\mathbf{p}_j}) + \mathbf{v}_{ij}$. We assume that $\mathbf{v}_{ij} \sim \mathcal{N}(0, \mathbf{R}_{ij})$ so that $\mathbf{z}_{ij} \sim \mathcal{N}(h_{ij}, \mathbf{R}_{ij})$, where $\mathbf{R}_{ij}$ is the observation error covariance matrix. Concatenating all of the observations, measurement functions and measurement covariances together, $\mathbf{z} = [\mathbf{z}_{10}^{\mathrm{T}}, \mathbf{z}_{11}^{\mathrm{T}}, \ldots, \mathbf{z}_{nm}^{\mathrm{T}}]^{\mathrm{T}}$, $h = [h_{10}^{\mathrm{T}}, h_{11}^{\mathrm{T}}, \ldots, h_{nm}^{\mathrm{T}}]^{\mathrm{T}}$, and $\mathbf{R} = \mathrm{diag}(R_{10}, R_{11}, \ldots, R_{nm})$, gives $\mathbf{z} \sim \mathcal{N}(h, \mathbf{R})$, which defines the measurement likelihood $p(\mathbf{z}|\mathbf{x})$. The first pose $\mathbf{x}_{\mathbf{p}_1}$ is a hyper-parameter that fixes the first pose and, thus, the entire system (this also removes the gauge freedom).

To be concrete, in this paper which uses stereo vision, h_{ij} projects the ith 3D landmark into the image taken from the jth pose and so z_{ij} is a pixel position (u, v).

We might also assume that we have *prior information* about the map and landmarks that can be represented by a Gaussian. Let $\hat{\mathbf{x}}_\Pi \sim \mathcal{N}(\mathbf{x}_\Pi, \boldsymbol{\Pi}^{-1})$ denote the prior information about the first pose and the map

$$\hat{\mathbf{x}}_\Pi = \begin{bmatrix} \hat{\mathbf{x}}_{\mathbf{m}} \\ \hat{\mathbf{x}}_{\mathbf{p}_1} \end{bmatrix},$$

$$\boldsymbol{\Pi} = \begin{bmatrix} \boldsymbol{\Pi}_{\mathbf{m}} & \boldsymbol{\Pi}_{\mathbf{pm}} \\ \boldsymbol{\Pi}_{\mathbf{pm}} & \boldsymbol{\Pi}_{\mathbf{p}} \end{bmatrix}.$$

By combining the process information with the prior information, we obtain the prediction PDF:

$$p(\mathbf{x}) = \mathcal{N}\left(\begin{bmatrix} \mathbf{x}_\Pi \\ f(\mathbf{x}) \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Pi}^{-1} & \\ & \mathbf{Q} \end{bmatrix} \right). \tag{1}$$

Under these Gaussian assumptions, the joint probability $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{z}|\mathbf{x})p(\mathbf{x})$ of the measurements and the state vector is

$$p(\mathbf{z}|\mathbf{x})p(\mathbf{x}) = \mathcal{N}\left(\begin{bmatrix} \mathbf{x}_\Pi \\ f(\mathbf{x}) \\ h(\mathbf{x}) \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Pi}^{-1} & & \\ & \mathbf{Q} & \\ & & R \end{bmatrix} \right). \tag{2}$$

Our goal is to compute the value of $\mathbf{x}$ which maximizes this density, with $\mathbf{z}$ being a fixed set of measurements.

Taking logs and ignoring constant terms that do not depend on $\mathbf{x}$, we see that maximizing $p(\mathbf{x}, \mathbf{z})$ is equivalent to minimizing

$$\ell(\mathbf{x}) = \frac{1}{2}(g(\mathbf{x})^{\mathrm{T}}\mathbf{C}^{-1}g(\mathbf{x})) = \frac{1}{2}||r(\mathbf{x})||^2, \tag{3}$$

where

$$g(\mathbf{x}) = \begin{bmatrix} g_\Pi(\mathbf{x}) \\ g_f(\mathbf{x}) \\ g_z(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \mathbf{x}_\Pi - \hat{\mathbf{x}}_\Pi \\ \mathbf{x}_\mathbf{p} - f(\mathbf{x}) \\ \mathbf{z} - h(\mathbf{x}) \end{bmatrix},$$

$$\mathbf{C} = \begin{bmatrix} \boldsymbol{\Pi}^{-1} & & \\ & \mathbf{Q} & \\ & & \mathbf{R} \end{bmatrix},$$

and we have lumped the sensor model, process model and prior information terms together.

The goal is to find the choice of $\mathbf{x}$ which minimizes the quadratic non-linear cost functional $\ell(\mathbf{x})$. Writing the normal equations associated with the Gauss–Newton method for solving non-linear least squares gives us an insight into the structure of the problem. Let $\mathbf{g}_p$ and $\mathbf{g}_m$ be the right-hand side vectors corresponding to the robot path and map, respectively. The Gauss–Newton update can be expressed as a 2×2 system of equations:

$$\begin{bmatrix} \boldsymbol{\Lambda}_\mathbf{m} & \boldsymbol{\Lambda}_\mathbf{mp} \\ \boldsymbol{\Lambda}_\mathbf{mp}{}^{\mathrm{T}} & \boldsymbol{\Lambda}_\mathbf{p} \end{bmatrix} \begin{bmatrix} \delta\mathbf{x}_\mathbf{m} \\ \delta\mathbf{x}_\mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{g}_\mathbf{m} \\ \mathbf{g}_\mathbf{p} \end{bmatrix}.$$

Taking advantage of this sparse structure, the system of equations is typically solved by forward-then-backward substitution, either of the *path-onto-the-map* or of the *map-onto-the-path* (Triggs et al. 2000).

Depending on the process noise and the prior, the system matrix, $\boldsymbol{\Lambda}$, can take on different sparsity patterns that affect the complexity of finding a solution. An infinite process noise covariance would mean that the motion model does not contribute information to the system, which would reduce the process block of the system matrix to block diagonal, which is $O(m + n^3)$ to solve. Similarly, without prior information (i.e. $\boldsymbol{\Pi} = 0$) the map block is also block diagonal, which is $O(m^3 + n)$ to solve. Without information from the motion model and without prior information the problem is equivalent to the bundle adjustment (BA) problem in photogrammetry, which can be solved in either $O(m^3 + n)$ or $O(m + n^3)$ (Brown 1976). It is interesting to note that in this form (no motion model, no prior), the first optimal solution using cameras appears to have been developed by Brown (1958). Brown was also the originator of what has come to be known as the Tsai camera model (Tsai 1987). When converted to a recursive least-squares framework, the computational costs mentioned above can typically be reduced to quadratic (Bar-Shalom and Fortmann 1988).

2.2. The Sliding Window Filter

For locally optimal trajectory and map estimation we employ a Sliding Window Filter (SWF), which is an approximation to the full feature-based batch non-linear least-squares SLAM problem (Sibley et al. 2006, 2007). The SWF concentrates computational resources on an accurate estimation of the spatially immediate map and trajectory from a sliding time window of the most recent sensor measurements. To keep computation tractable, old poses and landmarks that are not visible from the currently active sliding window of poses are marginalized out. After marginalization, the remaining non-linear least-squares problem is solved via a sparse Gauss–Newton method with a robust Huber-cost function.

Marginalizing out the parameters we wish to remove is equivalent to applying the *Schur complement* to the least-squares equations (Jordan 2003; Sibley 2006). For example, given the system

$$\begin{bmatrix} \mathbf{\Lambda_a} & \mathbf{\Lambda_b} \\ \mathbf{\Lambda_b}^{\mathrm{T}} & \mathbf{\Lambda_c} \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta}\mathbf{x_a} \\ \boldsymbol{\delta}\mathbf{x_b} \end{bmatrix} = \begin{bmatrix} \mathbf{g_a} \\ \mathbf{g_b} \end{bmatrix},$$

reducing the parameters $\mathbf{x_a}$ onto the parameters $\mathbf{x_b}$ gives

$$\begin{bmatrix} \mathbf{\Lambda}_a & \mathbf{\Lambda}_b \\ 0 & \mathbf{\Lambda}_c - \mathbf{\Lambda}_b^{\mathrm{T}}\mathbf{\Lambda}_a^{-1}\mathbf{\Lambda}_b \end{bmatrix} \begin{bmatrix} \mathbf{x}_a \\ \mathbf{x}_b \end{bmatrix} = \begin{bmatrix} \mathbf{g}_a \\ \mathbf{g}_b - \mathbf{\Lambda}_b^{\mathrm{T}}\mathbf{\Lambda}_a^{-1}\mathbf{g}_a \end{bmatrix},$$

where the term $\mathbf{\Lambda_b}^{\mathrm{T}}\mathbf{\Lambda_a}^{-1}\mathbf{\Lambda_b}$ is called the Schur complement of $\mathbf{\Lambda_a}$ in $\mathbf{\Lambda_b}$. After this forward substitution step, the smaller lower-right system

$$[\mathbf{\Lambda_c} - \mathbf{\Lambda_b}^{\mathrm{T}}\mathbf{\Lambda_a}^{-1}\mathbf{\Lambda_b}][\mathbf{x_b}] = [\mathbf{g_b} - \mathbf{\Lambda_b}^{\mathrm{T}}\mathbf{\Lambda_a}^{-1}\mathbf{g_a}]$$

can be solved for updates to $\mathbf{x_b}$. The SWF solves incrementally only for the smaller system, which is how it maintains constant time operation. Using back-substitution, the full system can be solved at any point, for instance, at loop closure if we desire a global solution. However, we find the global solution is more readily computed with pose-graph relaxation techniques described in Section 5, and do not use the SWF for loop closure.

2.2.1. SWF Overview

We now give a brief synopsis of the SWF algorithm.

Adding New Pose Parameters. First, after completing $m-1$ steps, the command $\mathbf{u}_m$ is used to drive the system forward via the process model, $\mathbf{x}_{\mathbf{p}_m} = f(\mathbf{x}_{\mathbf{p}_{m-1}}, \mathbf{u}_m)$, which adds six new pose parameters to $\mathbf{x_p}$. Recall that in the Gauss–Newton method the covariance matrix is approximated by the inverse of the Hessian matrix (Bell and Cathey 1993). Thus, after applying the process model but *before* incorporating any new measurements, we can use the Gauss–Newton method to compute an updated information matrix, which is simply the Hessian associated with the *maximum likelihood estimate* (MLE) solution. This operation is a linearized error propagation, affects only the process-block of the information matrix and can be computed in constant time.

Removing Parameters. Next, if there are now more than k poses active (for a k-step SWF), then the we marginalize out the oldest pose parameters using the Schur complement. If $k = 1$ then this step is algebraically equivalent to the extended Kalman filter SLAM timestep, and there is only ever a single active pose. Note that marginalizing affects the right-hand side of the system equations. In conjunction with the error propagation described above, this step transforms the state and information matrix identically to the first order discrete extended Kalman filter timestep, i.e. error propagation to a new pose followed by marginalizing old pose parameters is equivalent to the extended Kalman filter timestep. At this point, to keep the state vector size bounded, we also marginalize out invisible landmarks that are no longer visible from the active poses.

Updating Parameters. Before a complete measurement update is computed, parameters are added to $\mathbf{x_m}$ to represent any newly observed landmarks (initial values are computed via stereo), and $\mathbf{\Lambda_m}$ is extended (with zeros) appropriately. Finally, all of the measurements within the time window are used to update the least-squares solution. This step requires solving the non-linear least-squares problem, which we do via a sparse robust Gauss–Newton method.

Depending on the number of poses in the sliding window, the SWF can scale from the offline, optimal batch least-squares solution to a fast online incremental solution. For instance, if the sliding window encompasses all poses, the solution is algebraically equivalent to full SLAM; if only one time-step is maintained, the solution is algebraically equivalent to the extended Kalman filter SLAM solution (Lina María Paz et al. 2008). If robot poses and environment landmarks are slowly marginalized out over time such that the state vector ceases to grow, then the filter becomes constant time, like visual odometry (VO). The sliding window method also enables reversible data association (Bibby and Reid 2007), out-of sequence measurement updates, and robust estimation across multiple timesteps, all of which help the overall performance of our system.

This approach allows us to decouple our loop closure system from the core pose estimator, and hence concentrates computational resources on improving the local result. With high bandwidth sensors (such as cameras) focusing on the local problem is clearly important for computational reasons; this is especially true if we wish to fuse all of the sensor data (or a significant portion thereof). However, even with this local focus, once a loop closure is identified, global optimization over the sequence left behind can be a good match to the global batch solution.

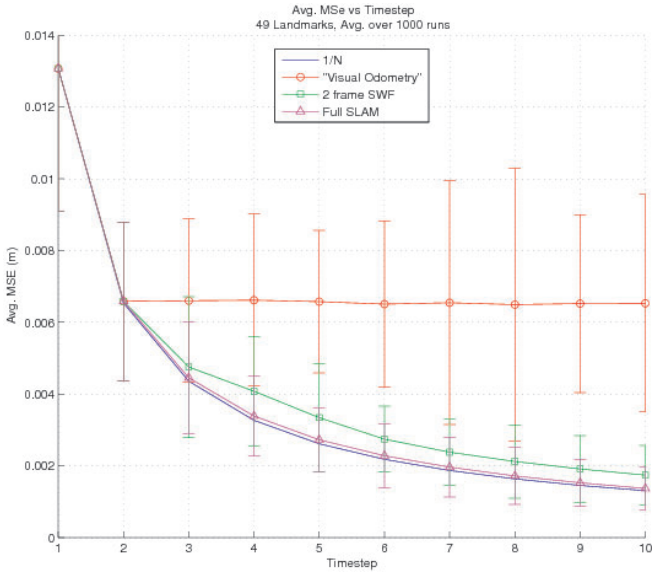


Fig. 4. The average mean squared error performance for VO compared with the batch solution, as well as the SWF solution. The SWF can be seen as strictly superior to VO with the same computational complexity as VO but with near optimal convergence.

