Visual-Inertial Navigation, Mapping and Localization: A Scalable Real-Time Causal Approach

Eagle S. Jones Stefano Soatto

Submitted to the Intl. J. of Robotics Research, August 27, 2009
Revised May 10, 2010; Accepted September 23, 2010

Abstract

We present a model to estimate motion from monocular visual and inertial measurements. We analyze the model and characterize the conditions under which its state is observable, and its parameters are identifiable. These include the unknown gravity vector, and the unknown transformation between the camera coordinate frame and the inertial unit. We show that it is possible to estimate both state and parameters as part of an on-line procedure, but only provided that the motion sequence is "rich enough," a condition that we characterize explicitly. We then describe an efficient implementation of a filter to estimate the state and parameters of this model, including gravity and camera-to-inertial calibration. It runs in real-time on an embedded platform, and its performance has been tested extensively. We report experiments of continuous operation, without failures, re-initialization, or re-calibration, on paths of length up to 30Km. We also describe an integrated approach to "loop-closure," that is the recognition of previously-seen locations and the topological re-adjustment of the traveled path. It represents visual features relative to the global orientation reference provided by the gravity vector estimated by the filter, and relative to the scale provided by their known position within the map; these features are organized into "locations" defined by visibility constraints, represented in a topological graph, where loop closure can be performed without the need to re-compute past trajectories or perform bundle adjustment. The software infrastructure as well as the embedded platform is described in detail in a technical report (Jones and Soatto (2009).)

1 Introduction

Reliable estimation of the trajectory of a moving body ("ego-motion") is key to a number of applications, particularly to autonomous navigation of ground and air vehicles. This problem is sometimes referred to as "odometry" or "navigation" in different contexts. We are particularly interested in applications where motion is used to close a control loop, for instance in autonomous navigation, and therefore an estimate has to be provided in real-time with minimal latency. A variety of methods and sensory modalities have been brought to bear to tackle this problem. The most common sensors are *inertial* (IMU, accelerometers and gyroscopic rate sensors), that have the advantage of high frame-rate and relatively small latency, but only provide relative motion and are subject to significant drift (or significant cost). Where possible, inertial sensors are often complemented by *global positioning* sensors (GPS), typically measuring time-of-flight to precisely localized satellites. These have the advantage of providing a global position at a relatively high frame-rate, but with lower accuracy and precision, and with availability issues due to the presence of multi-path reflection in urban areas or complete denial of service indoor and in adversarial environments. Vision presents a natural complement to inertial sensors, in that it is highly accurate, provides semi-local position estimate (relative to visibility constraints), but has a significant latency, computational complexity, and robustness issues depending on the complexity of the visual environment. Vision can also provide, in addition to ego-motion, estimates of the three-dimensional (3-D) layout of the scene, that can in turn be useful for obstacle avoidance and planning. It is no surprise that vision and inertial sensors are ubiquitous in nature, especially in animals with a high degree of mobility.

Other modalities, such as range data from active sensors (time-of-flight, lidar, radar) present challenges in applications where cost and interference are significant issues, such as in civilian transportation, and simpler sensors, such as encoder-based odometers or acoustic proximity sensors, have limited applicability although they can under certain circumstances be used effectively.

In this work we focus on visual and inertial sensors, their integration, and their use in ego-motion estimation, localization and mapping. We address some of the key issues that have hindered progress, including managing calibration[1] and unknown model parameters, making the estimation process stable, and dealing with practical issues such as handling failure modes from each modality.

In section 2 we describe the model we use to estimate ego-motion from vision and inertial sensors. The state of this model includes the unknown motion, and in the presence of uncertainty and noise the problem can be cast as a stochastic filtering problem. Because we are looking for a point-estimate (we are interested in the motion of *one* vehicle), we assume that the posterior density of the state is unimodal, and seek for an approximation of the mode. The presence of spurious modes is due to measurements that violate the assumptions implicit in the data formation process, and are therefore rejected in a robust-statistical hypothesis testing framework that we describe in sect. 3 and evaluate empirically.

One of the necessary conditions for a stochastic filter to converge is that the state be *observable*. We study the observability of motion in section 2.2.1. However, in addition to the unknown state, the data formation model also includes unknown parameters, such as the gravity vector or the relative position between optical and inertial sensors (camera-to-inertial calibration), that are not directly measured. Therefore, we study the problem of *parameter identifiability* to ascertain whether such parameters can be estimated on-line along with the state of the filter. The answer to this question depends on the motion undergone by the platform, which yields the notion of a *sufficiently exciting* motion sequence, that we characterize analytically.

In section 3 we describe the implementation of a filter, including issues that relate to the handling of outliers and other failure modes. In section 4 we present empirical results on both indoor and outdoor image sequences of length up to 30Km. This only refers to "open loop" navigation, and discards the estimates of 3-D structure that are nevertheless provided, in real-time, by the filter.

In section 5 we show how the previous results can be used for the problem of building a coherent map of the 3-D environment that can be used for *localization*. When one or more images are available, one can use them to determine whether this location has been previously visited, and if so to localize the position and orientation, relative to the map, from which the image(s) were taken. When a previously seen location is detected, one can re-align the estimated trajectories, a process known as "loop-closing." However, geometric alignment would require a costly post-processing of the data. Instead, we perform *topological loop closing*, by adding an edge in a graph where each node represents a "location," defined by co-visibility, and edges represent geometric constraints between locations. Thus the path is now a closed loop in the graph, although the composition of all the transformations along the edges of the loop does not not necessarily yield a closed trajectory in space. The geometric inconsistency, if any, is pushed to the farthest end of the map, and bears no consequences in real-time control applications, including path planning, obstacle avoidance etc. One can always run a post-mortem bundle adjustment to close the trajectory in space if so desired.

