Vast-scale Outdoor Navigation Using Adaptive Relative Bundle Adjustment

Gabe Sibley, Christopher Mei, Ian Reid and Paul Newman

The International Journal of Robotics Research 2010 29: 958 originally published online 4 May 2010
DOI: 10.1177/0278364910369268

Gabe Sibley
Christopher Mei
Ian Reid
Paul Newman

Department of Engineering Science,
University of Oxford, Oxford OX1 3PJ, UK
{gsibley, cmei, ian, pnewman}@robots.ox.ac.uk

Vast-scale Outdoor Navigation Using Adaptive Relative Bundle Adjustment

Abstract

In this paper we describe a relative approach to simultaneous localization and mapping, based on the insight that a continuous relative representation can make the problem tractable at large scales. First, it is well known that bundle adjustment is the optimal non-linear least-squares formulation for this problem, in that its maximum-likelihood form matches the definition of the Cramer–Rao lower bound. Unfortunately, computing the maximum-likelihood 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 maximum-likelihood solution incrementally in constant time, even at loop closure. Our approach is, by definition, everywhere locally Euclidean, and we show that the local Euclidean estimate matches that of traditional bundle adjustment. Our system operates online in realtime using stereo data, with fast appearance-based loop closure detection. We show results on over 850,000 images that indicate the accuracy and scalability of the approach, and process over 330 GB of image data into a relative map covering 142 km of Southern England. To demonstrate a baseline sufficiency for navigation, we show that it is possible to find shortest paths in the relative maps we build, in terms of both time and distance. Query images from the web of popular landmarks around London, such as the London Eye or Trafalgar Square, are matched to the relative map to provide route planning goals.

The International Journal of Robotics Research
Vol. 29, No. 8, July 2010, pp. 958–980
DOI: 10.1177/0278364910369268

KEY WORDS—robotics, visual SLAM, stereo mapping, bundle adjustment

1. Introduction

Bundle adjustment is the optimal non-linear least-squares solution to the "full" simultaneous localization and mapping problem (SLAM), in that it solves for the maximum-likelihood solution given all measurements over all time (Triggs et al. 2000). The goal in bundle adjustment is to minimize error between observed and predicted image-measurements of n three-dimensional (3D) landmarks sensed from m 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 that make use of the *normal equations* (Sorenson 1980). The linearized system matrix that appears in this process is recognized as the Fisher information matrix, which in turn defines the Cramer–Rao lower bound that is used to assess estimator consistency and optimality. A consequence is that bundle adjustment is the optimal non-linear least-squares SLAM algorithm.

The cost of optimizing the bundle adjustment objective function is cubic in complexity (in either m, the number of frames, or n, the number of landmarks). Even if sparsity in the problem-structure can be exploited (Krauthausen et al. 2006; Agarwal et al. 2009), for large and growing problems, the cost can quickly prohibit realtime solutions. This is especially true during loop closure, when 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 be arbitrarily large, which in turn make 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).

There is no imperative to estimate everything in a single coordinate frame; for instance, most problems of autonomous

Fig. 1. The London dataset. This shows the 121-km path taken between Oxford in the upper left and London in the bottom right. We compute visual estimates for 89.4% of this trajectory and fall back on inertial sensing for the remainder. Loops are closed using appearance-based place recognition (Cummins and Newman 2008). The graph begins in an office in Oxford, and proceeds with various forms of transport including: foot, bicycle, train, subway, lift, escalator, rickshaw, punt and ferris wheel. Note that, in the presence of sensor drift and noise, we cannot accurately estimate true position in the global frame. Such situations are common in practice, for instance, when traveling on a train or subway.

navigation, such as path planning, obstacle avoidance or object manipulation, can be addressed within the confines of a manifold. Taking this route, we structure the problem as a graph of relative poses with landmarks specified in relation to these poses. In three dimensions this graph defines a connected Riemannian manifold with a distance metric based on shortest paths. Note that this is not a sub-mapping approach (Bosse et al. 2004; Eade and Drummond 2008), 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 (Olson et al. 2006; Grisetti et al. 2007), 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 parameter vector, we show the relative maximum-likelihood estimate (MLE) solution in the manifold can be closely approximated using a constant-time incremental 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 as 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.

We have applied the relative approach to ∼850,000 frames of stereovision data gathered in and around Oxford and London, England (see Figures 1, 2 and 3). The data begin in an office in Oxford, and proceed with various forms of transport including: foot, bicycle, train, subway, lift, escalator, rickshaw, punt and ferris wheel. Note that many of these transport modes constitute unobservable moving reference frames that simply cannot be handled in a conventional monolithic single-Euclidean-frame approach, a point to which we will return later.

In the presence of sensor noise and drift, moving reference frames make it impossible to accurately estimate global position. For instance, when traveling on a train or subway, motion with respect to the global frame is effectively unobservable. In contrast, we show that the relative approach can handle such difficulties, and produces topometric maps that are still useful for navigation. To highlight this, we show sequences in which current state-of-the art global SLAM solutions fail due to unsensed ego-motion, yet optimization in the relative representation proceeds without hindrance.

To demonstrate that the continuous relative representation is sufficient for navigation, we show that it is possible to find shortest paths in the relative maps we build – both in terms of time and distance. Query images from Google image-search of popular landmarks around London, such as the London Eye or Trafalgar Square, are matched to the relative map to provide route planning goals.

In the next section we describe the related literature. In Section 3 we derive the new relative objective function. Results from simulation and results on real sequences are presented in

Fig. 2. The route taken around London with topologically interesting places for navigation. Between Paddington and Piccadilly the user is underground in the subway. From Piccadilly to Trafalgar Square and the London Eye the user is on foot. One loop was closed around the Eye. From the London Eye the user took the southern route West across the Thames, at which point he took a rickshaw to Trafalgar Square and Piccadilly Circus. From Piccadilly Circus the user walked across Hyde Park to the Natural History Museum, at which point the batteries died, approximately 7 hours into the experiment.

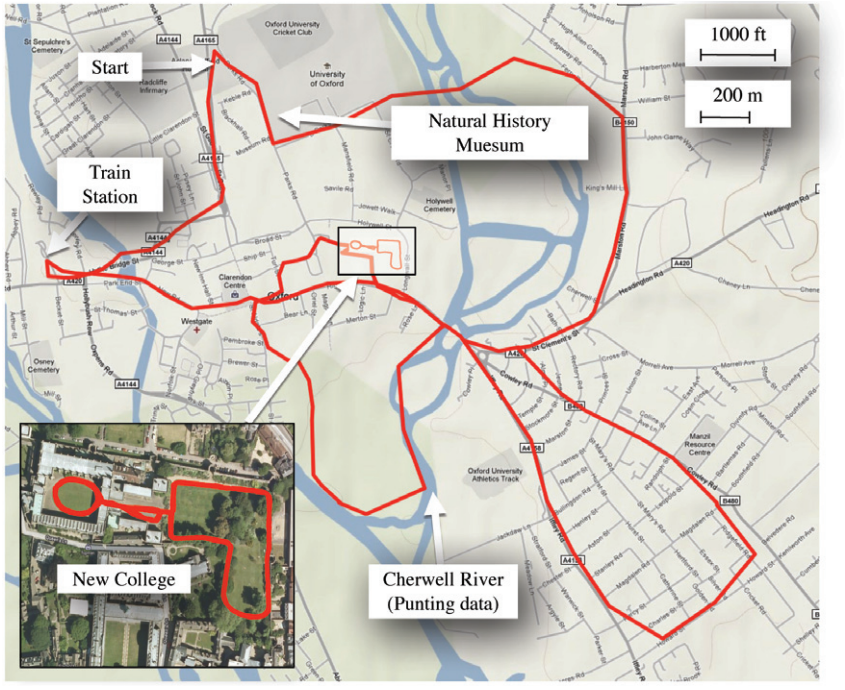


Fig. 3. The "Garden" sequence: 13-km path taken around Oxford with inset showing New College portion (a separate 2.2-km trajectory around New College is also shown in Figure 4). Scenes from the Natural History Museum, Oxford-London train, and punting on the Cherwell are shown later in Figures 10 and 11.

Section 4. We conclude with a discussion of the pros and cons of the relative approach.

2. Related Work

Topological navigation is a well-studied problem that was first addressed in robotics by Kuipers and Byun (1988). Both Kuipers and Byun (1988) and the later work by Choset and Nagatani (2001) seek to describe places of interest as nodes in a graph of relative representations that encode metric information. In this context path planning is a matter of graph search combined with local obstacle avoidance; based on this, topological representations have been used extensively for path planning. Recently, numerous authors have recognized the benefits of vision for topological navigation and mapping (Nister et al. 2004; Fraundorfer et al. 2007; Goedem'e et al. 2007; Steder et al. 2007), although often with the explicit goal of producing a globally embedded solution. The topological relationship between places often use appearance-based recognition based on bag-of-words image matching (Cummins and Newman 2008). Topological mapping, in conjunction with bag-of-words place recognition, has also recently been placed on firm probabilistic ground (Ranganathan et al. 2006; Ranganathan 2008).

From the metric estimation perspective, there has been much interest in Gaussian non-linear least-squares solutions based on "full-SLAM" or bundle adjustment (Triggs et al. 2000; Fitzgibbon and Zisserman 2004; Dellaert 2005; Thrun et al. 2005; Konolige and Agrawal 2008), although the problem is an old one (Brown 1958; Mikhail 1983). 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 (Deans 2005), and there are methods that approximate the full solution with various forms of marginalization (Sibley et al. 2007; Sibley 2007; Konolige and Agrawal 2008), or by ignoring small dependency information (McLauchlan 1999; Thrun et al. 2002b). Recently some researchers have successfully employed techniques from the linear algebra and numerical optimization communities to greatly reduce the cost of finding the full solution (Kaess et al. 2008). Many successful techniques use key frames to reduce complexity, although at the expense of accuracy (Engels et al. 2006; Mouragnon et al. 2006; Klein and Murray 2008). All of these techniques suffer from computational complexity issues during loop closure.

In the context of long-term autonomy, roboticists recognize the need for online, realtime, navigation and mapping algorithms. This means that localization and mapping algorithms must operate within a constant-time budget at each step. Driven by this need, many authors have recognized the benefit of relative representations and manifolds (Guivant and Nebot 2001; Bosse et al. 2004; Howard et al. 2006; Martinelli et al. 2007; Eade and Drummond 2008; Howard 2008). On the other hand, the drawbacks of single-frame solutions have been recognized for some time (Brooks 1985). The most common solution is probably sub-mapping (Bosse et al. 2004; Davison et al. 2007; Pinies and Tardos 2007; Eade and Drummond 2008), 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 (RBA) we propose can be seen as a *continuous* sub-mapping approach that avoids these complications.

To solve large Euclidean 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 (Olson et al. 2006; Grisetti et al. 2007). Starting with the linearized full-SLAM "$Ax = b$" normal equations, a generally sparse set of pose constraints can be constructed by forward substituting all landmark parameters onto the remaining pose parameters (Eustice et al. 2005; Thrun et al. 2005). 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 simply 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 landmark parameters, and iterates the procedure to convergence. This fact highlights a substantial difference between pose graphs and full-SLAM. Like other methods, Euclidean pose-graph solvers have worst-case complexity at loop closure that is dependent on the length of the loop.