It is interesting to note what happens if we simply delete parameters from the estimator instead of marginalizing them out. For a sliding window of size k, the error converges as $1/k$, just as we would expect the batch estimator to do. However, after k steps, the error stops converging as we delete information from the back of the filter. With such deleting and a sliding window of $k = 1$, we end up with a solution that is nearly identical to previous forms of VO (Matthies and Shafer 1987; Olson et al. 2001; Nister et al. 2004). The graph in Figure 4 shows the average mean squared error (MSE) performance for this type of VO compared with the batch solution, as well as the SWF solution. Given this insight, the SWF can be seen as strictly superior to VO: it has the same computational complexity as VO, yet it (1) shows near optimal convergence and (2) does not suffer from stationary drift. In practice, the SWF is most often used in this constant-time regime.

The SWF is an approach that can scale from exhaustive batch solutions to fast incremental solutions by tuning a time window of active parameters. If the window encompasses all time, the solution is algebraically equivalent to full SLAM; if only one time-step is maintained, the solution is algebraically equivalent to the extended Kalman filter SLAM solution. From this point on we simply refer to the case of $k = 1$ with landmark marginalization as VO.

2.3. The Provenance of the SWF

The SWF is a non-linear least-squares approach to navigation and mapping inspired by results from the photogrammetry community, dating back to the late 1950s (Brown 1958; Mikhail 1983), and later derivatives such as the variable state dimension filter (McLauchlan and Murray 1995; McLauchlan 1999), VO (Matthies and Shafer 1987; Nister et al. 2004), and of course extended Kalman filter SLAM (Smith et al. 1990). The techniques of photogrammetry were gradually adopted or rediscovered as VO and shape from motion in the computer vision community (Matthies and Shafer 1987; Triggs et al. 2000; Fitzgibbon and Zisserman 2004) and SLAM in the robotics community (Lu and Milios 1997; Thrun et al. 2005). These are all least-squares estimators, often expressing algebraically equivalent solutions.

Since the original development of the SWF (Sibley et al. 2006), some similar techniques have been developed in the computer vision literature based on BA (Engels et al. 2006; Mouragnon et al. 2006). The high frame rates achieved in Engels et al. (2006) are largely due to short feature track lengths; furthermore, the effect of marginalization and including prior information is not addressed, and it is assumed that fixing old frames is reasonable. As frames are removed and only certain keyframes are kept, the results cannot converge to the optimal batch solution. Similarly, the results of Mouragnon et al. (2006) do not include all of the data, but instead only use a selected sub-set of keyframes, and hence cannot match full SLAM. In contrast, the SWF attempts to match the full solution by rolling parameters into prior information.

Brown's photogrammetric BA is the original image-based batch maximum likelihood solution to the full SLAM problem from the iterative non-linear least-squares perspective (Brown 1958). Brown's sparse (and therefore fast) solution to BA does not include dense prior information or a process model, which can be useful for SLAM. The work by Mikhail (1983) gives an incremental/recursive algorithm that can include arbitrary functional relationships between parameters (e.g. a process model) as well as including prior information matrices. However, to facilitate faster run-times Mikhail employs the same sparse optimizations as Brown. Brown's sparse system of equations does not capture the temporal evolution of the PDF if there is prior information induced by marginalization.

GraphSLAM (Thrun et al. 2005), exactly sparse delayed state filters (ESDSFs) (Eustice et al. 2005), smoothing and mapping (SAM) (Dellaert and Kaess 2006), and recent work of Konolige and Agrawal (2007) are all examples of non-linear least-squares techniques similar to BA. SAM solves the system equations efficiently by variable re-ordering, which is also a well-known technique in photogrammetry (Triggs et al. 2000). The success of this approach depends critically on the structure of the least-squares system matrix, which generally cannot be known beforehand since it depends on how the robot goes about observing the world. General re-ordering algorithms that

are optimal for arbitrary system equations are known to be NP-complete (Yannakakis 1981). GraphSLAM is an offline solution and is typically tackled with available numerical sparse solvers.

Both GraphSLAM and ESDSFs factor the map onto the path, thereby producing a "pose-graph", which can then be solved for the optimal robot trajectory. Fast pose-graph optimization methods are a recent development (Frese and Duckett 2003; Olson et al. 2006; Grisetti et al. 2007). By finding the maximum likelihood configuration of a sequence of interrelated poses, these approaches can solve impressively large problems. Note, however, that pose-graph methods do not compute an optimal structure estimate and instead focus on computing the optimal vehicle trajectory.

ESDSFs are a *view-based* approach inspired by both the Variable State Dimension Filter (VSDF; see later) and Sparse Extended Information Filters (Thrun et al. 2002; Eustice et al. 2005). ESDSFs are efficient approximations to the full SLAM solution, although they rely on view-matching raw data, so the assumption of independent measurement noise in the sensor model may be violated: an eye must be kept on the "double counting data" issue.

In some sense, SWFs are the opposite of GraphSLAM and delayed state filters: where these methods factor the map onto the path, the SWF slowly factors the path onto the map. This has important implications for the run-time complexity as the algorithm progresses. In GraphSLAM, as the map is factored onto the path, the induced structure in the path block, $\mathbf{\Lambda_p}$, can grow to be arbitrarily complex. This stems from the fact that there are an infinite variety of paths through an environment, and usually we will not know how the robot is going to move beforehand. On the other hand, marginalizing the path onto the map only ever induces a structure with a *bounded complexity* as there is a limited number of landmark-to-landmark conditional dependencies induced. Fundamentally, while there is an infinite variety of paths through the environment, there is just one environment. This point is a crucial distinction between methods that factor onto the path and methods that do not.

The VSDF (McLauchlan and Murray 1995; McLauchlan 1999) combines the benefits of batch least squares with those of recursive estimation. Interestingly, both the SWF and the VSDF are very similar to Mikhail's "Unified Adjustment" technique (Mikhail 1983). Mikhail's work is a general and complete treatment of least-squares adjustment, whereas the SWF and VSDF are specific examples applied to SLAM and structure from motion (SFM). The VSDF is a mixed formulation, taking inspiration from the sparse Levenberg–Marquardt method used in BA (More 1978; Hartley and Zisserman 2000), and also from the traditional extended Kalman filter used in SLAM (Smith et al. 1990). For computational efficiency, the VSDF ignores conditional dependencies that are induced from marginalizing out old parameters, and, similarly to Brown's BA, it also ignores conditional dependencies that exist between adjacent pose parameters, especially the block tridiagonal matrix structure of the process block. In comparison, the least-squares formulation for full SLAM captures this information naturally. Neglecting conditional dependencies can be detrimental; in SLAM it will lead to divergence (Newman 1999).

The recent work of Deans (2005) is also inspired by the least-squares approach, and similarly to VSDF and SWF aims at online implementation by focusing the computation on the most recent set of measurements by removing parameters from consideration. However, instead of incrementally marginalizing the solution pose by pose, the formulation breaks the problem into sets of adjacent batch problems.

2.4. Feature Selection and Matching: The Image Processing Front-end

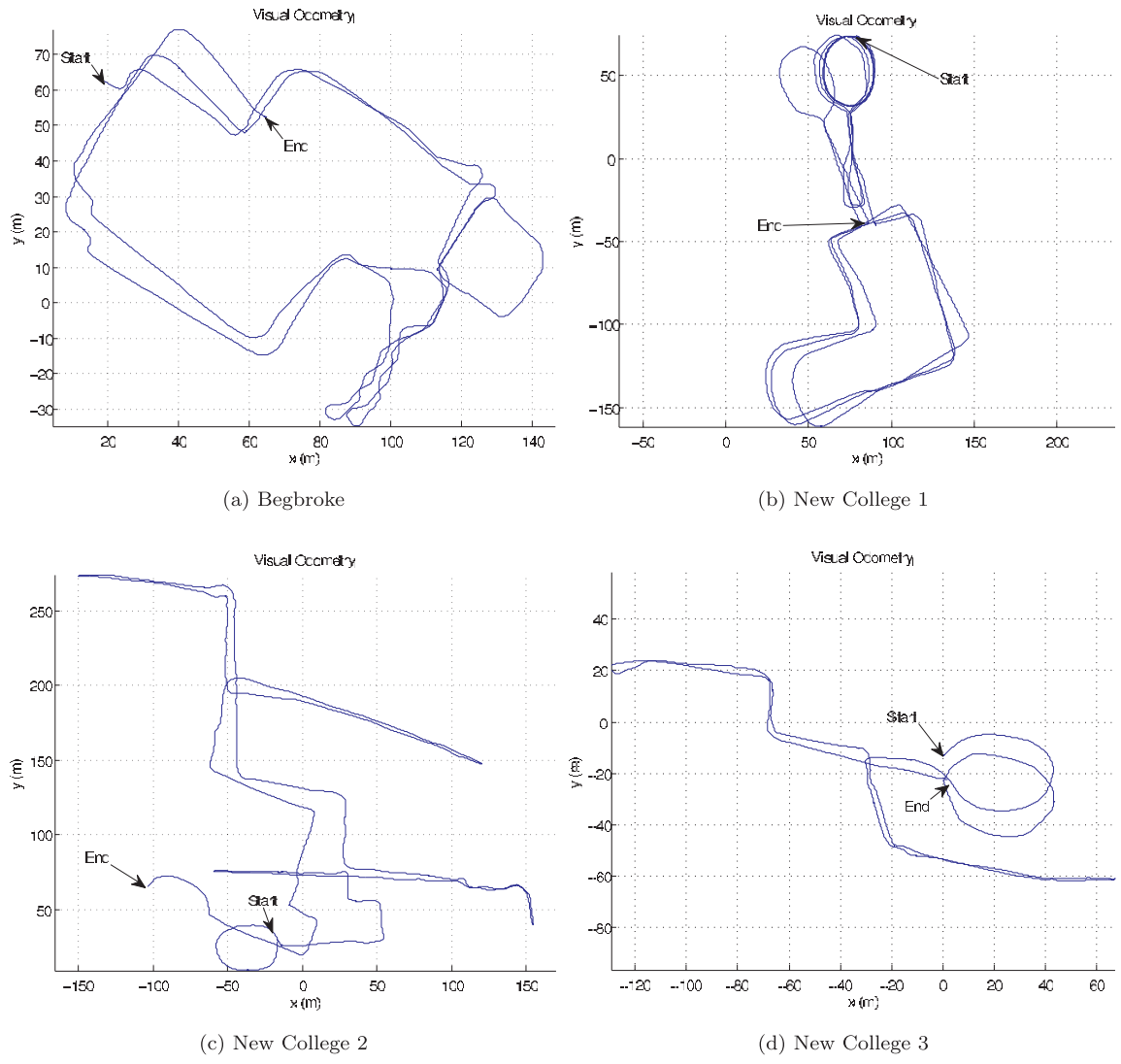
This section describes the underlying image processing for a feature-based visual tracker essential for tracking features between stereo frames, and is joint work with Mei and Reid of the Active Vision Lab at Oxford. The steps have similarities with other works in the field, e.g. Eade and Drummond, but here are adapted to the processing of stereo images. We begin with a top-level view. For each incoming frame, the following steps are undertaken.

(i) *Feature extraction.* The features used in this work are provided by the FAST corner extractor (Rosten and Drummond 2005). This extractor provides good repeatability at a small computational cost. FAST corners are extracted at different "pyramid levels" (scales). The pyramid provides robustness to motion blur and enables point matching in larger regions of the image.

(ii) *Pose initialization.* To provide robustness to strong inter-frame rotation, a sum of squared difference (SSD) gradient descent algorithm (Mei et al. 2008), applied at the highest pyramid level, is used to estimate the 3D rotation between two time-steps. The assumption of pure rotation is valid if the inter-frame translation is small with respect to the landmark depths and at 20 Hz frame rate this is indeed the case.

(iii) *Temporal feature matching.* The 3D landmarks (the map) are projected alternatively into the left and right images and matched in a fixed-sized window to the extracted FAST corners using mean SAD (sum of absolute difference with the mean removed for better resilience to lighting changes). A maximal accepted score is set to provide a first pass robustness to outliers. Point correspondences between image pairs are obtained by a scan line search in the already rectified images.

(iv) *Localization.* After the map points have been matched, a localization step minimizes the 6-DOF of the camera pose using m-estimators for robustness. After the



(a) Begbroke

(b) New College 1

(c) New College 2

(d) New College 3

Fig. 5. VO results for the four datasets detailed in Tables 2 and 3.

minimization, the landmark measurements with strong reprojection errors are removed from the system. This step proved important to enable early removal of outliers and the possibility of adding new, more stable landmarks.

(v) *Left–right matching.* To achieve a high frame rate with good accuracy around 50–100 features are tracked at each time-step. The feature selection process follows the assumption that we desire distinctive features with a uniform distribution in the image (irrespective of the underlying tracking uncertainty). A quadtree is used to represent the distribution of the measurements at each time-step. It contains the number of measurements in the image and the maximal amount of points allowed in the different parts of the image to ensure a uniform distribution of features. It is used in the following way.

(a) During temporal matching, the matched map points are inserted into the quadtree according to their measurement image locations.

(b) To add new features, FAST corners are extracted from the left and right images and ordered by a distinctiveness score (in this work we used Harris scores). To decide which features to add, the best features are taken in order and their image location is checked in the quadtree to ensure the maximal amount of allowed points has not been exceeded.

Table 2. VO Results for Begbroke and New College 1 Datasets

	Begbroke			New College 1		
	Average	Minimum	Maximum	Average	Minimum	Maximum
Distance travelled (km)	—	—	1.08	—	—	2.26
Frames processed	—	—	23,268	—	—	51,263
Velocity (m s^{-1})	0.93	0.00	1.47	0.94	9.46×10^{-4}	1.53
Angular velocity ($^\circ$ s^{-1})	9.49	0.0	75.22	7.08	4.12×10^{-3}	69.00
Frames per second	22.2	10.6	31.4	20.6	10.3	30.0
Features per frame	93	44	143	95	37	142
Feature track length	13.42	2	701	11.59	2	717
Reprojection error	0.17	2.74×10^{-3}	0.55	0.13	0.03	1.01

Table 3. VO Results for New College 2 and New College 3 Datasets.

