FrameSLAM: from Bundle Adjustment to Realtime Visual Mappping

Kurt Konolige and Motilal Agrawal

***Abstract*—Many successful indoor mapping techniques employ frame-to-frame matching of laser scans to produce detailed local maps, as well as closing large loops. In this paper, we propose a framework for applying the same techniques to visual imagery. We match visual frames with large numbers of point features, using classic bundle adjustment techniques from computational vision, but keep only relative frame pose information (a *skeleton*). The skeleton is a reduced nonlinear system that is a faithful approximation of the larger system, and can be used to solve large loop closures quickly, as well as forming a backbone for data association and local registration. We illustrate the working of the system with large outdoor datasets (10 km), showing large-scale loop closure and precise localization in real time.**

I. Introduction

Visual motion registration is a key technology for many applications, since the sensors are inexpensive and provide high information bandwidth. We are interested in using it to construct maps and maintain precise position estimates for mobile robot platforms indoors and outdoors, in extended environments with loops of more than 10 km, and in the absence of global signals such as GPS. This is a classic SLAM (simultaneous localization and mapping) problem. In a typical application, we gather images at modest frame rates, and extract hundreds of features in each frame for estimating frame to frame motion. Over the course of even 100 m, moving at 1 m/sec, we can have a thousand images and half a million features. The best estimate of the frame poses and feature positions, even for this short trajectory, is a large nonlinear optimization problem. In previous research using laser rangefinders, one approach to this problem was to perform frame-to-frame matching of the laser scans, and keep only the constraints among the frames, rather than attempting to directly estimate the position of each scan reading (feature) [15], [25], [27].

Using frames instead of features reduces the nonlinear system by a large factor, but still poses a problem as frames accumulate over extended trajectories. To efficiently solve large systems, we reduce the size of the system by keeping only a selected subset of the frames, the *skeleton*. Most importantly, and contrary to almost all current research in SLAM, the skeleton system consists of nonlinear constraints. This property helps it to maintain accuracy even under severe reductions, up to several orders of magnitude smaller than the original system. Figure 1 shows an example from an urban round-about scene. The original system has 700 stereo frames and over 100K 3D features. A skeleton graph at 5m intervals eliminates the features in favor of a small number of frame-frame links, and reduces the number of frames by almost an order of magnitude. The full nonlinear skeleton can be solved in 35 ms.

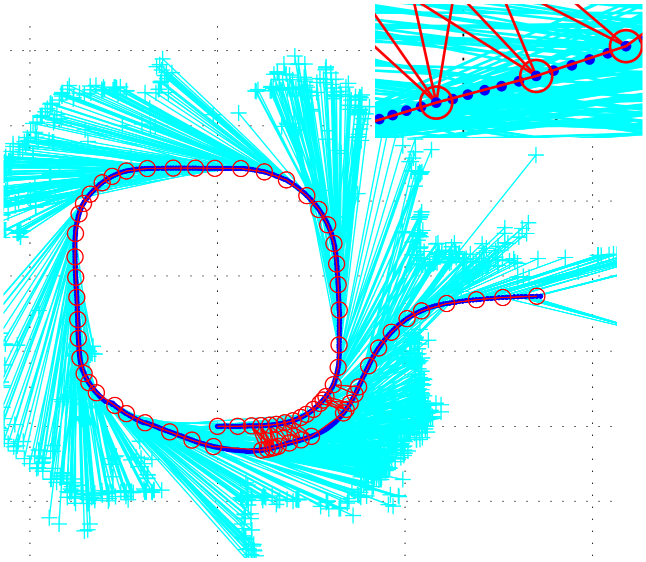

Fig. 1. Skeleton reduction of a 100 meter urban scene. Full Bayes graph is 700 frames (blue dots) and ∼100K features (cyan crosses). Frame-feature links are in cyan – only one in 200 are shown for display. Original frames are in blue (see inset for a closeup). The 133 reduced frames and their links are in red. The reduced graph is solved in 35 ms.

In this paper, we present frameSLAM, a complete visual mapping method that uses the skeleton graph as its map representation. Core techniques implemented in the system include:

- Precise, realtime visual odometry for incremental motion estimation.
- Nonlinear least-squares estimation for local registration and loop closure, resulting in accurate maps.
- Constant-space per area map representation. The skeleton graph is used for data association as well as map estimation. Continuous traversal of an area does not cause the map to inflate.
- Experimental validation. We perform small and large-scale experiments, in indoor, urban, and rough-terrain settings, to validate the method and to show realtime behavior.

A. Related Work

There has been a lot of recent work in visual SLAM, most of it concentrated on global registration of 3D features (undelayed SLAM). One approach, corresponding to classical EKF SLAM, is to use a large EKF containing all of the features [7], [8], [32]; another, corresponding to fastSLAM, is to use a set of particles representing trajectories and to attach a small EKF to each feature [10], [26]. EKF SLAM has obvious problems in representing larger environments, because the size of the EKF filter grows beyond realtime computation. Some recent work has split the large filter into submaps [30], which can then deal with environments on the order of 100 m, with some indications of realtime behavior.

A scale problem also afflicts fastSLAM approaches: it is unclear how many particles are necessary for a given environment, and computational complexity grows with particle set size. Additionally, the 6 DOF nature of visual SLAM makes it difficult for fastSLAM approaches to close loops. For these reasons, none of these approaches has been tried in the type of large-scale, rough-terrain geometry that we present here.

A further problem with feature-based systems is that they lose the power of consensus geometry to give precise estimates of motion and to reject false positives. The visual odometry backbone that underlies frameSLAM is capable of errors of less than 1% over many hundreds of meters [23], which has not yet been matched by global feature-based systems.

In delayed SLAM, camera frames are kept as part of the system. Several systems rely on this framework [13], [18], [19]. The iSAM filter approach [19] uses an information filter for the whole system, including frames and features. A nice factorization method allows fast incremental update of the filter. While this approach works for modest-size maps (∼1000 features), other techniques must be used for the large numbers of features found in visual SLAM.

The delayed filter of [13], [18], like our approach, keeps *only* the frames – visual feature matching between frames induces constraints on the frames. These constraints are maintained as a large, sparse information filter, and used to reconstruct underwater imagery over scales of 200-300m. It differs from our work in using a large linear filter instead of a reduced skeleton of nonlinear constraints: the incremental cost of update grows linearly with map size, and is not proposed as a realtime approach.

Since these techniques rely on linear systems, they could encounter problems when closing large loops, where the linearization would have to change significantly.

A very recent paper by Steder et al. [33], and earlier work by Kelly [37] has a very similar approach to skeleton systems. They also keep a constraint network of relative pose information between frames, based on stereo visual odometry, and solve it using nonlinear least square methods. However, their motion is basically restricted to 4 degrees of freedom, and the matching takes place on near-planar surfaces with downward-looking cameras, rather than the significantly more difficult forward-looking case.

Another related research area is place recognition for long-range loop closure. Recently, several new systems have emerged that promise realtime recovery of candidate place matches over very large databases [5], [29].

II. Skeleton Systems

We are interested in localizing and mapping using just stereo cameras, over large distances. There are three tasks to be addressed:

1) Local registration. The system must keep track of its position locally, and build a registered local map of the environment.
2) Long-range tracking. The system must compute reasonable long-range trajectories with low error.
3) Global registration. The system must be able to recognize previously-visited areas, and re-use local map information.

