Towards Life Long Mapping

Paul Newman, Gabe Sibley, Mark Cummins, Mike Smith, Alastair Harrison,
Christopher Mei, Ingmar Posner ,Ian Reid

Oxford University Mobile Robotics Group

Abstract In this paper we describe our work towards mapping arbitrary workspaces using stereo vision for ego motion estimation and laser for geometry acquisition over extended periods of time and especially over multiple sessions. We show how a fast appearance based technique allows us to detect intersections between independently created maps (different days with different, unplanned, metric origins) and hence we can fuse trajectories and dense maps built over long time scales. We show how a simple interpretation of the pose graph induced by the vehicle's motion as a chain pinched together by loop-closing and intersection constraints leads to a parallelizable optimisation problem. We conclude by presenting and analyzing results generated from multiple data sets gathered over many kilometers in a park and campus setting

1 Introduction

In this paper we describe our work on building a vision-based system capable of mapping large workspaces using vision for pose estimation and laser for map creation. Our ultimate goal is to engineer a suite of software and hardware that can operate in an environment for an arbitrarily long period of time. This of course is a vast task and we cannot address all the issues involved at once. This paper focuses on the software components of this task and in particular the information processing algorithms that might enable a robot to continually build and refine a model of its environment. Our system is designed to enable unsupervised map refinement to occur over multiple sessions - an important aspect of "life long" operation. In this sense our system is opportunistic, the appearance based place recognition system we employ, allows map intersections and overlaps to be discovered as and when they occur without recourse to metric estimates. Another aspect that must be considered is how one might evaluate the quality of the maps that have been built with the ultimate aim of revisiting poorly modeled areas or focussing computational ef-

fort on them. In our results section we present some preliminary work on assessing the quality of dense point clouds created under our visual navigation scheme.

2 Vehicle Trajectory Generation

Pose and trajectory estimation is a fundamental requirement for our work. In this work we use a fast, accurate and robust vision based system which is well suited to the vehicle shown in Figure 6. It is based on the Sliding Window Filter of Sibley [26] and is driven by robust inter-frame feature tracking across sequential stereo image-pairs. Our motivation for pushing the vision-based system over our 3D laser-based system which we have used in previous work is threefold : firstly, stereo cameras are cheap; secondly, 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 (e.g texture or corners) rather than aligning points two sets of points sampled from the workspace's surfaces — points which, because of the vehicle's motion, do not correspond to the same real world object.

2.1 The Sliding Window Filter

For locally optimal trajectory estimation we employ a Sliding Window Filter (SWF), which is an approximation to the full feature-based batch non-linear least squares SLAM problem [25, 26]. The SWF concentrates computational resources on accurately estimating 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.

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 [17]. 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. The sliding window method also enables reversible data association [2], 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 (like 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.

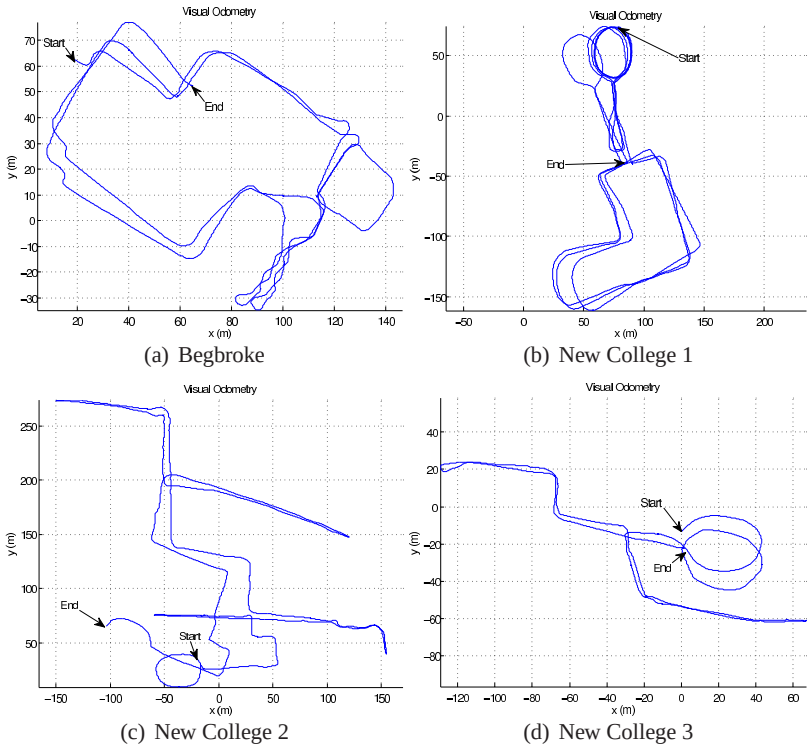
It is interesting to note what happens if we simply delete parameters from the estimator instead of marginalizing them out. For a sliding window of size k, the error converges like $1/k$ — just as we would expect the batch estimator to do. However, after k steps, the error stops converging as we delete 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 Visual Odometry (V.O.) [18, 20, 22]. 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 shall simply refer to the case of $k = 1$ as "Visual Odometry".