While we and others have studied the observability of ego-motion from monocular image sequences and inertial measurements,[2] to the best of our knowledge this is the first time the analysis includes the effects of unknown gravity and camera-to-inertial calibration parameters. In particular, we show that – under general position conditions – both gravity and camera-to-inertial calibration are identifiable. The general position conditions require that the motion undergone by the sensing platform has non-zero linear acceleration and non-constant rotational axis. Under such "sufficiently exciting" motion, the gravity vector is observable, and so is the camera-to-inertial calibration. Unfortunately, typical motions in many relevant applications violate such conditions. These include planar motion or driving at constant speed. Under these conditions, gravity

[1] Note that throughout this manuscript, autocalibration refers to the camera-to-IMU calibration, not to the intrinsic parameters of the camera – those can be inferred through standard calibration procedures as customary in this application domain (Tsai, 1987).

[2] See, for instance (Baldwin et al., 2007, Euston et al., 2008, Konolige et al., 2007, Jones et al., 2007, Kelly and Sukhatme, 2009) and the June 2006, vol. 26(6) special issue of the International Journal of Robotics Research describing the proceedings of the INNERVIS workshop.

and calibration parameters cannot be estimated, and therefore the filter is saturated to prevent drift. On the practical side, we describe the implementation of a system that can estimate ego-motion in real-time as well as estimate the 3-D position of hundreds of point features and achieves competitive performance in test sequences of larger scope than previously experimented with. We have implemented all the algorithms described in a modular software infrastructure that is described in detail in a technical report (Jones and Soatto, 2009).

Related literature will be referenced throughout the paper. We focus on *monocular* vision and inertial sensors operating in real-time, and even then there is a considerable body of related work. At one end of the spectrum are "vision-heavy" approaches that perform frame-rate updates of vision-based estimates, and use the IMU to address situations where visual tracking is lost because of fast motion, occlusions, or sudden changes of illumination. A representative example is (Zhu et al., 2007), based on (Nister, 2003). At the opposite end of the spectrum are "inertial-heavy" approaches that rely on frame-rate updates of inertial measurements, and use vision to reduce drift at a reduced frame-rate. A representative example is (Mourikis and Roumeliotis, 2007). Although our focus is in the integrated modeling and inference of a representation that is suitable for navigation and mapping, rather than on the particular algorithm to search large maps for loop-closure, our manuscript is closely related to existing work on large-scale localization. This includes (Bosse et al., 2004), that focuses on the mapping aspects and is not specific to a particular sensing mechanism, (Eade and Drummond, 2007a) that uses monocular vision only and therefore cannot exploit a global orientation and scale reference, (Guivant and Nebot, 2001) that focuses on real-time implementation based on range sensors, (Klein and Murray, 2007) that focuses on augmented reality and therefore does not address large spaces, long-term drift, scale and gravity issues, (Konolige and Agrawal, 2008, Mouragnon et al., 2006) that similarly addresses monocular vision, and (Nebot and Durrant-Whyte, 1999) that integrates inertial and bearing sensors for IMU calibration. Our work follows in the footsteps of (Jin et al., 2000), that demonstrated real-time structure from motion using a hand-held camera, based on the model of (Chiuso et al., 2002) where stability of the error dynamics was proven, (Favaro et al., 2001, 2003) that used a photometric representation to perform loop-closure and (Jones et al., 2007) that performed an analysis of the observability of ego-motion jointly with identifiability of gravity and camera-to-inertial calibration parameters.

Our implementation achieves results comparable to the state of the art in terms of overall drift (localization error as a percentage of distance traveled), but on tests performed in significantly larger sequences. Direct comparison has been performed with (Mourikis and Roumeliotis, 2007), who have kindly provided us with their data, on which we have tested our algorithm. Other algorithms cited above do not provide data to perform a direct comparison. We have made both our data and our implementation available at the website `http://vision.ucla.edu/corvis`, together with a description of our software platform.

In addition to accuracy and robustness, latency is an important factor in real-time applications. Most work that relies on the traditional epipolar geometry pipeline (Ma et al., 2003), including various sorts of bundle-adjustment, introduces delays in visual processing, because one has to wait for a small batch of images to be collected that have "sufficient parallax." This also requires handling exceptions, for instance when the platform stops, singular motions (planar scenes, quadrics), and a choice of the number of frames in a batch. To the best of our knowledge, all existing schemes for integrating visual and inertial sensors assume that gravity is known. Our system processes all data in a causal fashion, and does not require accurate knowledge of gravity or of a geo-referenced system. As a result, it is considerably more flexible and easier to use.

2 Ego-motion Estimation

There have been many attempts in the computer vision community to build a robust "visual odometry" module, including (Jin et al., 2000, Nister, 2003, Chiuso et al., 2002, Yang and Pollefeys, 2003, Davison, 2003, Goedeme et al., 2007). To the best of our knowledge, the first real-time demonstration was Jin et al. (2000). Some incorporate inertial measurements, either as inputs to the model (Roumeliotis et al., 2002), or as states (Qian et al., 2001, Dickmanns and Mysliwetz, 1992). More recently, (Veth et al., 2006, Veth and Raquet, 2006) presented a tightly-coupled vision and inertial model aimed at precision geolocation, that however requires complex calibration, a known terrain model and known landmarks. Another vision and

inertial model that we have already mentioned is (Mourikis and Roumeliotis, 2006, 2007). Most vision-inertial integration approaches use an Earth-centered, Earth-fixed (ECEF) coordinate model, rather than a local one. While each approach has its merits, the local coordinate model is more general and flexible, and does not require external knowledge of a global position and orientation. Of course, where global positioning information becomes available, it can be easily integrated into the measurement system.