As the backbone for our system, we use *visual odometry* (VO) to determine incremental motion of stereo cameras. The principle of VO is to simultaneously determine the pose of camera frames and position of world features by matching the image projections of the features (Figure 2), a well-known technique in computational vision. Our research in this area has yielded algorithms that are accurate to within several meters over many kilometers, when aided by an IMU [1], [2], [23].

In this paper, we concentrate on solving the first and third tasks. VO provides very good incremental pose results, but like any odometry technique, it will drift because of the composition of errors. To stay registered in a local area, or to close a large loop, requires recognition of previous frames, and the ability to integrate current and previous frames into a consistent global structure. Our approach is to consider the large system consisting of all camera frames and features as a Bayes net, and to produce reduced versions – skeletons – that are easily solvable but still accurate.

A. Skeleton Frames

Let us consider the problem of closing a large loop, which is at the heart of the SLAM technique. This loop could contain thousands of frames and millions of features – for example, one of our outdoor sets has 65K stereo pairs, each of which generates 1000 point features or more. We can't consider closing this loop in a reasonable time; eventually, we have to reduce the size of the system when dealing with large-scale consistency. At the same time, we want to be able to keep locally dense information for precise navigation. Recognition of this problem has led to the development of *sub-maps*, small local maps that are strung together to form a larger system [3], [30]. Our system of reductions has a similar spirit, but with the following key differences.

- No explicit submaps need to be built; instead, *skeleton frames* form a reduced graph structure on top of the system.
- The skeleton structure can scale to allow optimization of any portion of the system.

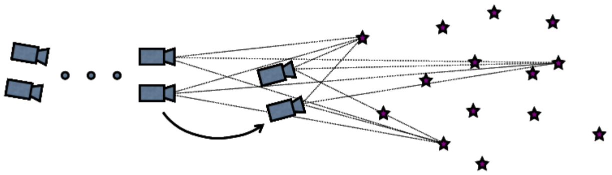

Fig. 2. Stereo frames and 3D points. VO estimates the pose of the frames and the positions of the 3D points at the same time, using image projections.

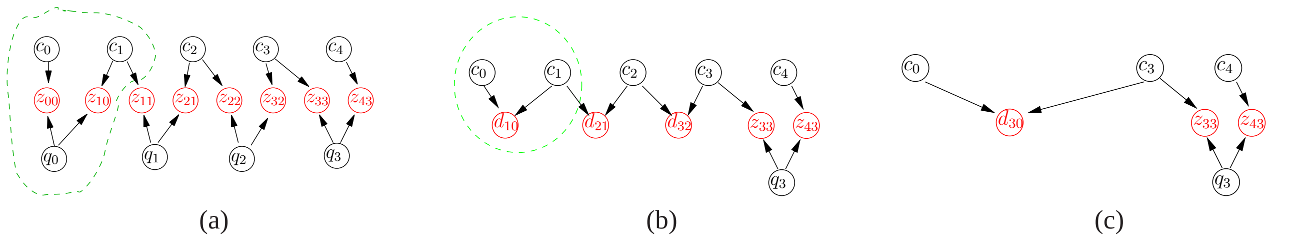

Fig. 3. Skeleton reduction as a Bayes net. System variables are in black, measurements are in red: each measurement represents one constraint. The initial net (a) contains camera frames c_i and features q_i that give rise to image points. In (b), most of the features are marginalized out, yielding measurements between the frames. In (c), some of the intermediate frames have also been marginalized.

- The skeleton frames support constraints that do not fully specify a transformation between frames. Such constraints arise naturally in the case of bearing-only landmarks, where the distance to the landmark is unknown.

Since there are many more features than frames, we want to reduce feature constraints to frames constraints; and after this, to reduce further the number of frames, while still maintaining fidelity to the original system. The general idea for computing a skeleton system is to convert a subset of the constraints into an appropriate single constraint. Consider Figure 3, which shows a Bayes net representation of image constraints. When a feature q_j is seen from a camera frame c_i, it generates a measurement z_{ij}, indicated by the arrows from the variables. Now take the subsystem of two measurements z_{00} and z_{10}, circled in (a). These generate a multivariate gaussian PDF $p(c_0, c_1, q_0|z_{00}, z_{01})$. We can reduce this PDF by marginalizing q_0, leaving $p(c_0, c_1|z_{00}, z_{01})$. This PDF corresponds to a *synthetic constraint* between c_0 and c_1, which is represented in Figure 3(b) by the circled nodes. In a similar manner, a PDF involving $c_0 - c_3$ can be reduced, via marginalization, to a PDF over just c_0 and c_3, as in Figure 3(c).

It is clear that the last system is much simpler than the first. As with any reduction, there is a tradeoff between simplicity and accuracy. The reduced system will be close to the original system, as long as the frame variables are not moved too far from their original positions. When they are, the marginalization that produced the reduction may no longer be a good approximation to the original PDF. For this reason, the form of the new constraint is very important. If it is tied to the global position of the frames, then it will become unusable if the variables are moved from their original global position, say in closing a loop. But, if the constraint uses relative positions, then the frames can be moved anywhere, as long as their relative positions are not displaced too much. The key technique introduced in this paper is the derivation of reduced relative constraints that are an accurate approximation of the original system.

A reduced system can be much easier to solve than the original one. The original system in Figure 1 has over 100K (vector) variables, and our programs run out of space trying to solve it; while its reduced form has just 133 vector variables, and can be solved in 35 ms.

III. Nonlinear Least Squares Estimation

The most natural way to solve large estimation problems is nonlinear least squares (NLSQ). There are several reasons why NLSQ is a convenient and efficient method. First, it offers an easy way to express image constraints and their uncertainty, and directly relates them to image measurements. Second, NLSQ has a natural probabilistic interpretation in terms of Gaussian multinormal distributions, and thus the Bayes net introduced in the previous section can be interpreted and solved using NLSQ methods. This connection also points the way to reduction methods, via the theoretically sound process of marginalizing out variables. Finally, by staying within a nonlinear system, it is possible to avoid problems of premature linearization, which are especially important in loop closure.

These properties of NLSQ have been of course been exploited in previous work, especially in structure-from-motion theory of computer vision (Sparse Bundle Adjustment or SBA [36]), from which this research draws inspiration. In this section, we describe the basics of the measurement model, NLSQ minimization, variable reduction by marginalization, and the "lifting" of linear to nonlinear constraints.

A. Frames and Features

Referring to Figure 2, there are two types of system variables, camera frames c_i and 3D features q_j. Features are parameterized by their 3D position; frames by their position and Euler roll, pitch, and yaw angles:

$$q_j = [x_j, y_j, z_j]^\top \tag{1}$$

$$c_i = [x_i, y_i, z_i, \phi_i, \psi_i, \theta_i]^\top. \tag{2}$$

Features project onto a camera frame via the projection equations. Each camera frame c_i describes a point transformation from world to camera coordinates as a rotation and translation $p_i = R_i p_w + t_i$; we abbreviate as the 3x4 matrix $T_i = [R_i|t_i]$. The projection $[u, v]^\top$ onto the image is given by

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K T_i \begin{bmatrix} q_i \\ 1 \end{bmatrix}, \tag{3}$$

where K is the 3x3 camera intrinsic matrix [16].

For stereo, Equation (3) describes projection onto the reference camera, which by convention we take to be the left one. A similar equation for $[u', v']^\top$ holds for the right camera, with $t'_i = t_i + [B, 0, 0]^\top$ as the translation part of the projection matrix.[1] We will label the image projection $[u, v, u', v']^\top$ from c_i and q_j as z_{ij}.