While the worst-case complexity for full-SLAM is $O(m^3)$, in practice there is often substantial sparsity in the problem, and this structure can be exploited to great effect (Triggs et al. 2000; Steedly et al. 2003; Krauthausen et al. 2006; Agarwal et al. 2009). Regardless, in the face of an ever-expanding set of nested observations, the cost of solving the full Euclidean solution continues to grow. As an alternative, the relative approach presented in this paper is designed to avoid this complexity entirely.

The work most similar to RBA is the relative formulation given by Eade and Drummond (2008) and Konolige and Agrawal (2008). 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 Euclidean 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 for-

mulates 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 optimization approaches have been explored within the privileged Euclidean frame paradigm (Steedly and Essa 2001; Ranganathan et al. 2007). These techniques, together with all of the methods presented in this section, are not constant time at loop closure, and all but one (Bosse et al. 2004) 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.

3. Methods

In this section we first describe the continuous relative representation and how to optimize within this framework. Second we describe our robust stereo front-end processing pipeline. With this system we are able to achieve the kinds of metric accuracy shown in Figure 4(a) and (b) and produce reconstructions such as that shown in Figure 4(c). Finally, we describe a graph search strategy that serves to demonstrate path-planning sufficiency.

3.1. Problem Formulation

Instead of optimizing an objective function parameterized in a single privileged coordinate frame, we now derive a completely relative formulation. Recall that 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 $l_{j,k}$, $k \in 1, \ldots, n$, $j \in 1, \ldots, m$ be a set of n 3D landmarks each parameterized relative to some *base frame* j. Let t_j, $j \in 1, \ldots, m$ 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, although globally it is not embedded in a single Euclidean space. The relationship between frame α and 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 matrix defined by t_j. An example trajectory and graph with this notation is shown in Figure 5(b).

Each t_j parameterizes an infinitesimal 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

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

the sensor model for a single measurement is

$$\begin{aligned} h_{i,k}(l_{j,k}, t_i, \ldots, t_j) &= \mathcal{K}\left(T_{j,i}^{-1} l_{j,k}\right) \\ &= \mathcal{K}\left(g_{i,k}(l_{j,k}, t_{j+1}, \ldots, t_i)\right), \end{aligned}$$

where $g_{i,k} : \mathbb{R}^{\dim(x)} \to \mathbb{R}^4$, $x \mapsto T_{j,i}^{-1} l_{j,k}$ transforms the homogeneous point $l_{j,k}$ from base frame j to the observation frame i, and $\mathcal{K} : \mathbb{R}^4 \to \mathbb{R}^2$, is the standard perspective projection function (Hartley and Zisserman 2000).

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 assumption that measurements $z_{i,k}$ are independent and normally distributed: $z_{i,k} \sim N(h_{i,k}, R_{i,k})$. The cost function we associate with this formulation is

$$\begin{aligned} J &= \sum_{k=1}^{n} \sum_{i \in m_k} \left(z_{i,k} - h_{i,k}(x)\right)^{\mathrm{T}} R_{i,k}^{-1} \left(z_{i,k} - h_{i,k}(x)\right), \\ &\quad (m_k : \text{set of frames that see landmark } k) \\ &= \|z - h(x)\|_{R^{-1}}, \end{aligned} \tag{1}$$

which depends on the landmark estimate, $l_{j,k}$ and *all of the transform estimates* $t_{j+1}, \ldots, t_i$ *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 MLE (subject to local minima). Projecting via kinematic chains in this manner is novel, but it changes the sparsity patterns in the system Jacobian. Compared with normal bundle adjustment, this new pattern increases the cost of solving the sparse normal equations for updates δx to the parameter vector x, although, as we show, the ultimate computational complexity is the same if we use key frames.

Note that *any* edge in the underlying *co-observability* graph can be used for optimization. Empirically we have found it sufficient to optimize edges along the robot trajectory, together with loop-closure edges selected by the breadth-first-search mechanism described in Section 3.5. Selecting which edges are the best to optimize is an open issue.

3.2. Sparse Solution

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

$$H^{\mathrm{T}} R^{-1} H \delta x = H^{\mathrm{T}} R^{-1} (z - h(x)), \tag{2}$$

where δx is the parameter update we are solving for, $H = \partial h / \partial x$ is the Jacobian of the sensor model, and R is the

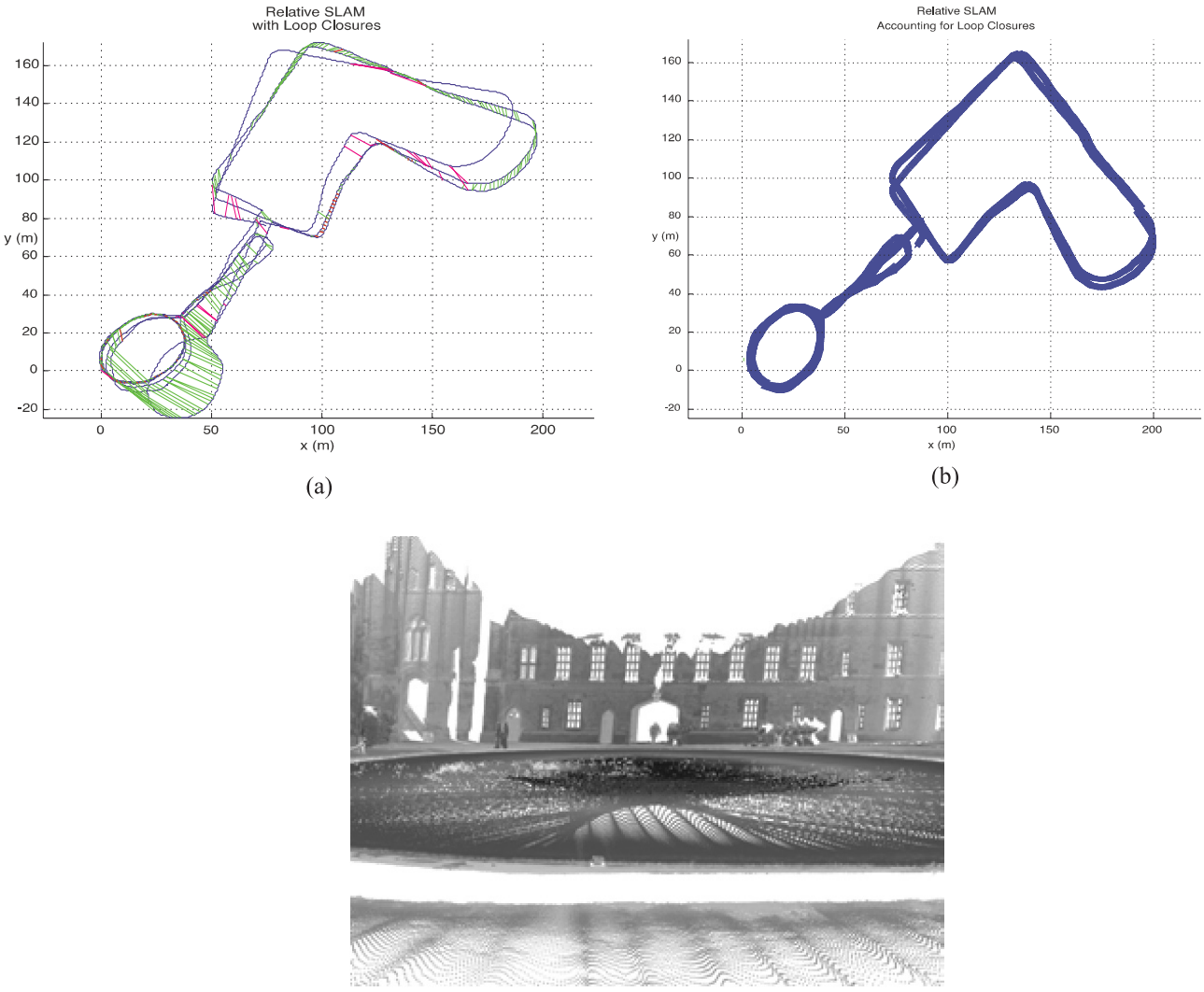


Fig. 4. Example of the metric pose estimate output by our system from the 2.26 km November 3, 2008 New College dataset (Smith et al. 2009). (a) The estimate before, and (b) after, taking loop closure into account. In this case trajectory error before loop closure is 15–25 m in the (xy)-plane and ∼15 m in z; after loop closure the error is ∼10 cm in the $(xy$-plane, ∼10 cm in z. The laser data rendered from the relative trajectory in (c) indicates the local accuracy. Clearly, metric structure is available in the relative approach. In our experience this is sufficient for path planning, obstacle avoidance and local scene analysis (Newman et al. 2009).

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 6 and Figure 7 we see that $H^{\mathrm{T}} = \left[H_l^{\mathrm{T}} H_t^{\mathrm{T}}\right]$ and $\delta x = [\delta l;\ \delta t]$, which exposes a well-known 2×2 block structure for Equation (2),

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

where δl and δt are parameter updates for the map and edge transforms that we are solving for;

$$\begin{aligned} r_l &= H_l^{\mathrm{T}} R^{-1}(z - h(x)), \\ r_t &= H_t^{\mathrm{T}} R^{-1}(z - h(x)), \\ V &= H_l^{\mathrm{T}} R^{-1} H_l, \\ W &= H_l^{\mathrm{T}} R^{-1} H_t, \end{aligned}$$

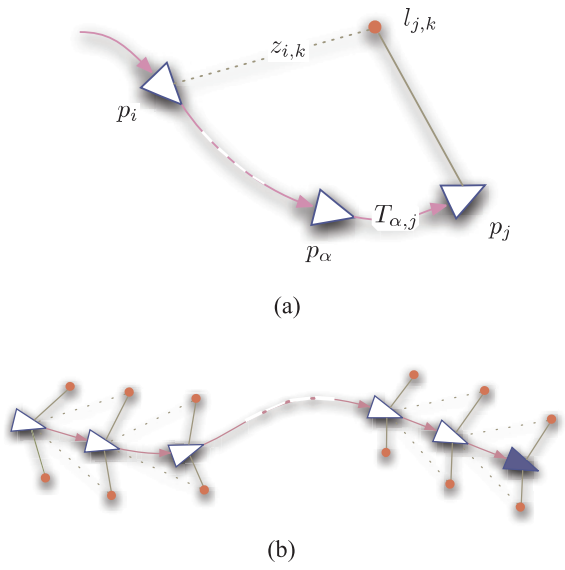


Fig. 5. (a) Notation: poses are indicated with triangles, landmarks with red circles. Landmark base frames are indicated with solid lines, e.g. here landmark k is stored relative to frame j. Each inter-pose edge in the graph includes an error-state transform defined about $t_j = 0$, that is, $T_{\alpha,j} = \hat{T}_{\alpha,j} T_{(t_j)}$, where $\hat{T}_{\alpha,j}$ is the current estimate of the relative transform between frame α and frame j. Note 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. (b) How landmarks are stored relative to the poses; clearly there is no reference to a privileged global coordinate frame.