In the next section we introduce our notation and describe the simplest model for ego-motion based on standard rigid-body assumption.

2.1 Modeling ego-motion

Our exposition employs notation that is fairly standard and described in detail in Chapter 2 of (Murray et al., 1994) or (Ma et al., 2003). We represent the motion of the rigid body holding both the vision sensor (camera) and inertial sensors (IMU) via $g = (R, T) \in SE(3)$, with $R \in SO(3)$ a rotation (orthogonal, positive unit-determinant) matrix, and $T \in \mathbb{R}^3$ a translation vector. $\widehat{V}^b = g^{-1}\dot{g} \in se(3)$ is the so-called "body velocity," i.e., the velocity of the moving body relative to the inertial frame, written in the coordinates of the moving body's reference frame. In homogeneous coordinates, we have

$$g = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}; \quad \widehat{V}^b = \begin{bmatrix} \widehat{\omega}^b & v^b \\ 0 & 0 \end{bmatrix}; \; V^b = \begin{bmatrix} \omega^b \\ v^b \end{bmatrix} \tag{1}$$

where $\widehat{\omega} \in so(3)$ is the skew-symmetric matrix constructed from the coordinates of $\omega \in \mathbb{R}^3$, and $v \in \mathbb{R}^3$ is the translational velocity. For a motion with constant rotational velocity ω, we have $R(t) = \exp(\widehat{\omega}t)$ if $R(0) = I$. The null rigid motion is $e = (I, 0)$. When writing the change of coordinates from a frame, say "b" for the moving body, to another frame, say "s" for the spatial (inertial) frame, we use a subscript g_{sb}, again following (Murray et al., 1994). With this notation in place, we proceed to formalize the problem of estimating body pose and velocity.

We denote with $X_0^i \in \mathbb{R}^3$ the coordinates of a point in the inertial frame, and $y^i(t) \in \mathbb{R}^2$ its projection onto the (moving) image plane. Along with the pose g_{sb} of the body relative to the spatial frame and (generalized) body velocity V_{sb}^b, these quantities evolve via

$$\begin{cases} \dot{X}_0^i = 0, \quad i = 1, \dots, N \\ \dot{g}_{sb}(t) = g_{sb}(t)\widehat{V}_{sb}^b(t), \quad g_{sb}(0) = e \end{cases} \tag{2}$$

which can be broken down into the rotational and translational components $\dot{R}_{sb}(t) = R_{sb}(t)\widehat{\omega}_{sb}^b(t)$ and $\dot{T}_{sb}(t) = R_{sb}(t)v_{sb}^b(t)$. The translational component of body velocity, v_{sb}^b, can be obtained from the last column of the matrix $\frac{d}{dt}\widehat{V}_{sb}^b(t)$. That is, $\dot{v}_{sb}^b = \dot{R}_{sb}^T\dot{T}_{sb} + R_{sb}^T\ddot{T}_{sb} = -\widehat{\omega}_{sb}^b v_{sb}^b + R_{sb}^T\ddot{T}_{sb} \doteq -\widehat{\omega}_{sb}^b v_{sb}^b + \alpha_{sb}^b$, which serves to define $\alpha_{sb}^b \doteq R_{sb}^T\ddot{T}_{sb}$. An ideal inertial measurement unit would measure $\omega_{sb}^b(t)$ and $\alpha_{sb}^b(t) - R_{sb}^T(t)\gamma$ where γ denotes the gravity vector in the inertial frame. An ideal vision algorithm capable of maintaining correspondence and overcoming occlusions, with the body frame and the camera frame coinciding, would measure $y^i(t) = \pi\left(R_{sb}^T(t)(X_0^i - T_{sb}(t))\right)$. Here $\pi : \mathbb{R}^3 \to \mathbb{R}^2;\; X \mapsto y = [X_1/X_3,\; X_2/X_3]^T$ is a canonical central (perspective) projection. To summarize, a simplistic, idealized model to generate the sensed data y_{imu} and y^i is given by

$$\begin{cases} \dot{X}_0^i = 0, \quad i = 1, \dots, N \\ \dot{R}_{sb}(t) = R_{sb}(t)\widehat{\omega}_{sb}^b(t), \quad R_{sb}(0) = I \\ \dot{T}_{sb}(t) = R_{sb}(t)v_{sb}^b(t), \quad T_{sb}(0) = 0 \\ \dot{v}_{sb}^b(t) = -\widehat{\omega}_{sb}^b(t)v_{sb}^b(t) + \alpha_{sb}^b(t) \\ y_{imu}(t) = \begin{bmatrix} \omega_{sb}^b(t) \\ \alpha_{sb}^b(t) - R_{sb}^T(t)\gamma \end{bmatrix} \\ y^i(t) = \pi\left(R_{sb}^T(t)(X_0^i - T_{sb}(t))\right). \end{cases} \tag{3}$$