B. *Measurement Model*

For a given frame c_i and feature q_j, the expected projection is given by

$$z_{ij} = h(x_{ij}) + v_{ij}, \tag{4}$$

where v_{ij} is gaussian noise with covariance W_{ij}^{-1}, and for convenience we let x_{ij} stand for c_i, q_j. Here the measurement function h is the projection function of Equation 3. Typically W^{-1} is diagonal, with a standard deviation of a pixel.

For an actual measurement $\bar{z}_{ij}$, the induced error or cost is

$$\Delta z(x_{ij}) = \bar{z}_{ij} - h(x_{ij}), \tag{5}$$

and the weighted square cost is

$$\Delta z(x_{ij})^\top W_{ij} \Delta z(x_{ij}). \tag{6}$$

We will refer to the weighted square cost as a *constraint*. Note that the PDF associated with a constraint is

$$p(z|x_{ij}) \propto \exp(\frac{1}{2}\Delta z(x_{ij})^\top W_{ij} \Delta z(x_{ij})). \tag{7}$$

Although only frame-feature constraints are presented, there is nothing to prevent other types of constraints from being introduced. For example, regular odometry would induce a constraint $\Delta z(c_i, c_j)$ between two frames.

C. *Nonlinear Least Squares*

The optimization problem is to find the best set of model parameters x – camera frames and feature positions – to explain vectors of observations $\bar{z}$ – feature projections on the image plane. The nonlinear least squares method estimates the parameters by finding the minimum of the sum of the constraints (Sum of Squared Errors, or SSE):

$$f(x) = \sum_{ij} \Delta z(x_{ij})^\top W_{ij} \Delta z(x_{ij}). \tag{8}$$

A more convenient form of f eliminates the sum by concatenating each error term into a larger vector. Let $\Delta z(x) \equiv [\Delta z(x_{00})^\top, \cdots, \Delta z(x_{mn})^\top]^\top$ be the full vector of measurements, and $W \equiv \mathrm{diag}(W_{00}, \cdots, W_{mn})$ the block-diagonal matrix of all the weights. Then the SSE equation (8) is equivalent to the matrix equation

$$f(x) = \Delta z(x)^\top W \Delta z(x). \tag{9}$$

Assuming the measurements are independent under x, the matrix form can be interpreted as a multinormal PDF $p(z|x)$, and by Bayes' rule $p(x|z) \propto p(z|x)p(x)$. To maximize the likelihood $p(x|z)$, we minimize the cost function (9) [36].

Since (9) is nonlinear, finding the minimum typically involves reduction to a linear problem in the vicinity of an initial solution. After expanding via Taylor series and differentiating, we get the incremental equation

$$H\delta\mathbf{x} = -g, \tag{10}$$

where g is the gradient and H is the Hessian of f with respect to x. Finally, after getting rid of some second-order terms, we can write Equation (10) in the *Gauss-Newton* form

$$J^\top W J\, \delta x = -J^\top W \Delta z, \tag{11}$$

with J the Jacobian $\partial h/\partial x$, and the Hessian H approximated by $J^\top W J$.

In the nonlinear case, one starts with an initial value x_0, and iterates the linear solution until convergence. In the vicinity of a solution, convergence is quadratic. Under the Maximum Likelihood (ML) interpretation, at convergence H is the inverse of the covariance of x, that is, the information or precision matrix. We will write $J^\top W J$ as Λ to emphasize its role as the information matrix. In the delayed-state SLAM formulation [13], Λ serves as the system filter in a non-iterated, incremental version of the SSE problem.

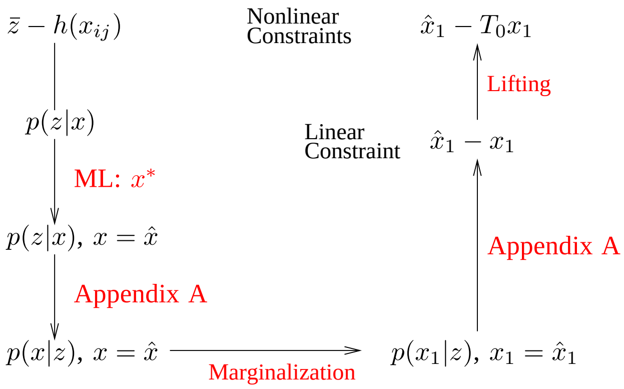

Fig. 4. Reduction process diagram. Nonlinear measurements constraints induce a Gaussian PDF over z, which is converted to a PDF over the system variables x. Reduction gives a smaller PDF over just x_1, which corresponds to the linear constraint $\hat{x}_1 - x_1$. Lifting relativizes this to $\hat{x}_1 - T_0 x_1$.

D. *Nonlinear Reduction*

At this point we have the machinery to accomplish the reduction shown in Figure 3, eliminating variables from the constraint net. Consider all the constraints involving the first two frames ($\Delta z(c_i, q_j)$ for $i = 0, 1$). Figure 4 diagrams the process of reducing this to a single nonlinear constraint $\Delta z(c_0, x_1)$. On the left side, the NSLQ process induces a PDF over the variables x, with an ML value of $\hat{x}$. Marginalization of all variables but x_1 gives a PDF over just x_1, which corresponds to the linear constraint $\hat{x}_1 - x_1$. The "lifting" process relativizes this to a nonlinear constraint. Mathematical justification for several of the steps is given in Appendix A.

The following algorithm specifies the process in more detail.

[1] We assume a standard calibration for the stereo pair, where the internal parameters are equal, and the right camera is displaced along the camera frame X axis by an amount B.

Constraint Reduction

Input: set of constraints $\Delta z(x_i, \cdots)^\top W \Delta z(x_i, \cdots)$ in variables $c_0, x_{1,n}$, where c_0 (at least) is a frame variable.

Output: constraint $\Delta z(c_0, x_1)^\top W' \Delta z(c_0, x_1)$ that represents the same PDF for c_0, x_1.

1) Fix $c_0 = 0$ (the origin).
2) Solve Eq. (11) in $\Delta z(x_i, \cdots)$ to get an estimated mean $\hat{x}_i$ and Hessian $\hat{\Lambda}$.
3) Convert $\hat{\Lambda}$ to a reduced information matrix $\hat{\Lambda}_1$ for x_1 by marginalizing all variables $x_{2,n}$.
4) Lift the linear constraint in x_1 to a nonlinear constraint $\Delta z(c_0, x_1) = \hat{x}_1 - T_0 x_1$ with weight $\hat{\Lambda}_1$ – T_0 is the transformation to c_0's coordinates.

In step 1, we constrain the system to have c_0 as the origin. Normally the constraints leave the choice of origin free, but we want all variables relative to c_0's frame.

The full system (minus c_0, which is fixed) is then solved in step 2, generating estimated values $\hat{\mu}$ for all variables, as well as the information matrix $\hat{\Lambda}$. These two represent a gaussian multivariate distribution over the variables $x_{1,n}$. Next, in step 3, the distribution is marginalized, getting rid of all variables except x_1. The reduced matrix $\hat{\Lambda}_1$ represents a PDF for x_1 that summarizes the influence of the other variables.