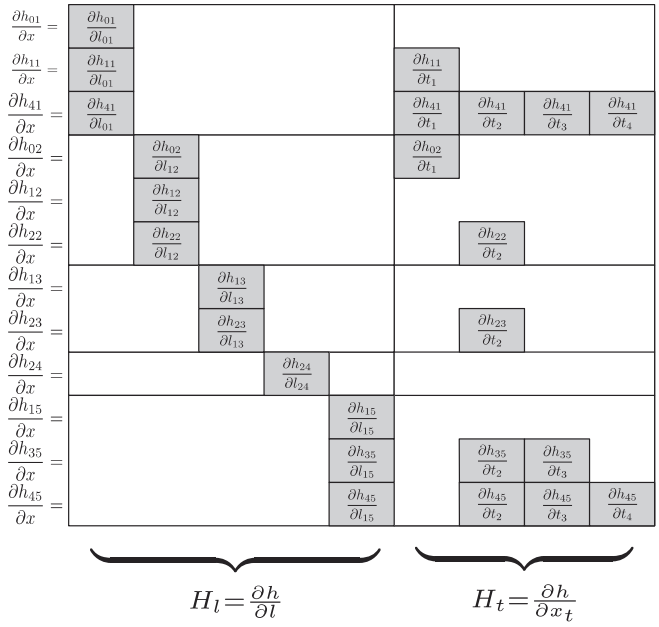


Fig. 6. Example relative bundle adjustment Jacobian structure for the sequence of 12 observations in Figure 7. Gray indicates non-zero entries. The horizontal stripes in the right-hand H_t term above correspond to projections that rely on transforming parameter estimates along kinematic chains from frame j to frame i. These stripes are the only difference in sparsity pattern between the relative formulation and traditional bundle adjustment.

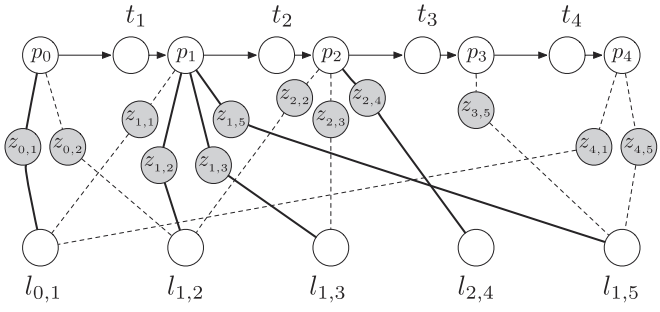


Fig. 7. Graphical model for the sequence of 12 observations, $\{z_{0,1}, z_{1,1}, z_{4,1}\}$, $\{z_{0,2}, z_{1,2}, z_{2,2}\}$, $\{z_{1,3}, z_{2,3}\}$, $\{z_{2,4}\}$, and $\{z_{1,5}, z_{3,5}, z_{4,5}\}$. There are five poses, $p_{0,\ldots,4}$, four edge estimates $t_{1,\ldots,4}$, and five landmarks $l_{0,1}$, $l_{1,2}$, $l_{1,3}$, $l_{2,4}$ and $l_{1,5}$. This example has the Jacobian $H = \partial h / \partial x$ that is depicted in Figure 6. Bold lines from poses indicate which frames are base frames.

and

$$U = H_t^{\mathrm{T}} R^{-1} H_t.$$

This sparse system is solved by forward substitution using the Schur compliment (Sibley 2006),

$$\begin{bmatrix} V & W \\ 0 & U - W^{\mathrm{T}} V^{-1} W \end{bmatrix} \begin{bmatrix} \delta l \\ \delta t \end{bmatrix} = \begin{bmatrix} r_l \\ r_t - W^{\mathrm{T}} V^{-1} r_l \end{bmatrix},$$

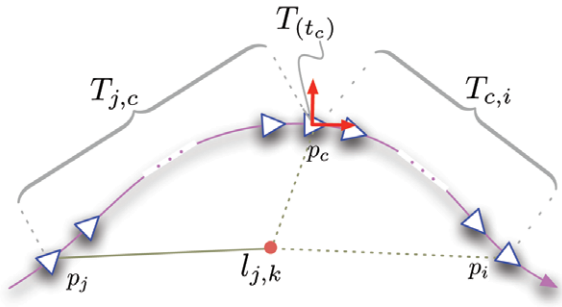
solving for δt,

$$\delta t = \left[U - W^{\mathrm{T}} V^{-1} W\right]^{-1} r_t - W^{\mathrm{T}} V^{-1} r_l,$$

and then back substitution to solve for δl,

$$\delta l = V^{-1}(r_l - W \delta t).$$

After each iteration, δt is used to update the relative edge transform between poses, e.g. $\hat{T}_{\alpha,j} = \hat{T}_{\alpha,j} T_{(\delta t_j)}$. We store edge-transforms as 4×4 matrices, which is a convenient non-singular representation, and care is taken to keep the rotation matrix orthonormal in the face of rounding errors. Building



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

Fig. 8. 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 $h_{i,k} = \mathcal{K}(g_{i,k}(x))$, where $g_{i,k}(x) = T_{j,i}^{-1}l_{j,k} = \left(T_{j,c}T_{(c)}T_{c,i}\right)^{-1} l_{j,k}$, 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 error-state transform $T_{(t_c)}$ indicated in red. The terms of interest when computing derivatives are (1) the transform parameters t_c and (2) the landmark parameters $l_{j,k}$.

this linear system is the dominant cost in solving each iteration, which makes it important to compute the sparse Jacobian of h efficiently.

3.3. Relative Jacobians

Owing to the functional dependence of the projection model on the kinematic chain of relative poses, the Jacobian in the relative formulation is very different from its Euclidean counterpart. With reference to Figure 8, focus for a moment on a single infinitesimal transform $T_{(t_c)}$ that is somewhere along the kinematic chain from frame i to j. The individual derivatives shown in Figure 6 are

$$\frac{\partial h_{i,k}}{\partial l_{j,k}} = \frac{\partial \mathcal{K}}{\partial g_{i,k}} \frac{\partial g_{i,k}}{\partial \bar{l}_{j,k}}$$

and

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

where $\partial\mathcal{K}/\partial g_{i,k}$ is the 2×3 Jacobian of the perspective projection function (Hartley and Zisserman 2000).

The 4×3 Jacobian of $g_{i,k}$ with respect to the non-homogeneous 3D point $\bar{l}_{j,k}$ is

$$\frac{\partial g_{i,k}}{\partial \bar{l}_{j,k}} = T_{i,j}^{-1}[1, 1, 1, 0]^{\mathrm{T}} = \begin{bmatrix} R_{i,j} \\ 0 \end{bmatrix}.$$

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

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

where $\partial T_{(t_c)}/\partial t_c$ are the canonical generators of SE(3) (these simplify computing the Jacobians; see the Appendix). We now address the cost of solving each update.

3.4. 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 parameter updates efficiently:

1. *Build linear system*, computing the terms U, V, W, r_t and r_l. Complexity is $O(m^2n)$ using key frames.
2. *Forward substitute*, computing $A = U - W^{\mathrm{T}}V^{-1}W$, and $b = r_t - W^{\mathrm{T}}V^{-1}r_l$. 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 substantially different in the relative framework so we describe it in more detail in Algorithm 1. The overall complexity for all steps 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 (Huber 1964). Section 4 gives results of applying this sparse optimization routine to large real and simulated sequences.

Finally, note 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

Algorithm 1 Build linear system. Computes U, V, W, r_t, and r_l in $O(m^2 n)$.

Clear U, V, W, r_t, and r_l
for all landmarks k **do**
  for all key-frames i with a measurement of landmark k **do**
    Compute $\frac{\partial h_{i,k}}{\partial l_{j,k}}$
    $e_{i,k} = z_{i,k} - h_{i,k}(x)$
    $w_{i,k} = R_{i,k}^{-1}$
    $V_k = V_k + \frac{\partial h_{i,k}}{\partial l_{j,k}}^{\mathrm{T}} w_{i,k}^{-1} \frac{\partial h_{i,k}}{\partial l_{j,k}}$
    $r_{l_k} = r_{l_k} + \frac{\partial h_{i,k}}{\partial l_{j,k}}^{\mathrm{T}} w_{i,k}^{-1} e_{i,k}$
    for all $p \in Path(i, j)$ **do**
      Compute $\frac{\partial h_{i,k}}{\partial t_p}$
      $r_{t_P} = r_{t_P} + \frac{\partial h_{i,k}}{\partial t_p}^{\mathrm{T}} w_{i,k} e_{i,k}$
      $W_{k,p} = W_{k,p} + \frac{\partial h_{i,k}}{\partial l_{j,k}}^{\mathrm{T}} w_{i,k} \frac{\partial h_{i,k}}{\partial t_p}$
      for all $q \in Path(p, j)$ **do**
        Compute $\frac{\partial h_{i,k}}{\partial t_q}$
        $U_{p,q} = U_{p,q} + \frac{\partial h_{i,k}}{\partial t_p}^{\mathrm{T}} w_{i,k} \frac{\partial h_{i,k}}{\partial t_q}$
        $U_{q,p} = U_{q,p} + \frac{\partial h_{i,k}}{\partial t_q}^{\mathrm{T}} w_{i,k} \frac{\partial h_{i,k}}{\partial t_p}$
      end for
    end for
  end for
end for

back substitution (steps 2–4) should be approximately equivalent.

3.5. Adaptive Updates

To reduce computation, it is important to optimize only those parameters that might change in light of new information (Steedly and Essa 2001; Ranganathan et al. 2007). In the following we outline one approach to limit the parameters that are actively optimized.

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 $\Delta\epsilon$. This search is ultimately limited by machine precision; in practice, values for $\Delta\epsilon$ of the order of 1×10^{-2} track the *MLE* solution closely. 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 form a slightly larger set that 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 9.

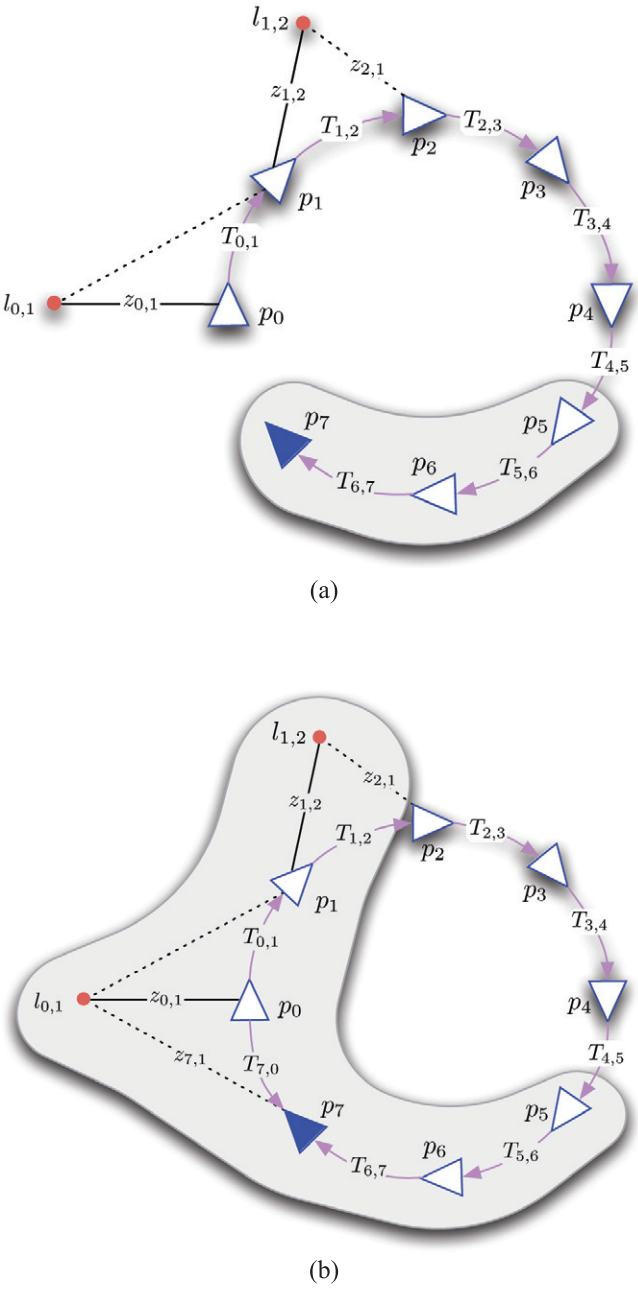


Fig. 9. Discovery of local active region: (a) pre-loop closure; (b) post-loop closure. In (a), re-projection errors have changed by more than $\Delta\epsilon$ in the local frames p_5, p_6 and p_7. In (b) a new edge $T_{7,0}$ is added during loop closure, and the graph search leads to a larger active region with frames p_0, p_1, p_5, p_6 and p_7.