These equations can be simplified by defining a new linear velocity, v_{sb}, which is neither the body velocity v^b_{sb} nor the spatial velocity v^s_{sb}, but instead $v_{sb} \doteq R_{sb}v^b_{sb}$, the body velocity relative to the spatial frame. Consequently, we have that $\dot{T}_{sb}(t) = v_{sb}(t)$ and $\dot{v}_{sb}(t) = \dot{R}_{sb}v^b_{sb} + R_{sb}\dot{v}^b_{sb} = \ddot{T}_{sb} \doteq \alpha_{sb}(t)$ where the last equation serves to define the new linear acceleration α_{sb}; as one can easily verify we have that $\alpha_{sb} = R_{sb}\alpha^b_{sb}$. The vision measurements remain unaltered, whereas the linear component of the inertial measurements becomes $R^T_{sb}(t)(\alpha_{sb}(t)-\gamma)$. If we model rotational acceleration $w(t) \doteq \dot{\omega}^b_{sb}$ and translational jerk $\xi(t) \doteq \dot{\alpha}_{sb}(t)$ as Brownian motions, our random walk model, with biases and noises, and with all subscripts removed, is

$$\begin{cases} \dot{X}^i_0 = 0, \quad i = 1, \dots, N \\ \dot{R}(t) = R(t)\widehat{\omega}(t), \quad R(0) = I \\ \dot{T}(t) = v(t), \qquad T(0) = 0 \\ \dot{\omega}(t) = w(t) \\ \dot{v}(t) = \alpha(t) \\ \dot{\alpha}(t) = \xi(t) \\ y_{imu}(t) = \begin{bmatrix} \omega(t) \\ R^T(t)(\alpha(t) - \gamma) \end{bmatrix} + \begin{bmatrix} \omega_{bias} \\ \alpha_{bias} \end{bmatrix} + n_{imu}(t) \\ y^i(t) = \pi\left(R^T(t)(X^i_0 - T(t))\right) + n^i(t). \end{cases} \tag{4}$$

where $v \doteq v_{sb} \doteq R_{sb}v^b_{sb}$, $\alpha \doteq \alpha_{sb} \doteq R_{sb}\alpha^b_{sb}$, $R \doteq R_{sb}, T \doteq T_{sb}$ and $\omega \doteq \Omega^b_{sb}$. In reality, the frames of the IMU and the camera do not coincide, and the IMU measurements are replaced with

$$y_{imu}(t) = R^T_{bi}\begin{bmatrix} \omega(t) \\ R^T(t)(\alpha(t) - \gamma + \ddot{R}(t)T_{bi}) \end{bmatrix} + \begin{bmatrix} \omega_{bias} \\ \alpha_{bias} \end{bmatrix} + n_{imu}(t) \tag{5}$$

where g_{bi} denotes the (constant) body-to-camera transformation, since we attach the body frame to the camera. Our choice of the camera frame as the body origin slightly complicates this model, but simplifies the following analysis. The results, of course, are identical whether derived in the camera or the IMU reference frame.

Both (3) and (4) are dynamical models with noise inputs which can be used to determine the likelihood of their outputs; estimating body pose $g_{sb}(t)$ and velocities v^b_{sb}, ω^b_{sb} is equivalent to inferring the state of such a model from measured outputs. This is a *filtering* problem (Jazwinski, 1970) when we impose *causal processing*, that is, the state at time t is estimated using only those measurements up to t, as is necessary in closed-loop applications. These requirements admit a variety of estimation techniques, including sum-of-Gaussian filters (Alspach and Sorenson, 1972), numerical integration, projection filters (Brigo et al., 1998), unscented filters (Julier and Uhlmann, 1997), particle filters and extended Kalman filters (Jazwinski, 1970). The condition which must be satisfied for *any* of these approaches to work is that the model be *observable*. We address this issue in the next section.

2.2 Analysis

In this section we study the observability of the state of the model above, as well as the identifiability of its unknown parameters, namely the camera-to-inertial calibration and gravity. The reader uninterested in the development of the analysis can read the claims within and skip the rest of this section.

2.2.1 Observability of 3-D motion and structure, identifiability of gravity and calibration frames

The observability of a model refers to the possibility of uniquely determining the state trajectory (in our case the body pose) from output trajectories (in our case, point feature tracks and inertial measurements). Observability is independent of the amount of (input or output) noise, and it is a necessary condition for *any*

filtering algorithm to converge (Jazwinski, 1970). When the model is not observable, the estimation error dynamics are unstable, and therefore eventually blow up.

It is well-known that pose is *not* observable from (monocular) vision-only measurements (Chiuso et al., 2002), because of an arbitrary gauge transformation (a scaling and a choice of Euclidean reference frame (McLauchlan, 1999)). The model can be made observable by fixing certain states, or adding pseudo-measurement equations (Chiuso et al., 2002). It is immediate to show that pose is also not observable from inertial-only measurements, since the model consists essentially of a double integrator. The art of inertial navigation, without the aid of visual or global positioning measurements, is to make them blow up as slowly as possible.

For the purpose of analysis, we start with a simplified version of (4) with no camera-IMU calibration (see Sect. 2.2.3), no biases[3] $\omega_{bias} = 0; \alpha_{bias} = 0$, no noises $\xi(t) = 0; w(t) = 0;\ n_{imu}(t) = 0; n^i(t) = 0$, since they have no effect on observability, and known gravity (see Sect. 2.2.4 otherwise).

The observability of a linear model can be determined easily with a rank test (Kailath, 1980). Analysis of non-linear models is considerably more complex (Isidori, 1989), but essentially hinges on whether the initial conditions of (4) are *uniquely* determined by the output trajectories $\{y_{imu}(t), y^i(t)\}_{t=1,\dots,T;i=1,\dots,N}$. If it is possible to determine the initial conditions, the model (4) can be integrated forward to yield the state trajectories. On the other hand, if two different sets of initial conditions can generate the same output trajectories, then there is no way to distinguish their corresponding state trajectories based on measurements of the output.