Step 4 is the most difficult to understand. The information matrix $\hat{\Lambda}_1$ is derived under the condition that c_0 is the origin. But we need a constraint that will hold when intermixed with other constraints, where c_0 may be nonzero. The final step lifts the result from a fixed c_0 to *any* pose for c_0. Here's how it works. Steps 2 and 3 produce a mean $\hat{x}_1$ and information matrix $\hat{\Lambda}_1$ such that $\exp[\frac{1}{2}(\hat{x}_1 - x_1)^\top \hat{\Lambda}_1 (\hat{x}_1 - x_1)]$ is a PDF for x_1. This PDF is equivalent to a *synthetic observation* on x_1, with the linear measurement function $h(x_1) = x_1$. Now replace $h(x_1)$ with the relativized function $h(c_0, x_1) = T_0 x_1$, where T_0 transforms x_1 into c_0's coordinates. When $c_0 = 0$, this gives exactly the same PDF as the original $h(x_1)$. And for any $c_0 \neq 0$, we can show that the constraint $\Delta z(c_0, x_1)$ produces the same PDF as the constraints $\Delta z(x_i, \cdots)$ (see Appendix A).

What is interesting about $\Delta z(c_0, x_1)$ is its nonlinear nature. It represents a spring connecting c_0 and x_1, with precisely the right properties to accurately substitute for the larger nonlinear system $\Delta z(c_0, x_1, \cdots)$. The accuracy is affected by how closely the reduced system follows two assumptions that were made:

- The displacement between c_0 and x_1 is close to $\hat{x}_1$.
- None of the variables $x_{2,n}$ participate in other constraints in the system.

These assumptions are not always fully met, especially the second one. Nonetheless, we will show in experiments with large outdoor systems that even very drastic variable reductions, such as those in Figure 1, give accurate results.

E. Marginalization

In Step 3 we require the reduction of an information matrix $\hat{\Lambda}$ to extract a marginal distribution between two of the vector variables. Just deleting the unwanted variable rows and columns would give the *conditional distribution* $p(c_0, x_1 | x_2, \cdots)$. This distribution significantly overestimates the confidence in the connection, compared to the marginal, since the uncertainty of the auxiliary variables is disregarded [12]. The correct way to marginalize is to convert $\hat{\Lambda}$ to its covariance form by inversion, delete all but the entries for c_0 and x_1, and then invert back to the information matrix. A shortcut to this procedure is the *Schur complement* [14], [31], [21], which is also useful in solving sparse versions of Equation (9). We start with a block version of this equation, partitioning the variables into frames c and features q:

$$\begin{bmatrix} H_{11} & H_{12} \\ H_{12}^\top & H_{22} \end{bmatrix} \begin{bmatrix} \delta c \\ \delta q \end{bmatrix} = \begin{bmatrix} -J_c^\top W_c \Delta z(c) \\ -J_q^\top W_q \Delta z(q) \end{bmatrix} \qquad (12)$$

Now we define a reduced form of this equation:

$$\bar{H}_{11} \delta c = -g, \qquad (13)$$

with

$$\bar{H}_{11} \equiv H_{11} - H_{12} H_{22}^{-1} H_{12}^\top \qquad (14)$$
$$-g \equiv -J_c^\top W_c \Delta z(c) - H_{12} H_{22}^{-1} J_q^\top W_q \Delta z(q). \qquad (15)$$

Here the matrix equation (13) involves only the variables c. There are two cases where this reduction is useful.

1) Constraint reduction. The reduced Hessian $\bar{H}_{11}$ is the information matrix for the variables c, with variables q marginalized out. Equation (15) gives a direct way to compute the marginalized Hessian in Step 3 of constraint reduction.
2) Visual odometry. For the typical situation of many features and few frames, the reduced equation offers an enormous savings in computing NLSQ, with the caveat: H_{22} must be easily invertible. Fortunately, for features the matrix H_{22} is diagonal, since features only have constraints with frames, and thus the Jacobian J in $J^\top W J$ affects just H_{12}.

IV. Data Association and Landmarks

The raw material for constraints comes from data association between image features. We have implemented a method for matching features across two camera frames that serves a dual purpose. First, it enables incremental estimation of camera motion for tracking trajectories (visual odometry). Second, on returning to an area, we match the current frame against previous frames that serve as landmarks for the area. These landmark frames are simply the skeleton system that is constructed as the robot explores an area – a reduced set of frames, connected to each other by constraints. Note that landmark frames are *not* the same as feature landmarks normally used in non-delayed EKF treatments of VSLAM. Features are only represented implicitly, by their projections onto camera frames. Global and local registration is done purely by matching images between frames, and generating frame-frame constraints from the match.

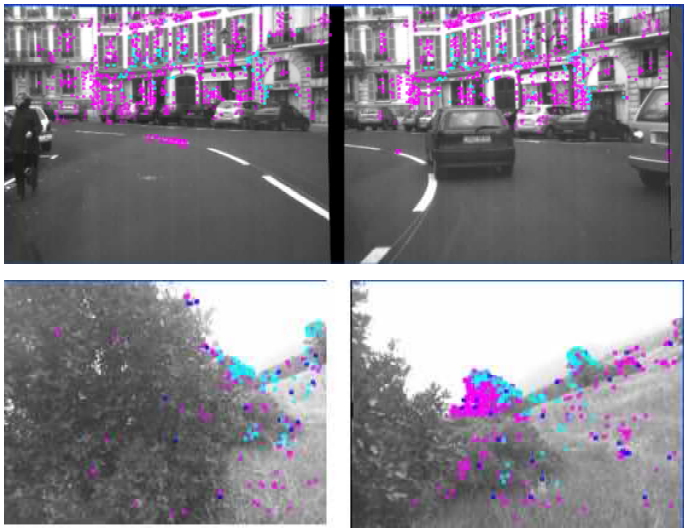

Fig. 5. Matching between two frames in an urban scene (top) and rough terrain (bottom). Objects that are too close, such as the bush in the bottom right image, cannot be found by stereo, and so have no features. Inlier matches for the best consensus estimate are in cyan; other features found but not part of the consensus are in magenta. The upper pair is about 5m distance between frames, and there are moving objects. The lower pair has significant rotation and translation.

A. *Matching Frames*

Consider the problem of matching stereo frames that are close spatially. Our goal is to precisely determine the motion between the two frames, based on image feature matches. Even for incremental frames, rapid motion can cause the same feature to appear at widely differing places in two successive images; the problem is even worse for wide-baseline matching. The images of Figure 5 show some difficult examples from urban and offroad terrain. Note the significant shift in feature positions.

One important aspect of matching is using image features that are stable across changes of scale and rotation. While SIFT [24] and SURF [17] are the features of choice, they are not suitable for real-time implementations (15 Hz or greater). Instead, we use a novel multiscale center-surround feature called CenSure. In previous research, we have shown that this operator has the requisite stability properties, but is just slightly more expensive than the Harris operator [23].

In any image feature matching scheme, there will be false matches. In the worst cases of long-baseline matching, sometimes only 10% of the matches are good ones. We use the following robust matching algorithm to find the best estimate, taking advantage of the geometric constraints imposed by rigid motion.

Consensus Match

1) Extract features from the left image.
2) Perform stereo to get corresponding feature positions in the right image.
3) Match to features in previous left image using normalized cross correlation.
4) Form consensus estimate of motion using RANSAC on three points.
5) Use NLSQ to polish the result over the two frames (four images).