	New College 2			New College 3		
	Average	Minimum	Maximum	Average	Minimum	Maximum
Distance travelled (km)	—	—	2.05	—	—	0.82
Frames processed	—	—	49,114	—	—	29,489
Velocity (m s^{-1})	0.83	4.55×10^{-4}	3.05	0.56	1.63×10^{-4}	1.26
Angular velocity ($^\circ$ s^{-1})	7.13	8.23×10^{-3}	62.56	4.83	5.24×10^{-3}	59.75
Frames per second	21.5	7.4	29.8	20.3	7.4	28.6
Features per frame	91	45	142	93	49	146
Feature track length	14.43	2	622	27.76	2	1363
Reprojection error	0.12	0.028	0.91	0.10	0.024	0.29

If it passes the test, the corresponding point in the other stereo pair is searched along the same scanline.

2.5. Visual Odometry Results

We present results from two venues, "Begbroke" and "New College", the latter taken over multiple days. The datasets are summarized in Tables 2 and 3 and the estimated trajectories are shown in Figures 5(a)–5(d).

3. Closing Loops with FAB-MAP

Loop closure detection is a well-known difficulty for metric SLAM systems. Our system employs an appearance-based approach to detect loop closure, i.e. using sensory similarity to determine when the robot is revisiting a previously mapped area. Loop closure cues based on sensory similarity are independent of the robot's estimated position, and so are robust even in situations where there is significant error in the metric position estimate, for example after traversing a large loop where turning angles have been estimated poorly.

Our approach, FAB-MAP, previously described in Cummins and Newman (2007, 2008b,a, 2009), is based on a probabilistic notion of similarity and incorporates a generative model for typical place appearance which allows the system to correctly assign loop closure probability to observations even in environments where many places have similar sensory appearance, a problem known as perceptual aliasing.

Appearance is represented using the bag-of-words model developed for image retrieval systems in the computer vision community (Sivic and Zisserman 2003; Nister and Stewenius 2006) which has recently been applied to mobile robotics for loop closure detection by several authors (Filliat 2007; Angeli et al. 2008). More generally, appearance has been used in loop closure detection and localization tasks by many authors (Kröse et al. 2001; Lamon et al. 2001; Wang et al. 2005; Wolf et al. 2005; Chen and Wang 2006; Schindler et al. 2007). At time k, our appearance map consists of a set of n_k discrete locations, each location being described by a distribution over

which appearance words are likely to be observed there. Incoming sensory data is converted into a bag-of-words representation; for each location, we can then ask how likely it is that the observation came from that location's distribution. We also find an expression for the probability that the observation came from a place not in the map. This yields a PDF over location, which we can use to make a data association decision and either create a new place model or update our belief about the appearance of an existing place. Essentially, this is a SLAM algorithm in the space of appearance, which runs parallel to our metric SLAM system.

3.1. A Bayesian Formulation of Location from Appearance

Calculating position, given an observation of local appearance, can be formulated as a recursive Bayes estimation problem. If L_i denotes a location, Z_k the kth observation and $\mathcal{Z}^k$ all observations up to time k, then

$$p(L_i|\mathcal{Z}^k) = \frac{p(Z_k|L_i, \mathcal{Z}^{k-1})p(L_i|\mathcal{Z}^{k-1})}{p(Z_k|\mathcal{Z}^{k-1})}. \tag{4}$$

Here $p(L_i|\mathcal{Z}^{k-1})$ is our prior belief about our location, $p(Z_k|L_i, \mathcal{Z}^{k-1})$ is the observation likelihood, and $p(Z_k|\mathcal{Z}^{k-1})$ is a normalizing term. An observation Z is a binary vector, the ith entry of which indicates whether or not the ith word of the visual vocabulary was detected in the current scene. The key term here is the observation likelihood, $p(Z_k|L_i, \mathcal{Z}^{k-1})$, which specifies how likely each place in our map was to have generated the current observation. Assuming current and past observations are conditionally independent given location, this can be expanded as

$$\begin{aligned} p(Z_k|L_i) \;=\; & p(z_n|z_1, z_2, \ldots, z_{n-1}, L_i)\ldots \\ & p(z_2|z_1, L_i)p(z_1|L_i). \end{aligned} \tag{5}$$

This expression cannot be evaluated directly because of the intractability of learning the high-order conditional dependencies between appearance words. The simplest solution is to use a naive Bayes approximation; however, we have found that results improve considerably if we instead employ a Chow Liu approximation (Chow and Liu 1968) which captures more of the conditional dependencies between appearance words. The Chow Liu algorithm locates a tree-structured Bayesian network that approximates the true joint distribution over the appearance words. The approximation is guaranteed to be optimal within the space of tree-structured networks. For details of the expansion of $p(Z_k|L_i)$ using the Chow Liu approximation we refer readers to Cummins and Newman (2007).

3.2. Loop Closure or New Place?

One of the most significant challenges for appearance-based loop closure detection is calculating the probability that the current observation comes from a place not already in the map. This is particularly difficult due to the repetitive nature of many real-world environments: a new place may look very similar to a somewhere visited previously. While many appearance-based localization systems exist, this extension beyond pure localization makes the problem considerably more difficult (Gutmann and Konolige November 1999). The key is a correct calculation of the denominator of Equation (4), $p(Z_k|\mathcal{Z}^{k-1})$. If we divide the world into the set of mapped places M and the unmapped places $\overline{M}$, then

$$\begin{aligned} p(Z_k|\mathcal{Z}^{k-1}) = & \sum_{m\in M} p(Z_k|L_m)p(L_m|\mathcal{Z}^{k-1}) \\ & + \sum_{u\in\overline{M}} p(Z_k|L_u)p(L_u|\mathcal{Z}^{k-1}), \end{aligned} \tag{6}$$

where we have applied our assumption that observations are conditionally independent given location. The first summation is simply the likelihood of all of the observations for all places in the map. The second summation is the likelihood of the observation for all possible unmapped places. Clearly we cannot compute this term directly because the second summation is effectively infinite. We have investigated a number of approximations. A mean field-based approximation has reasonable results and can be computed very quickly; however, we have found that a sampling-based approach yields the best results. If we have a large set of randomly collected place models L_u (readily available from previous runs of the robot), then we can approximate the term by

$$\begin{aligned} p(Z_k|\mathcal{Z}^{k-1}) \approx & \sum_{m\in M} p(Z_k|L_m)p(L_m|\mathcal{Z}^{k-1}) \\ & + p(L_{\text{new}}|\mathcal{Z}^{k-1}) \sum_{u=1}^{n_s} \frac{p(Z_k|L_u)}{n_s}, \end{aligned} \tag{7}$$

where n_s is the number of samples used, $p(L_{\text{new}}|\mathcal{Z}^{k-1})$ is our prior probability of being at a new place, and the prior probability of each sampled place model L_u with respect to our history of observations is assumed to be uniform. Note here that in our experiments the places L_u do not come from the current workspace of the robot, rather they come from previous runs of the robot in different locations. They hold no specific information about the current workspace but rather capture the probability of certain generic repeating features, such as foliage and brickwork. Figures 6 and 7 show typical loop closure results obtained using our method. Note the high degree of confidence despite marked changes in scene and lighting. Figure 8 shows the compute time per new image added as a function of topological map size. Note that these results are generated with a FAB-MAP implementation described by Cummins and Newman (2008a) and much faster compute times are reported in Cummins and Newman (2009).

Fig. 6. Place recognition results generated by FAB-MAP. Probability of loop closure is calculated to be 0.9986. (Note that a stitched panorama view is shown here; the algorithm is applied directly to the unstitched frames.)

Fig. 7. Example place recognition result generated by FAB-MAP under markedly different lighting conditions. Probability of loop closure is calculated to be 0.9519.

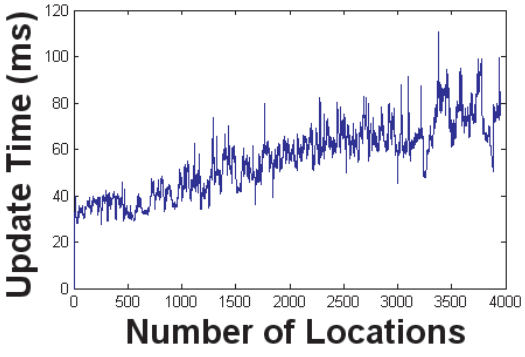


Fig. 8. Inference time for FAB-MAP. Generating the SURF features adds a fixed overhead of 716 ms on average. The mean inference time is 56 ms, so the total mean processing time per panoramic image is 772 ms. The robot collects a panoramic image on average every 1.7 seconds, so this is faster than real time.

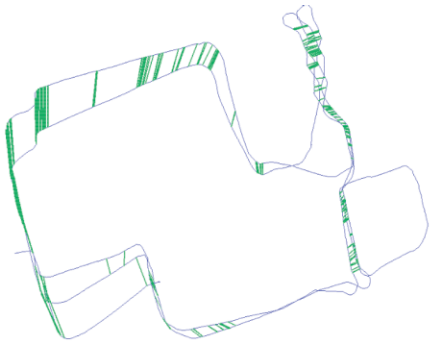
In this paper we have used a Ladybug panoramic camera because the 360° views it provides allow loop closure detections when revisiting a place in the opposite direction. However, there is nothing about our system that explicitly requires 360° views. Indeed, we could (and have) use the relatively narrow field of view images from the stereo pair but we would expect an increase in the false negative rate.

4. Upgrading from Topological Loop Closures to Metric Constraints

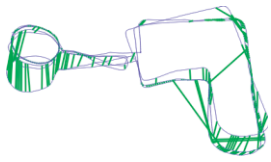
The FAB-MAP algorithm takes a collection of images as input (each image in our case is a five-image panorama taken from a Ladybug camera). Images are presented sequentially and at each time-step the algorithm returns a $(N + 1)$ bin PDF over places (images) representing the probability that the latest image corresponds to each of N previous places (images) or a "new place". This allows us to generate topological loop closure notification when the probability of a match becomes substantial. The precision-recall and spatial regularity of the detected loop closures is shown in Figures 10 and 11. There is a marked difference in recall performance between the Begbroke and New College runs. The Begroke sequence was well lit and diverse in appearance. In contrast, the New College dataset (Smith et al. 2009) is far more challenging containing marked changes in lighting and many opportunities for spatial aliasing (false positives) something which FAB-MAP is designed to be resistant to. Note however that for both datasets one in two poses are within 2 m of a correctly identified loop closure constraint.

Loop closures are detected using a multi-view camera giving 360° of view. They take the form of a tuple $\langle t_a, t_b \rangle$ where t_a and t_b are two times at which the vehicle appeared to be in the same place. We refer to a and b as "loop closure ends". Figure 9(a) illustrates the distribution of loop closures detected on the Begbroke dataset. Only loop closures with a 99% probability are indicated.

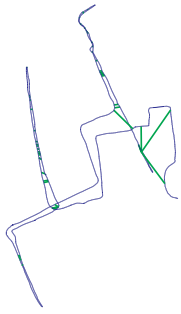
The question now is how does one apply this loop closure constraint to our metric VO derived trajectory. For any loop closure $\langle t_a, t_b \rangle$ we require a metric parameterization of the 6-DOF transformation ${}^{a}T_{b}$ between the poses of the vehicle at the times t_a and t_b. We currently use two options: pose recovery from two pairs of stereo images and laser point cloud matching.



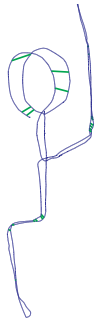
(a) Begbroke



(b) New College 1

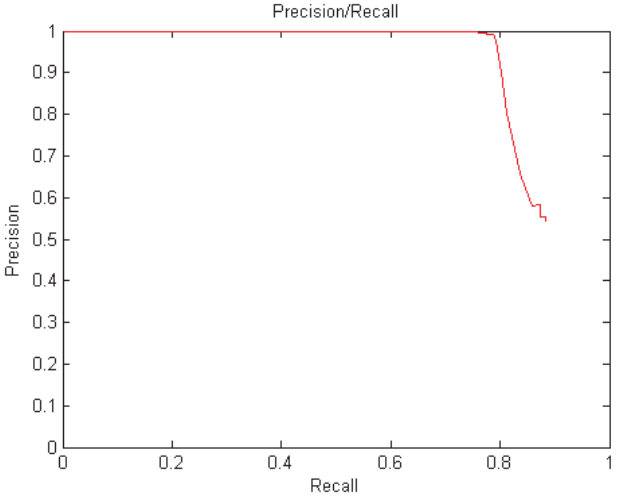


(c) New College 2

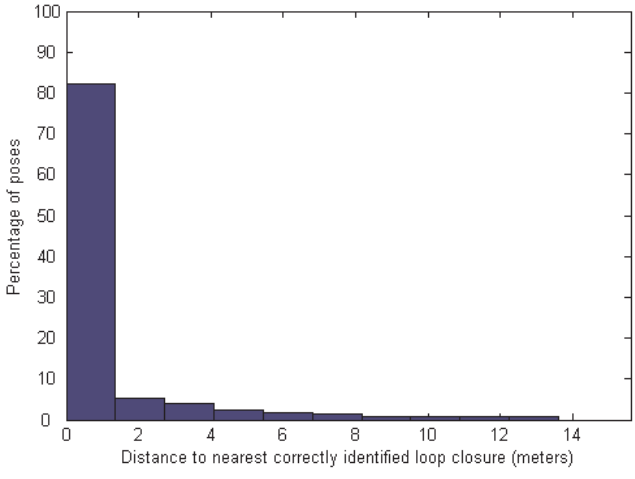


(d) New College 3

Fig. 9. VO results for the four datasets detailed in Tables 2 and 3 with detected loop closures shown in green. Only loop closures with a 99% probability are indicated. Note that in contrast to the Begbroke dataset where lighting was ideal, there are false positives in the processing of the New College 1 dataset, which must be removed with geometric consistency checks.