2.2.2 Indistinguishable trajectories

As a gentle introduction to our analysis we first show that, when only inertial measurements are available, the model (4) is not observable. To this end, consider an output trajectory $y_{imu}(t)$ generated from a particular acceleration $\alpha(t)$. We integrate the model to obtain $v(t) = \int_0^t \alpha(\tau)d\tau + \bar{v}$, and we can immediately see that any initial velocity $\bar{v}$ will give rise to the same exact output trajectory. Hence, from the output, we will never be able to determine the translational velocity, and therefore the position of the body frame, uniquely.

Claim 1 (Inertial only) *Given inertial measurements $\{y_{imu}(t)\}_{t=1,\dots,T}$ only, the model (4) is not observable. If $\{R(t), T(t), \omega(t), v(t), \alpha(t) \neq 0\}$ is a state trajectory, then for any $\bar{v}, \bar{T}, \bar{R}$ identical measurements are produced by*

$$\begin{cases} \tilde{R}(t) = \bar{R}R(t) \\ \tilde{T}(t) = \bar{R}T(t) + \bar{v}t + \bar{T} \\ \tilde{v}(t) = \bar{R}v(t) + \bar{v} \\ \tilde{\alpha}(t) = \bar{R}\alpha(t) \\ \tilde{\gamma} = \bar{R}\gamma. \end{cases} \tag{6}$$

If the gravity vector γ is known, then from $\tilde{\gamma} = \gamma$ we get that $\bar{R} = \exp(\widehat{\gamma})$, so the rotational ambiguity reduces to one degree of freedom. The claim can be easily verified by substitution to show that $\tilde{R}^T(t)(\tilde{\alpha}(t) - \tilde{\gamma}) = R^T(t)(\alpha(t) - \gamma)$, and assumes that $\|\tilde{\gamma}\| = \|\gamma\|$ is enforced. Note that if we impose $\tilde{R}(0) = R(0) = I$, then $\bar{R} = I$, and $\bar{T} = 0$, but we still have the ambiguity $\tilde{T}(t) = \exp(\widehat{\gamma})T(t) + \bar{v}t$, $\tilde{v}(t) = \exp(\widehat{\gamma})v(t) + \bar{v}$ and $\tilde{\alpha}(t) = \exp(\widehat{\gamma})\alpha(t)$. We will discuss the case $\alpha(t) = 0\ \forall\ t$ shortly. The volume of the unobservable set grows with time even if we enforce $(R(0), T(0)) = (I, 0)$, as $\|T(t) - \tilde{T}(t)\| = \|(I - \bar{R})T(t) - \bar{v}t - \bar{T}\| = \|\bar{v}t\| \to \infty$. Vision measurements alone are likewise insufficient to make the model observable.

Claim 2 (Vision only) *Given only vision measurements $\{y^i(t)\}_{i=1,\dots,N;t=1,\dots,T}$ of N points in general position (Chiuso et al., 2002), the model (4) is not observable. Given any state trajectory $\{X_0, R(t), T(t), \omega(t), v(t), \alpha(t)\}$,*

[3]Biases make the model trivially unobservable, so the results that follow are valid so long as biases are negligible or suitably compensated for.

for any rigid motion $(\bar{R}, \bar{T})$ *and positive scalar* $\lambda > 0$*, identical measurements are produced by*

$$\begin{cases} \tilde{X}_0^i = \lambda(\bar{R}X_0^i + \bar{T}) \\ \tilde{R}(t) = \bar{R}R(t) \\ \tilde{T}(t) = \lambda(\bar{R}T(t) + \bar{T}) \\ \tilde{v}(t) = \lambda\bar{R}v(t) \\ \tilde{\alpha}(t) = \lambda\bar{R}\alpha(t) \end{cases} \tag{7}$$

This can be verified by substitution. Note that $\dot{\tilde{X}}_0^i = 0$, so $\lambda, \bar{R}, \bar{T}$ are arbitrary *constants*. Even if we enforce $(R(0), T(0)) = (I, 0)$, the unobservable set can grow unbounded, for instance $\|\tilde{T}(t) - T(t)\| = \|(I - \lambda\bar{R})T(t) - \lambda\bar{T}\| = |1 - \lambda|\|T(t)\|$.

We now fix the global reference frame, or equivalently the initial conditions $(R(0), T(0))$, by constraining three directions determined by three points on the image plane, as described in (Chiuso et al., 2002). In the combined vision-inertial system, this is sufficient to simultaneously restrain the motion of the IMU (given that the camera and IMU move together as a rigid body). This leaves us with an ambiguity in the scale factor only; that is, $\tilde{R} = R$ and $\tilde{T} = \lambda T$ (therefore $\tilde{\omega} = \omega$ and $\tilde{\alpha} = \lambda\alpha$). We do not yet have constraints on gravity, nor the transformation between camera and IMU. We seek to determine what, if any, substitutions λ, $\tilde{g}_{bi}$, and $\tilde{\gamma}$ can be made for the true values $\lambda = 1$, g_{bi}, and γ while leaving the measurements (5) unchanged.