The details of the feature tracking and efficient optimisation scheme employed by our VO system are beyond the scope of this paper and can be found in recently published work [27, 19]. However figures present headline results from two venues, "Begbroke" and "New College" — the latter taken over multiple days. The data sets are summarised in Tables 1 and 2 and the estimated trajectories are shown in Figures 1(a)-1(d).

Table 1 Visual odometry results for Begbroke and first New College data sets.

	Begbroke			New College 1		
	Avg.	Min.	Max.	Avg.	Min.	Max.
Distance Travelled (km)	—	—	1.08	—	—	2.26
Frames Processed	—	—	23,268	—	—	51263
Velocity (m/s)	0.93	0.00	1.47	0.94	9.46*e-4*	1.53
Angular Velocity (deg/s)	9.49	0.0	75.22	7.08	4.12*e-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×*1e-3*	0.55	0.13	0.03	1.01



(a) Begbroke (b) New College 1

(c) New College 2 (d) New College 3

Fig. 1 Visual Odometry results for the four data sets detailed in Tables 1 and 2.

Table 2 Visual odometry results for second and third New College data sets.

	New College 2			New College 3		
	Avg.	Min.	Max.	Avg.	Min.	Max.
Distance Travelled (km)	—	—	2.05	—	—	0.82
Frames Processed	—	—	49,114	—	–	29,489
Velocity (m/s)	0.83	4.55*e-4*	3.05	0.56	1.63*e-4*	1.26
Angular Velocity (deg/s)	7.13	8.23*e-3*	62.56	4.83	5.24*e-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

3 Closing Loops with FABMAP

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 poorly estimated.

Our approach, FABMAP (Fast Appearance Based Mapping), previously described in [5, 8, 7, 9], 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 [28, 21] and has recently been applied to mobile robotics for loop closure detection by several authors [10, 1]. More generally appearance has been used in loop closure detection and localisation tasks by many authors [15, 16, 3, 24, 14, 31]. 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 k^{th} observation and $\mathscr{Z}^k$ all observations up to time k, then:

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

Here $p(L_i|\mathscr{Z}^{k-1})$ is our prior belief about our location, $p(Z_k|L_i,\mathscr{Z}^{k-1})$ is the observation likelihood, and $p(Z_k|\mathscr{Z}^{k-1})$ is a normalizing term. An observation Z is a binary vector, the i^{th} entry of which indicates whether or not the i^{th} word of the visual vocabulary was detected in the current scene. The key term here is the observation likelihood, $p(Z_k|L_i,\mathscr{Z}^{k-1})$, 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:

$$p(Z_k|L_i) = p(z_n|z_1,z_2,...,z_{n-1},L_i)...p(z_2|z_1,L_i)p(z_1|L_i) \tag{2}$$

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 re-

sults improve considerably if we instead employ a Chow Liu approximation [4] 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 [6].