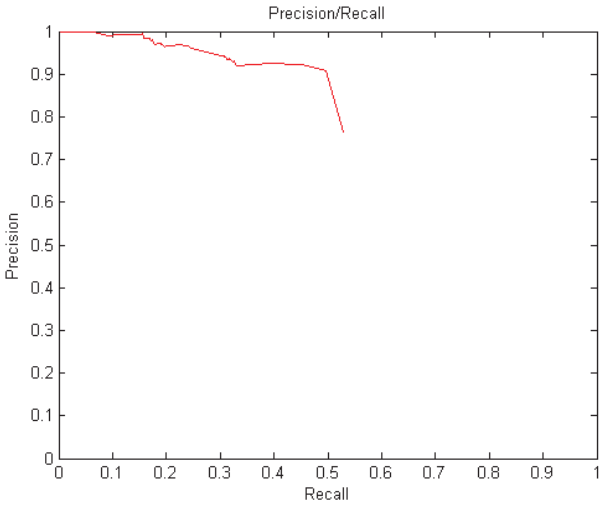


(a) Precision-recall

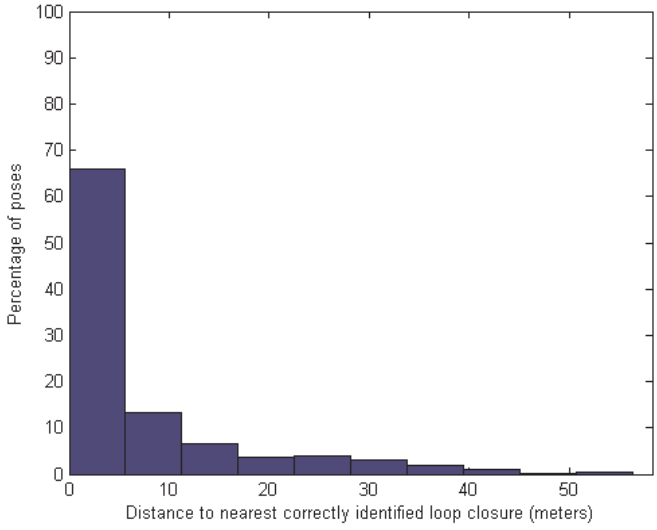


(b) Spatial regularity

Fig. 10. Quality evaluation of the FAB-MAP loop closure detections. The precision-recall curve for FAB-MAP loop closure detection for the Begbroke dataset is shown in (a); 74% of possible loop closures are detected correctly, without false positives. The spatial distribution of the loop closure detections is shown in (b). For parts of the trajectory where loop-closing occurs (defined as the paths being within 7.5 m), 85% of poses are either detected as loop closures or are within 2 m of a detected loop closure.



(a) Precision-recall



(b) Spatial regularity

Fig. 11. Quality evaluation of the FAB-MAP loop closure detections. The precision-recall curve for FAB-MAP loop closure detection for the New College 1 dataset is shown in (a); 16% of possible loop closures are detected correctly, with 99.5% precision. The spatial distribution of the loop closure detections is shown in (b). For parts of the trajectory where loop-closing occurs (defined as the paths being within 7.5 m), 50% of poses are either detected as loop closures or are within 2 m of a detected loop closure.

4.1. Pose Recovery from Stereo Pairs

A version of our VO front-end is used to verify loop-closures from FAB-MAP. The approach described in Section 2.4 is used to select 500 well-distributed image points. Scale-invariant feature transform (SIFT) descriptors are then extracted (with scale provided by depth from stereo), and n-to-n matching is performed between left images to establish temporal correspondence. RANSAC is used to find the initial transformation between frames (with three 3D points used to produce potential models). The final RANSAC estimate is then used to seed a Gauss–Newton MLE estimate with a Huber kernel for further robustness. Typical stereo loop closure images are shown in Figures 12 and 13. Estimates that have more than 50 matches and a reprojection error less than 0.2 pixels are kept as valid. These uncertain loop-closure transforms are used during pose graph relaxation as described in Section 5. Figure 14 shows an interesting and important case in which the FAB-MAP algorithm gives a false positive which is caught by this visual geometry test.

4.2. Pose Recovery via Point Cloud Matching

Recovering the relative pose from stereo yields excellent results, however it cannot be run on all loop closures. It is not always the case that loop closures bind points in the vehicle's trajectory in which the vehicle is travelling in the same direction, for example the first pass through a region may have been a north–south traversal, while the second is south–north. The FAB-MAP loop closure is insensitive to changes in the direction of travel, it considers all the visual words seen in a 360° panorama, but the two views from the stereo rig are wildly different and there is little hope of finding an alignment between the two poses. In these cases we resort to using iterative closest point (ICP) (Besl and McKay 1992) between two point clouds generated from short (a few seconds) segments of the vehicle's motion around each end of the loop closure.

ICP is not guaranteed to converge, especially if the initial guessed alignment between the point clouds is in gross error (often the case with loop closures). A technique capable of matching 2D point clouds under such conditions was proposed by Bosse and Zlot (2008) and it is our intention to extend this to the 3D case which we need here. However, for the results given in this paper we implemented a simple (conservative) threshold-based classifier capable of rejecting incorrect alignments based on the final absolute residual norm, inlier to outlier ratio and rate of change of residual norm over the optimization. Figures 15(a) and 15(b) show the effect of scene shape on the outcome of the ICP alignment. Convergence problems with ICP are well known and we do not dwell more on them here. However, were it not for an ICP fall back, we would not be able to deduce the metric loop closures in the New College 1 dataset.

Before moving on to discussing how metric loop closure measurements are used, Figure 16 shows the loop closures which were upgraded from topological to metric form by both

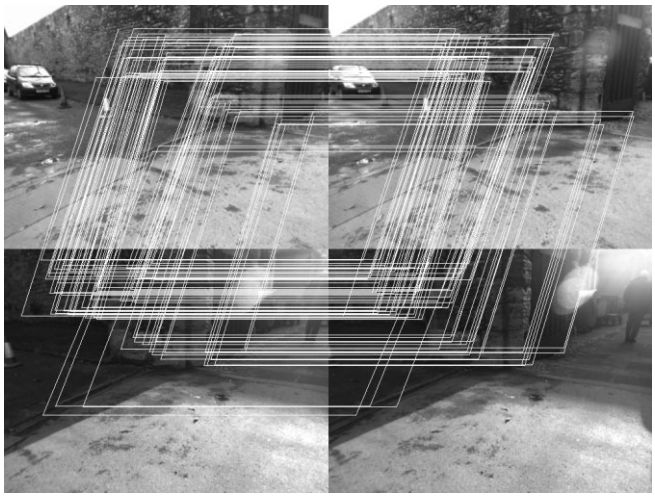


Fig. 12. Feature correspondences at loop closure are found and verified by relative stereo pose estimation. Loop closures presented by FAB-MAP must pass a geometric check: we typically require 50 correspondences with an average reprojection error less than 0.2 pixels before we accept the metric loop closure measurement as valid. Note that this does not mean the output of the FAB-MAP process is false, just that there are not enough geometric features to upgrade from a topological to metric constraint. Such inferred uncertain relative poses are used in the pose-relaxation technique described in Section 5. The figure shows the intra-pair matches (left, right) and the inter-pose matches (top, bottom).

Fig. 13. A FAB-MAP true positive rejected by stereo registration due to lack of correspondences. It is possible to generate more matches; here we chose to err on the conservative side when it comes to computing metric information from loop closure notifications: incorrect loop closures are dire.

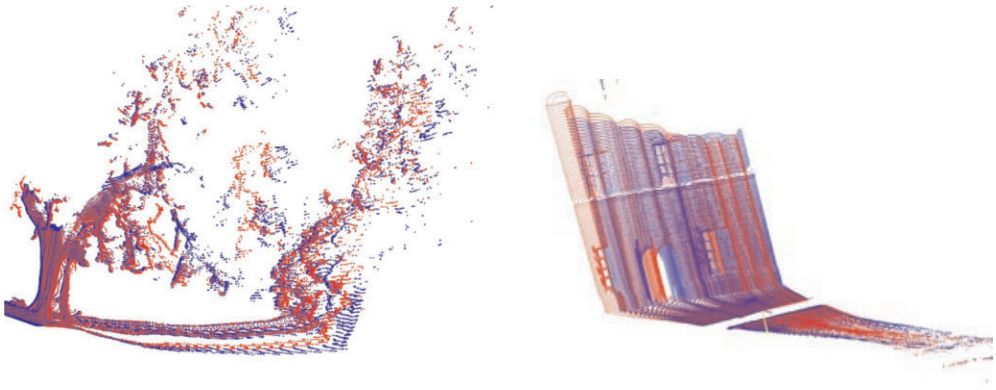
Fig. 14. FAB-MAP false positive rejected by stereo registration due to lack of correspondences. The two scenes are clearly not identical although they do share a common appearance.

stereo and ICP. Note that point cloud matching was only invoked for the cases in which the stereo method failed, generally because of a reverse traverse.

5. Pose Graph Relaxation

The VO subsystem produces a chain of 6-DOF vehicle poses linked by relative transformations which should be thought of as uncertain metric constraints. The combination of the FAB-MAP output and metric pose recovery methods just described provides additional constraints between poses, resulting in a graph of vehicle poses. Figure 17 illustrates the structure of a typical pose graph.

We wish to "relax" this graph, perturbing the edges to accommodate, in a minimum error sense, the metric information in both VO and loop closure constraints. Several authors have examined methods for pose graph relaxation in recent years, e.g. Thrun and Montemerlo (2006) and Grisetti et al. (2007). The particular size and structure of our graphs motivated us to use classical non-linear optimization techniques taking care at implementation time to make full use of the sparse properties of the problem. We note with reference to Figure 17 that the VO system produces a chain of relative transformations (and poses) through the center of the graph. This chain corresponds to the vehicle's smooth trajectory through the workspace. Loop closure constraints pinch this chain together via single edges between disparate poses. We chose to optimize not over the set of poses in the graph but rather over the relative poses between them. Define $\mathcal{V} = \{v_1, v_2, \ldots\}$ to be the set of inter-pose transformations along the trajectory chain such that v_i is the



(a) Good ICP alignment (b) Failed ICP alignment

Fig. 15. Scenes with rich geometry commonly lead to excellent 6-DOF alignment but when presented with largely flat scenes ICP commonly converges to a local minima. Here the boughs of the tree (final loop around the ground of the New College 1 dataset) provide a well-defined minima and an excellent match between two point clouds (red and blue). In the case of a facade of a building the alignment has snapped to an incorrect alignment, understandable as a spatial aliasing problem.

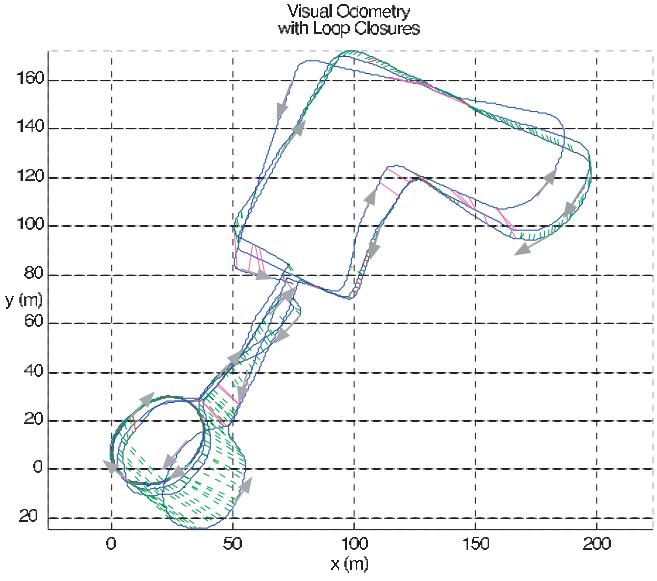


Fig. 16. The successful stereo-metric and laser ICP-metric loop closures that survived the geometric verification stages. Stereo successes are in dashed green and "fall back" laser cases are in solid purple. The direction of travel of the vehicle has been indicated with arrows.

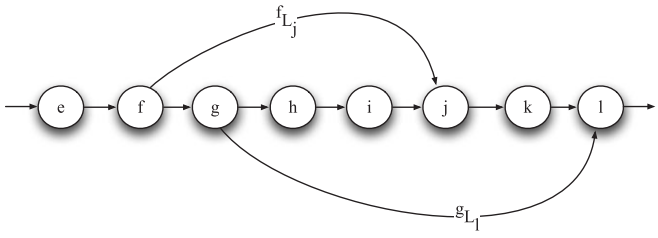


Fig. 17. A section of a typical pose graph. Poses (e, f...l) are denoted as nodes (circles) and edges are relative transformations. There is a chain of relative transformations flowing through the graph created by the VO system. Loop closure transformations ${}^{i}L_{j}$ are single edges linking disparate nodes (i and j) of this chain.

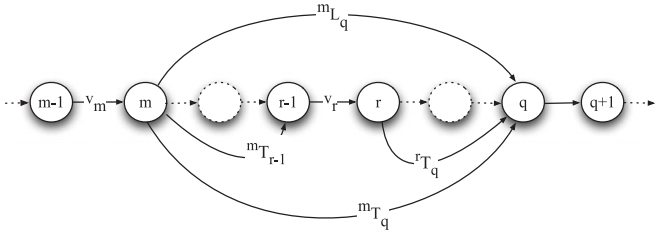


Fig. 18. A section of pose graph showing a loop closure between vehicle poses m and q and a state of interest v_r. Note that the pose graph optimization is over transformations between vehicle poses and not the poses themselves. The dotted circles represent an arbitrary number of poses.

transformation between pose $i-1$ and pose i. Furthermore, define $V = [v_1^{\mathrm{T}}, v_2^{\mathrm{T}}, \ldots]^{\mathrm{T}}$ to be a stacked vector of parameterizations of these relative transformations, this will be our state vector which we wish to optimize.