Three matched points give a motion hypothesis between the frames [16]. The hypothesis is checked for inliers by projecting all feature points onto the two frames (Equation 3). Features that are within 2 pixels of their projected position are considered to be inliers – note that they must project correctly in all four images. The best hypothesis (most inliers) is chosen and optimized using the NLSQ technique of Section III-E. Some examples are shown in Figure 5 with matched and inlier points indicated.

This algorithm is used for both incremental tracking and wide-baseline matching, with different search parameters for finding matching features. Note that we do not assume any motion model or other external aids in matching.

B. *Visual Odometry*

Consensus matching is the input to a visual odometry process for estimating incremental camera motion. To make incremental motion estimation more precise, we incorporate several additional methods.

1) Key frames. If the estimated distance between two frames is small, and the number of inliers is high, we discard the new frame. The remaining frames are called key frames. Typical distances for key frames are about 0.1 - 0.5 meters, depending on the environment.
2) Incremental bundle adjustment. A sliding window of about 10 keyframes (and their features) is input to the NLSQ optimization. Using a small window significantly increases the accuracy of the estimation [11], [23], [28], [35].
3) IMU data. If an Inertial Measurement Unit is available, it can decrease the angular drift of VO, especially tilt and roll, which are referenced to gravity normal [23], [34].

For a small enough set of frames, recent research has shown that incremental bundle adjustment can be done very efficiently using Hessian reduction [11], [28].

The third item is an interesting addition to NLSQ estimation. The following equations describe IMU measurements of gravity normal and yaw angle increments:

$$g_i = h_g(c_i) \tag{16}$$

$$\Delta\psi_{i-1,i} = h_{\Delta\psi}(c_{i-1}, c_i) \tag{17}$$

The function $h_g(c)$ returns the deviation of the frame c in pitch and roll from gravity normal. $h_{\Delta\psi}(c_{i-1}, c_i)$ is just the yaw angle difference between the two frames. Using accelerometer data acting as an inclinometer, with a very high noise level to account for unknown accelerations, is sufficient for (16) to constrain roll and pitch angles and completely avoid drift. For yaw angles, only a good IMU can increase the accuracy VO estimates. In the experiments section, we show results both with and without IMU aiding.

C. *Wide Baseline Matching*

To register a new frame against a landmark frame, we use the same consensus matching technique as for visual odometry.

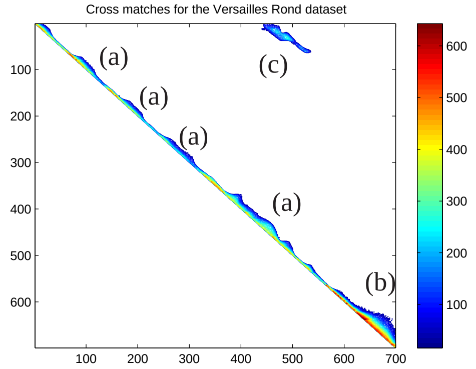

Fig. 6. Matching statistics on a 700 image urban scene. The number of inliers in matches between frames is color-coded; inliers counts below 30 were eliminated. Only the upper right triangle is used. Note the longer matching areas where the vehicle goes along a straight stretch (a), and the very long matches at the end as the car slows down along a straightway (b). The off-diagonal (c) is the set of matched frames closing the loop.

This procedure has several advantages.

- Sensitivity over scale. The CenSure features are scale-independent, and so are stable when there has been significant viewpoint change.
- Efficiency. The CenSure features and stereo are already computed for visual odometry, so wide baseline matching just involves steps 3-5 of the Consensus Match algorithm. This can be done at over 30 Hz.
- Specificity. The geometric consistency check almost guarantees that there will be no false positives, even using a very low inlier threshold.

Figure 5 shows an example of wide baseline matching in the upper urban scene. The distance between the two frames is about 5m, and there are distractors such as cars and pedestrians. The buildings are very self-similar, so the geometric consistency check is very important in weeding out bad matches. In this scene, there are about 800 features per image, and only 100 inliers for the best estimate match.

To analyze sensitivty and selectivity, we computed the inlier score for every possible cross-frame match in the 700 frame urban sequence shown in Figure 1. Figure 6 shows the results, by number of inliers. Along the diagonal, matching occurs for several frames, to an average of 10m along straight stretches. The only off-diagonal matching occurs at the loop closure. The lower scores on closure reflect the sideways offset of the vehicle from its original trajectory. Consensus matching produced essentially perfect results for this dataset, giving no false positives, and correctly identifying loop closures.

D. Place Recognition

We implement a simple but effective scheme for recognizing places that have already been visited, using the consensus match just presented. This scheme is not intended to do "kidnapped" recognition, matching an image against a large database of places [5], [29]. Instead, it functions when the system has a good idea of where it is relative to the landmark frames that have been accumulated. In a local area, for example, the system always stays well-registered, and has to search only a small number of frames for matches. Over larger loops, the method is linear in the size of the area it must search.

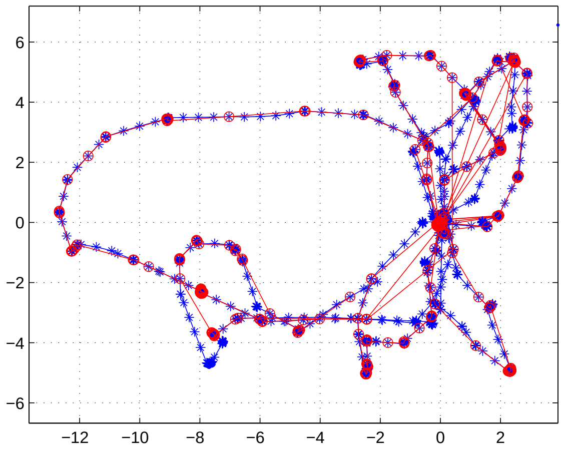

Fig. 7. Skeleton graph for an indoor dataset (dataset courtesy of Robert Sim). Distances are in meters. Dataset has 8,490 key frames (a small set of these are shown in blue). The skeleton graph is shown in red; some of the longer-range links are from landmark frame matching, and some are from reduction eliminating frames. NLSQ computation for the skeleton graph takes 65 ms.

The main problem that arises is which landmark frames should serve as candidates to match to the current frame. Ideally, we would use the relative covariance estimates computed from the Bayes net as a gate, that is, $\Delta x^\top W \Delta x < d$, where Δx is the distance between the current frame c_n and a candidate landmark c_i [9]. However, computing relative covariance involves marginalizing all variables in the system Hessian, and is too expensive to be done online [12]. Instead, we use the skeleton to provide an approximate covariance that is conservative. In the skeleton net, we find the shortest path from c_n to c_i, and then compose the incremental covariances along the path to get an estimate of the relative covariance. An efficient search of the net ($O(n \log n)$) can be done using Dijkstra's method [33]. In practice this method needs several milliseconds on the largest graphs we have attempted; for very large spaces, a hierarchical skeleton would be appropriate to keep computational costs reasonable.

One property we would like to observe is that the space consumed by the skeleton should be proportional to the area visited. On continual movement in an area, the skeleton will continue to grow unless frames are removed. Our method here is to marginalize out any frames that are within a distance d and angle a of an existing skeleton frame. As an example, we show an indoor dataset consisting of 8,490 keyframes in a small 12m x 12m area (Figure 7). The skeleton graph was produced using a landmark distance of 2 meters and 10 degrees, reducing the graph to 272 frames with 23 cross-frame links. The final skeleton graph is solved by NLSQ in 65 ms.