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 previously visited one. While many appearance-based localization systems exist, this extension beyond pure localization makes the problem considerably more difficult [13]. The key is a correct calculation of the denominator of Equation 1, $p(Z_k|\mathscr{Z}^{k-1})$. If we divide the world into the set of mapped places M and the unmapped places $\overline{M}$, then

$$p(Z_k|\mathscr{Z}^{k-1}) = \sum_{m\in M} p(Z_k|L_m)p(L_m|\mathscr{Z}^{k-1}) + \sum_{u\in\overline{M}} p(Z_k|L_u)p(L_u|\mathscr{Z}^{k-1}) \qquad (3)$$

where we have applied our assumption that observations are conditionally independent given location. The first summation is simply the likelihood of all 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

$$p(Z_k|\mathscr{Z}^{k-1}) \approx \sum_{m\in M} p(Z_k|L_m)p(L_m|\mathscr{Z}^{k-1}) + p(L_{new}|\mathscr{Z}^{k-1}) \sum_{u=1}^{n_s} \frac{p(Z_k|L_u)}{n_s} \qquad (4)$$

where n_s is the number of samples used, $p(L_{new}|\mathscr{Z}^{k-1})$ 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 2 and 3 show typical loop closure results obtained using our method. Not

the high degree of confidence despite marked changes in scene and lighting. OUr current implementation of FABMAP can compare an incoming panoramic image to the impressions of 5 million previously mapped places in a second.

Fig. 2 Place recognition results generated by FABMAP. 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. 3 Example place recognition result generated by FABMAP under markedly different lighting conditions. Probability of loop closure is calculated to be 0.9519.

4 Pose Graph Relaxation

The VO subsystem produces a chain of 6DOF vehicle poses linked by relative transformations which should be thought of as uncertain metric constraints. The combination of the FABMAP output and metric pose recovery methods just described provides additional constraints between poses, resulting in a graph of vehicle poses. Figure 4 illustrates the structure of a typical pose graph.

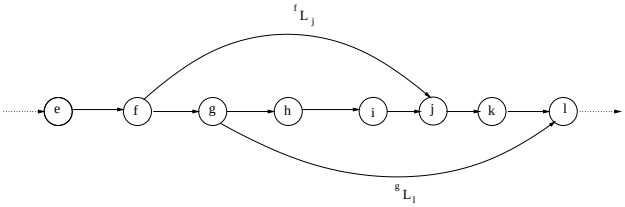


Fig. 4 A section of typical pose graph. Poses (e, f...l) are denoted as nodes (circles) and edges are relative transformations. There is a chain of relative transformations flowing through the graph created by the visual odometry system. Loop closure transformations $^{i}L_{j}$ are single edges linking disparate nodes (i and j) of this chain.

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. [11, 29, 23, 12]. The particular size and structure of our graphs motivated us to use classical non-linear optimisation techniques taking care at implementation time to make full use of the sparse properties of the problem. We note with reference to Figure 4 that the visual odometry 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 optimise not over the set of poses in the graph but rather over the relative poses between them. Define $\mathscr{V} = \{v_1, v_2 \cdots\}$ to be the set of inter-pose transformations along the trajectory chain such that v_i is the transformation between pose $i-1$ and pose i. Furthermore define $V = [v_1^T, v_2^T \cdots]^T$ to be a stacked vector of parameterisations of these relative transformations — this will be our state vector which we wish to optimise.

Consider now Figure 5 which shows a loop closure constraint between two poses m and q. We note that the transformation, ${}^m T_q$ between and two poses m and q is simply the integration of all the individual transformations between poses:

$$ {}^m T_q = v_{m+1} \oplus v_{m+2} \cdots \oplus v_q \tag{5} $$

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 $\mathscr{L} = \{L_1 \cdots L_n\}$ where L_i is between pose $a(i)$ and $b(i)$ (a and b are look up functions), and m interpose visual odometry measurements $\mathscr{VO} = \{vo_1 \cdots vo_m\}$, then the cost metric we wish to impose on the whole graph and then minimise is

$$ C(V|\mathscr{L}, \mathscr{VO}) = \sum_i^n ||L_i - {}^{a(i)} T_{b(i)}||^2 + \sum_i^m ||vo_i - v_i||^2 \tag{6} $$

where we note that the prediction ${}^{a(i)} T_{b(i)}$ is itself a function of V. The quadratic cost function in Equation 6 is well suited to classical non-linear minimisation techniques. Many of these techniques require the calculation of the derivative of the measurement prediction with respect to the state vector being optimised. We will now consider the form of this derivative.