3.6. Processing Pipeline

In this section we describe the engineering effort required to achieve precision, robustness and speed in the visual processing pipeline. Our approach is key-frame based and, similar to parallel tracking and mapping (PTAM) (Klein and Murray 2008), the bundle adjuster runs in a separate thread. The image processing pipeline includes the following steps:

1. Pre-processing: includes rectification to allow scanline searching for left–right correspondences. Images are shifted to obtain the same mean and variance. We use FAST features (Rosten and Drummond 2006) extracted at different levels of a scale-space pyramid for robustness to image blur. Detection thresholds are modified at each timestep to keep the number of detected features at a desired level independent of the scene.

2. Dense alignment: an estimate of the 3D rotation is obtained using the sum-of-squared-distance of image intensity using an efficient second-order gradient descent minimization (ESM) as described by Mei et al. (2008). This greatly helps in cases with perceptual aliasing, such as bricks, tiles and picket fences; it also reduces the search range for establishing feature correspondence.

3. Matching in time: the 3D coordinates of the landmarks are projected into the left and right images and 9×9 patches are matched using a mean shifted sum-absolute-difference error metric. Finally, ESM sub-pixel refinement is performed. Once matched, the current motion is estimated with a standard combination of RANSAC and a final robust m-estimation step (Fischler and Bolles 1981; Hartley and Zisserman 2000).

4. Starting new landmarks: we typically track 100–150 features and use a multi-level quad-tree to distribute features evenly. At each pyramid level, a quad-tree captures how many features project into each cell. From these counts we can ensure an even spatial distribution across the image. Finally, upon initialization a SIFT (Lowe 2004) descriptor is computed which can be used during re-localization and loop closure. Features are sorted by their Harris corner score (Harris and Stephens 1988), and those with higher scores are instantiated first.

These steps help ensure robustness to the types of challenging operating conditions illustrated in Figures 10 and 11. Further details can be found in Mei et al. (2009). Another implementation detail to note is that, for scalability while maintaining frame-rate performance, we have had to implement an out-of-core graph class which transparently saves and loads the map from disk. This out-of-core data structure enables maps that are only limited by secondary storage capacity.

3.7. Loop Closure and Place Recognition

For loop closure and place recognition we rely on FAB-MAP, which represents each place using the bag-of-words model developed for image retrieval systems by the computer vision community Sivic and Zisserman (2006); Cummins and Newman (2008). At time k the appearance map consists of a set of n_k discrete locations. Each location is described by a distribution over the existence of artifacts that could generate a visual word in an image. Incoming sensory data is converted into a bag-of-words representation; for each location, a query is made that returns how likely it is that the observation came from that location or from a new place. This allows us to determine whether we are revisiting previously visited locations. In a filtering framework, incorrect loop closures are catastrophic as the statistical estimates are corrupted. The continuous relative representation enables recovery from erroneous loop closures as removing the incorrect graph link and bad measurements returns the system to its previous state.

3.8. Path Planning

Path planning consists of finding shortest paths in the relative map with edges weighted either by distance or time. We use the magnitude of inter-frame motion (excluding orientation) to compute edge weights for distance-based searches. Time-based searches use an edge weight that is simply the time between key frames, with an average value used for loop-closure edges.

4. Results

In this section we provide details of the simulation and real-world experimental results that show the scalability and accuracy of the relative SLAM solution.

4.1. Relative Bundle Adjustment Timing

The iterative non-linear least-squares solution that exploits the sparse relative structure and the four steps in Section 3.4 results in the run-time breakdown shown in Figure 12. This illustrates that building the sparse system of equations is the dominant cost.

4.2. Simulation Results

To determine the performance of the relative framework, a batch of simulations was run. The sequence contains a realistic trajectory, landmark distribution, and a 1-pixel standard deviation Gaussian measurement noise (see Figure 13).

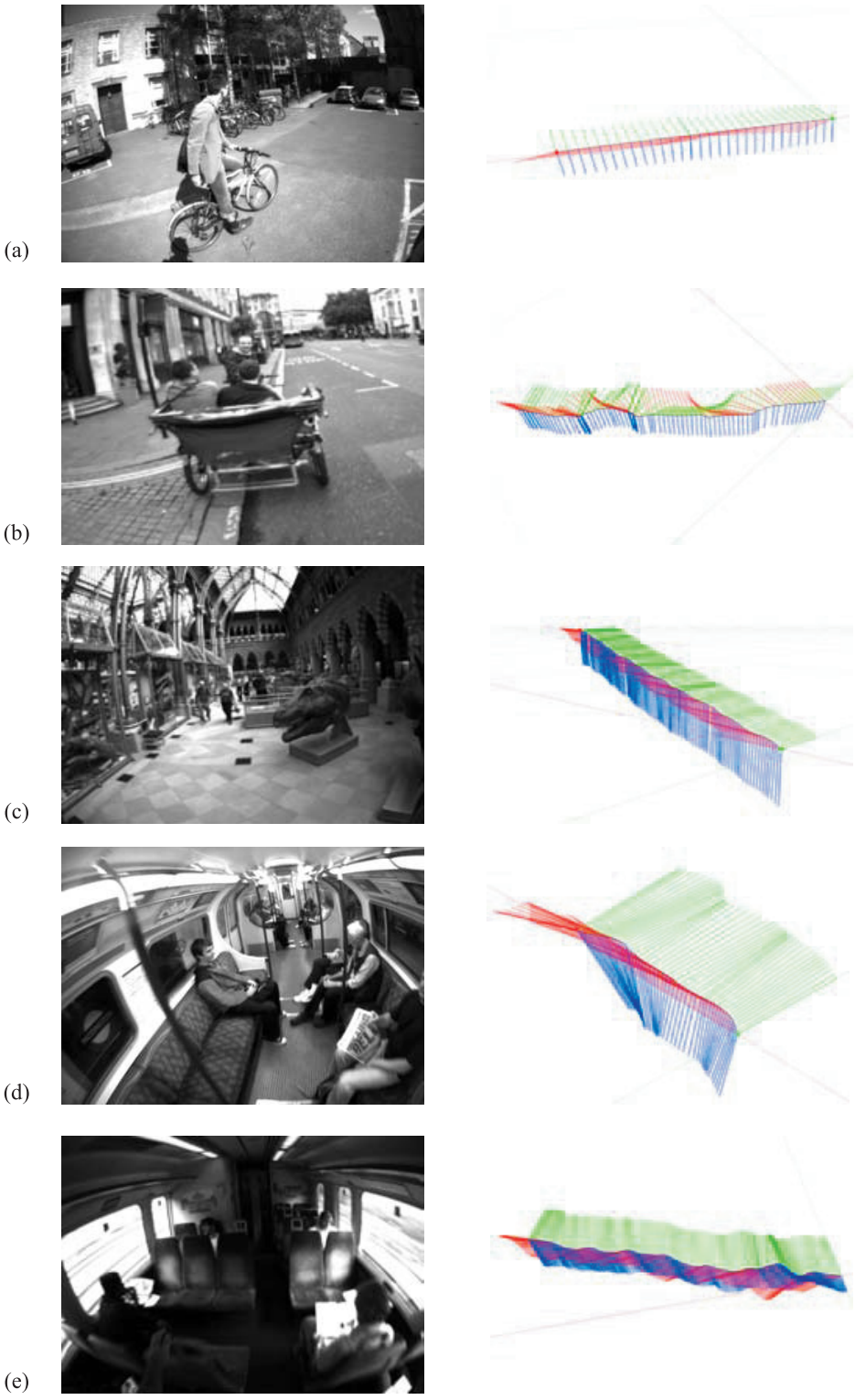


Fig. 10. Excerpts from the London dataset showing typical trajectory estimates for various modes of transport. (a) A smooth constant velocity trajectory from a bicycle. (b) Track estimated from the rickshaw showing user head swivel. Not as fast as the bicycle. (c) Walking in the Oxford Natural History Museum. Note the clearly visible gait. (d) Walking into a subway car. Note that at some point in this trajectory, the car begins to move, a fact not visually discernible here. Detecting linear acceleration with off-the-shelf inertial sensors is difficult in this situation. (e) Walking on a train looks like walking anywhere else. Note the extreme motion blur in the windows from the quickly passing external terrain.

