

Adaptive Relative Bundle Adjustment

Gabe Sibley, Christopher Mei, Ian Reid, Paul Newman

Department of Engineering Science

University of Oxford, OX1 3PJ, Oxford, UK

{gsibley, cmei, ian, pneman}@robots.ox.ac.uk

Abstract—It is well known that bundle adjustment is the optimal non-linear least-squares formulation of the simultaneous localization and mapping problem, in that its maximum likelihood form matches the definition of the Cramer Rao Lower Bound. Unfortunately, computing the ML solution is often prohibitively expensive – this is especially true during loop closures, which often necessitate adjusting all parameters in a loop. In this paper we note that it is precisely the choice of a single privileged coordinate frame that makes bundle adjustment costly, and that this expense can be avoided by adopting a completely relative approach. We derive a new relative bundle adjustment, which instead of optimizing in a single Euclidean space, works in a metric-space defined by a manifold. Using an adaptive optimization strategy, we show experimentally that it is possible to solve for the full ML solution incrementally in constant time – even at loop closure. Our system also operates online in real-time using stereo data, with fast appearance-based loop closure detection. We show results for sequences of 23k frames over 1.08km that indicate the accuracy of the approach.

I. INTRODUCTION

Bundle adjustment is the optimal solution to the so-called “full” simultaneous localization and mapping problem, in that it solves for the maximum likelihood solution given all measurements over all time. The goal in bundle adjustment is to minimize error between observed and predicted image-measurements of n 3D landmarks sensed from m 6D sensor poses (or frames). Measurements and parameter estimates are usually considered to be normally distributed, and the problem is typically tackled with non-linear least-squares optimization routines like Levenberg–Marquardt or the Gauss-Newton method. The linearized system matrix that appears in this process matches the form of the Fisher Information matrix, which in turn defines the Cramer Rao Lower Bound that is used to assess estimator consistency and optimality. It is not surprising therefore that bundle adjustment is the optimal non-linear least-squares simultaneous localization and mapping algorithm.

The cost of optimizing the bundle adjustment objective-function is cubic in complexity (in either m or n). For large and growing problems, this can quickly become prohibitive. This is especially true during loop-closure, when often all parameters in the loop must be adjusted. In a single coordinate frame, the farther the robot travels from the origin, the larger position uncertainty becomes. Errors at loop closure can therefore become arbitrarily large, which in turn makes it impossible to compute the *full* maximum likelihood solution in constant time (here the “full” solution is the one that finds the optimal estimates for all parameters).

It is not clear that it is *necessary* to estimate everything in a single coordinate frame – for instance most problems of autonomous navigation, such as path planning, obstacle avoidance or object manipulation, can be addressed within the confines of a *metric manifold*. Taking this route, we structure the problem as a graph of relative poses with landmarks specified in relation to these poses. In 3D this graph defines a connected Riemannian manifold with a distance metric based on shortest paths. Notice that this is not a sub-mapping approach, as there are no distinct overlapping estimates, and there is only one objective function with a *minimal* parameter vector; similarly, this is not a pose-graph relaxation approach, as it solves for landmark structure as well.

Together with an adaptive optimization scheme that only ever solves for a small sub-portion of the state vector, we find evidence that the full maximum likelihood solution in the manifold can be found using an incrementally constant time algorithm. Crucially, this appears true *even at loop closure*. We stress at the outset that the relative solution is not equivalent to the normal Euclidean-space solution and it does not produce an estimate that can be easily embedded in a single Euclidean frame. Converting from the relative manifold into a single Euclidean space is a difficult problem that we argue is best handled by external resources that do not have constant run-time requirements - e.g. by operator computers, not on the robot.

In the next section we describe the related literature. In Section III we derive the new relative objective function. Results from simulation and initial results on real sequences are presented in Section IV. We conclude with a discussion of the pros and cons of the relative approach.

II. RELATED WORK

There has been much interest in Gaussian non-linear least-squares solutions to SLAM based on “full-SLAM” or bundle adjustment [29][31][8][12][19], though the problem is an old one [3][22]. The full SLAM problem tries to optimize the joint vehicle trajectory and map structure simultaneously given all measurements ever made. There are approximate incremental solutions that only optimize a small local subset of the map [7], and there are methods that approximate the full solution with various forms of marginalization [19][27], or by ignoring small dependency information [30][21]. Recently some have successfully employed techniques from the linear algebra and numerical optimization communities to greatly reduce the cost of finding the full solution [17]. Many use key-frames to reduce complexity, though at the expense of accuracy