Consider again Figure 5 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

$$ \delta\, {}^m T_q = \sum_{r=m+1}^{q} \frac{\partial\, {}^m T_q}{\partial\, v_r}\, \delta v_r \tag{7} $$

where δv_r is an incremental change in the r_{th} component of the state vector V — the relative transformation between pose $r-1$ and pose r. Considering the partial

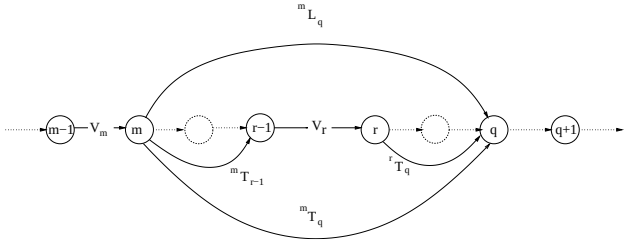


Fig. 5 A section of pose graph showing a loop closure between pose m and q and a state of interest v_r

derivative in the summation and substituting Equation 5 we have

$$\frac{\partial\, {}^{m}T_q}{\partial\, v_r} = \frac{\partial \left\{ v_{m+1} \oplus v_{m+2} \cdots \oplus v_q \right\}}{\partial v_r} \tag{8}$$

$$= \frac{\partial \left\{ {}^{m}T_{r-1} \oplus v_r \oplus {}^{r}T_q \right\}}{\partial v_r} \tag{9}$$

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

$$\frac{\partial\, {}^{m}T_q}{\partial\, v_r} = \mathscr{J}_1({}^{m}T_{r-1} \oplus v_r, {}^{r}T_q)\, \mathscr{J}_2({}^{m}T_{r-1}, v_r) \tag{10}$$

where

$$\mathscr{J}_1(x,y) = \frac{\partial x \oplus y}{\partial x} \tag{11}$$

$$\mathscr{J}_2(x,y) = \frac{\partial x \oplus y}{\partial y} \tag{12}$$

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

Equation 7 can be written in matrix form

$$\delta\, {}^{m}T_q = \mathbf{h}_{m,q} \delta V \tag{13}$$

where δV is a vector of small changes in V and $\mathbf{h}$ is a row-matrix where the k^{th} sub block ($m < k < q$) is given by Equation 10 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

$$\mathbf{h}_{m,q} \delta V = \delta {}^{m}L_q \tag{14}$$

If we have n loop closure constraints we will have n such constraints to fulfill each in the form of Equation 14 yielding

$$\mathbf{H} \delta V = \delta L \tag{15}$$

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

$$\begin{bmatrix} \mathbf{H} \\ \mathbf{I} \end{bmatrix} \delta V = \begin{bmatrix} \delta L \\ Z \end{bmatrix} \tag{16}$$

where $Z = [vo_1^T, vo_2^T \cdots]^T$ 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 $\left[\mathbf{H}^T \mathbf{I}\right]^T$ is large and we do not wish to create or store it in memory — to yield incremental adjustments in the pose graph's edges. Optimisation ceases when the perturbations in V become small. Note also that each loop closure constraint produces a new densely filled in block-row in $\mathbf{H}$ in our implementations we have found that it is the calculation of this matrix which is the bottle neck and not the solve itself - certainly this operation is easily parallelized as each row can be calculated independently.