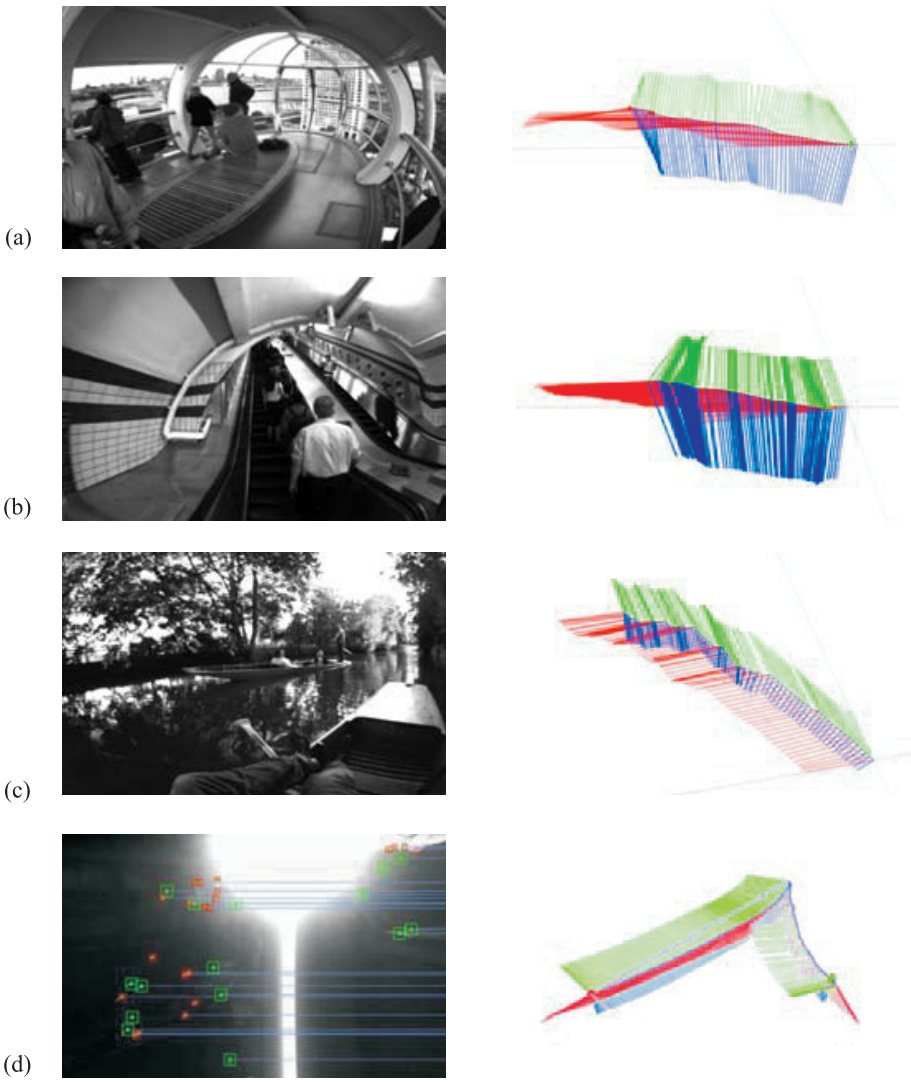


Fig. 11. More trajectories for various modes of transport. (a) Walking on the London Eye. It is difficult to detect loop closure metrically in the trip around the wheel. (b) Challenging conditions on an escalator. This is a failure case in which tracked motion oscillates between moving and stationary (note the bunched axes where it is stationary). Detecting linear acceleration with off-the-shelf inertial sensors is difficult in this situation. (c) Visual motion estimation while punting is challenging due to reflections which appear to cause a slight instability. (d) Successful motion estimation in challenging lighting conditions (tracked FAST corners are indicated). Parts (a) and (b) are again from the London dataset, (c) is from the Punting dataset, and (d) is an example from the 1,000 km dataset reported by Cummins (2009).

We need to be careful when measuring performance in the relative representation. It is important to base our error metric on geometric invariants that are *coordinate frame independent*, such as relative distance or angles. The breadth-first-search error reported here is one way to do this.

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; note that these rips do not exist in the manifold and are simply an artifact of projection to a single Euclidean frame. Finally, the total trajectory registration error is computed as the average Euclid-

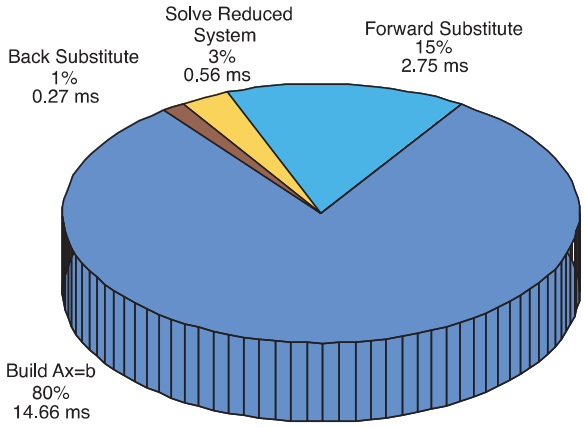


Fig. 12. Average run times for the main steps of relative bundle adjustment on an Intel Core 2 Duo 2.8 GHz processor. The average adaptive region from the simulation was 4.6 frames. Note that it is the cost of building the linear system of equations that dominates the cubic complexity of solving for the adaptive region of poses.

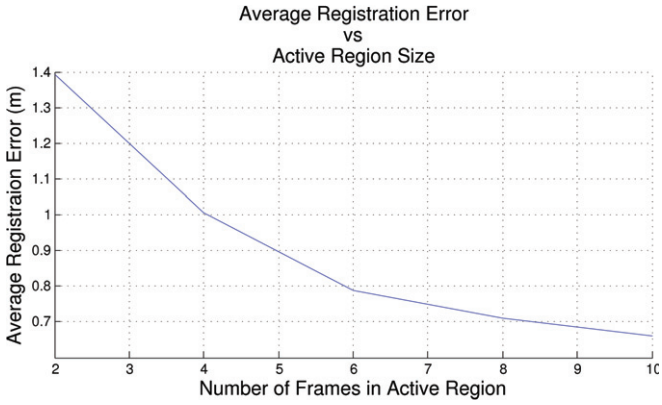


Fig. 14. Average registration error versus number of frames being updated. In the relative formulation, as the local region grows, the average root mean square error drops quickly toward the same as when computed with all frames active. This motivates the use of an adaptive region that allows parameters to vary only if it has an effect on the cost function.

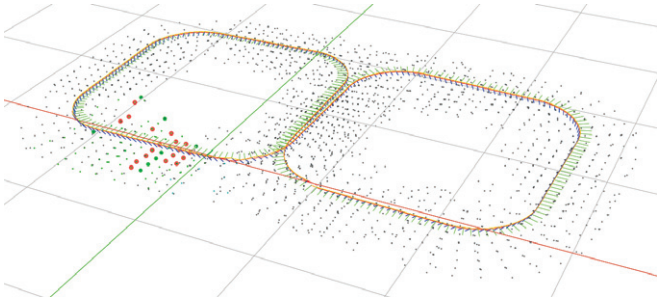


Fig. 13. Figure-of-eight sequence used in Monte Carlo simulations. This sequence has 288 frames, 3,215 landmarks and 12,591 measurements with 1-pixel standard deviation Gaussian measurement noise added.

ean distance between ground truth and the localized frames. The average of all frames and all registrations is the error plotted. Not surprisingly, results in Figure 14 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 four and five poses, except at loop closure where it jumps to accommodate the larger region of poses found by the breadth-first search. Figure 15 shows the adaptive region size discovered for two different loop closures, one 50 m long and another 100 m 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 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.

Numerous authors have shown that SLAM can be constant time in practice during the exploration phase (Thrun et al. 2002a; Ranganathan et al. 2007; Kaess 2008). The relative approach presented here is explicitly designed to take advantage of this phenomenon at loop closure, as well as during exploration. The adaptive optimization is only successful in conjunction with the relative formulation, otherwise we would see computation spike at loop closure, as reported by Ranganathan et al. (2007).

4.3. Real Data

The system operates online at 20–40 Hz, this includes all image processing, feature tracking, robust initialization routines, and calls to FABMAP (Cummins and Newman 2007) to detect loop closures. We have run it successfully on numerous sequences including those listed in Table 1. Table 2 gives an indication of typical system performance.

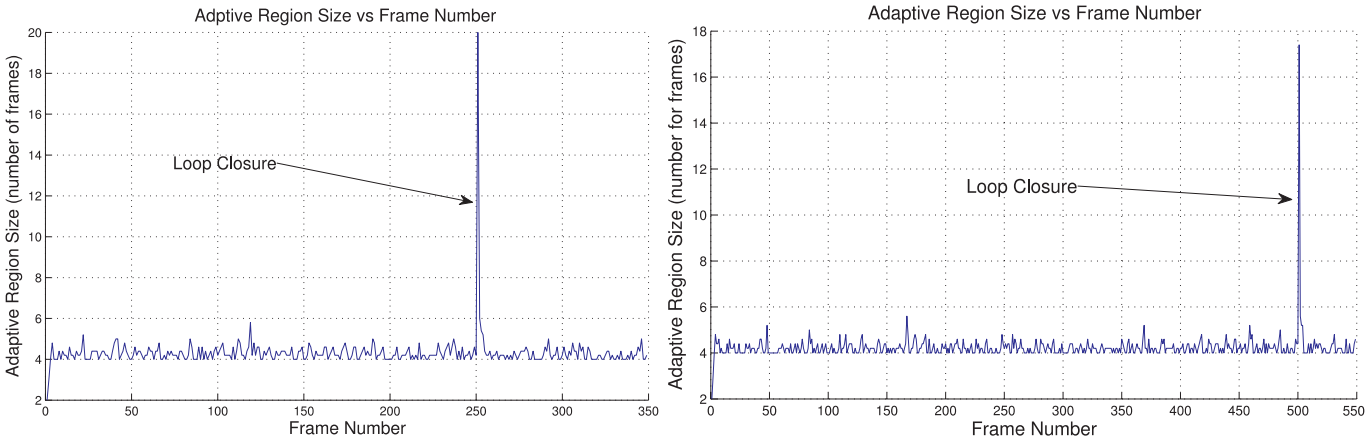


Fig. 15. How the number of frames in the adaptive region fluctuates over time and during loop closure: (a) 50 m loop with closure at frame 250; (b) 100 m loop with closure at frame 500. During loop closure the size of the adaptive region jumps to accommodate all of the local frames that have been added to the active region, as well as any neighboring frames that will be affected. Note that errors do not propagate all the way around the loop, and only a fraction of the parameter 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 75 cm 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.

Table 1. List of Datasets. All Images are 512 × 384 grayscale, captured at 20 Hz with a PointGrey BumbleBee2 Camera. The London, Garden, New College and Punting Datasets Overlap, which Allows Multi-session Mapping as we have Shown in Shown in Newman et al. (2009).

Name	*Frames*	*Space (GB)*	*Date*	*Distance (km)*
London	479,729	188.6	10/9/09	121
Garden	185,797	73.1	8/10/ 09	13.1
Punting	114,736	45.1	17/9/09	5.1
New College	52,510	20.6	3/11/ 08	2.3
Science Park	23,467	9.2	17/9/08	1.1
TOTAL	856,239	333.6	—	142

Table 2. Typical Performance of the Online System for the Begbroke Science Park Dataset Processed on an Intel Core 2 Duo 2.8 GHz.

	Average	*Minimum*	*Maximum*
Distance traveled (km)	—	—	1.08
Frames processed	—	—	23,268
Velocity (m s^{-1})	0.93	0.0	1.47
Angular velocity ($\circ$ s^{-1})	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	0	0.55

4.4. Comparison with Bundle Adjustment

Both RBA and traditional bundle adjustment seek to minimize re-projection error, given a parametric model of the world. Figure 16 shows that the local Euclidean re-projection error for RBA matches that of traditional bundle adjustment. This implies that the metric world structure is similar and we have shown in Holmes et al. (2009) that the normalized L^2 difference between RBA and traditional bundle adjustment pose estimates remains constant and less than 1.4×10^{-4}.

4.5. The Importance of Loop Closure

In addition to storage requirements (which are handled with out-of-core data structures to ensure a constant size memory

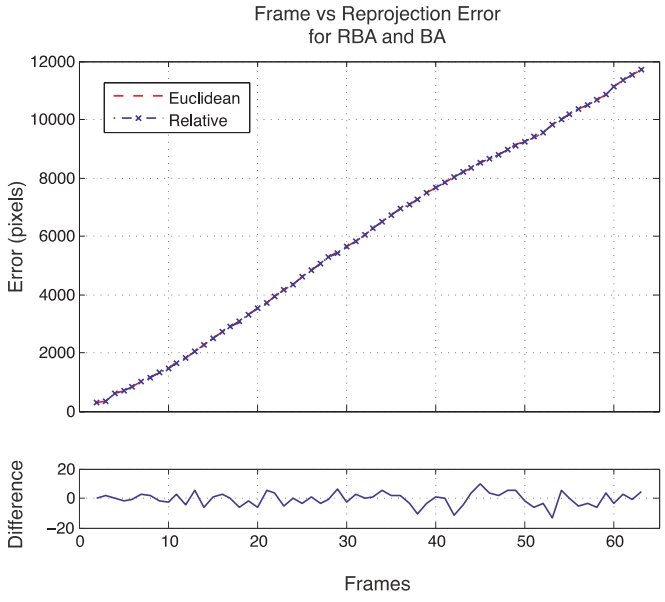


Fig. 16. Comparison between RBA and Euclidean BA. This graph shows the expected linear increase in re-projection error using both the relative and global bundle adjustment as more key-frames are added in a sequence. Like traditional bundle adjustment, the relative objective function is defined in terms of re-projection error – it is evident that both representations converge to approximately the same re-projection error.