Let us define $\bar{R} \doteq \tilde{R}_{bi}R_{bi}^T$ and $\bar{T} \doteq \frac{1}{\lambda}(\tilde{T}_{bi} - \bar{R}T_{bi})$. This allows us to write $\tilde{R}_{bi} = \bar{R}R_{bi}$ and $\tilde{T}_{bi} = \bar{R}T_{bi} + \lambda\bar{T}$ without loss of generality. The constraint $\tilde{\omega} = \omega$ and the IMU's measurement of angular velocity tell us that $R_{bi}^T\omega(t) = \tilde{R}_{bi}^T\tilde{\omega}(t) = R_{bi}^T\bar{R}^T\omega(t)$, so $\omega(t) = \bar{R}^T\omega(t)$. Hence $\bar{R}$ is forced to be a rotation around the ω axis; it is easy to verify that this implies

$$\bar{R}\hat{\omega} = \hat{\omega}\bar{R}. \tag{8}$$

The accelerometer measurements require that

$$R_{bi}^T R^T(t)\left(\alpha(t) - \gamma + \ddot{R}(t)T_{bi}\right) = \tilde{R}_{bi}^T R^T(t)\left(\tilde{\alpha}(t) - \tilde{\gamma} + \ddot{R}(t)\tilde{T}_{bi}\right). \tag{9}$$

This is satisfied only by assigning

$$\tilde{\gamma} = R\bar{R}R^T(t)\gamma + \left(\lambda I - R(t)\bar{R}R^T(t)\right)\alpha(t) + \ddot{R}(t)\lambda\bar{T}. \tag{10}$$

Note that $R^T(t)\ddot{R}(t) = \hat{\dot{\omega}}(t) + \hat{\omega}^2(t)$, so (8) allows us to write $\bar{R}R^T(t)\ddot{R}(t) = R^T(t)\ddot{R}(t)\bar{R}$. This identity may be used to verify (10) by substitution into (9). We can now fully describe the ambiguities of the system.

Claim 3 (Observability of Combined Inertial-Vision System) *Provided the global reference frame is fixed as in (Chiuso et al., 2002), two state trajectories for the system (4-5) are indistinguishable if and only if, for constants* $\lambda \in \mathbb{R}$ *and* $(\bar{R}, \bar{T}) \in SE(3)$*,*

$$\begin{cases} \tilde{X}_0^i = \lambda X_0^i \\ \tilde{R}(t) = R(t) \\ \tilde{T}(t) = \lambda T(t) \\ \tilde{R}_{bi} = \bar{R}R_{bi} \\ \tilde{T}_{bi} = \bar{R}T_{bi} + \lambda\bar{T} \\ \tilde{\omega}(t) = \omega(t) = \bar{R}\omega(t) \\ \tilde{\gamma} = R\bar{R}R^T(t)\gamma + (\lambda I - R(t)\bar{R}R^T(t))\alpha(t) + \ddot{R}(t)\lambda\bar{T}, \end{cases} \tag{11}$$

We now examine a few scenarios of interest. First, in a simple case when gravity and calibration are known, the ambiguity reduces to $0 = (\lambda - 1)\alpha(t)$, which tells us that scale is determined so long as acceleration is non-zero.

Claim 4 (Inertial & Vision) *The model (4) is locally observable provided that $\alpha(t) \neq 0$ and that the initial conditions $(R(0), T(0)) = (I, 0)$ are enforced.*

We emphasize that unless the global reference is fixed by saturating the filter along three visible directions, following the analysis in (Chiuso et al., 2002), the choice of initial pose is not sufficient to make the model observable since it is not actively enforced by the filter.

The term "locally observable" refers to the fact that infinitesimal measurements are sufficient to disambiguate initial conditions; local observability is a stronger condition than global observability, and can be verified by computing the rank of the observability co-distribution (Isidori, 1989).

2.2.3 Observability with unknown calibration

Measuring the transformation between the IMU and the camera precisely requires elaborate calibration procedures, and maintaining it during motion requires tight tolerances in mounting. To the best of our knowledge there is no study that characterizes the effects of camera-IMU calibration errors on motion estimates. Consider the simplified case of known gravity, and correct rotational calibration, but a small translational miscalibration (for example, due to expansion or contraction of metals with temperature). Our constraint becomes $(1-\lambda)\alpha(t) = \ddot{R}(t)\lambda\bar{T}$, where $\bar{T}$ is the miscalibration. For general motion, this is clearly not satisfiable, and can cause divergence of the filter. In this section we show that such errors can be made to have a negligible effect; indeed, we advocate forgoing such a calibration procedure altogether. Instead, a filter should be designed to automatically calibrate the camera and IMU.

First consider the ambiguity in rotational calibration, $\bar{R}$. Since $\bar{R}\omega(t) = \omega(t)$, $\bar{R}$ must be the identity when $\omega(t)$ is fully general.[4] This reduces the second constraint to $(1-\lambda)\alpha(t) = \ddot{R}\lambda T$. If $\alpha(t)$ is non-zero and not a function of $\ddot{R}$, then $\lambda = 1$ and the model is observable. These are "general position conditions", in the sense that they are satisfied except for a set of measure zero in the space of all possible motions. Unfortunately, this "thin set" includes constant-velocity motion, that is quite common in practical applications. Therefore, this case will have to be dealt with separately.

The general position conditions relate to the concept of "sufficient excitation" in parameter identification, whereby the driving inputs to the system (noise or, in this case, the externally-driven platform motion) are sufficiently complex as to excite all the modes of the system. In our case, based on the analysis above, "sufficiently exciting" refers to motion sequences that have non-zero linear acceleration, $\alpha(t) \neq 0$, and non-constant rotational axis $\omega(t) \times \omega(\tau) \neq 0$ for $t \neq \tau$.

Claim 5 (Identifiability of camera-to-inertial calibration) *The model (4), augmented with (5) and T_{bi}, R_{bi} added to the state with constant dynamics, is locally observable, so long as motion is sufficiently exciting and the global reference frame is fixed.*

Note that we refer to observability of the augmented model as being equivalent to the identifiability of the model parameters. This follows customary practices of adding the unknown parameters to the state of the model with a trivial dynamics (constant, or slowly-varying, or random walk), thereby transforming the filtering/identification problem into a pure filtering problem.