5 Results

5.1 Platform

All the algorithms, systems and results in this paper have been applied to data gathered by the vehicle shown in Figure 6. While there is nothing vehicle-specific in our work, it is worthwhile swiftly summarising the vehicle's characteristics. The vehicle is actuated by a RMP200 base from Segway. It has 4 internal PC's at 1.6 GHz with around 1TB of total storage. Images streamed at 2Hz from a Point Grey Ladybug camera (5 panoramic images) are used in our appearance-based loop closure (FABMAP) algorithm. Stereo pairs read at 20Hz 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 75Hz mode to capture the far field geometry. The vehicle can run for approximately 90 minutes on a single battery charge with all systems powered.

The trajectory estimation described in this paper (see for example 7(a)) is entirely vision-based (apart from cases where we need to fall back to ICP registration to infer loop closure geometry). We map the 3D structure of the workspace by rendering laser range data and stereo depth maps from the estimated trajectory.

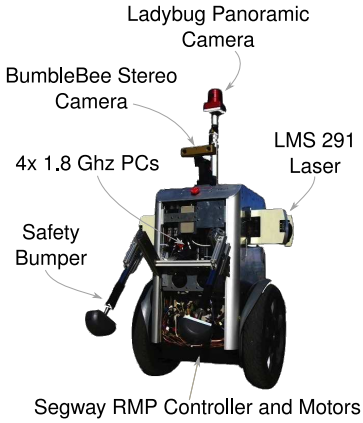
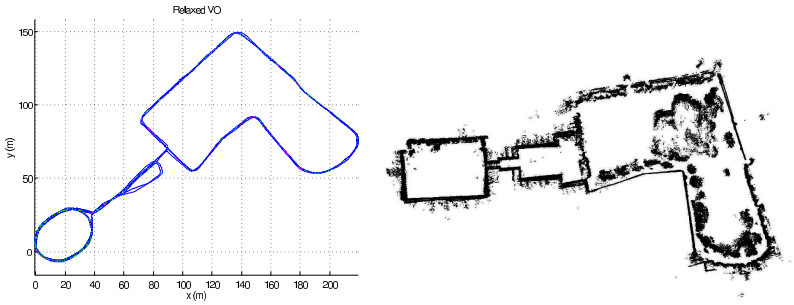


Fig. 6 The results in this paper correspond to data gathered from the modified Segway platform shown above. The vehicle has a sensor payload of 2 SICK lasers, an XSens inertial sensor, a GARMIN GPS, a Point Grey stereo "Bumblebee" camera and a "Ladybug 2" panoramic camera. It carries small form factor PCs linked with a GBit internal network. Total onboard storage is of the order of 1TB.

Table 4 Summary of the salient properties of the two data sets used in this paper

Data Set Properties		
Name	Measure	Value
Begbroke	Size	9.3GB
	Laser	no
	Stereo	20Hz at 512 by 384 mono
	Omnicam	2Hz, 5 images color
	Distance Driven	1.08 km
	Sessions	single shot
New College	Size	Laser: 2.9 GB, Images 53GB
	Laser	2×75Hz over 90 degrees at 0.25 deg resolution
	Stereo	20Hz at 512 by 384 mono
	Omnicam	2Hz, 5 images color
	Distance Driven	5.13 km
	Sessions	multiple over three days



(a) Optimised trajectory (b) Rendered Laser

Fig. 7 The optimised trajectory of the first New College data set (2.3km) using visual constraints and ICP matching on laser data when stereo correspondences are insufficient to upgrade a topological loop closure (from FABMAP) to a metric interpose transformation and a complete "bird's-eye" view of the 1st New College data set with the map rendered from an optimised pose-graph.

5.2 Laser Map Generation

Our vehicle is equipped with two LMS 291 lasers mounted vertically on its sides. The lasers are set to 0.5 degree resolution resulting in an "angel wing" beam pattern. By capturing the intensity of the reflected laser pulses and careful time synchronisation (Tables 1 and 2 indicate the angular velocities experienced by our vehicle) we are able to generate detailed 3D point clouds. Figure 8 shows the typical detail produced in real time from our full 6DOF platform. Figure 7(b) shows a thinned point cloud of the entire 1st New College data set.