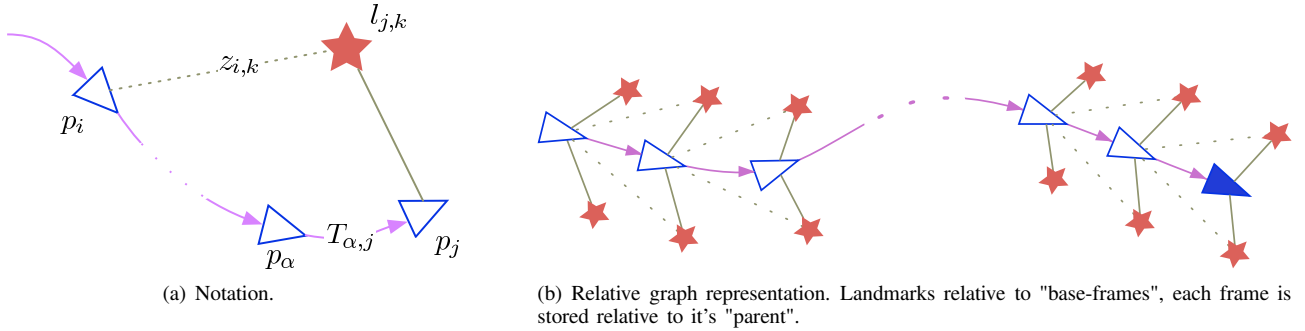


Figure 1. (a) Notation for a simple trajectory: poses are indicated with triangles, landmarks with red stars. Landmark base-frames are indicated with solid lines — e.g. here landmark k is stored relative to frame j . Each inter-pose edge in the graph includes an error-state transform defined about $t_j = 0$ — that is, $T_{\alpha,j} = \hat{T}_{\alpha,j} T_{(t_j)}$, where $\hat{T}_{\alpha,j}$ is the current estimate of the relative transform between frame α and frame j . Notice that predicting the measurement $z_{i,k}$ of landmark k in frame i will rely on all parameters in the kinematic chain from p_j to p_i . Figure (b) shows how landmarks are stored relative to the poses; clearly there is no reference to a privileged global coordinate frame.

[10][23][18]. All these techniques suffer from computational complexity issues during loop closures.

In the context of long-term autonomy, roboticists recognize the need for online, real-time, navigation and mapping algorithms. This means that localization and mapping algorithms must operate incrementally within a constant-time budget. Driven by this need, many authors have recognized the benefit of relative representations [2][9][19][1][15][4][13][20]. The most common solution is probably sub-mapping [2][25][6][9], which breaks the estimation into many smaller mapping regions, computes individual solutions for each region, and then estimates the relationships between these sub-maps. Many difficult issues arise in sub-mapping, including map overlap, data duplication, map fusion and breaking, map alignment, optimal sub-map size, and consistent global estimation in a single Euclidean frame. The relative bundle adjustment we propose can be seen as a *continuous* sub-mapping approach that avoids these complications.

To solve large SLAM problems with many loops, the most successful methods currently are the pose-graph optimization algorithms. Instead of solving the full SLAM problem, these methods optimize a set of relative pose constraints [24][14]. This is attractive because using forward substitution it is possible to transform full SLAM into a generally sparse set of pose constraints [11][29], and even to make the resulting system of equations relative [19]. Note that, given the assumed Gaussian problem structure, this kind of forward substitution to a pose-graph is algebraically equivalent to marginalization; methods that marginalize landmark parameters onto pose parameters so as to define a pose-graph are executing the forward substitution phase of sparse bundle adjustment. In this light, pose-graph relaxation, which solves for the optimal path estimate, can be seen as one-half of one iteration of full SLAM, because full SLAM also back-substitutes for the map parameters, and iterates the procedure to convergence. Like other methods, pose-graph solvers have worst-case complexity at loop closure that is dependent on the length of the loop.

The work most similar to relative bundle adjustment is the relative formulations given by Eade [9] and Konolige [19]. The former is akin to sub-mapping methods with constraints to enforce global Euclidean consistency at loop closure; the latter formulates the cost function relative to a single Eu-

clidean frame and then makes a series of approximations to produce a sparse relative pose-graph. Neither method derives the purely relative objective function (incrementally, both rely on some form of single-reference frame), neither formulates the objective function completely without privileged frames, and both methods carry the burden of finding a globally consistent estimate in a single Euclidean frame. Our approach is substantially different because of the completely relative underlying objective function that we derive.

Finally, a number of adaptive region approaches have been explored within the privileged Euclidean frame paradigm [28][26]. These techniques, together with all of the methods presented in this section, are not constant time at loop closure, and all but one [2] solve for a solution in a single Euclidean space. We find that using adaptive region estimation in conjunction with the relative formulation is the key that enables constant time operation.

III. METHODS

Instead of optimizing an objective function parameterized in a single privileged coordinate frame, we now derive a completely relative formulation.

A. Problem Formulation

Bundle adjustment seeks to minimize error between the observed and predicted measurements of n landmarks sensed from m sensor poses (or frames). Likewise we minimize the difference between predicted and measured values. Let $l_{j,k}$, $k \in 1, \dots, n$, $j \in 1, \dots, m$ be a set of n 3D landmarks each parameterized relative to some *base-frame* j . Let t_j , $j \in 1, \dots, m$ be a set of m 6D relative pose relationships associated with edges in an undirected graph of frames. The graph is built incrementally as the vehicle moves through the environment, and extra edges are added during loop closure. The graph defines a connected Riemannian manifold that is by definition everywhere locally Euclidean, though globally it is not embedded in a single Euclidean space. The relationship between parent-frame α and child-frame j is defined by a 4×4 homogeneous transform matrix, $T_{\alpha,j} = \hat{T}_{\alpha,j} T_{(t_j)}$, where $\hat{T}_{\alpha,j}$ is the current estimate and $T_{(t_j)}$ is the 4×4 homogeneous