Consider now Figure 18 which shows a loop closure constraint between two poses m and q. We note that the transformation, ${}^{m}T_{q}$ between two poses m and q is simply the

integration of all of the individual transformations between poses:

$$ {}^{m}T_q = v_{m+1} \oplus v_{m+2} \oplus \cdots \oplus v_q, \tag{8} $$

where $\oplus$ denotes the transformation composition operator. This then constitutes a prediction of the loop closure constraint ${}^{m}L_q$ and $\|{}^{m}L_q - {}^{m}T_q\|^2$ is a measure of the compatibility of the graph edges with the loop closure measurements. More generally, if we have a set of n loop closures $\mathcal{L} = \{L_1 \cdots L_n\}$ where L_i is between pose $a(i)$ and $b(i)$ (a and b are look up functions), and m interpose VO measurements $\mathcal{VO} = \{vo_1 \cdots vo_m\}$, then the cost metric we wish to impose on the whole graph and then minimize is

$$ C(V|\mathcal{L}, \mathcal{VO}) = \sum_{i}^{n} \|L_i - {}^{a(i)}T_{b(i)}\|^2 + \sum_{i}^{m} \|vo_i - v_i\|^2 \tag{9} $$

where we note that the prediction ${}^{a(i)}T_{b(i)}$ is itself a function of V. The quadratic cost function in Equation (9) is well suited to classical non-linear minimization techniques. Many of these techniques require the calculation of the derivative of the measurement prediction with respect to the state vector being optimized. We now consider the form of this derivative.

Consider again Figure 18 which shows one loop closure between pose m and pose q. We can write an incremental change in the prediction of ${}^{m}T_q$ as

$$ \delta\, {}^{m}T_q = \sum_{r=m+1}^{q} \frac{\partial\, {}^{m}T_q}{\partial\, v_r}\, \delta v_r, \tag{10} $$

where δv_r is an incremental change in the rth component of the state vector V: the relative transformation between pose $r-1$ and pose r. Considering the partial derivative in the summation and substituting Equation (8) we have

$$ \frac{\partial\, {}^{m}T_q}{\partial\, v_r} = \frac{\partial \left\{ v_{m+1} \oplus v_{m+2} \oplus \cdots \oplus v_q \right\}}{\partial v_r} \tag{11} $$

$$ = \frac{\partial \left\{ {}^{m}T_{r-1} \oplus v_r \oplus^{r} T_q \right\}}{\partial v_r}, \tag{12} $$

where ${}^{m}T_{r-1}$ and ${}^{r}T_q$ are rigid kinematic chains. This allows us to write via the chain rule

$$ \frac{\partial\, {}^{m}T_q}{\partial\, v_r} = \mathcal{J}_1({}^{m}T_{r-1} \oplus v_r, {}^{r}T_q)\mathcal{J}_2({}^{m}T_{r-1}, v_r), \tag{13} $$

where

$$ \mathcal{J}_1(x, y) = \frac{\partial x \oplus y}{\partial x}, \tag{14} $$

$$ \mathcal{J}_2(x, y) = \frac{\partial x \oplus y}{\partial y}, \tag{15} $$

are the Jacobians of the composition operator $\oplus$ for arbitrary transformations x and y.

Equation (10) can be written in matrix form

$$ \delta\, {}^{m}T_q = \mathbf{h}_{m,q}\delta V, \tag{16} $$

where δV is a vector of small changes in V and $\mathbf{h}$ is a row-matrix where the kth sub block ($m < k < q$) is given by Equation (13) and zero for all k outside this range. Writing the error between predicted transformation ${}^{m}T_q$ and the measured value of the loop closure ${}^{m}L_q$ as $\delta^{m}L_q$ we seek a change in V, δV, such that

$$ \mathbf{h}_{m,q}\delta V = \delta^{m}L_q. \tag{17} $$

If we have n loop closure constraints we will have n such constraints to fulfill each in the form of Equation (17) yielding

$$ \mathbf{H}\delta V = \delta L \tag{18} $$

where δL is a stacked vector of loop closure measurements. As it stands this system of equations is almost certainly under-constrained; there will typically be many fewer loop closures than poses (we typically drop a pose every 50 ms). The system is made to be observable by adding in the visual odometry measurements between poses such that the complete problem becomes

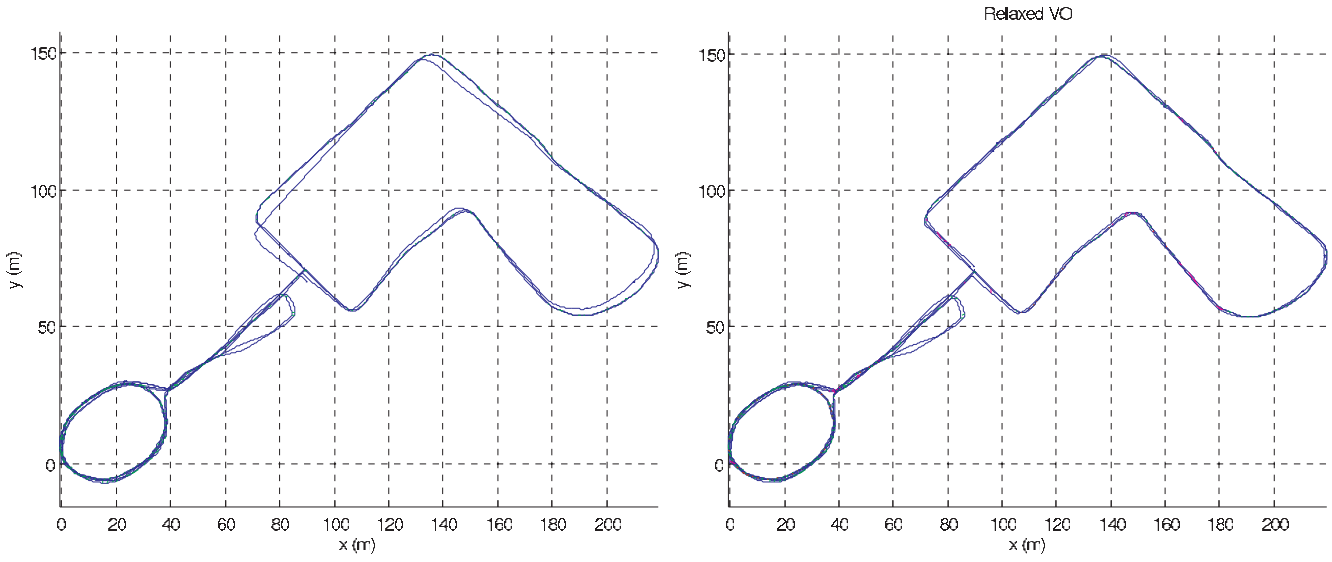
$$ \begin{bmatrix} \mathbf{H} \\ \mathbf{I} \end{bmatrix} \delta V = \begin{bmatrix} \delta L \\ Z \end{bmatrix} \tag{19} $$

where $Z = [vo_1^{\mathrm{T}}, vo_2^{\mathrm{T}}, \ldots]^{\mathrm{T}}$ is a stacked vector of visual odometry measurements between poses. This linear form can then be solved swiftly using standard techniques (we use preconditioned conjugate gradient because $\left[\mathbf{H}^{\mathrm{T}}\mathbf{I}\right]^{\mathrm{T}}$ is large and we do not wish to create or store it in memory) to yield incremental adjustments in the pose graph's edges. Optimization ceases when the perturbations in V become small.

Figure 19(a) shows the results of applying our relaxation approach to the trajectory shown in Figure 5(b) using only stereo metric constraints. The final loop around the grounds was made in the opposite direction to those that came before and so no FAB-MAP loop closures could be upgraded metrically. Figure 19(b) shows the advantages of being able to fall back on laser-based ICP matching. Where no stereo metric constraints could be found, point clouds rendered from the VO trajectory are matched in 6 DOFs and used to constrain the pose graph. Figures 20(a) and 20(b) show relaxed trajectories for the New College 2 and New College 3 datasets.

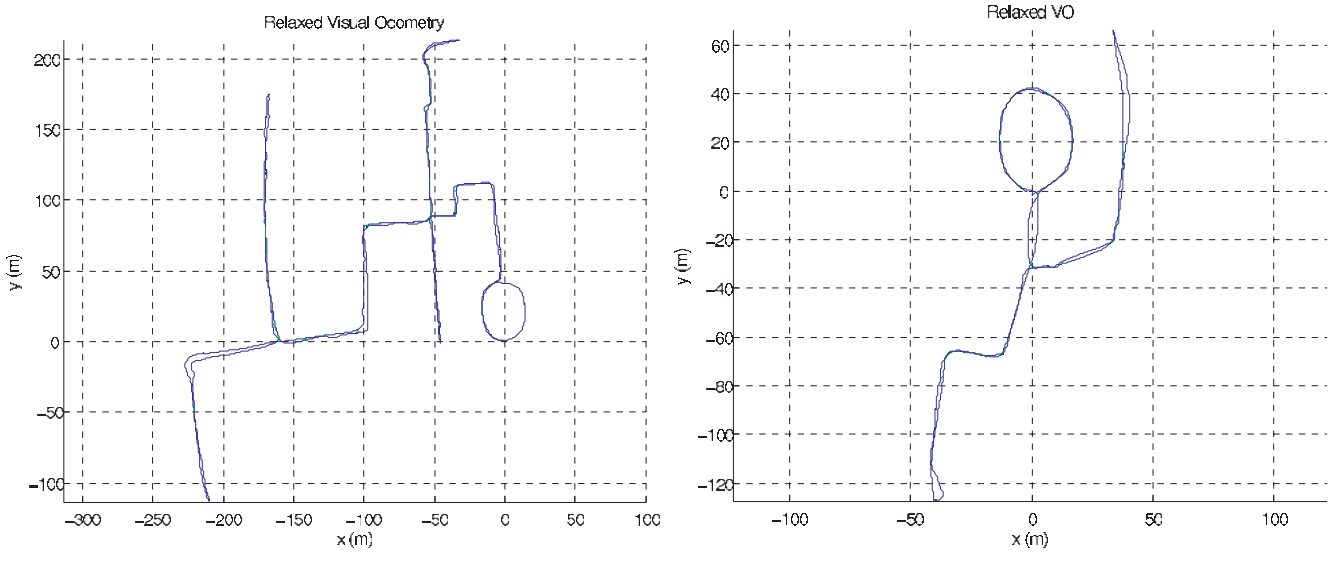
6. Map Generation and Quality Assessment

The trajectory estimation described in this paper is entirely vision-based (apart from cases where we need to fall back to ICP registration to infer loop closure geometry; see Section 4.2). We map the 3D structure of the workspace by rendering laser range data and stereo depth maps from the estimated trajectory.



(a) New College 1 vision only (b) New College 1 vision and laser

Fig. 19. Left: the optimized trajectory of the New College 1 dataset (2.3 km) using only visual constraints (no laser ICP). Note the final loop around the ground appears to be in error because no stereo matching was possible due to the opposite traversal direction. Right: the optimized trajectory using both visual constraints and ICP matching. Note how in comparison to Figure 19(a) the final excursion around the grounds is properly constrained.



(a) New College 2 (b) New College 3

Fig. 20. The optimized trajectory of the New College 2 dataset (2.1 km) and the New College 3 dataset (0.8 km).

6.1. Laser Map Generation

Our vehicle is equipped with two LMS 291 lasers mounted vertically on its sides. The lasers are set to 0.5° resolution resulting in an "angel wing" beam pattern. By capturing the intensity of the reflected laser pulses and careful time synchronization (Tables 2 and 3 indicate the angular velocities experienced by our vehicle) we are able to generate detailed 3D point clouds. Figure 21 shows the typical detail produced in real time from our full 6-DOF platform.

Fig. 21. Close detail of a point cloud built by rendering range and reflectance data from the estimated trajectory of a moving Segway platform (New College dataset).

Fig. 22. View of the buildings in the quad of the New College dataset rendered from the 6-DOF estimated trajectory.

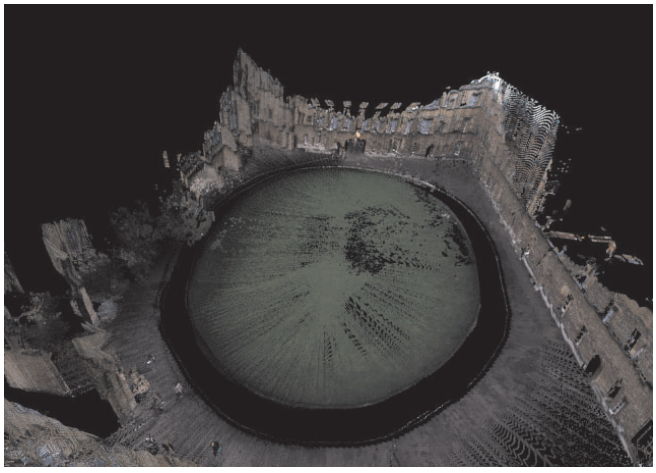


Fig. 23. A view of the New College dataset with color derived from back-projecting laser points into the images taken by the panoramic camera used for loop closure detection.