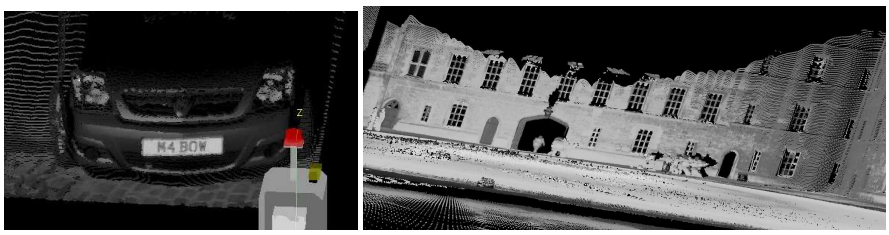


Fig. 8 Close detail of a point cloud built by rendering range and reflectance data from the estimated trajectory of moving Segway platform (New College data set).

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 will analyse 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, ${}^1p_{1:k}$, 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 ${}^{1:N}\mathcal{W}_i \quad i = 1:k$. We are assuming here that the maps being analysed 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. Firstly we calculate the normal ${}^j\hat{\mathbf{n}}_i$ of each wall patch ${}^j\mathscr{W}_i$ via an SVD of its scatter matrix and also the centroids ${}^j\mathbf{c}_i$, $j = 1:N \quad i = 1:k$. 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 5 presents statistics of these quantities.

Table 5 Analysis of the quality of New College Quad Point Cloud

Property	**Value**
maximum intra-cluster angle over all $\mathscr{W}$	9.1^o
minimum intra-cluster angle over all $\mathscr{W}$	0.32^o
maximum average intra-cluster angle over all $\mathscr{W}$	4.86^o
minimum average intra-cluster angle over all $\mathscr{W}$	0.66^o
average intra-cluster angle over all $\mathscr{W}$	3.6^o
maximum intra-cluster displacement over all $\mathscr{W}$	$0.6m$
minimum intra-cluster displacement over all $\mathscr{W}$	$0.02m$
maximum average intra-cluster displacement over all $\mathscr{W}$	$0.14m$
minimum average intra-cluster displacement over all $\mathscr{W}$	$0.33m$
average intra-cluster displacement over all $\mathscr{W}$	$0.21m$

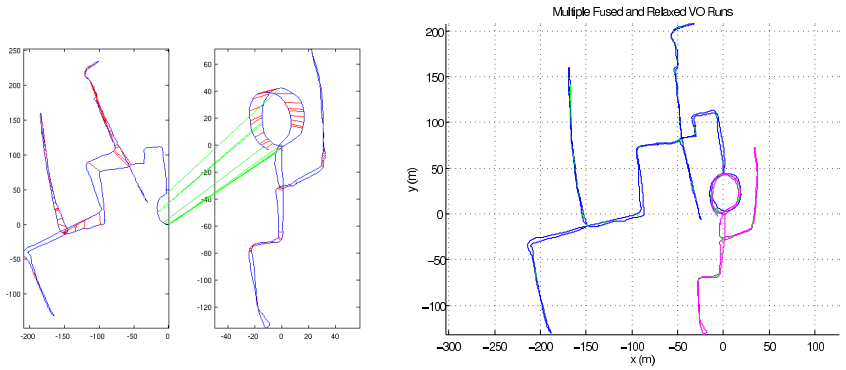
The results are promising although not perfect and this is an area requiring further work. In particular it would be advantageous and interesting to add extra constraints to the pose graph as a function of the measured quality of the maps - this is an area of current research.

6 Multi Session Mapping

The FABMAP 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 means loop closures can be found between data sets gathered days apart and because the operation is purely appearance-based, we need not worry about aligning metric coordinate frames. Figure 9(a) shows loop closures found between the 2nd and 3rd New College data sets.

Section 4 shows how the graph relaxation can be viewed as relaxing a chain of poses layed 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 9(b) shows the result of applying this technique to the co-joined trajectories shown in Figure 9(b).