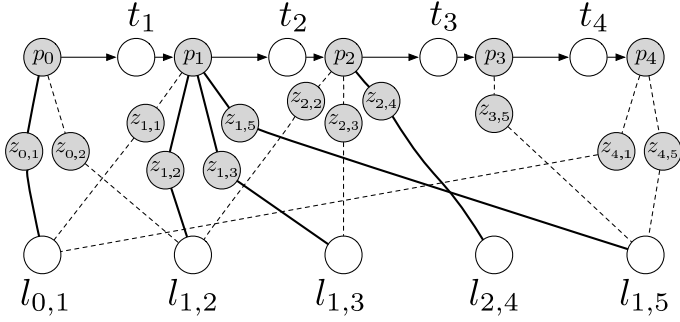


Figure 2. Graphical example for the sequence of 12 observations, $\{z_{0,1}, z_{1,1}, z_{4,1}\}$, $\{z_{0,2}, z_{1,2}, z_{2,2}\}$, $\{z_{1,3}, z_{2,3}\}$, $\{z_{2,4}\}$, $\{z_{1,5}, z_{3,5}, z_{4,5}\}$. There are five poses, p_0, \dots, p_4 , four edge estimates t_1, \dots, t_4 , and five landmarks $l_{0,1}, l_{1,2}, l_{1,3}, l_{2,4}$ and $l_{1,5}$. This example has the Jacobian $H = \frac{\partial h}{\partial x}$ that is depicted in Figure 3. Bold lines from poses indicate which frames are base-frames.

matrix defined by t_j . An example trajectory and graph with this notation is shown in Figure I.

Each t_j parameterizes an infinitesimal delta transform applied to the relationship from its parent frame in the graph (i.e. an error-state formulation). The kinematic chain from frame j to frame i is defined by a sequence of 4×4 homogeneous transforms

$$T_{ji} = \hat{T}_{j,j+1} T_{(t_{j+1})} \hat{T}_{j+1,j+2} T_{(t_{j+2})}, \dots, \hat{T}_{i-1,i} T_{(t_i)};$$

the sensor model for a single measurement is

$$\begin{aligned} h_{i,k}(l_{j,k}, t_i, \dots, t_j) &= Proj(T_{j,i}^{-1} l_{j,k}) \\ &= Proj(g_{i,k}(l_{j,k}, t_{j+1}, \dots, t_i)) \end{aligned}$$

where $g_{i,k} : \mathbb{R}^{\dim(x)} \rightarrow \mathbb{R}^4$, $x \mapsto T_{j,i}^{-1} l_{j,k}$ transforms the homogeneous point $l_{j,k}$ from base-frame j to the observation frame i . This describes how landmark k , stored relative to base-frame j , is transformed into sensor frame i and then projected into the sensor. We make the usual assumption that measurements $z_{i,k}$ are normally distributed: $z_{i,k} \sim N(h_{i,k}, R_{i,k})$. The cost function we associate with this formulation is

$$\begin{aligned} J &= \sum_{k=1}^n \sum_{i \in \mathcal{I}}^{m_k} (z_{i,k} - h_{i,k}(x))^T R_{i,k}^{-1} (z_{i,k} - h_{i,k}(x)) \quad (1) \\ &= \|z - h(x)\|_{R^{-1}}, \quad (2) \end{aligned}$$

which depends on the landmark estimate, $l_{j,k}$ and *all the transform estimates t_{j+1}, \dots, t_i on the kinematic chain from the base-frame j to the measurement-frame i* . This problem is solved using iterative non-linear least-squares Gauss-Newton minimization for the values of x that minimize re-projection error — this yields the *maximum likelihood* estimate (subject to local minima). Projecting via kinematic chains like this is novel, but it changes the sparsity patterns in the system Jacobian. Compared to normal bundle adjustment, this new pattern increases the cost of solving the sparse normal equations for updates δx to the state vector x — though as we will see, the ultimate computational complexity is the same.

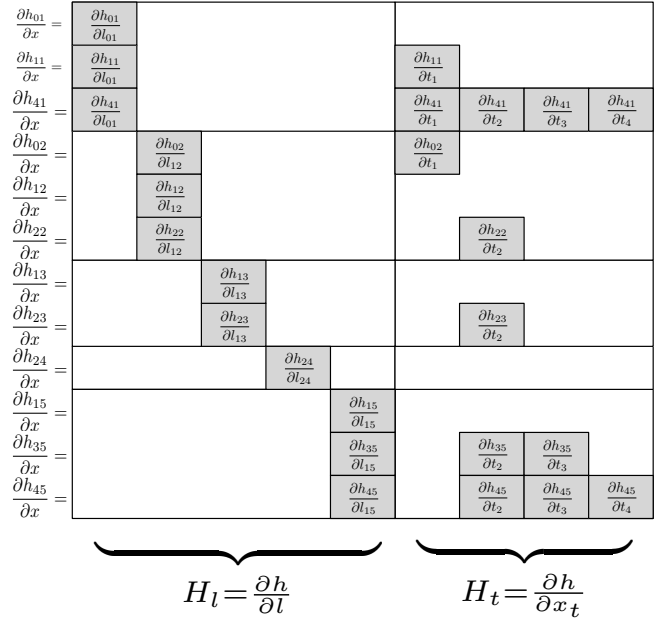


Figure 3. Example relative bundle adjustment Jacobian structure for the sequence of 12 observations in Figure 2. Grey indicates non-zero entries. The horizontal stripes in the right hand H_t term above correspond to projections that rely on transforming state estimates along kinematic chains from frame j to frame i . These stripes are the only difference in sparsity pattern between the relative formulation and traditional bundle adjustment.