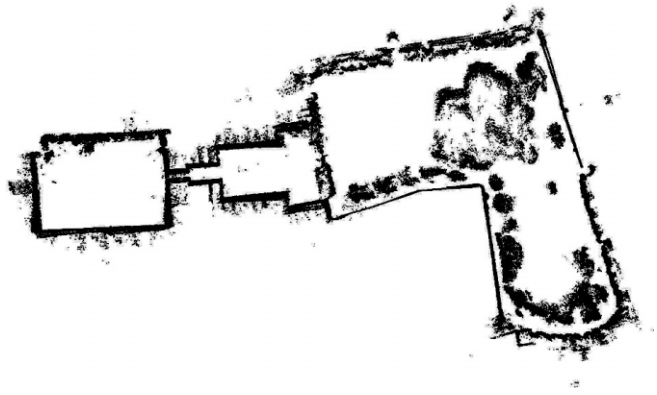


Fig. 24. A complete "bird's eye" view of the New College 1 dataset with the map rendered from an optimized pose-graph.

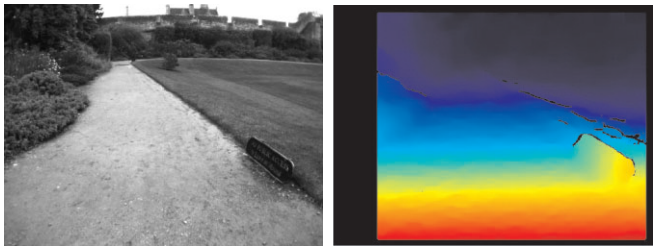
Figure 22 shows a view of part of the map built from the New College dataset (Smith et al. 2009) (front quad) rendered from the estimated trajectory. The "up" gravity vector has been aligned using the ground plane detection described in Section 6.2. Figure 24 shows a thinned point cloud of the entire New College 1 dataset.

With an assembled 3D point cloud in hand, it is possible to produce a colored version like that shown in Figure 23 by back-projecting laser points into the view of a camera and looking up the required color. It is at this point that the importance of high-quality lens distortion removal, timing and 6-DOF pose estimation becomes evident: poor spatial and temporal alignment lead to disappointing results. While this produces appealing results it is not an end in itself. Rather it is an important precursor to the semantic labeling step described in Section 8.

6.2. Dense Stereo Map Generation

As well as using the stereo rig to estimate vehicle motion, we are able to generate disparity maps in real time. This will enable us to undertake obstacle avoidance and motion planning tasks. At present we use the disparity maps to fill in the 3D structure of the scene not observed by the scanning lasers on our vehicle shown in Figure 2. The orientation and field of view (90°) of the lasers means that a stripe of workspace is unobserved underneath the vehicle and near each side (note the black stripe in Figure 23).

We implement a local, window-based stereo algorithm employing a number of disparity refinement and error detection stages. Stereo images from the Point Grey BumbleBee2 camera are undistorted and rectified using the factory calibration stored onboard the camera. To compensate for any photometric



(a) Left image (b) Disparity map

Fig. 25. The result of disparity map calculation on a stereo pair from the New College dataset. The color of pixels in the disparity map indicate depth: red pixels are close to the camera, dark blue are far away. Pixels for which no disparity could be calculated are black.

variation between the two images, we process the images using a Laplacian of Gaussian filter (Marr and Hildreth 1980). Taking the left image as the reference image, we calculate correlation scores using a sum of absolute differences over the correlation window (typically 11×11 pixels^2). These disparities are refined using the multiple supporting windows technique described by Hirschmüller et al. (2002). This helps to compensate for errors introduced by a correlation window which overlaps depth discontinuities. Five supporting windows are used for speed of computation and the best (lowest) three scores contribute to the refined correlation score.

For each pixel, a search of the corresponding discrete correlation curve is performed, looking for the minimum correlation score. A left/right consistency check, as proposed by Fua (1993), performs the correlation search twice by reversing the roles of the two images. A disparity is marked as invalid if the two correlation curve minima do not agree.

A sharply defined minimum is strongly indicative of a correct correspondence match. A flat, or close to flat correlation curve indicates a region of low texture in which it is inherently difficult to find a correct match using a window-based stereo algorithm. We therefore disregard disparities for pixels where the relative difference between the lowest and second lowest points of the curve falls below an empirically determined threshold.

Subpixel interpolation is performed by fitting a parabola to the correlation minimum and the two neighboring values: the minimum of this curve is taken to be the subpixel disparity. Finally, we consider the eight-way connected components of each pixel in the resulting refined disparity map, discarding pixels which are not connected to a minimum number of pixels with similar disparities. This step helps to remove isolated incorrect pixels. An example result of our disparity map calculation is shown in Figure 25.

We convert the disparity maps into 3D point clouds and, using the 6-DOF poses from VO (Section 2), orient them in a global coordinate frame. A simple RANSAC (Fischler and Bolles 1981) plane fitting method is used to detect the ground plane in each point cloud. Results are shown with the ground plane highlighted in Figure 26. We choose to only store 3D points which are located within 5 m of the camera. This is due to the triangulation uncertainty in the conversion from disparity to depth becoming more pronounced with more distant points (Matthies and Shafer 1987). An average of 79% of possible pixels in each 512×384 input image are given valid disparity values by our implementation, and of these 58% fall within our 5 m threshold.

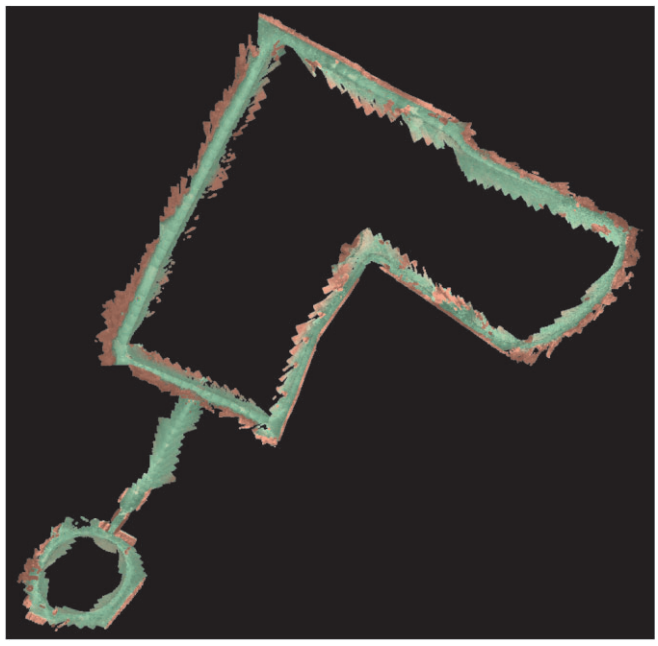


Fig. 26. Dense 3D point cloud from stereo using the New College 1 dataset rendered from the VO trajectory, with the ground plane in green. An average of 79% of possible pixels in each 512×384 input image are given valid disparity values by our implementation, and of these 58% fall within 5 m of the camera.

6.3. Assessing Map Quality

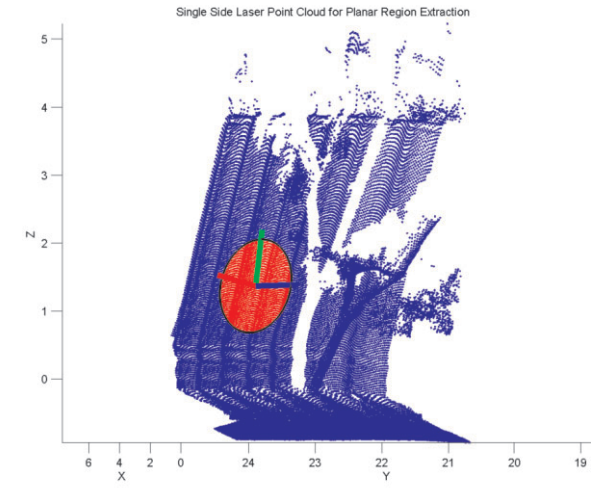
Although the 3D point clouds are visually compelling, it is important to assess their intrinsic quality. In the long term we want to use measures of map quality to deduce additional pose graph constraints required to create a high-quality model of the workspace. In this section we analyze the quality of the map built inside the New College quad. The quadrangle was circumnavigated four times and a perfect map would have all four walls lining up perfectly after each orbit. Our approach is to measure how far from this ideal our map really is. We

begin by finding planar sets of points from walls which were observed on multiple loops using the following two steps.

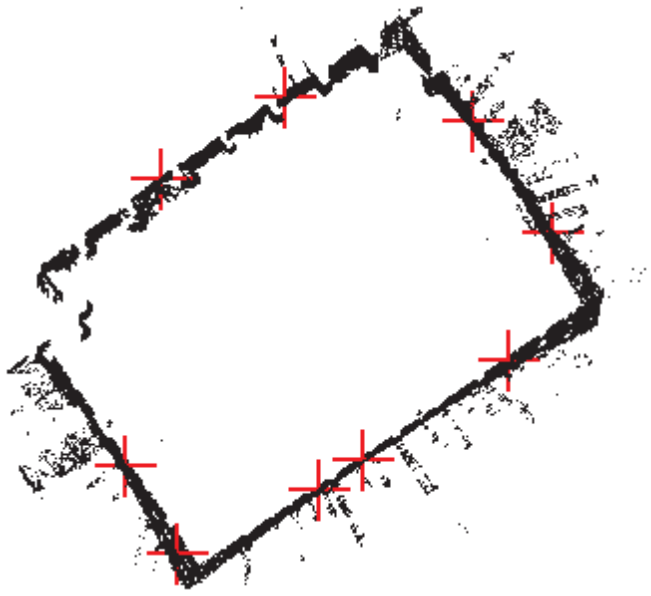
- *Region of interest selection.* The user is presented with a 3D point cloud of the *initial* pass of a environment and selects k test points, ${}^{1}p_{1:k}$, on a wall and expands a capture radius r_i round each such that the set of points, ${}^{1}\mathcal{W}_i$ within r_i of p_i lie within a plane. Here we are using a superscripted prefix to indicate the pass of the workspace: 1 being the first pass, 2 being the second and so on.
- *Interest expansion*. A script is run which searches over the entire map to find additional planar point sets that correspond to the same patch of wall but from subsequent passes. If there were N complete passes through the environment we would expect N point sets for each of the k user-selected test points ${}^{1:N}\mathcal{W}_i$, $i = 1:k$. We are assuming here that the maps being analyzed are not in gross error, otherwise finding correspondences across passes will be hard.

We are now able to calculate statistics on how consistent the geometry of the wall patches are as they are mapped again and again. First we calculate the normal ${}^{j}\hat{\mathbf{n}}_i$ of each wall patch ${}^{j}\mathcal{W}_i$ via a singular value decomposition of its scatter matrix and also the centroids ${}^{j}\mathbf{c}_i$, $j = 1:N$, $i = 1:k$. For each possible pairing of planes corresponding to the same physical patch of wall we calculate the angle between the surface normals and the distance between centroids. We refer to these quantities as intra-cluster alignment and displacement. Table 4 presents statistics of these quantities.

The results are promising although not perfect, and this is an area requiring further work. In particular it would be advantageous and interesting to add extra constraints to the pose graph as a function of the measured quality of the maps — this is an area of current research.



(a) A user-selected seed point in a planar region



(b) Locations in which seed points were selected

Fig. 27. In the left image a user has selected a point on a wall (beside a tree) using laser points only from the first pass past it and a planar region has been detected and selected around it. The right-hand image shows, with red crosses, where these test points were selected to generate the statistics shown in Table 4.

7. Multi-session Mapping

The FAB-MAP architecture can easily be applied across data gathered from multiple outings. The input to the algorithm can be batch or sequential. Presented with a collection of images, it generates a list of loop closure notifications between images which are themselves time-stamped. This that means loop closures can be found between datasets gathered days apart and because the operation is purely appearance-based, we need not worry about aligning metric coordinate frames. Figure 28 shows loop closures found between the New College 2 and New College 3 datasets.

Section 5 shows how the graph relaxation can be viewed as relaxing a chain of poses laid down by the vehicle's motion which is pinched together by loop closure edges. This notion can be simply extended to multi-session scenarios by modeling the change of location between the end of day k and the start of day $k+1$ as a single link joining two trajectory chains, but of which we have infinite uncertainty. Figure 28 shows the result of applying this technique to the co-joined trajectories shown in Figure 29.