2.2.4 Dealing with gravity

We now turn our attention to handling the unknown gravity vector. Because γ has a rather large magnitude, even small estimation errors in R_{sb} will cause a large innovation residual $n_{imu}(t)$. Dealing with gravity is an art of the inertial navigation community, with many tricks of the trade developed over the course of decades of large scale applications. We will not review them here; the interested reader can consult (Kayton and Fried, 1996). Rather, we focus on the features of vision-inertial integration. Most techniques already in use in inertial navigation, from error statistics to integration with positioning systems, can be incorporated if one so desires.

[4] Special cases include not only $\omega(t) = 0$, but also $\omega(t)$ spanning less than two independent directions.

Our approach is to simply add the gravity vector to the state of the model (4) with trivial dynamics $\dot{\gamma} = 0$ and small model error covariance. Note that this is *not* equivalent to the slow-averaging customarily performed in navigation filters – the disambiguation of the gravity vector comes from the coupling with vision measurements. Assuming known calibration, we have that $\tilde{\gamma} = \gamma + (\lambda - 1)\alpha(t)$. Since γ and $\tilde{\gamma}$ are constants, λ must be unity as long as $\alpha(t)$ is non-constant.

Claim 6 (Identifiability of gravity) *The gravity vector, if added to the state of (4) with trivial dynamics $\dot{\gamma} = 0$, is locally observable provided that $\alpha(t)$ is not constant and the global reference frame is fixed.*

The claims just made may be combined if gravity and calibration are unknown.

Claim 7 (Observability of calibration and gravity) *The model (4-5) and T_{bi}, R_{bi}, γ added to the state with constant dynamics, is locally observable, so long as motion is sufficiently exciting and the global reference frame is fixed.*

As we have mentioned, "cruising" and other common motions are *not* sufficiently exciting, in the sense that the constraints (11) are non-trivially satisfied. A full derivation of the constraints and a more complete analysis of degenerate cases are available in Appendix A of the technical report (Jones and Soatto, 2009).

3 Implementation

The model (4) used for analysis is simplified by removing noise, biases, calibration, and by writing the dynamics in continuous time. In practice, estimates are only required at discrete time instants. Therefore, for the purpose of implementation, we use a discrete-time version of the model (4), modified in a number of ways. First, because the uncertainty in the coordinates X_0^i is uneven, we represent each point with its projection $y_0^i = \pi(X_0^i)$ and its log-depth ρ^i, so that $X_0^i = \bar{y}_0^i e^{\rho^i}$, where the bar $\bar{y}$ denotes the homogeneous coordinates of y (we will forgo the bar from now on since it is clear from the context whether the point y is expressed in Euclidean or homogeneous coordinates). We then represent linear jerk (the derivative of acceleration) and rotational acceleration, as well as the unknown gravity and the translational and rotational components of the calibration parameters, as Brownian motions driven by white zero-mean Gaussian processes whose covariance is a design parameter that is set during a tuning procedure as customary in non-linear filtering (Jazwinski, 1970). The resulting model is

$$\begin{cases} y_0^i(t+1) = y_0^i(t) + n_0^i(t) \quad i = 4, \dots, N(t) \\ \rho^i(t) = \rho^i(t) + n_\rho^i(t) \quad i = 1, \dots, N(t) \\ T(t+1) = T(t) + v(t), \quad T(0) = 0 \\ \Omega(t+1) = Log_{SO(3)}(\exp(\widehat{\Omega}(t)) \exp(\widehat{\omega}(t)), \quad R(0) = I \\ v(t+1) = v(t) + \alpha(t) \\ \omega(t+1) = \omega(t) + w(t) \\ \alpha(t+1) = \alpha(t) + \xi(t) \\ \xi(t+1) = \xi(t) + n_\xi(t) \\ w(t+1) = w(t) + n_w(t) \\ \gamma(t+1) = \gamma(t) + n_\gamma(t) \\ T_{cb}(t+1) = T_{cb}(t) + n_{T_{cb}}(t) \\ \Omega_{cb}(t+1) = \Omega_{cb}(t) + n_{\Omega_{cb}}(t) \\ y^i(t) = \pi\left(e^{\widehat{\Omega}_{cb}(t)} e^{-\widehat{\Omega}(t)} (e^{-\widehat{\Omega}_{cb}(t)} (y_0^i(t) e^{\rho^i(t)} - T_{cb}(t)) - T(t)) + T_{cb}(t)\right) + n^i(t) \\ y_{imu}(t) = \begin{bmatrix} \omega(t) + \omega_{bias} \\ e^{-\widehat{\Omega}(t)}(\alpha(t) - \gamma(t)) + \alpha_{bias} \end{bmatrix} + n_{imu}(t) \\ norm_\gamma = \|\gamma(t)\|. \end{cases} \tag{12}$$

When the motion is sufficiently exciting, for instance during an auto-calibration procedure where the platform is moved freely in space, gravity is observable, so the covariance of the error of the last (pseudo-measurement) equation is set to a very large number, while all other covariances are tuned. Once the calibration parameters have converged, during normal operation, to avoid drift while the platform traverses regions of motion-space that are not sufficiently exciting, we "lock" the calibration states by reducing the covariance of $n_{T_{cb}}$ and $n_{\Omega_{cb}}$ to zero, and so for the pseudo-measurement equation that fixes the magnitude of the gravity vector. We also note that gravity becomes trivially unobservable when the biases are unknown. Therefore, we lock the biases (by saturating the corresponding states in the filter) during the short calibration sequence, and then for long-term operation we lock the norm of gravity and let the biases float.