B. Sparse Solution

The *normal equations* associated with the iterative non-linear least squares Gauss-Newton solution to equation (1) are

$$H^T R^{-1} H \delta x = H^T R^{-1} (z - h(x)). \quad (3)$$

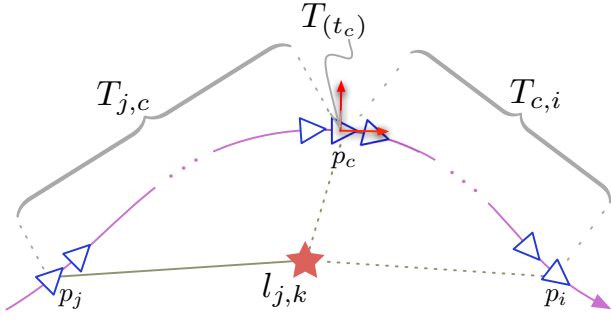
where $H = \frac{\partial h}{\partial x}$ is the Jacobian of the sensor model, R is the block diagonal covariance matrix describing the uncertainty of the collective observation vector z (the stacked vector of all measurements). Referring to the example in Figure 3 we see that $H^T = [H_l^T H_t^T]$ and $\delta x = [\delta l; \delta t]$, which exposes a well known 2×2 block structure for equation (3),

$$\begin{bmatrix} V & W \\ W^T & U \end{bmatrix} \begin{bmatrix} \delta l \\ \delta t \end{bmatrix} = \begin{bmatrix} r_l \\ r_t \end{bmatrix},$$

where δl and δt are state updates for the map and edge transforms that we are solving for; $r_l = H_l^T R^{-1} (z - h(x))$, $r_t = H_t^T R^{-1} (z - h(x))$, $V = H_l^T R^{-1} H_l$, $W = H_l^T R^{-1} H_t$, and $U = H_t^T R^{-1} H_t$. Building this linear system is the dominant cost in solving each iteration, which makes it important to compute the sparse Jacobian of h efficiently.

C. Relative Jacobians

Due to the functional dependence of the projection model on the kinematic chain of relative poses, the Jacobians in the relative formulation are very different from their Euclidean counterpart. With reference to Figure 4, focus for a moment on a single infinitesimal transform $T_{(t_c)}$ that is somewhere



$$g_{i,k}(x) = (T_{j,c}T_{(t_c)}T_{c,i})^{-1}l_{j,k}$$

Figure 4. This diagram shows the sensor following a path through p_j and p_i while making measurements of landmark $l_{j,k}$ (indicated with dashed lines). Landmark k is stored relative to frame j (indicated by a solid line). To compute the projection of landmark k in frame i , we evaluate $h_{i,k} = Proj(g_{i,k}(x))$, where $g_{i,k}(x) = T_{j,i}^{-1}l_{j,k} = (T_{j,c}T_{(t_c)}T_{c,i})^{-1}l_{j,k}$, which encapsulates projection along the kinematic chain between frame j and frame i . To help understand how the relative formulation Jacobian is computed, this diagram focuses on the the error-state transform $T_{(t_c)}$ indicated in red. The state-vector terms of interest when computing derivatives are 1) the transform parameters t_c , and 2) the landmark parameters $l_{j,k}$.

along the kinematic chain from frame i to j . The individual derivatives shown in Figure 3 are

$$\frac{\partial h_{i,k}}{\partial l_{j,k}} = \frac{\partial k}{\partial g_{i,k}} \frac{\partial g_{i,k}}{\partial l_{j,k}},$$

and

$$\frac{\partial h_{i,k}}{\partial t_c} = \frac{\partial k}{\partial g_{i,k}} \frac{\partial g_{i,k}}{\partial t_c}$$

where $\frac{\partial k}{\partial g_{i,k}}$ is the Jacobian of the perspective projection function (using the standard K intrinsic camera calibration matrix).

The Jacobian of $g_{i,k}$ with respect to the 3D point $l_{j,k}$ is

$$\frac{\partial g_{i,k}}{\partial l_{j,k}} = \begin{bmatrix} R_{i,j} \\ 0 \end{bmatrix}.$$

The Jacobian of $g_{i,k}$ with respect to t_c has three cases that depend on the direction of the transform $T_{(t_c)}$ on the path from frame i to j

$$\frac{\partial g_{i,k}}{\partial t_c} = \begin{cases} T_{i,c} \frac{\partial T_{(t_c)}}{\partial t_c} T_{c,j} l_{j,k} & \text{if } T_{(t_c)} \text{ points towards } j \\ T_{i,c} \frac{\partial T_{(-t_c)}}{\partial t_c} T_{c,j} l_{j,k} & \text{if } T_{(t_c)} \text{ points towards } i \\ 0 & \text{if } i = j \end{cases}$$

and $\frac{\partial T_{(t_c)}}{\partial t_c}$ is the canonical generators of $SE(3)$ (a $4 \times 4 \times 6$ tensor). We now address the cost of solving each update.

D. Complexity of computing the relative sparse solution

Similar to sparse bundle adjustment, the following steps are used to exploit the structure of H to compute the *normal equations* and state-updates efficiently:

- 1) *Build linear system*, computing the terms U , V , W , r_t , and r_l . Complexity is $O(m^3n)$.
- 2) *Forward substitute*, computing $A = U - W^T V^{-1} W$, and $b = r_t - W^T V^{-1} r_l$. Complexity is $O(m^2n)$.
- 3) *Solve reduced system* of equations, $A\delta t = b$ for the update δt . Complexity is $O(m^3)$.
- 4) *Back substitute* to solve for the map update, $\delta l = V^{-1}(r_l - W\delta t)$. Complexity is $O(mn)$.