The optimization of our pose graphs is an offline process: it takes about 20 minutes to optimize a 50,000 node graph with a few hundred loop closures. The question of finding the correct weighting between loop closure interpose constraints is deli-

Table 4. Analysis of the Quality of New College Quad Point Cloud

Property	Value
Maximum intra-cluster angle over all $\mathcal{W}$	9.1°
Minimum intra-cluster angle over all $\mathcal{W}$	0.32°
Maximum of the average intra-cluster angle over all $\mathcal{W}$	4.86°
Minimum of the average intra-cluster angle over all $\mathcal{W}$	0.66°
Average intra-cluster angle over all $\mathcal{W}$	3.6°
Maximum intra-cluster displacement over all $\mathcal{W}$	0.6 m
Minimum intra-cluster displacement over all $\mathcal{W}$	0.02 m
Maximum of the average intra-cluster displacement over all $\mathcal{W}$	0.33 m
Minimum of the average intra-cluster displacement over all $\mathcal{W}$	0.14 m
Average intra-cluster displacement over all $\mathcal{W}$	0.21 m

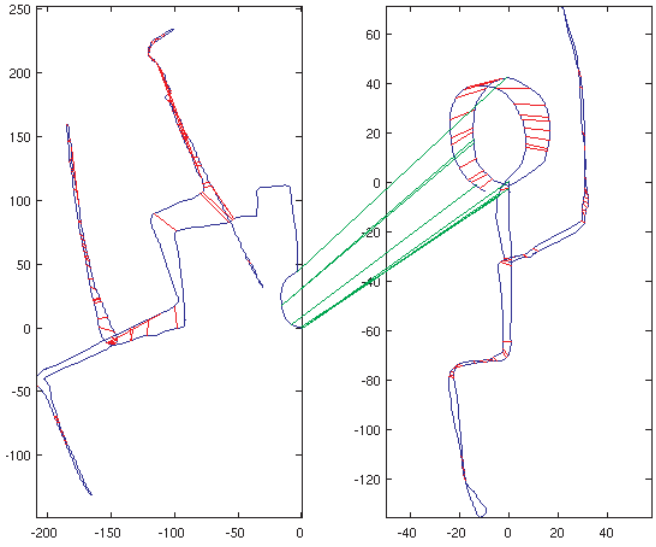


Fig. 28. Loop closure links found within and *between* the New College 2 and New College 3 datasets. Inter-day loop closures are shown in green.

cate and needs further research. Certainly, one must model the correlations between linear and rotational motion for a non-holonomic vehicle. Also, if the optimization is seeded with an atrocious first guess then convergence to a reasonable trajectory is far from assured. As always, local minima are a hazard and these often take the form of tight knots in the vehicle trajectory. To undo one of these knots (and from there reach a

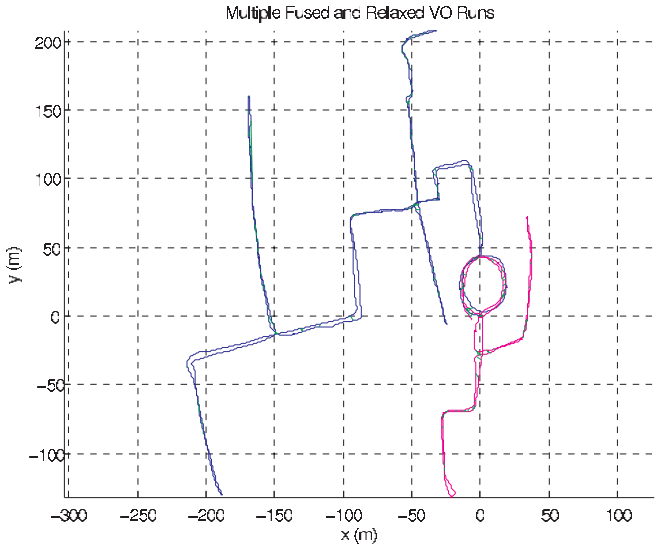


Fig. 29. Relaxed multi-session trajectories between the New College 2 (blue) and New College 3 (pink) datasets. Note that fusion and relaxation is done with no manual alignment of coordinate frames: the alignment is discovered automatically by applying loop closure constraints.

global minima) appears to require a temporary increase in the cost $C(V|\mathcal{L}, \mathcal{VO})$ as defined in Equation (9), something that gradient-based optimizers are unable to do.

8. Semantic Labeling

The maps we produce are agglomerations of laser points: at this point they are well registered and colored and make for appealing images such as Figure 23, but we wish to do more. We want to move towards understanding *what* is in the map, where it is and what that might mean to a user and for the operation of the vehicle. Particularly when navigating in an urban context, a more informative, higher-order representation of the environment is indispensable: if only because self-preservation dictates avoidance of highly dynamic regions such as roads. Robust localization can be helped by distinguishing features beyond the recognition of ubiquitous general objects such as "ground", "wall" or "house". This motivates the definition of desired classes: in an urban environment places can be distinguished by the type of ground that is present, the color and texture of surrounding houses (or, more appropriately, of surrounding walls) and the presence or absence of other features such as bushes or trees. Our goal is to add value to maps built by mapping algorithms by augmenting them with such higher-order, semantic labels. We achieve this by using both *scene appearance and geometry* to produce a composite description of the local area. The following presents an overview of the

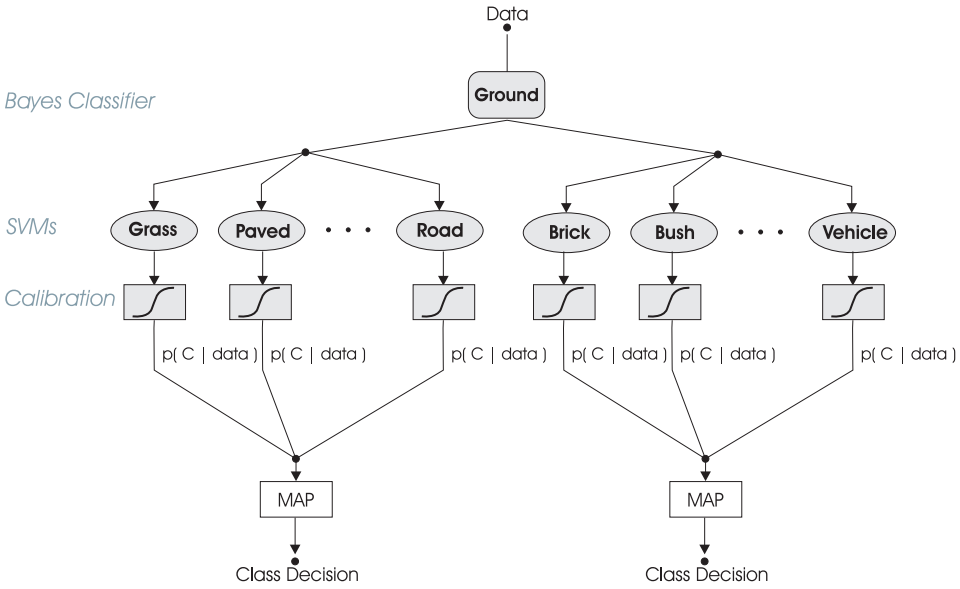


Fig. 30. The classification hierarchy employed in this work employing both a Bayesian classifier (to separate ground and non-ground classes) and a bank of SVMs.

classification framework employed as well as the data processing involved. The system was first introduced and evaluated extensively by Posner et al. (2008). It is worth noting that the classifiers employed here originate from a different vehicle with a different sensor payload and setup: the classifiers were trained originally on an ATRV-Junior vehicle using data from a forward-looking LMS 200 unit mounted in a reciprocating cradle. However, the general nature of the features used for classification provide for acceptable classification performance without necessitating a customization of the classification framework or even retraining of the classifiers for the Segway-based platform in Figure 2.

8.1. The Labeling Pipeline

Our scene labeling engine is based on both appearance and geometric features extracted from cross-calibrated camera–laser pairs. In this case, on both sides of the vehicle one of the sideways-looking cameras of the Ladybug unit was calibrated against the LMS unit on that same side. This allows for the projection of gathered laser data into the corresponding images. Thus equipped, the processing pipeline proceeds by first performing a plane segmentation on a laser point cloud associated with a particular scene. The choice of a plane as a geometric primitive is tolerable because of its ubiquitous use in man-made environments, but it is something our latest work does not require. This segmentation provides us with a robust estimate of local 3D geometry associated with every laser datum identified as part of a plane in the environment. These data are then projected into the corresponding camera images.

While the next section provides a more detailed outline of the classification framework employed, we mention here our choice of a majority voting scheme in the resulting classifications to motivate the next step in the processing pipeline. As described in detail by Posner et al. (2008), the initial plane segmentation in laser space is refined based on an off-the-shelf image segmentation algorithm. The result of this processing step are image patches, or *superpixels*, which, by way of containing laser data, have 3D geometric information associated with them. For each of these superpixels, standard appearance features are associated with each of the projected laser data. In this case, a histogram for both the hue and saturation channel is calculated over a fixed-size neighborhood around each interest point. The laser data associated with each superpixel as well as the corresponding feature vectors form the input to the classification stage of the system.

8.2. Classification Framework

The classification framework adopted here operates on individual laser data grouped by superpixel membership and results in the classification of entire superpixels in an image by means of majority consensus of individual classifications. In order to classify individual laser data, we employ a hierarchical combination of two distinct discriminative approaches. An illustration is given in Figure 30. At the top of the hierarchy a threshold classifier is employed to distinguish between ground and non-ground classes, based on the Bayes decision rule. The

Table 5. Classification Results for an Independent Validation Set (Reproduced with permission from (Posner et al. 2008))

Class details			Performance	
Name	Number of patches	Number of points	Precision (%)	Recall (%)
Grass	99	5,393	96.6	98.1
Tarmac/paved	1,373	77,256	97.7	89.0
Dirt patch	147	7,988	46.4	84.8
Textured wall	2,240	69,541	82.7	73.5
Smooth wall	906	29,881	56.9	64.4
Bush/foliage	181	8,364	60.6	62.8
Vehicle	169	4,499	43.7	80.1

decision is based on the height (from the ground) of an individual laser datum as well as the orientation of the plane segment of which the datum is a member. The appropriate thresholds for this classification stage were learned from labelled training data.

The second level of the classification hierarchy consists of a bank of support-vector machines (SVMs) for the ground and the non-ground classes, respectively. SVMs are a popular choice where the model parameters are found by solving a convex optimization problem. This is a desirable property since it implies that the final classifier is guaranteed to be the best feasible discriminant given the training data.

While SVMs are inherently binary decision makers, multi-class classification within a bank of classifiers is performed by comparing the outputs of the individual SVMs trained as one-versus-all. This is a common extension to the binary case (Burges 1998). In order to ensure that the classifier outputs are of the same scale, and are thus comparable, a *probabilistic calibration* is performed in which the class posterior from the raw SVM output is estimated such that the final classification amounts to a maximum *a posteriori* decision amongst the individual classes (Platt 2000). Finally, majority consensus amongst all of the individual laser classifications within an image patch provides the label for that superpixel.

The system was trained and evaluated on an ATRV Junior platform using laser and vision data from over 16 km of track through an urban environment. Individual SVMs were trained using a Gaussian kernel, which is a common choice and has been found to perform well in a variety of applications. The kernel width as well as a trade-off parameter specifying a tolerance for misclassifications during training were determined using five-fold cross-validation over a grid in parameter space. To provide an indication of typical system performance classification results on a validation set are presented in Table 5. For this dataset scene classification was carried out on average in 4.8 s per frame. For a detailed description and analysis of the performance of the classification framework the reader is referred to Posner et al. (2008).

Typical output from this system when applied to data gathered by the Segway at various positions around the New College Quad (dataset 1) is shown in Figure 31.

9. Future Work

This paper documents our progress in producing a reliable large-scale navigation system. While very few published methods tackle the trajectory lengths we do here (Konolige and Agrawal (2008) and Nister et al. (2006) being clear exceptions) and at our frame density, much remains to be done. While we certainly have the parts in place to achieve our aims, we are not at the stage at which long-term operation is reliable. If we were to pick one aspect of this research that needs attention it would be introspection: the ability to look back over past decisions, measurements and optimizations and, armed with several metrics, decide that all is not well and, ideally, plan and execute remedial action. This goes beyond the commonplace day-to-day data association problem where we search for the best way to interpret a given set of measurements (including rejecting them). We should be looking at the final global properties of maps and trajectories (for example, compatibility between camera pixels and laser range images) to assess online performance and drive exploration strategies. Our work on map quality analysis is a start down this path, but much remains to be done to provide SLAM systems with the nagging, persistent self-doubt that we believe will lead to the robust implementations we desire. Looking to the future, our motivation is to move up from pixels and laser pulses through geometry and image patches and up to useful structural and semantic labels of workspaces. We wish to generate symbols with sufficient diversity and richness that allow a connection with computational linguistics. Indeed, a mid-term goal is to reach a state of systems maturity in which it becomes sensible to engage in problems of life-long learning and principled human machine communication via natural language. We have some way to go before achieving this, but we believe the bedrock

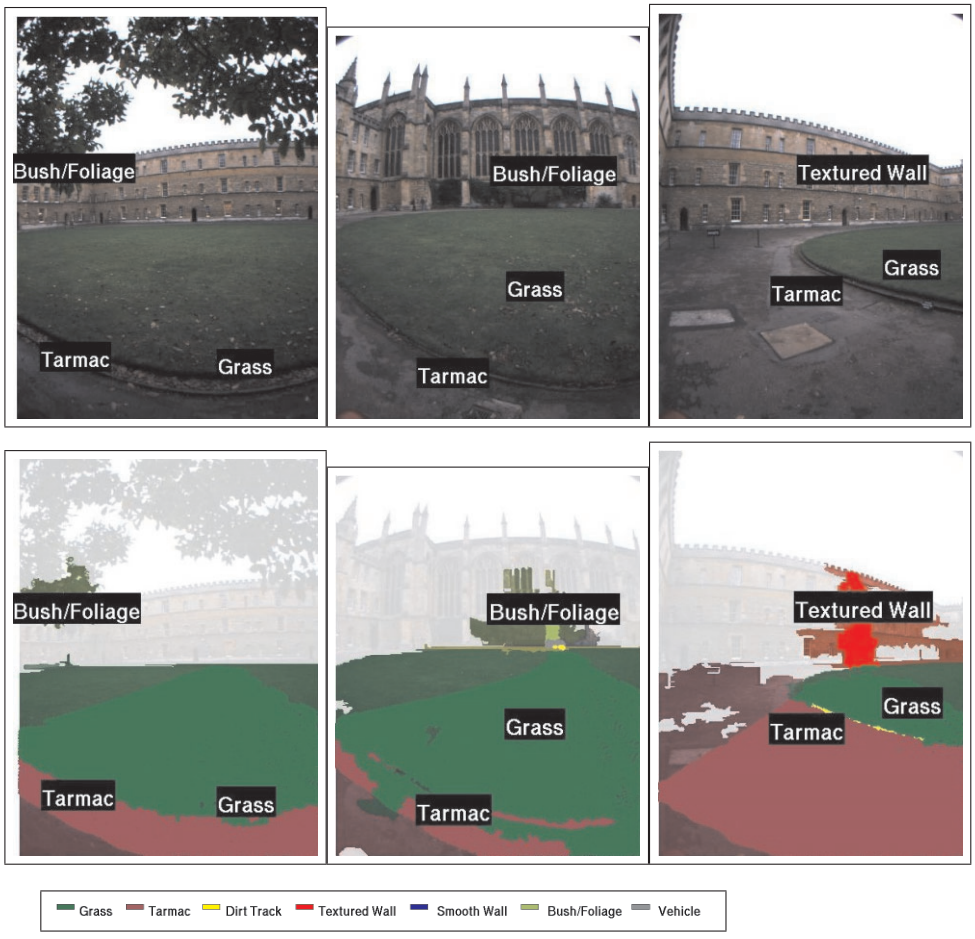


Fig. 31. Typical output from the scene labeling engine employed around the New College quad. The top row presents the original images. The bottom row presents the corresponding superpixel classifications and shows the projected laser data for each image. The labels are generated automatically. While not all of the classes mentioned in the legend are represented in the images, the full legend has been provided to give an intuition as to the classes catered for by the current system. A more detailed evaluation of the system employed here can be found in Posner et al. (2008).

must be a robust, long-lived ability to localize, map and label workspaces from a moving platform.