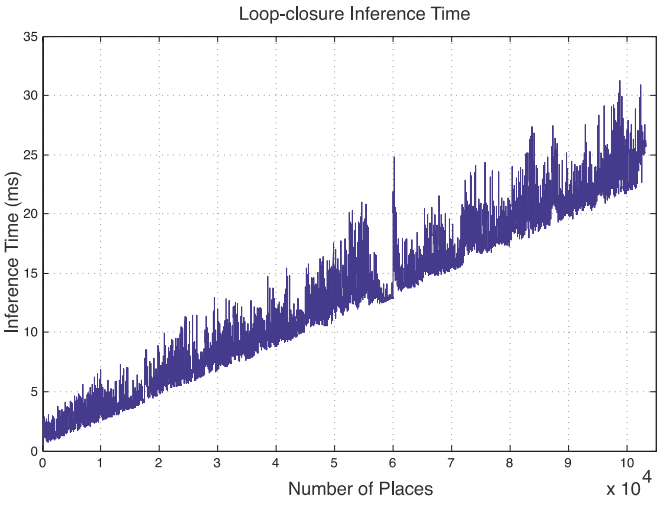


Fig. 17. Loop-closure detection times from a 1,000 km dataset with 103,256 places in the map at the end of the run (Cummins 2009). The best fit is 1.04 ms baseline + 207 ns per place. Maximum processing time for any observation is 31.3 ms.

footprint) loop-closure detection is the only non-constant time aspect of our SLAM system. It is therefore important that loop closure be fast and scale well. Figure 17 indicates the scalability of FAB-MAP, and shows the potential to query millions of locations per second: a scale we have yet to reach. Clearly, realtime loop-closure detection is not a bottleneck.

Identifying loop closures in the relative framework and then projecting to a single Euclidean frame with a breadth-first search leads to significant improvement in map error in a single global frame. For instance, Figure 18 shows that loop-closure drastically improves mapping error, even without applying an expensive global optimization.

4.6. Path Planning

To plan a route in the graph we begin with a web image from Trafalgar Square matched to an image from the graph with FAB-MAP (for example see Figure 19). Given this query, we can find a path from Oxford to London based on either metric or temporal distance. Note however that information can also go in the other direction – that is, given such appearance-based matches from the Internet, it is possible to label relative map locations with search words describing places.

Two paths from the Natural History Museum in Oxford to the London Eye are computed. The first path, based on a desired shortest travel, takes the *southern* bridge from West to East (see Figure 2). This route is shorter due to the rickshaw that the user rode on during exploration. The second path, which gives shortest distance, traverses the loop closure at Trafalgar Square, and then takes the *northern* bridge to the London Eye, which was traversed on foot and is indeed the more direct route (see Figure 20 for representative loop closure hypothesis from FAB-MAP). Clearly, paths planned in the relative representation can take advantage of metric and temporal information.

Autonomous traversal of these paths will clearly require obstacle avoidance and knowledge of the various forms of transport used: these problems are not addressed in this paper; we simply wish to point out that it is possible to find paths in the graph and that these paths can be informed by the underlying topological and metric map structure.

4.7. Unsensed Ego Motion in the Real World

Autonomous navigation in human working environments is an important problem and, in the process, unsensed ego motion is a common phenomenon. Navigation in the real world frequently requires travel on and inside various forms of moving reference frames, where it is impossible to sense the global frame in a drift-free fashion. Using current state-of-the-art SLAM estimation techniques, it is not possible to compute a consistent global representation while undergoing unsensed

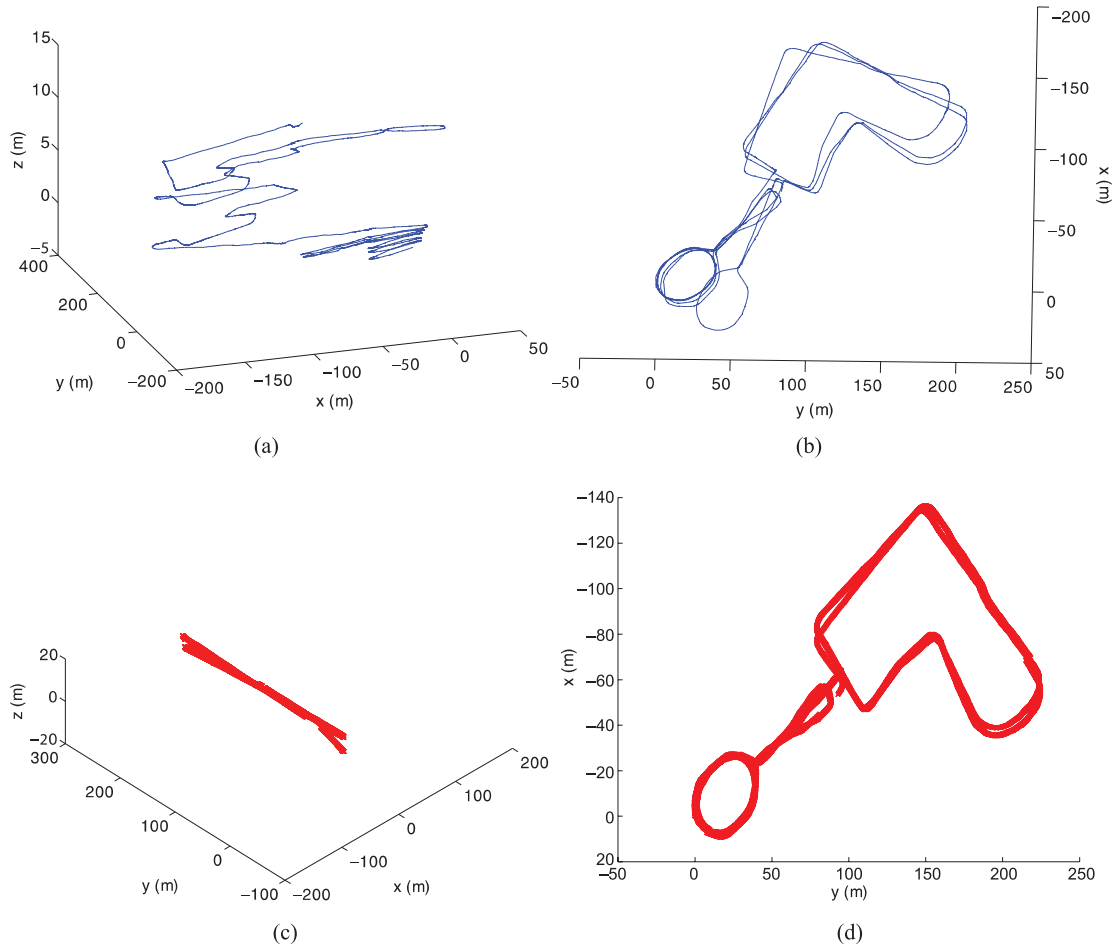


Fig. 18. Side view of New College data set (a) before and (c) after loop closure and top view of New College data set (b) before and (d) after loop closure. These figures illustrate that loop closure substantially reduces overall mapping error, even before we apply global relaxation. In this case trajectory error before loop closure is 15–25 m in the (xy)-plane and ∼15 m in z; after loop closure the error is ∼10 cm in the (xy)-plane, ∼10 cm in z.

Fig. 19. Image from the web of Trafalgar Square. FAB-MAP hypothesis match from the graph before any geometric checks have been applied. Given this query, we find a path from Oxford to London based on either metric or temporal distance. Note, however, that information may also go in the other direction; that is, for highly distinctive places such as the London Eye, Piccadilly Circus, Trafalgar Square or the Natural History Museum, it is possible to label relative map locations with relevant information from the Internet (Sivic and Zisserman 2006).

Fig. 20. Loop closure candidates at Trafalgar Square and Piccadilly Circus. Loop closure allows shorter paths to be found in the map.

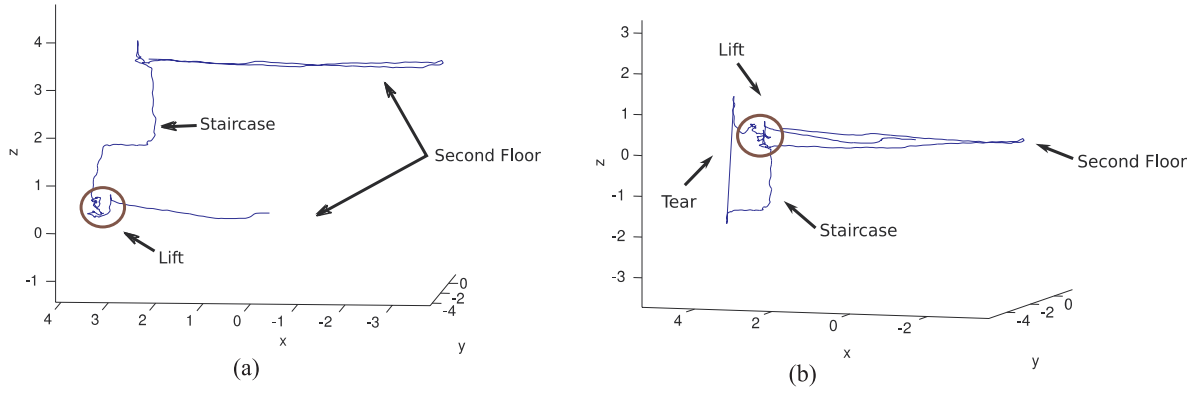


Fig. 21. Lift sequence demonstrating the complexity of unobservable ego motion with respect to a global inertial frame. Even with inertial sensing, transportation portals such as subways, lifts, trains, etc., frustrate efforts to estimate position with respect to the global inertial frame. In (a), after first exploring a floor, then taking a flight of stairs followed by a lift, the robot returns to the same floor. No loop closure has currently been detected. In (b) a loop closure has been triggered, and now the trajectory cannot be represented in Cartesian space, which causes global SLAM optimization routines to fail; a "tear" appears (in this example in the staircase). The size of the tear is related to the distance traveled in the lift. Note well that this tear does not exist in the underlying manifold, which is still useful for navigation because of its relative nature. This is a key advantage of the relative approach.

ego-motion, such as the lift example in Figure 21 or the numerous cases shown in Figures 10 and 11.

The lift example in Figure 21 also highlights that one cannot simply apply a sliding window local Euclidean bundle adjustment, because to do so one would first have to identify the transition on and off the lift and prevent optimization across these boundaries. Without finding these transitions first, and under noisy or biased sensing, any attempt to compute a global solution will face difficulty, as the visual measurements are inconsistent with a single Euclidean embedding. The relative approach avoids this problem.