The first step is completely different in the relative framework so we describe it in more detail in algorithm 1. The overall complexity is $O(m^3)$, which matches traditional sparse bundle adjustment. Note that it is easy to convert algorithm 1 into a robust m -estimator by replacing the weights, $w_{i,k}$, with robust weight kernels, $w_{i,k} = R_{i,k}^{-1} \mathcal{W}(e_{i,k})$ — for example we use the Huber kernel [16]. Section IV gives results of applying this sparse optimization routine to large simulated and real sequences.

algorithm 1 Build linear system. Computes U , V , W , r_t , and r_l in $O(m^3)$

Clear U , V , W , r_t , and r_l

for all landmarks k **do**

for all frames i with a measurement of landmark k **do**

 Compute $\frac{\partial h_{i,k}}{\partial l_{j,k}}$

$$e_{i,k} = z_{i,k} - h_{i,k}(x)$$

$$w_{i,k} = R_{i,k}^{-1}$$

$$V_k += \frac{\partial h_{i,k}}{\partial l_{j,k}}^T w_{i,k}^{-1} \frac{\partial h_{i,k}}{\partial l_{j,k}}$$

$$r_{l_k} += \frac{\partial h_{i,k}}{\partial l_{j,k}}^T w_{i,k}^{-1} e_{i,k}$$

for all $p \in Path(i, j)$ **do**

 Compute $\frac{\partial h_{i,k}}{\partial t_p}$

$$r_{t_p} += \frac{\partial h_{i,k}}{\partial t_p}^T w_{i,k} e_{i,k}$$

$$W_{k,p} += \frac{\partial h_{i,k}}{\partial l_{j,k}}^T w_{i,k} \frac{\partial h_{i,k}}{\partial t_p}$$

for all $q \in Path(p, j)$ **do**

 Compute $\frac{\partial h_{i,k}}{\partial t_q}$

$$U_{p,q} += \frac{\partial h_{i,k}}{\partial t_p}^T w_{i,k} \frac{\partial h_{i,k}}{\partial t_q}$$

$$U_{q,p} += \frac{\partial h_{i,k}}{\partial t_q}^T w_{i,k} \frac{\partial h_{i,k}}{\partial t_p}$$

end for

end for

end for

end for

Finally, notice that if feature tracks are contiguous over numerous frames (which they typically are), then the sparsity pattern in W will be the same in the relative-formulation as it is in the traditional one – hence the relative-formulation cost of forward-substitution, solving the reduced system, and back-substitution (steps 2-4) should be approximately equivalent.

E. Adaptive Updates

To reduce computation, it is important to optimize only those parameters that might change in light of new information [26][28]. Below we outline one approach to limit the parameters that are actively optimized.

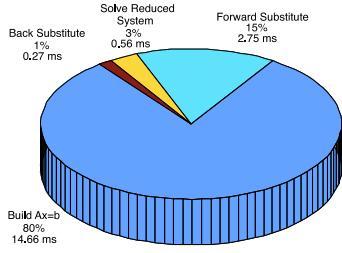


Figure 5. Average run-times for the main steps of relative bundle adjustment on an Intel Core 2 Duo 2.8GHz processor. The average adaptive region from the Monte Carlo simulation was 4.6 frames. Note that it is the cost of building the linear system of equations that dominates the cubic complexity of solving for the adaptive region of poses.

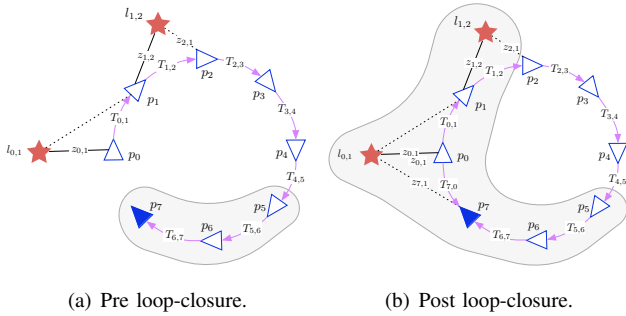


Figure 6. Discovery of local active region. In (a), re-projection errors have changed by more than $\Delta\epsilon$ in the local frames p_5 , p_6 , and p_7 . In (b) a new edge $T_{7,0}$ is added during loop-closure, and the graph search leads to a larger active region with frames p_0 , p_1 , p_5 , p_6 , and p_7 .

A breadth-first-search from the most recent frame is used to discover local parameters that might require adjustment. During the search, all frames in which the average re-projection error changes by more than a threshold, $\Delta\epsilon$, are added to an *active region* that will be optimized. The search stops when no frame being explored has a change in re-projection error greater than $\Delta\epsilon$. Landmarks visible from active frames are activated, and all non-active frames that have measurements of these landmarks are added to a list of static frames, which forms a slightly larger set we call the static region. Measurements made from static frames are included in the optimization, but the associated relative pose-error parameters are not solved for. Example active regions are shown in Figure 6.

IV. RESULTS

The iterative nonlinear least-squares solution that exploits the sparse relative structure and the four steps in section III-D results in the run-time break-down shown in Figure 5. This illustrates that building the sparse system of equations is the dominant cost.

A. Simulation results

To determine the performance of the relative framework, a batch of Monte Carlo simulations were run. The sequence contains a realistic trajectory, landmark distribution, and a 1 pixel standard deviation Gaussian measurement noise (see Figure 7).

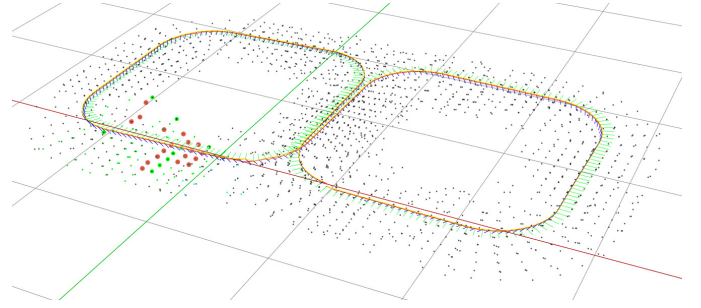


Figure 7. Figure-of-eight sequence used in Monte Carlo simulation. This sequence has 288 frames, 3,215 landmarks and 12,591 measurements with 1 pixel standard deviation Gaussian measurement noise added.