E. Realtime Implementation

Our VO system has been implemented and runs at 15 Hz on 512x384 resolution stereo images using a 2 GHz processor, and is in habitual use in demo runs on an outdoor robot [1],

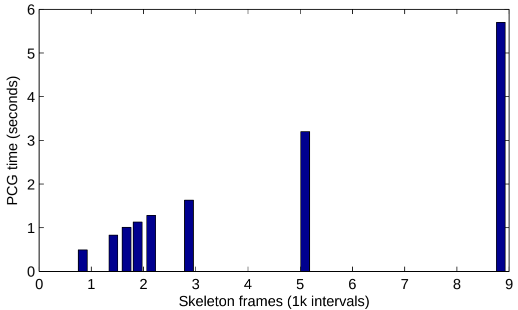

Fig. 8. Large-scale timings for a skeleton graph. The trajectory of the graph was ∼10 km. The NLSQ time is linear in the size of the skeleton for this graph.

[2], [23]. At this point we have implemented the rest of the frameSLAM system only in processing logs, for which we report timing results; we are transitioning the system to an outdoor robot, and will report system results soon.

The main strategy for adding registration matches is to use a dual-core machine: run VO on one core, and wide-baseline matching and skeleton computations on another. Wide-baseline matching is on an "anytime" basis: we match the current keyframe against a set of candidates that pass the Mahalanobis gate, until the next keyframe comes in, when we start matching that. Whenever a good match is found, the system performs an NLSQ update, either on the whole skeleton, or a smaller area around the current keyframe, depending on the situation.

Of course, this strategy does not address the significant problems of frame storage and retrieval for very large systems, as done by recent place-recognition algorithms [5], [29]. It may also miss some matches, since it does not explore all possible candidates. But for skeletons of less than 10k frames, where the frames can be kept in RAM, it works well.

For efficient calculation of the NLSQ updates, we use a preconditioned conjugate gradient algorithm (PCG) that has been shown to have good computational properties – in many cases, the complexity increases linearly with the size of the skeleton [20]. For the large outdoor dataset, Figure 8 plots the PCG time against the size of the skeleton. Note that these are for a very large loop closure of a combined 10 km trajectory – typically only a small local area needs to be optimized.

V. Experiments

It is important to test VSLAM systems on data gathered from real-world platforms. It is especially important to test under realistic conditions: narrow FOV cameras, full 3D motion, and fast movement, as these present the hardest challenge for matching and motion estimation. For this research, we used three widely different sources of stereo frames.

1) An indoor sequence consisting of 22K frames in a small area, moving slowly (courtesy of Robert Sim [10]). The stereo system had a wide FOV, narrow baseline, and was purely planar motion.
2) An outdoor automobile sequence, the Versailles Rond dataset (courtesy of Andrew Comport [4]). This dataset has 700 frames with fast motion, 1 m baseline, narrow FOV, covering about 400 meters.

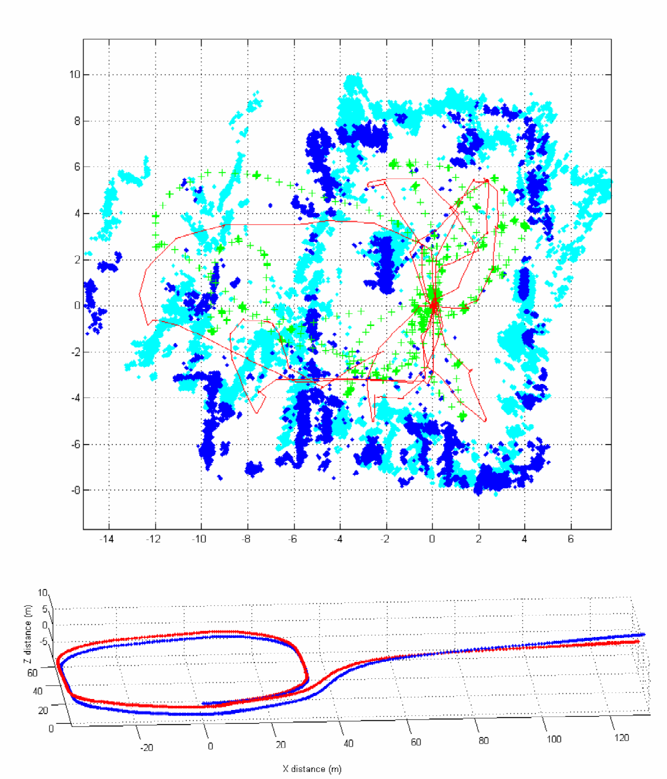

Fig. 9. Top: Indoor dataset showing raw VO (green crosses are frames and cyan dots are features) vs. frameSLAM result (red trajectory and blue features). Note that the frameSLAM features correspond to a rectangular set of rooms, while the VO results are skewed after the loop. Bottom: Versailles Rond urban dataset. Blue is raw VO, red is frameSLAM result (see Figure 1 for cross-frame links). Note that the Z offset of the loop has been corrected.

3) Two outdoor rough-terrain sequences of about 5 km each, from the Crusher project [6]. Baseline is 0.5 m, narrow FOV, and fast, full 3D motion with lots of bouncing on rough terrain. These datasets offer a unique opportunity, for two reasons. First, they are autonomous runs through the same waypoints; they overlap and cross each other throughout, and end up at the same place for a large loop closure. Second, the dataset is instrumented with both IMU and RTK GPS data, and our frameSLAM results can be computed for both aided and unaided VO, and compared to ground truth.

   A certain number of frames, about 2%, cannot be matched for VO in this dataset. We fill in these values with IMU data.

A. Planar Datasets

The indoor and Versailles Rond datasets were used throughout the paper to illustrate various aspects of the frameSLAM system. Because there is no ground truth, they offer just anecdotal evidence of performance: the indoor dataset illustrates that the skeleton does not grow with continued traversal of a region; the Versailles Rond dataset shows loop closure over a significant distance. From the plots of Figure 9, the improvement in fidelity to the environment is apparent. One possible measure of the improvement is the planarity of the trajectories. The table below lists the relevant statistics for the two runs. The timings are for NLSQ optimization of the entire skeleton.

	Length	Key	Skeleton	Planarity (m)		Time
	(m)	frames	frames	VO	fS	(ms)
Indoor	150	8.2K	272	0.15	0.11	65
Versailles	370	700	133	0.19	0.15	35

B. Crusher Datasets

The Crusher data comes from two autonomous 5 km runs, which overlap significantly and form a loop. There are 20K keyframes in the first run, and 22K in the second. Over 20 million features are found and matched in the keyframe images, and roughly 3 times that in the total image set. Figure 10 (top) gives an idea of the results from raw VO on the two runs. There is significant deviation in all dimensions by the end of the loop (circled in red). With a skeleton of frames at 5m intervals, there were a total of 1978 reduced frames, and 169 wide-baseline matches between the runs using consensus matching. These are shown as red links in the top plot.

The middle plot shows the result of applying frameSLAM to a 5m skeleton. Here the red trajectory is ground truth for the blue run, and it matches the two runs at the beginning and end of the trajectory (circled on the left). The two runs are now consistent with each other, but still differ from ground truth at the far right end of the trajectory. This is to be expected: the frameSLAM result will only be as good as the underlying odometry when exploring new areas.