Many challenging motion sequences from the 121-km London dataset are shown in Figures 10 and 11. In contrast to the data in Figures 4 and 23 this data was collected with a

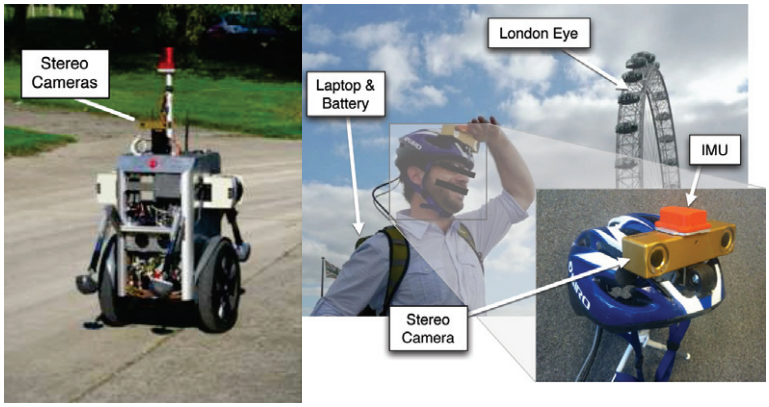


Fig. 22. Processing data collected from human-like autonomous navigation in urban spaces (right) is very different, and substantially more challenging than processing stable robot data (left). In both settings we capture 512 × 384 grayscale images at 20 Hz. The robot camera has 65° field of view (FOV) lenses and the wearable rig has 100° FOV lenses. Both rigs are PointGrey BumbleBee2 cameras (inset-left).

Table 3. Results for London Human-collected Data. Note the Difference in Linear and Angular Velocity: this Reflects the Fact that Head Swivels Result in Very Fast Visual Motion Estimates. This Type of Motion is Exactly the Kind of Challenge *Not* Faced in Typical Robot Data.

	Average	*Minimum*	*Maximum*
Distance traveled (km)	—	—	125
Frames processed	—	—	479,726
Velocity (m s^{-1})	1.3	0	8.2
Angular velocity ($\circ$ s^{-1})	8.6	0	191.5
Frames per second	33.7	21.3	46.4

Table 4. Robot Data Collected on a Segway RMP (see Figure 22). Note the Difference in Linear and Especially Angular Velocity in Table 3.

	Average	*Minimum*	*Maximum*
Distance Traveled (km)	—	—	0.8
Frames Processed	—	—	29,489
Velocity (m/s)	0.6	0	1.3
Angular Velocity (deg/s)	4.8	0	59.8
Frames Per Second	20.3	7.4	28.6

head-mounted stereo system, shown in Figure 22. No special care was taken to collect "clean" data that would lend itself to easy processing; the data reflects a typical human experience of the world. To highlight this, Tables 3 and 4 compare robot-collected with human-collected data. Processing such data is challenging, and there are numerous difficulties encountered, which include but are not limited to: unsensed ego motion, motion blur, dynamic lighting changes, dropped frames, lens flare, dynamic obstacles, obstructed views, non-overlapping frames and power failures.

During the experiment from London to Oxford we are able to compute relative metric motion estimates 89.4% of the time, falling back on a constant velocity model and inertial orientation sensing for the remainder. Thus, 100% of the path is covered *topologically*, which makes it possible to plan paths through the map.

5. Discussion

We posit that the topometric relative formulation is sufficient for many mobile robot navigation tasks, and that a single global Euclidean representation is rarely necessary online over vast scales. Certainly the benefits afforded by incremental constant-time performance are tremendous, and in 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

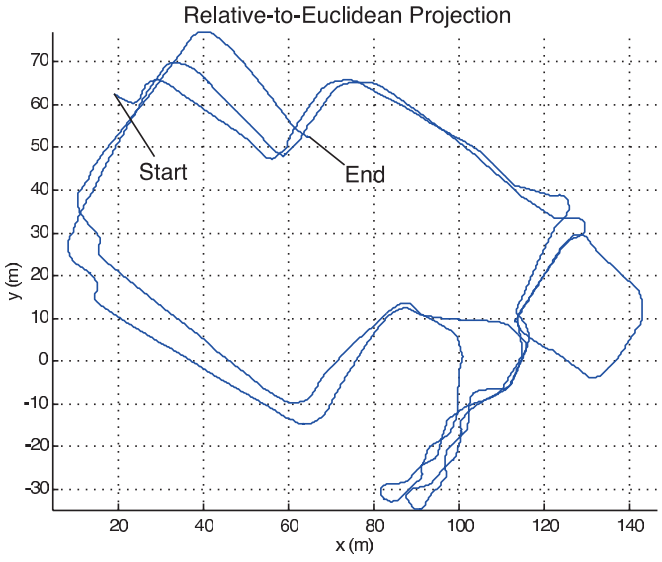


Fig. 23. A 1.08-km path over 23,000 frames estimated for the Begbroke Science Park sequence. Table 2 shows typical performance results.

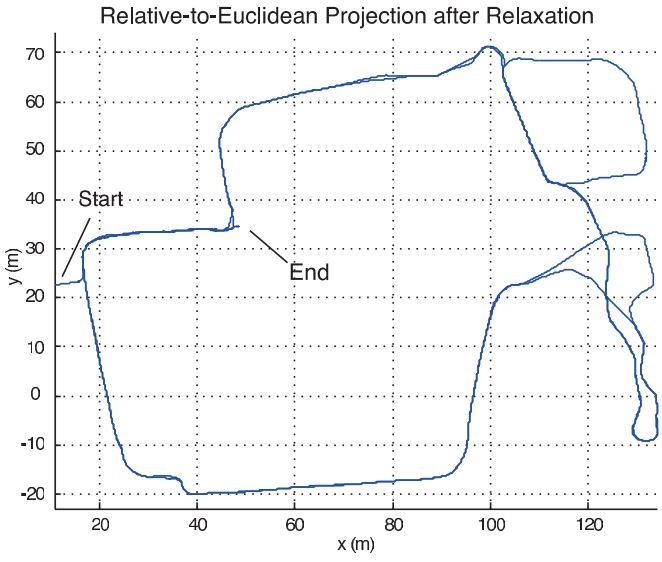


Fig. 24. A globally consistent relaxed view of the Begbroke Science Park sequence (using the pose-graph relaxation technique described in Newman et al. (2009)). To view relative estimates in a consistent fashion (single global frame) we have to transform from the relative representation to a single Euclidean coordinate system. The sequence here has 23,000 poses over 1.08 km which makes the conversion computationally expensive. This relative-to-global transformation process is designed to run on the user interface, *not* on the robot.

that process, for example, undertaking fast approximate pose-graph relaxation in order to render consistent results in a user interface (Olson et al. 2006; Grisetti et al. 2007).

For instance, Figures 23 and 24 show the result of transforming a large relative state estimate into a single Euclidean frame using pose-graph relaxation (Newman et al. 2009). 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 estimation is an inappropriate goal; what matters is relative metric accuracy and topological consistency, all of which can be attained with a relative manifold approach.

To be concrete: if the task is surveying, then it makes sense to estimate everything in a single coordinate frame. We are interested in autonomous navigation, so for us, and we suspect many roboticists, surveying is the not goal of SLAM. In this light, it is useful to distinguish between *SLAM-for-survey* and *SLAM-for-autonomy*: both use similar estimation machinery, but they have different goals.

Ultimately, algorithms that solve for robot position in the privileged inertial coordinate frame are very different from relative approaches: they have different objective functions and they solve for different quantities. Privileged-frame solutions seek to embed the entire robot trajectory in a single Euclidean space; relative solutions solve in a manifold. The relative manifold is a metric space, and distance between two points can be computed from shortest paths in the graph. We have shown that the relative representation is amenable to planning (because path-planning algorithms are commonly defined over graphs). Further, because the manifold is (by definition) locally Euclidean, we have access to highly accurate local metric structure at any time (see, for instance, Figure 4). Topometric solutions are sufficient for real-world navigation, and in our experience they are increasingly necessary as well. For instance, given unsensed ego motion, it is not possible to build consistent map structures in a privileged frame on which to navigate. We have shown that it is possible within a purely relative approach.

Note that claims of navigational path-planning sufficiency are based on the assumption that co-observability implies traversability. We also rely on the assumption that we know how to handle the transitions between transportation modes encountered during the traversal of paths from point a to point b. So, while it may be sufficient from the pure graph-search point of view, it is clearly limited because of the lack of higher-level knowledge about transportation modes; that is, path planning along routes that include trains, lifts, etc., must be informed about the temporal schedule of these transportation modes: it

needs to know how to board a train or a lift. This is a substantially harder problem, and one that we argue is necessary to solve if we are going to effectively use transportation modes for autonomous navigation. Presently, we side step this problem by simply defining sufficiency in terms of the ability to find shortest paths in a graph, that is, we use the traditional definition, even though it is no longer appropriate once we have unsensed ego motion in the form of moving reference frames. This is exciting ground for future work.

While it is certainly possible to build large-scale, consistent global world models (especially with the use of GPS), we find that there are numerous real-world situations where it is effectively impossible to do so; that is, even GPS is not sufficient. There are many examples in which position in the global inertial frame is extremely difficult to estimate accurately, places such as lifts, subways, trains, and these are places we want to navigate autonomously. This fact bears scrutiny, and helps us focus on much harder problems that we will have to solve in order to move forward. These are problems such as learning when and where unsensed ego motion becomes probable; that is, automatically discovering the location of *changing transportation modes*. It is interesting that one solution appears to be learning to recognize high-level semantic objects, such as lifts, escalators, planes, trains and automobiles. Given such labels then perhaps we can relate the topometric world to the global inertial frame.

6. 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. 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 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. To explore the feasibility and scalability of our approach, over 850,000 images and inertial data are processed to produce relative estimates covering more than 142 km of Southern England. We point out the numerous challenges we encounter, and highlight in particular the problem of unsensed ego motion, which occurs when the robot finds itself on or within a moving frame of reference. In contrast to global representations, we find that the continuous relative representation can naturally accommodate moving reference frames, without having to identify them first, and without inconsistency. We also show that a relative, topological approach to autonomous navigation is sufficient, in the sense that one can find shortest paths in a map. We conclude that the relative approach is a route towards autonomous navigation in environments with moving reference frames.

Acknowledgments

The work reported in this paper undertaken by the Mobile Robotics Group was funded by (1) the Systems Engineering for Autonomous Systems (SEAS) Defence Technology Centre and (2) the European Commission under grant agreement number FP7-231888-EUROPA. The work reported in this paper undertaken by the Active Vision Lab acknowledges the support of EPSRC grant GR/T24685/01. We would also like to thank Steven Holmes for his expert punt navigation on the Cherwell river.