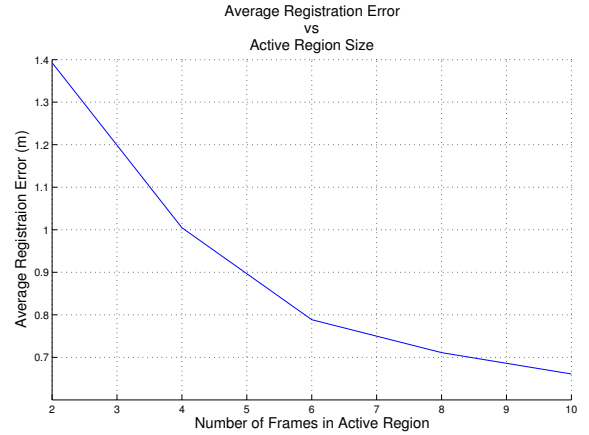
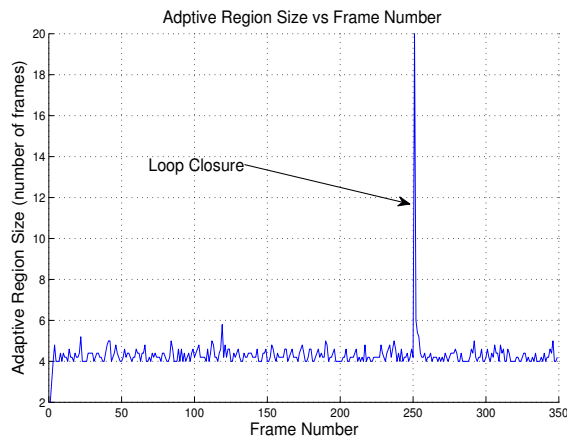


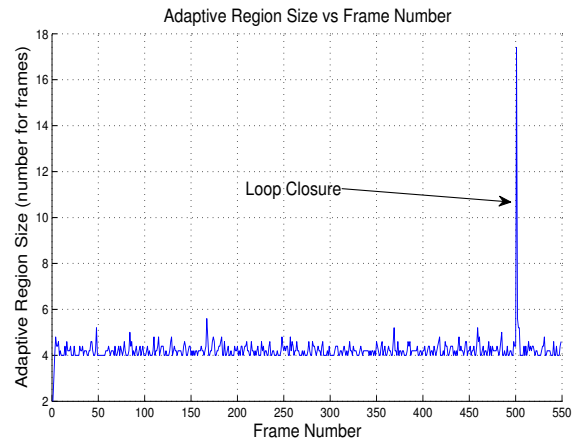
Figure 8. Average Registration Error vs. Number of Frames being updated. In the relative formulation, as the local region grows the average RMS error drops quickly toward the same as when computed with all frames active. This motivates the use of an adaptive region that allows parameters to vary only if it has an effect on the cost function.

We compute errors in the following way: for each pose in the trajectory, we register that pose to its ground truth counterpart, and then *localize* the rest of the relative trajectory in that frame. Note that “localizing” the relative trajectory is done with a breadth-first-search that computes each frame’s pose in the coordinate system of the root frame; this process projects from the relative manifold into a single Euclidean frame, and may cause “rips” to appear at distant loop closures. Finally, the total trajectory registration error is computed as the average Euclidean distance between ground truth and the localized frames. The average of all frames and all registrations is the error plotted. Not surprisingly, initial results in Figure 8 indicate that error reduces towards the full solution (in the relative space) as the local region increases in size.

The results here use an adaptive region threshold of $\Delta\epsilon=0.05$ pixels. With this threshold we find that the discovery of new frames to include in the active region quickly drops to between 4 and 5 poses, except at loop closure where it jumps to accommodate the larger region of poses found by the breadth-first-search. Figure 9 shows the adaptive region size discovered for two different loop closures, one 50m long and another 100m long. The point to note is that the discovered adaptive region is independent of loop size, and that errors do not propagate around the loop even though loop closure



(a) 50m loop with closure at frame 250.



(b) 100m loop with closure at frame 500.

Figure 9. This figure shows how the number of frames in the adaptive region fluctuates over time and during loop closure. During loop closure the size of the adaptive region jumps to accommodate all the local frames that have been added to the active region, as well as any neighboring frames that will be affected. Notice that errors do not propagate all the way around the loop, and only a fraction of the state vector needs to be updated. Loop closure at 250 and 500 frames induces updates in approximately the same number of parameters, which strongly indicates that optimization at loop closure will remain constant time, independent of loop size. Before loop closure, the average metric position error is over 75cm for the 500 frame loop. Using the same adaptive region criteria, Euclidean bundle adjustment would require adjusting *all* parameters in the loop - whereas the adaptive relative approach only adjusts 20 poses.

error is ~ 75 cm on average for the 500 frame sequence. Using the same adaptive region criteria, Euclidean bundle adjustment would require adjusting *all* parameters in the loop - whereas the adaptive relative approach adjusts just 20 poses.

Our adaptive strategy for discovering the active region is designed to have a rippling effect: when parameter estimates change, it effects the re-projection error in nearby frames, which, if greater than $\Delta\epsilon$, will add those parameters to the active region, potentially causing them to change... etc. A key result of the relative formulation is that these errors *stop propagating* and balance out with distance from the new information - that is, the network of parameters is critically damped.