If VO is aided by an IMU (Section IV-B), global error is reduced dramatically. The bottom plot shows the frameSLAM result using the aided VO – note that the blue run virtually overlays the RTK GPS ground truth trajectory.

How well does the frameSLAM system reduce errors from open-loop VO? We should not expect any large improvement in long-distance drift at the far point of trajectories, since SLAM does not provide any global input that would correct such drift. But, we should expect dramatic gains in relative error, that is, between frames that are globally close, since SLAM enforces consistency when it finds correspondences. To show this, we compared relative pose of every frame pair to ground truth, and plotted the results as a function of distance between the frames. Figure 11 shows the results for both raw and aided VO. For raw VO (top plot), the open-loop errors are very high, because of the large drift at the end of the trajectories (Figure 10, top). With the cross-links enforcing local consistency, frameSLAM gives much smaller errors for short distances, and degrades with distance, a function of yaw angle drift. Note that radical reductions in the size of the skeleton, from 1/4 to 1/400 of the original keyframes, have negligible effect, proving the accuracy of the reduced system.

A similar story exists for IMU-aided VO. Here the errors are much smaller because of the smaller drift of VO. But the same gains in accuracy occur for small frame distances, and again there is almost no effect from severe skeleton reductions until after 300 meters.

VI. Conclusions

We have described frameSLAM, a system for visual SLAM that is capable of precise, realtime estimation of motion, and also is able to keep track of local registration and global consistency. The key component is a skeleton system of visual frames, that act both as landmarks for registration, and as a network of constraints for enforcing consistency. frameSLAM has been validated through testing in a variety of different environments, including large-scale, challenging offroad datasets of 10 km.

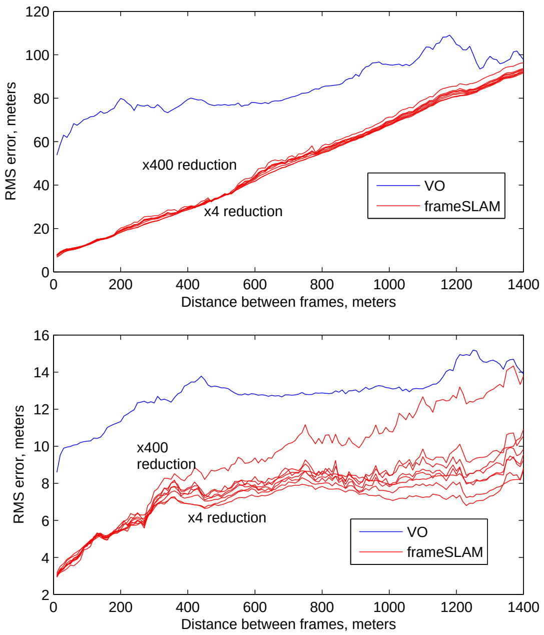

Fig. 11. RMS error as a function of distance. For every pair of frames, the error in their relative pose is plotted as a function of the distance between the frames. Top: Unaided VO. The blue line shows poor open-loop VO performance, even for short distances; frameSLAM (red lines) gives excellent results for these distances. Skeleton reduction factor has negligible influence. Bottom: IMU-aided VO.

We are currently porting the system to two live robot platforms [6], [22], with the intent of providing completely autonomous offroad navigation using just stereo vision. The VO part of the system has already been proven over a year of testing, but cannot eliminate the long-term drift that accrues over a run. With the implementation of the skeleton graph, we expect to be able to assess the viability of the anytime strategy for global registration presented in Section IV-E.

Appendix A
Nonlinear constraint "lifting"

Let c_0, x_1 and $\mathbf{q}$ be a set of variables with measurement cost function

$$\Delta z^\top W_i \Delta z \tag{18}$$

and measurement vector $\bar{z}$. For c_0 fixed at the origin, let $\hat{\Lambda}_1$ be the Hessian of the reduced form of (18), according to Step 3 of the Constraint Reduction algorithm. We want to show that the cost function

$$\Delta z'^\top \hat{\Lambda}_i \Delta z' \tag{19}$$

has approximately the same value at the ML estimate $\hat{x}_1$, where $z'(c_0, x_1) = T_0 x_1$ and $\bar{z}' = \hat{x}_1$. To do this, we show that the likelihood distributions are approximately the same.

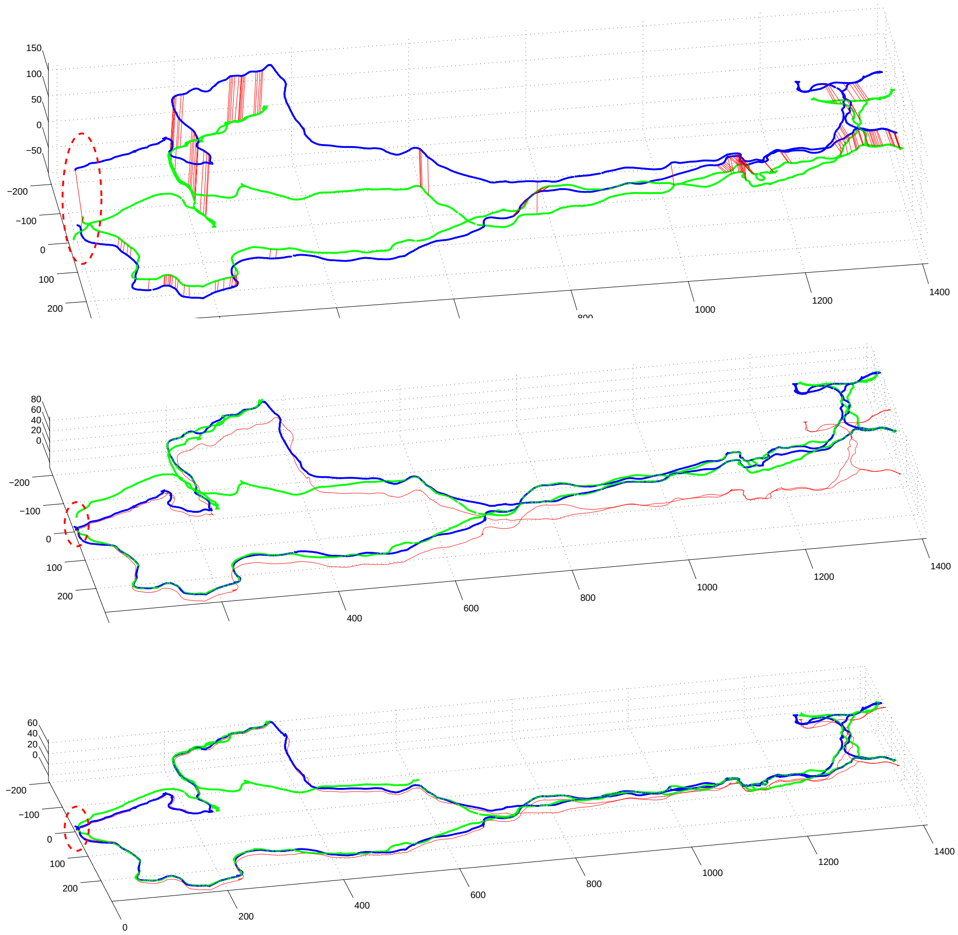

Fig. 10. XYZ plot of two Crusher trajectories (blue and green) of about 5 km each. Top shows the raw VO, with cross-matched frames with red links. The start and finish of both runs is at the left, circled in red; the runs are offset vertically by 20 m at the begnninng to display the links. Note the loop closure between the end of the blue run and the beginning of the green run. Middle shows the frameSLAM-corrected system for a 5m skeleton. The ground truth for the blue run is in red. The relative positions of the green and blue runs have been corrected, and the loop closed. The bottom shows the excellent result for IMU-aided VO.