The optimisation of our pose graphs is an offline process — it currently takes about 20 minutes to optimise a $50,000$ node graph with a few hundred loop clo-



(a) Link Discovery (b) Post Relaxation

Fig. 9 Loop closure links found within and *between* the second and third New College data sets. Inter-day loop closures are shown in pink. Relaxed multi-session trajectories between the second (blue) and third (pink) New College data sets. Note that fusion and relaxation is done with no manual alignment of coordinate frames — the alignment is automatically discovered by applying loop closure constraints.

sures. The question of finding the correct weighting between loop closure interpose constraints is a delicate one and needs further research. Certainly, one must model the correlations between linear and rotational motion for a non-holonomic vehicle. Also, if the optimisation 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 global minima) appears to requires a temporary increase in the cost $C(V|\mathscr{L},\mathscr{VO})$ as defined in Equation 6 — something gradient based optimisers are unable to do.

7 Conclusions and Future Plans

In this paper we have summarized the key components of an infrastructure free vision system that is able to recover consistent 6DOF trajectories over 1000's of meters in real time and also opportunistically fuse maps built on different days. Because our system contains topological and metric elements allows us to begin to address issue of life long mapping and learning about the workspace. The transparent discovery of inter-session alignments by FABMAP allows us to assemble ever more complex and substantial maps. Additionally the relative formulation of the output of the Sliding Window Filter [27] lends itself to both a purely relative formulation in which a single globally refined map is never explicitly calculated and also traditional single-frame metric maps like those shown in the results section. In the case of the latter it becomes important to assess in some way the quality of the map in the absence of ground truth. In the results section we presented initial work

considering how we can use the alignment and sharpness of detected planar surfaces in the workspace to provide meaningful metrics about map quality. This last point, we believe, speaks to a bigger issue — what needs to be done to enable life long learning in a truly meaningful sense? Certainly introspection is crucial and while several researchers are working on this problem, see in particular recent work by Tipaldi [30], much remains to be done – one cannot learn unless errors are detected. We also have an eye on how we might attempt to learn configuration policies for the sets of parameters which govern our often complicated subsystems by running them again and again, day after day benefitting from sporadic and opportunistic human input. We hope to show that given a prototype working system we can achieve an increase competency and longevity by smart, possibly dynamic modulation of system parameters. Persistence of navigation in both space and time is an interesting big problem and it goes well beyond researching the original SLAM problem — no bad thing because surely now SLAM is more "tool" than "problem".

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,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.)

References

1. A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer. A fast and incremental method for loop-closure detection using bags of visual words. *IEEE Transactions On Robotics, Special Issue on Visual SLAM*, 2008.
2. C. Bibby and I. Reid. Slam in dynamic environments with reversible data association. In *Robotics Science and Systems Conference*, 2007.
3. Cheng Chen and Han Wang. Appearance-based topological Bayesian inference for loop-closing detection in a cross-country environment. *The International Journal of Robotics Research*, 25(10):953–983, 2006.
4. C.K. Chow and C.N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, IT-14(3), May 1968.
5. Mark Cummins and Paul Newman. Probabilistic appearance based navigation and loop closing. In *Proc. IEEE International Conference on Robotics and Automation (ICRA'07)*, Rome, April 2007.
6. Mark Cummins and Paul Newman. Probabilistic appearance based navigation and loop closing. In *Proc. IEEE International Conference on Robotics and Automation (ICRA'07)*, Rome, April 2007.
7. Mark Cummins and Paul Newman. Accelerated appearance-only SLAM. In *Proc. IEEE International Conference on Robotics and Automation (ICRA'08)*, Pasadena,California, April 2008.
8. Mark Cummins and Paul Newman. FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. *The International Journal of Robotics Research*, 27(6):647–665, 2008.
9. Mark Cummins and Paul Newman. Highly scalable apppearance-only SLAM - FAB-MAP 2.0. In *Proc. Robotics Science And Systems*, Seattle, 2009. To appear.
10. D. Filliat. A visual bag of words method for interactive qualitative localization and mapping. *Robotics and Automation, 2007 IEEE International Conference on*, pages 3921–3926, April 2007.