B. Real Data

The system operates online at 20-40Hz, this includes all image processing, feature tracking, robust initialization routines, and calls to FABMAP[5] to detect loop closures. We have run it successfully on sequences with up to 110K frames over tens of kilometers. Figure 10 shows a 1.08 kilometer trajectory computed from 23K frames. Table I gives an indication of typical system performance.

V. DISCUSSION

The privileged-frame approach and the relative formulations are very different; their objective functions are different and they solve for different quantities. The former embeds the trajectory in a single Euclidean space; the latter in a connected Riemannian manifold. At first reading it may appear that the lack of a simple Euclidean distance metric between two points, and the fact that we cannot render the solution very easily, is a disadvantage of the relative formulation. Note however that the manifold is a metric space, and distance between two points can be computed from shortest paths in the graph. With this in mind, the relative representation should *still* be amenable to

	Science Park		
	Avg.	Min.	Max.
Distance Traveled (km)	—	—	1.08
Frames Processed	—	—	23,268
Velocity (m/s)	0.93	0.0	1.47
Angular Velocity (deg/s)	9.49	0.0	75.22
Frames Per Second	22.2	10.6	31.4
Features per Frame	93	44	143
Feature Track Length	13.42	2	701
Re-projection Error	0.17	$2.74 \times 1e-3$	0.55

Table I
TYPICAL PERFORMANCE OF OUR ONLINE SYSTEM FOR THE BEGBROKE SCIENCE PARK DATA SET PROCESSED ON AN INTEL CORE 2 DUO 2.8GHZ.

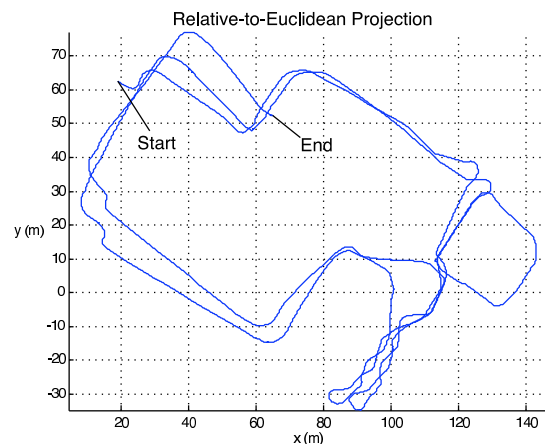


Figure 10. 1.08km path over 23K frames estimated for the Begbroke Science Park sequence. Table I shows typical performance results.

planning algorithms which are commonly defined over graphs in the first place. Furthermore, because the manifold is (by definition) locally Euclidean, algorithms that require precise *local* metric estimates, such as obstacle avoidance or object manipulation, can operate without impediment.

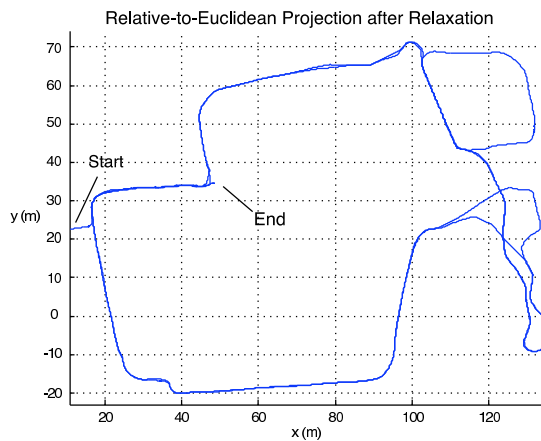


Figure 11. This figure shows a globally consistent relaxed view of the Begbroke Science Park sequence. To view relative estimates in a consistent fashion (single global frame) we have to transform from the relative representation to a single Euclidean coordinate system. The sequence here has 23K poses over 1.08 kilometers which makes the conversion computationally expensive. This relative-to-global transformation process is designed to run on the user interface, *not* on the robot.

We posit that a *topometric* relative formulation is sufficient for many mobile robot navigation tasks, and that a single global Euclidean representation is rarely necessary. Certainly the benefits afforded by incrementally constant-time performance are tremendous, and in the light of that, some inconvenience may be acceptable. If a unified global Euclidean picture is deemed essential by a particular external application or technique, our choice would be to push responsibility for generating the single Euclidean embedding into that process - for example undertaking fast approximate pose-graph relaxation in order to render consistent results in a user interface [24], [14].

A. Rendering in a single Euclidean space

As an example, Figure 11 shows the result of transforming a large relative state estimate into a single Euclidean frame using pose-graph relaxation. Note that even this state-of-the-art global Euclidean estimate fails to discover the true rectilinear structure. Arguably the best way to improve the map would be to *schedule* new measurements across the diagonal of the map, thereby considerably constraining the solution. While this interventionist approach is used extensively in surveying, we are not comfortable with placing such a requirement on a mobile platform — ideally navigation and mapping should be a quiet background task producing estimates for consumption by any interested client process. With this example in mind, perhaps accurate global Euclidean state estimates are the wrong goal to aim for — what matters is relative metric accuracy and topological consistency — all of which can be attained with a relative manifold approach.

VI. CONCLUSION

The fact that the variables in bundle adjustment are defined relative to a single coordinate frame has a large impact on the algorithm’s iterative convergence rate. This is especially true at loop closure, when large errors must propagate around the

entire loop to correct for global errors that have accumulated along the path. As an alternative, we have presented an adaptive relative formulation that can be viewed as a *continuous* sub-mapping approach – in many ways our relative treatment is an intuitive simplification of previous sub-mapping methods. Furthermore by solving all parameters within an adaptive region, the proposed method attempts to match the full maximum likelihood solution within the metric space defined by the manifold. In stark contrast to traditional bundle adjustment, our evaluations and results indicate that state updates in the relative approach are constant time, and crucially, remain so even during loop closure events.