The cost function (18) has the joint normal distribution

$$P(\hat{z}|x_1, \mathbf{q}) \propto \exp(-\frac{1}{2}\Delta z^\top W_i \Delta z) \tag{20}$$

We want to find the distribution (and covariance) for the variable x_1. Let $\mathbf{x} = x_1, \mathbf{q}$, and $f(\mathbf{x})$ the cost function (18). Expanding $f(\mathbf{x} + \delta\mathbf{x})$ in a Taylor series, the cost function becomes

$$\begin{aligned}
& (\hat{z} - f(\mathbf{x}))^\top W (\hat{z} - f(\mathbf{x})) && (21)\\
\simeq\ & (\hat{z} - f(\mathbf{x}) - J\delta x)^\top W (\hat{z} - f(\mathbf{x}) - J\delta x) && (22)\\
=\ & \delta x_1^\top \hat{\Lambda}_1 \delta x_1 - 2\Delta z W J \delta x + \text{const}, && (23)
\end{aligned}$$

where we have used the Schur equality to reduce the first term of the third line. As Δz vanishes at $\hat{x}$, the last form is quadratic in x_1, and so is a joint normal distribution over x_1. From inspection, the covariance is $\hat{\Lambda}_1^{-1}$. Hence the ML distribution is

$$P(x_1|\hat{z}) \propto \exp(-\frac{1}{2}(\hat{x}_1 - x_1)^\top \hat{\Lambda}_1 (\hat{x}_1 - x_1)). \tag{24}$$

The cost function for this PDF is (19) for c_0 fixed at the origin, as required. When c_0 is not the origin, the cost function (18) can be converted to an equivalent function by transforming all variables to c_0's coordinate system. The value stays the same because the measurements are localized to the positions of c_0 and x_1 – any global measurement, for example a GPS reading, would block the equivalence. Thus, for arbitrary c_0, (20) and (24) are approximately equal just when x_1 is given in c_0's coordinate system. Since (24) is produced by the cost function (19), we have the approximate equivalence of the two cost functions.

References

[1] M. Agrawal and K. Konolige. Real-time localization in outdoor environments using stereo vision and inexpensive gps. In *ICPR*, August 2006.

[2] M. Agrawal and K. Konolige. Rough terrain visual odometry. In *Proc. International Conference on Advanced Robotics (ICAR)*, August 2007.

[3] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, and S. Teller. An atlas framework for scalable mapping. In *ICRA*, 2003.

[4] A. Comport, E. Malis, and P. Rives. Accurate quadrifocal tracking for robust 3d visual odometry. In *ICRA*, 2007.

[5] M. Cummins and P. M. Newman. Probabilistic appearance based navigation and loop closing. In *ICRA*, 2007.

[6] DARPA Crusher Project. http://www.rec.ri.cmu.edu/projects/autonomous/index.htm.

[7] A. Davison. Real-time simultaneaous localisation and mapping with a single camera. In *ICCV*, pages 1403–1410, 2003.

[8] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *IEEE PAMI*, 29(6), 2007.

[9] M. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (slam) problem. *IEEE Trans. Robotics and Automation*, 17(3), 2001.

[10] P. Elinas, R. Sim, and J. J. Little. sigmaslam: Stereo vision slam using the rao-blackwellised particle filter and a novel mixture proposal distribution. In *ICRA*, 2007.

[11] C. Engels, H. Stewnius, and D. Nister. Bundle adjustment rules. *Photogrammetric Computer Vision*, September 2006.

[12] R. M. Eustice, H. Singh, and J. J. Leonard. Exactly sparse delayed-state filters for view-based SLAM. *IEEE Trans. Robotics*, 22(6), 2006.

[13] R. M. Eustice, H. Singh, J. J. Leonard, and M. R. Walter. Visually mapping the RMS Titanic: conservative covariance estimates for SLAM information filters. *Intl. J. Robotics Reserach*, 25(12), 2006.

[14] U. Frese. A proof for the approximate sparsity of slam information matrices. In *ICRA*, 2005.

[15] J. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *Proc. IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, pages 318–325, Monterey, California, November 1999.

[16] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[17] T. T. Herbert Bay and L. V. Gool. Surf: Speeded up robust features. In *European Conference on Computer Vision*, May 2006.

[18] V. S. Ila, J. Andrade, and A. Sanfeliu. Outdoor delayed-state visually augmented odometry. In *Proc. IFAC Symposium on Intelligent Autonomous Vehicles*, 2007.

[19] M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Fast incremental smoothing and mapping with efficient data association. In *ICRA*, Rome, 2007.

[20] K. Konolige. Large-scale map-making. In *Proceedings of the National Conference on AI (AAAI)*, 2004.

[21] K. Konolige and M. Agrawal. Frame-frame matching for realtime consistent visual mapping. In *Proc. International Conference on Robotics and Automation (ICRA)*, 2007.

[22] K. Konolige, M. Agrawal, R. C. Bolles, C. Cowan, M. Fischler, and B. Gerkey. Outdoor mapping and navigation using stereo vision. In *ISER*, 2007.

[23] K. Konolige, M. Agrawal, and J. Solà. Large scale visual odometry for rough terrain. In *Proc. International Symposium on Research in Robotics (ISRR)*, November 2007.

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

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

[26] T. K. Marks, A. Howard, M. Bajracharya, G. W. Cottrell, and L. Matthies. Gamma-slam: Stereo visual slam in unstructured environments using variance grid maps. In *ICRA*, 2007.

[27] M. Montemerlo and S. Thrun. Large-scale robotic 3-d mapping of urban structures. In *ISER*, 2004.

[28] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. Real time localization and 3d reconstruction. In *CVPR*, volume 1, pages 363 – 370, June 2006.

[29] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *CVPR '06*, 2006.

[30] L. Paz, P. Jensfelt, J. Tards, and J. Neira. Ekf slam updates in o(n) with divide and conquer slam. In *ICRA*, 2007.

[31] G. Sibley, G. S. Sukhatme, and L. Matthies. Constant time sliding window filter slam as a basis for metric visual perception. In *ICRA Workshop*, 2007.

[32] J. Solà, M. Devy, A. Monin, and T. Lemaire. Undelayed initialization in bearing only slam. In *ICRA*, 2005.

[33] B. Steder, G. Grisetti, C. Stachniss, S. Grzonka, A. Rottmann, and W. Burgard. Learning maps in 3d using attitude and noisy vision sensors. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2007.

[34] D. Strelow and S. Singh. Motion estimation from image and inertial measurements. *International Journal of Robotics Research*, 23(12), 2004.

[35] N. Sunderhauf, K. Konolige, S. Lacroix, and P. Protzel. Visual odometry using sparse bundle adjustment on an autonomous outdoor vehicle. In *Tagungsband Autonome Mobile Systeme*. Springer Verlag, 2005.

[36] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzibbon. Bundle adjustment - a modern synthesis. In *Vision Algorithms: Theory and Practice*, LNCS, pages 298–375. Springer Verlag, 2000.

[37] R. Unnikrishnan and A. Kelly. A constrained optimization approach to globally consistent mapping. In *Proceedings International Conference on Robotics and Systems (IROS)*, 2002.