11. G. Grisetti, C. Stachniss, S. Grzonka, and W. Burgard. A tree parameterization for efficiently computing maximum likelihood maps using gradient descent. In *Proceedings of Robotics: Science and Systems*, Atlanta, GA, USA, June 2007.
12. Giorgio Grisetti, Dario Lodi Rizzini, Cyrill Stachniss, Edwin Olson, and Wolfram Burgard. Online constraint network optimization for efficient maximum likelihood map learning. pages 1880–1885, 2008.
13. J. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, pages 318–325, Monterey, California, November 1999.
14. R. Cipolla J. Wang and Zha H. Vision-based global localization using a visual vocabulary. *Proceedings of Int. Conference on Robotics and Automation*, 2005.
15. Ben J. A. Kröse, Nikos A. Vlassis, Roland Bunschoten, and Yoichi Motomura. A probabilistic model for appearance-based robot localization. *Image and Vision Computing*, 19(6):381–391, 2001.
16. Pierre Lamon, Illah Nourbakhsh, Björn Jensen, and Roland Siegwart. Deriving and matching image fingerprint sequences for mobile robot localization. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Seoul, Korea, May 21-26 2001.
17. J. D. Tardós Lina María Paz, Pedro Piniés and J. Neira. Large-scale 6-dof slam with stereo-in-hand. *IEEE Transactions on Robotics*, 24(5):946–957, Oct. 2008.
18. L. Matthies and S. Shafer. Error modelling in stereo navigation. *IEEE Journal of Robotics and Automation*, 3(3):239–248, 1987.
19. Paul Newman, Gabe Sibley, Mike Smith, Mark Cummins, Alastair Harrison, Chris Mei, Ingmar Posner, Robbie Shade, Derik Schroeter, Liz Murphy, Winston Churchill, Dave Cole, and Ian Reid. Navigating, recognising and describing urban spaces with vision and laser. *The International Journal of Robotics Research (To Appear)*, 2009.
20. D. Nister, O. Naroditsky, and J. Bergen. Visual odometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–659, Washington, DC, 2004.
21. David Nister and Henrik Stewenius. Scalable recognition with a vocabulary tree. In *Conf. Computer Vision and Pattern Recognition*, volume 2, pages 2161–2168, 2006.
22. C. F. Olson, L. H. Matthies, M. Schoppers, and M. W. Maimone. Stereo ego-motion improvements for robust rover navigation. In *Proceedings of the IEEE Conference on Robotics and Automation*, pages 1099–1104, Washington, DC, 2001.
23. E. Olson, J. Leonard, and S. Teller. Fast iterative alignment of pose graphs with poor initial estimates. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2262–2269, 2006.
24. Grant Schindler, Matthew Brown, and Richard Szeliski. City-Scale Location Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7, 2007.
25. G. Sibley, G. Sukhatme, and L. Matthies. The iterated sigma point kalman filter with applications to long range stereo. In *Robotics: Science and Systems*, pages 263–270, 2006.
26. Gabe Sibley, Larry Matthies, and Gaurav Sukhatme. *A Sliding Window Filter for Incremental SLAM*. Springer Lecture Notes in Electrical Engineering, 2007.
27. Gabe Sibley, Christopher Mei, Ian Reid, and Paul Newman. Adaptive relative bundle adjustment. In *Robotics Science and Systems (RSS) (To Appear)*, Seattle, USA, June 2009.
28. J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proceedings of the International Conference on Computer Vision*, Nice, France, October 2003.
29. Sebastian Thrun and Michael Montemerlo. The graph slam algorithm with applications to large-scale mapping of urban structures. *Int. J. Rob. Res.*, 25(5-6):403–429, 2006.
30. Gian Diego Tipaldi. *Looking Inside for Mapping the Outside: Introspective Simultaneous Localization and Mapping*. PhD thesis, University of Rome La Sapienza, PhD Thesis, 2009. in press.
31. Jürgen Wolf, Wolfram Burgard, and Hans Burkhardt. Robust vision-based localization by combining an image-retrieval system with Monte Carlo localization. *IEEE Transactions on Robotics*, 21(2):208–216, 2005.