Appendix: Rotation Derivatives

Let $\theta = [r, p, q]$ represent roll, pitch and yaw. The associated rotation matrix using the *roll–pitch–yaw* angle convention (Sciavicco and Siciliano 1996) is

$$
\begin{aligned}
R &= R_q R_p R_r \\
&= \begin{bmatrix} c(q) & -s(q) & 0 \\ s(q) & c(q) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c(p) & 0 & s(p) \\ 0 & 1 & 0 \\ -s(p) & 0 & c(p) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(r) & -s(r) \\ 0 & s(r) & c(r) \end{bmatrix} \\
&= \begin{bmatrix} c(p)c(q) & -c(r)s(q)+s(r)s(p)c(q) & s(r)s(q)+c(r)s(p)c(q) \\ c(p)s(q) & c(r)c(q)+s(r)s(p)s(q) & -s(r)c(q)+c(r)s(p)s(q) \\ -s(p) & s(r)c(p) & c(r)c(p) \end{bmatrix}.
\end{aligned}
$$

where $s(\cdot)$ and $c(\cdot)$ are short for $\sin(\cdot)$ and $\cos(\cdot)$, respectively. The derivatives of R with respect to infinitesimal rotations ($\theta = 0$) are

$$
\begin{aligned}
\frac{\partial R}{\partial r} &= R_q R_p \frac{\partial R_r}{\partial r} \\
&= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & c'(r) & -s'(r) \\ 0 & s'(r) & c'(r) \end{bmatrix} \\
&= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix},
\end{aligned}
$$

$$\frac{\partial R}{\partial p} = R_q \frac{\partial R_p}{\partial p} R_q$$
$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c'(p) & 0 & s'(p) \\ 0 & 0 & 0 \\ -s'(p) & 0 & c'(p) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
$$= \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix},$$

and

$$\frac{\partial R}{\partial q} = \frac{\partial R_q}{\partial q} R_p R_r$$
$$= \begin{bmatrix} c'(q) & -s'(q) & 0 \\ s'(q) & c'(q) & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
$$\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

The six individual terms for the derivative of a transformation matrix $T_{(t)}$ with respect to $t = [x, y, z, r, p, q]$ evaluated at $\theta = 0$ are

$$\frac{\partial T_{(t)}}{\partial x} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = G_1,$$

$$\frac{\partial T_{(t)}}{\partial y} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = G_2,$$

$$\frac{\partial T_{(t)}}{\partial z} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} = G_3,$$

$$\frac{\partial T_{(t)}}{\partial r} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = G_4,$$

$$\frac{\partial T_{(t)}}{\partial p} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = G_5,$$

$$\frac{\partial T_{(t)}}{\partial q} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = G_6.$$

These are the canonical generators of SE(3). Together they give the full Jacobian, which is a $4 \times 4 \times 6$ tensor,

$$\frac{\partial T_{(t)}}{\partial t} = \begin{bmatrix} G_1 & G_2 & G_3 & G_4 & G_5 & G_6 \end{bmatrix}.$$

This tensor simplifies when right multiplied by a 4×1 homogeneous vector $v = [x, y, z, 1]^{\mathrm{T}}$

$$\frac{\partial T_{(t)}}{\partial t} v = -\begin{bmatrix} G_1 & G_2 & G_3 & G_4 & G_5 & G_6 \end{bmatrix} v$$
$$= \begin{bmatrix} I & [\bar{v}]_\times \\ 0 & 0 \end{bmatrix}$$

where $[\bar{v}]_\times$ is the 3×3 skew-symmetric matrix built from $\bar{v} = [v_1, v_2 v_3]^{\mathrm{T}}$. Note that this form greatly simplifies computing the Jacobians, $\partial g_{i,k}/\partial t_c$, in Section 3.3.

The inverse of a homogeneous transformation matrix is

$$\begin{bmatrix} R & \bar{v} \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} R^{\mathrm{T}} & -R^{\mathrm{T}}\bar{v} \\ 0 & 1 \end{bmatrix}.$$

The derivative of an inverse transform is thus

$$\frac{\partial T_{(t)}^{-1}}{\partial t} = \frac{\partial}{\partial t} \begin{bmatrix} R^{\mathrm{T}} & -R^{\mathrm{T}}\bar{v}^{\mathrm{T}} \\ 0 & 1 \end{bmatrix}$$
$$= -\begin{bmatrix} G_1 & G_2 & G_3 & G_4 & G_5 & G_6 \end{bmatrix}$$
$$= -\frac{\partial T_{(t)}}{\partial t}$$
$$= \frac{\partial T_{(-t)}}{\partial t}$$

because $R = I$ when we take the Jacobian with respect to $[x, y, z]$ and $[x, y, z] = 0$ when we take the Jacobian with respect to R.

References

Agarwal, S., Snavely, N., Simon, I., Seitz, S. M. and Szeliski, R. (2009). Building Rome in a day. *IEEE International Conference on Computer Vision*.

Bosse, M. C., Newman, P. M., Leonard, J. J. and Teller., S. (2004). SLAM in large-scale cyclic environments using the Atlas framework. *The International Journal of Robotics Research*, **23**(12): 1113–1139.

Brooks, R. (1985). Visual map making for a mobile robot. *IEEE International Conference on Robotics and Automation*.

Brown, D. (1958). *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).

Choset, H. and Nagatani, K. (2001). Topological simultaneous localization and mapping (SLAM): toward exact localization without explicit localization. *IEEE Transactions on Robotics and Automation*, **17**: 125–137.

Cummins, M. (2009). *Probabilistic Localization and Mapping in Appearance Space*. PhD thesis, University of Oxford.

Cummins, M. and Newman, P. (2007). Probabilistic appearance based navigation and loop closing. *IEEE Conference on Robotics and Automation*.

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

Davison, A., Reid, I., Molton, N. and Stasse, O. (2007). MonoSLAM: Realtime single camera SLAM. *IEEE Transactions Pattern Analysis and Machine Intelligence*, **29**(6): 1113–1139.

Deans, M. C. (2005). *Bearings-Only Localization and Mapping*. PhD thesis, School of Computer Science, Carnegie Mellon University.

Dellaert, F. (2005). Square root SAM. *Proceedings of Robotics: Science and Systems*, pp. 1181–1203.

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

Engels, C., Stewenius, H. and Nister, D. (2006). Bundle adjustment rules. *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*, pp. 57–64.

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.

Fraundorfer, F., Engels, C. and Nister, D. (2007). Topological mapping, localization and navigation using image collections. *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Goedem'e, T., Nuttin, M., Tuytelaars, T. and Gool, L. V. (2007). Omnidirectional vision based topological navigation. *International Journal of Computer Vision*, **74**(3): 219–236.

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*.

Guivant, J. and Nebot, E. (2001). Optimization of the simultaneous localization and map-building algorithm for real-time implementation. *IEEE Transactions on Robotics and Automation*, **17**(3): 242–257.

Harris, C. and Stephens, M. (1988). A combined corner and edge detector. *Proceedings of The Fourth Alvey Vision Conference*, Manchester, pp. 147–151.

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

Holmes, S. A., Sibley, G., Klein, G., and Murray, D. W. (2009). A relative frame representation for fixed-time bundle adjustment in monocular SFM. *Proceedings IEEE International Conference on Robotics and Automation*.

Howard, A. (2008). Real-time stereo visual odometry for autonomous ground vehicles. *IEEE Conference on Robots and Systems (IROS)*.

Howard, A., Sukhatme, G. S. and Mataric, M. J. (2006). Multirobot simultaneous localization and mapping using manifold representations. *Proceedings of the IEEE*, **94**(7): 1360–1369.

Huber, P. J. (1964). Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, **35**(2): 73–101.

Kaess, M. (2008). *Incremental Smoothing and Mapping*. PhD Thesis, Georgia Institute of Technology.

Kaess, M., Ranganathan, A. and Dellaert, F. (2008). iSAM: Incremental smoothing and mapping. *IEEE Transactions on Robotics and Automation*, **244**(6): 1365–1378.

Klein, G. and Murray, D. (2008). Improving the agility of keyframe-based SLAM. *European Conference on Computer Vision*.

Konolige, K. and Agrawal, M. (2008). FrameSLAM: From Bundle Adjustment to Real-Time Visual Mapping. *IEEE Transactions on Robotics and Automation, IEEE Journal*

of Robotics and Automation, International Journal of Robotics Research, **24**(5): 1066–1077.

Krauthausen, P., Dellaert, F. and Kipp, A. (2006). Exploiting locality by nested dissection for square root smoothing and mapping. *Proceedings of Robotics: Science and Systems*.

Kuipers, B. and Byun, Y. (1988). A robust qualitative method for spatial learning in unknown environments. *Proceedings of the National Conference on Artificial Intelligence*.

Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, **60**(2): 91–110.

Martinelli, A., Nguyen, V., Tomatis, N. and Siegwart, R. (2007). A relative map approach to SLAM based on shift and rotation invariants. *Robotics and Autonomous Systems*, **55**(1): 50–61.

McLauchlan, P. F. (1999). *The Variable State Dimension Filter Applied to Surface-based Structure from Motion*. Technical Report, University of Surrey.

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 and Automation*, **24**(6): 1352–1364.

Mei, C., Sibley, G., Cummins, M., Reid, I. and Newman, P. (2009). A constant-time efficient stereo SLAM system. *Proceedings of the British Machine Vision Conference*.

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

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., Sibley, G., Smith, M., Cummins, M., Harrison, A., Mei, C., Posner, I., Shade, R., Schroeter, D., Murphy, L., Churchill, W., Cole, D. and Reid, I. (2009). Navigating, recognizing and describing urban spaces with vision and lasers. *The International Journal of Robotics Research*, **1**: 1–28.

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.

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.

Pinies, P. and Tardos, J. D. (2007). Scalable SLAM building conditionally independent local maps. *IEEE Conference on Intelligent Robots and Systems*.

Ranganathan, A. (2008). *Probabilistic Topological Maps*. PhD Thesis, Georgia Institute of Technology.

Ranganathan, A., Kaess, M. and Dellaert, F. (2007). Loopy SAM. *International Joint Conferences on Artificial Intelligence*, pp. 2191–2196.

Ranganathan, A., Menegatti, E., and Dellaert, F. (2006). Bayesian inference in the space of topological maps. *IEEE Transactions on Robotics and Automation*, **22**(1): 92–107.

Rosten, E. and Drummond, T. (2006). Machine learning for high-speed corner detection. *European Conference on Computer Vision*.

Sciavicco, L. and Siciliano, B. (1996). *Modelling and Control of Robot Manipulators*. Berlin: Springer.

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. (2007). *Long Range Stereo Data-fusion from Moving Platforms*. PhD Thesis, University of Southern California.

Sibley, G., Matthies, L. and Sukhatme, G. (2007). *A Sliding Window Filter for Incremental SLAM*, chapter 7, (*Lecture Notes in Electrical Engineering*). Berlin, Springer, Volume 8. pp. 103–112.

Sivic, J. and Zisserman, A. (2006). Video Google: efficient visual search of videos. *Toward Category-Level Object Recognition*, Berlin, Springer: pp. 127–144.

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.

Sorenson, H. W. (1980). *Parameter Estimation: Principles and Problems*. New York, Marcel Dekker, Inc.

Steder, B., Grisetti, G., Grzonka, S., Stachniss, C., Rottmann, A. and Burgard, W. (2007). Learning maps in 3D using attitude and noisy vision sensors. *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Steedly, D. and Essa, I. (2001). Propagation of innovative information in non-linear least-squares structure from motion. *Proceedings of ICCV01*, pp. 223–229.

Steedly, D., Essa, I. and Dellaert, F. (2003). Spectral partitioning for structure from motion. *IEEE International Conference on Computer Vision*.

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

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

Thrun, S., Koller, D., Ghahmarani, Z. and Durrant-Whyte., H. (2002b). SLAM updates require constant time. *Workshop on the Algorithmic Foundations of Robotics*.

Triggs, B., McLauchlan, P. F., Hartley, R. I. and Fitzgibbon, A. W. (2000). Bundle Adjustment—A Modern Synthesis. ICCV '99: Proceedings of the International Workshop on Vision Algorithms. Berlin, Springer, pp. 298–375.