Acknowledgments

The work reported in this paper undertaken by the Mobile Robotics Group was funded by the Systems Engineering for Autonomous Systems (SEAS) Defence Technology Centre established by the UK Ministry of Defence, by Guidance Ltd, and by the UK EPSRC (CNA and Platform Grant EP/D037077/1). Christopher Mei and Ian Reid of the Active Vision Lab acknowledge the support of EPSRC grant GR/T24685/01. The authors would like to extend their thanks to Rohan Paul and Ian Baldwin for their cold wanderings collecting data around New College.

References

Angeli, A., Filliat, D., Doncieux, S. and Meyer, J.-A. (2008). A fast and incremental method for loop-closure detection using bags of visual words. *IEEE Transactions On Robotics*, **24**: 1027–1037.

Bar-Shalom, Y. and Fortmann, T. E. (1988). *Tracking and Data Association*. New York, Academic Press.

Bell, B. M. and Cathey, F. W. (1993). The iterated Kalman filter update as a Gauss–Newton method. *IEEE Transactions on Automatic Control*, **38**(2): 294–297.

Besl, P. J. and McKay, N. D. (1992). A method for registration of 3-d shapes. *IEEE Transactions Pattern Analysis and Machine Intelligence*, **14**(2): 239–256.

Bibby, C. and Reid, I. (2007). Simultaneous localisation and mapping in dynamic environments (SLAMIDE) with re-

versible data association. *Proceedings of Robotics: Science and Systems*, Atlanta, GA.

Bosse, M. and Zlot, R. (2008). Map matching and data association for large-scale two-dimensional laser scan-based slam. *The International Journal of Robotics Research*, **27**(6): 667–691.

Brown, D. (1958). A solution to the general problem of multiple station analytical stereotriangulation. *RCP-MTP Data Reduction Technical Report No. 43*, Patrick Air Force Base, Florida (also designated as AFMTC 58-8).

Brown, D. (1976). The bundle adjustment—progress and prospects. *XIIIth Congress of the International Society for Photogrammetry*.

Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, **2**(2): 121–167.

Chen, C. and Wang, H. (2006). Appearance-based topological Bayesian inference for loop-closing detection in a cross-country environment. *The International Journal of Robotics Research*, **25**(10): 953–983.

Chow, C. and Liu, C. (1968). Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, **14**(3): 462–467.

Cole, D. and Newman, P. M. (2006). Using laser range data for 3d SLAM in outdoor environments. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Orlando, FL.

Cummins, M. and Newman, P. (2007). Probabilistic appearance based navigation and loop closing. *Proceedings IEEE International Conference on Robotics and Automation (ICRA'07)*, Rome.

Cummins, M. and Newman, P. (2008a). Accelerated appearance-only SLAM. *Proceedings IEEE International Conference on Robotics and Automation (ICRA'08)*, Pasadena, CA.

Cummins, M. and Newman, P. (2008b). FAB-MAP: probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, **27**(6): 647–665.

Cummins, M. and Newman, P. (2009). Highly scalable apppearance-only SLAM—FAB-MAP 2.0. *Proceedings of Robotics Science and Systems*, Seattle, WA.

Deans, M. C. (2005). Bearings-only localization and mapping. *PhD Thesis*, School of Computer Science, Carnegie Mellon University.

Dellaert, F. and Kaess, M. (2006). Square Root SAM: simultaneous location and mapping via square root information smoothing. *The International Journal of Robotics Research*, **25**(12): 1181.

Eade, E. and Drummond, T. (2008). Unified loop closing and recovery for real time monocular SLAM. *Proceedings of 19th British Machine Vision Conference*, Leeds, UK.

Engels, C., Stewenius, H. and Nister, D. (2006). Bundle adjustment rules. *Proceedings of Photogrammetric Computer Vision*.

Eustice, R., Singh, H., Leonard, J., Walter, M. and Ballard, R. (2005). Visually navigating the RMS Titanic with SLAM information filters. *Proceedings of Robotics: Science and Systems*, Cambridge, MA.

Filliat, D. (2007). A visual bag of words method for interactive qualitative localization and mapping. *Proceedings 2007 IEEE International Conference on Robotics and Automation* pp. 3921–3926.

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, **24**: 381–395.

Fitzgibbon, A. W. and Zisserman, A. (2004). *Automatic Camera Recovery for Closed or Open Image Sequences*. Berlin, Springer.

Frese, U. and Duckett, T. (2003). A multigrid approach for accelerating relaxation-based SLAM. *Proceedings IJCAI Workshop on Reasoning with Uncertainty in Robotics (RUR 2003)*, Acapulco, Mexico, pp. 39–46.

Fua, P. (1993). A parallel stereo algorithm that produces dense depth maps and preserves image features. *Machine Vision and Applications*, **6**(1): 35–49.

Grisetti, G., Stachniss, C., Grzonka, S. and Burgard, W. (2007). A tree parameterization for efficiently computing maximum likelihood maps using gradient descent. *Proceedings of Robotics: Science and Systems*, Atlanta, GA.

Gutmann, J. and Konolige, K. (November 1999). Incremental mapping of large cyclic environments. *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, Monterey, CA, pp. 318–325.

Hartley, R. and Zisserman, A. (2000). *Multiple View Geometry in Computer Vision*. Cambridge, Cambridge University Press.

Hirschmüller, H., Innocent, P. R. and Garibaldi, J. (2002). Real-time correlation-based stereo vision with reduced border errors. *International Journal of Computer Vision*, **47**(1–3): 229–246.

Ho, K. L. and Newman, P. (2007). Detecting loop closure with scene sequences. *International Journal of Computer Vision*, 74(3): 261–286.

Iagnemma, K. and Buehler, M. (eds) (2006a). Special issue on the DARPA Grand Challenge, Part I. *Journal of Field Robotics*, **23**(8): 461–652. ISSN 0741-2223.

Iagnemma, K. and Buehler, M. (eds) (2006b). Special issue on the DARPA Grand Challenge, Part II. *Journal of Field Robotics* **23**(9): 655–835.

Jordan, M. (2003). *An Introduction to Graphical Models* (unpublished).

Konolige, K. and Agrawal, M. (2007). Frame–frame matching for realtime consistent visual mapping. *Proceedings 2007*

IEEE International Conference on Robotics and Automation, Rome, Italy.

Konolige, K. and Agrawal, M. (2008). FrameSLAM: from bundle adjustment to real-time visual mapping. *IEEE Transactions on Robotics*, **24**(5): 1066–1077.

Kröse, B. J. A., Vlassis, N. A., Bunschoten, R. and Motomura, Y. (2001). A probabilistic model for appearance-based robot localization. *Image and Vision Computing*, **19**(6): 381–391.

Lamon, P., Nourbakhsh, I., Jensen, B., and Siegwart, R. (2001). Deriving and matching image fingerprint sequences for mobile robot localization. *Proceedings of the IEEE International Conference on Robotics and Automation*, Seoul, Korea.

Lina María Paz, J. D. T., Piniés, P. and Neira, J. (2008). Large-scale 6-DOF SLAM with stereo-in-hand. *IEEE Transactions on Robotics*, **24**(5): 946–957.

Lu, F. and Milios, E. (1997). Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, **4**(4): 333–349.

Marr, D. and Hildreth, E. (1980). Theory of edge detection. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, **207**(1167): 187–217.

Matthies, L. and Shafer, S. (1987). Error modelling in stereo navigation. *IEEE Journal of Robotics and Automation*, **3**(3): 239–248.

McLauchlan, P. F. (1999). The variable state dimension filter applied to surface-based structure from motion. *Technical Report*, University of Surrey.

McLauchlan, P. F. and Murray, D. W. (1995). A unifying framework for structure and motion recovery from image sequences. *Proceedings of the International Conference on Computer Vision*, pp. 314–320.

Mei, C., Benhimane, S., Malis, E. and Rives, P. (2008). Efficient homography-based tracking and 3-D reconstruction for single viewpoint sensors. *IEEE Transactions on Robotics*, **24**: 1352–1364.

Mikhail, E. M. (1983). *Observations and Least Squares*. Rowman & Littlefield.

More, J. (1978). *The Levenberg–Marquardt Algorithm: Implementation and Theory* (*Lecture Notes in Mathematics*, Vol. 630). Berlin, Springer, pp. 105–116.

Mouragnon, E., Lhuillier, M., Dhome, M., Dekeyse, F. and Sayd, P. (2006). Real time localization and 3D reconstruction. *Proceedings of Computer Vision and Pattern Recognition*.

Newman, P. (1999). On the structure and solution of the simultaneous localisation and map building problem. *PhD Thesis*, The University of Sydney.

Newman, P. M., Cole, D. M. and Ho, K. (2006). Outdoor SLAM using visual appearance and laser ranging. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Orlando, FL.

Nister, D., Naroditsky, O. and Bergen, J. (2004). Visual odometry. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Washington, DC, pp. 652–659.

Nister, D., Naroditsky, O. and Bergen, J. (2006). Visual odometry for ground vehicle applications. *Journal of Field Robotics*, **23**(1): 3–20.

Nister, D. and Stewenius, H. (2006). Scalable recognition with a vocabulary tree. *Proceedings of the Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. 2161–2168.

Olson, C. F., Matthies, L. H., Schoppers, M. and Maimone, M. W. (2001). Stereo ego-motion improvements for robust rover navigation. *Proceedings of the IEEE Conference on Robotics and Automation*, Washington, DC, pp. 1099–1104.

Olson, E., Leonard, J. and Teller, S. (2006). Fast iterative alignment of pose graphs with poor initial estimates. *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2262–2269.

Platt, J. (2000). Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In: *Advances in Large Margin Classifiers*, A. Smola, P. Bartlett, B. Schoelkopf, and D. Schuurmans (eds), MIT Press, pp. 61–74.

Posner, I., Schroeter, D. and Newman, P. (2008). Online generation of scene descriptions in urban environments. *Robotics and Autonomous Systems*, **56**(11): 901–914.

Rosten, E. and Drummond, T. (2005). Fusing points and lines for high performance tracking. *Proceedings IEEE International Conference on Computer Vision*, Vol. 2, pp. 1508–1511.

Schindler, G., Brown, M. and Szeliski, R. (2007). City-Scale Location Recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–7.

Sibley, G. (2006). Sliding window filters for SLAM. *Technical Report CRES-06-004*, University of Southern California, Center for Robotics and Embedded Systems.

Sibley, G., Matthies, L. and Sukhatme, G. (2007). *A Sliding Window Filter for Incremental SLAM* (*Lecture Notes in Electrical Engineering*, Vol. 8). Berlin, Springer.

Sibley, G., Sukhatme, G. and Matthies, L. (2006). The iterated sigma point Kalman filter with applications to long range stereo. *Proceedings of Robotics: Science and Systems*, pp. 263–270.

Sivic, J. and Zisserman, A. (2003). Video Google: a text retrieval approach to object matching in videos. *Proceedings of the International Conference on Computer Vision*, Nice, France.

Smith, M., Baldwin, I., Churchill, W., Paul, R. and Newman, P. (2009). The new college vision and laser data set. *The International Journal of Robotics Research*, **28**(5): 595–599.

Smith, R. C., Self, M. and Cheeseman, P. (1990). Estimating uncertain spatial relationships in robotics. *Autonomous*

Robot Vehicles, I. J. Cox and G. T. Wilfong (eds). Berlin, Springer, pp. 167–193.

Thrun, S., Burgard, W. and Fox, D. (2005). *Probabilistic Robotics*. Cambridge, MA, MIT Press.

Thrun, S., Koller, D., Ghahmarani, Z. and Durrant-Whyte, H. (2002). Simultaneous mapping and localization with sparse extended information filters: Theory and initial results. *Proceedings of the Workshop on Algorithmic Foundations of Robotics*.

Thrun, S. and Montemerlo, M. (2006). The graph SLAM algorithm with applications to large-scale mapping of urban structures. *The International Journal of Robotics Research*, **25**(5–6): 403–429.

Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., et al. (2006). Stanley: the robot that won the DARPA Grand Challenge. *Journal of Field Robotics*, **23**(1): 661–692.

Triggs, B., McLauchlan, P., Hartley, R. and Fitzgibbon, A. (2000). Bundle adjustment—a modern synthesis. *Vision Algorithms: Theory and Practice*, W. Triggs, A. Zisserman and R. Szeliski (eds) (*Lecture Notes in Computer Science*). Berlin, Springer, pp. 298–375.

Tsai, R. Y. (1987). A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of Robotics and Automation*, **3**(4): 323–344.

Urmson, C., Anhalt, J., Bae, H., Bagnell, J. D., Baker, C., Bittner, R. E., et al. (2008). Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics Special Issue on the 2007 DARPA Urban Challenge, Part I*, **25**(1): 425–466.

Wang, J., Cipolla, R. and Zha, H. (2005). Vision-based global localization using a visual vocabulary. *Proceedings of the International Conference on Robotics and Automation*.

Wolf, J., Burgard, W. and Burkhardt, H. (2005). Robust vision-based localization by combining an image-retrieval system with Monte Carlo localization. *IEEE Transactions on Robotics*, **21**(2): 208–216.

Yannakakis, M. (1981). Computing the minimum fill-in is NP-complete. *SIAM Journal of Algebraic and Discrete Mathematics*, **2**: 77–79.