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). The work reported in this paper undertaken by the Active Vision Lab acknowledges the support of EPSRC grant GR/T24685/01.

REFERENCES

- [1] J. Blanco, J. Fernandez-Madriral, and J. Gonzalez. Toward a unified bayesian approach to hybrid metric-topological SLAM. *IEEE Transactions on Robotics and Automation*, 24(2):259–270, 2008.
- [2] M. C. Bosse, P. M. Newman, J. J. Leonard, and S. Teller. SLAM in large-scale cyclic environments using the atlas framework. *International Journal of Robotics Research*, 23(12):1113–1139, December 2004.
- [3] D.C. Brown. A solution to the general problem of multiple station analytical stereotriangulation. Technical report, RCP-MTP Data Reduction Technical Report No. 43, Patrick Air Force Base, Florida (also designated as AFMTC 58-8), 1958.
- [4] L. A. Clemente, A. J. Davison, I. Reid, J. Neira, and J. D. Tardos. Mapping large loops with a single hand-held camera. In *Robotics: Science and Systems*, 2007.
- [5] M. Cummins and P. Newman. Probabilistic appearance based navigation and loop closing. In *IEEE Conference on Robotics and Automation*, 2007.
- [6] A. Davison, I. Reid, N. Molton, and O. Stasse. MonoSLAM: Realtime single camera SLAM. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 29(6):1113–1139, 2007.
- [7] M. C. Deans. *Bearings-Only Localization and Mapping*. PhD thesis, School of Computer Science, Carnegie Mellon University, 2005.
- [8] F. Dellaert. Square root SAM. In *Proceedings of Robotics: Science and Systems*, pages 1181–1203, 2005.
- [9] E. Eade and T. Drummond. Unified loop closing and recovery for real time monocular SLAM. In *Proceedings British Machine Vision Conference*, September 2008.
- [10] C. Engels, H. Stewenius, and D. Nister. Bundle adjustment rules. In *Photogrammetric Computer Vision*, 2006.
- [11] R. Eustice, H. Singh, J. Leonard, M. Walter, and R. Ballard. Visually navigating the RMS Titanic with SLAM information filters. In *Robotics: Science and Systems*, pages 57–64, 2005.
- [12] A. W. Fitzgibbon and A. Zisserman. *Automatic Camera Recovery for Closed or Open Image Sequences*. Springer, 2004.
- [13] U. Frese and T. Duckett. A multigrid approach for accelerating relaxation-based SLAM. In *Proceedings IJCAI Workshop on Reasoning with Uncertainty in Robotics (RUR 2003)*, pages 39–46, Acapulco, Mexico, 2003.
- [14] G. Grisetti, C. Stachniss, S. Grzonka, and W. Burgard. A tree parameterization for efficiently computing maximum likelihood maps using gradient descent. In *Proceedings Robotics: Science and Systems*, 2007.
- [15] J.E. Guivant and E.M. Nebot. Optimization of the simultaneous localization and map-building algorithm for real-time implementation. *IEEE Transactions on Robotics and Automation*, 17(3):242–257, June 2001.

- [16] P. J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(2):73–101, 1964.
- [17] M. Kaess. *Incremental Smoothing and Mapping*. PhD thesis, Georgia Institute of Technology, 2008.
- [18] G. Klein and D. Murray. Improving the agility of keyframe-based SLAM. In *European Conference on Computer Vision*, 2008.
- [19] K. Konolige and M. Agrawal. FrameSLAM: from bundle adjustment to realtime visual mapping. *IEEE Transactions on Robotics and Automation*, *IEEE Journal of Robotics and Automation*, *International Journal of Robotics Research*, 24(5):1066–1077, 2008.
- [20] A. Martinelli, V. Nguyen, N. Tomatis, and R. Siegwart. A relative map approach to SLAM based on shift and rotation invariants. *Robotics and Autonomous Systems*, 55(1):50–61, 2007.
- [21] P. F. McLauchlan. The variable state dimension filter applied to surface-based structure from motion. Technical report, University of Surrey, 1999.
- [22] E. M. Mikhail. *Observations and Least Squares*. Rowman & Littlefield, 1983.
- [23] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyse, and P. Sayd. Real time localization and 3d reconstruction. In *Proceedings of Computer Vision and Pattern Recognition*, 2006.
- [24] 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.
- [25] P. Pinies and J. D. Tardos. Scalable slam building conditionally independent local maps. In *IEEE conference on Intelligent Robots and Systems*, 2007.
- [26] A. Ranganathan, M. Kaess, and F. Dellaert. Loopy SAM. In *International Joint Conferences on Artificial Intelligence*, pages 2191–2196, 2007.
- [27] G. Sibley, L. Matthies, and G. Sukhatme. *A Sliding Window Filter for Incremental SLAM*. Springer Lecture Notes in Electrical Engineering, 2007.
- [28] D. Steedly and I. Essa. Propagation of innovative information in non-linear least-squares structure from motion. In *ICCV01*, pages 223–229, 2001.
- [29] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, Cambridge, MA, 2005.
- [30] S. Thrun, D. Koller, Z. Ghahmarani, and H. Durrant-Whyte. SLAM updates require constant time. In *Workshop on the Algorithmic Foundations of Robotics*, December 2002.
- [31] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment – A modern synthesis. In W. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, LNCS, pages 298–375. Springer Verlag, 2000.