The vision measurement equation arises from the fact that $g(t) = (R(t), T(t)) = g_{sb}(t)$ is the transformation from the body frame to the spatial frame, which is the body frame at time $t = 0$, so the transformation mapping the initial measurement y_0^i to the current time is $g^{-1}(t)$, after it has been transformed to the camera frame, so we have $X_0^i = y_0^i e^{\rho^i}$, the point in space relative to the camera reference frame at time $t = 0$, and $X_t^i = g_{cb} g^{-1}(t) g_{cb}^{-1} X_0^i$, the point in space relative to the camera reference frame at the current time t. This is the point that is projected to give rise to the measurement $y^i(t)$. The IMU measurement equation has been derived in eq. (4). Note that this is different than the subsequent equation (5), where we have attached the body frame to the camera, which is convenient for analysis. In section we attach the body frame to the IMU that results in a simpler overall implementation. Thus g_{bi} has been supplanted by g_{cb} and the vision measurements are transformed rather than the IMU measurements. Notice that the number of visible features $N(t)$ can change with time, and the index i in the first equation (describing point feature positions in the camera at time t) starts from 4. Fixing the first three points[5] is equivalent to fixing three directions in space, which fixes the global reference as described in (Chiuso et al., 2002). Depths are represented using the exponential map to ensure that their estimates are positive. We remind the reader that $\omega = \omega_{sb}^b$, whereas $v = v_{sb}$ and $\alpha = \alpha_{sb}$ are defined as (4).

To overcome failures of the low-level feature tracking module, we can employ a robust version of an extended Kalman filter (EKF), similar to (Vedaldi et al., 2005), which allows a variable number of points.

In ideal conditions, the model (12) suffices to specify the design of a filter, whether an extended Kalman filter (EKF) or one of many variants of point-estimators such as the unscented Kalman filter (UKF), or various sum-of-Gaussian filters etc. In practice, however, there are more complex phenomena at play that we address in order.

3.1 Discretization

Measurements are provided by the sensors at different time intervals. Our IMU, for instance, produces readings at 100Hz, whereas the camera has a refresh-rate of 30Hz. This is straightforward to handle by introducing an elementary time step, dt, and multiplying the process noises by dt at each prediction step. The state can then be updated asynchronously as different measurements become available.

Unfortunately, measurements are not available as soon as the sensors produce them. Because of hardware constraints and communication interfaces, there is a delay between the time-stamp when a datum is produced, and the instant when the datum is available for processing by the filter. This delay is non-deterministic, and is different for different sensors and different interfaces. We have tested our system with fiber-optic and firewire camera interfaces, and serial and USB inertial measurement interfaces. For each sensor combination, we have performed an experiment whereby the platform was mounted on a balanced boom and moved in front of a checkerboard pattern. At each bounce we measured the peak acceleration and the inversion of image feature trajectories. The delay between measurements from the optical channel and the inertial channel becoming available has means ranging from 50ms to 80ms, with standard deviations in the order of 20ms. While taking into account the fluctuation of this delay requires dedicated hardware, with individual time-stamps, we forgo this step, and only compensate for the average delay. As described in Appendix B of (Jones and Soatto, 2009), in our software infrastructure measurements are written in a mapbuffer, together

[5] We choose the first three points for simplicity; in practice one may want to choose three points that are sufficiently far apart to avert the degeneracy of the fixed frame.

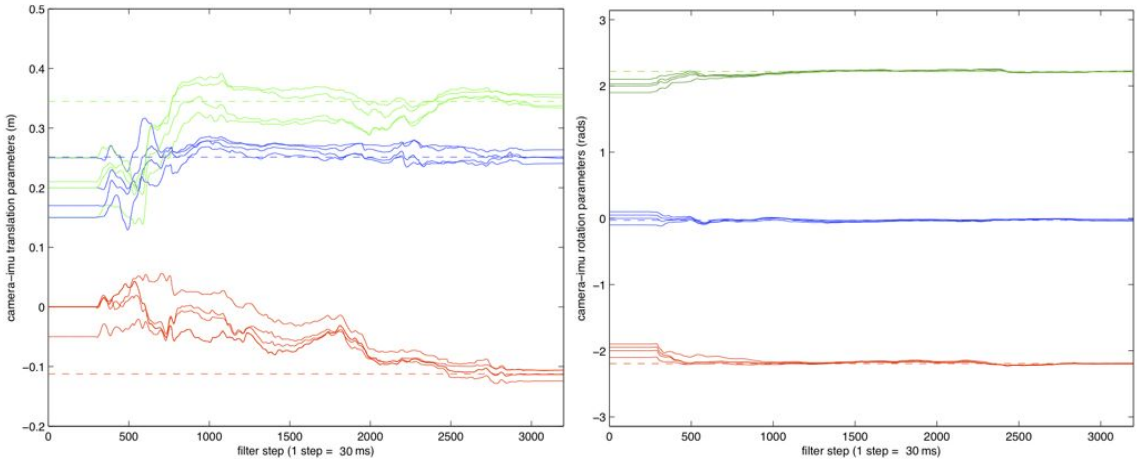

Figure 1: *Autocalibration experiments: translational (left) and rotational (right) calibration parameters, shown for multiple runs of a repeated experiment (5 trials, to facilitate visualization). Translation parameters are in meters, rotation parameters are the exponential coordinates of the rotation matrix in radians. Along the abscissa, each step is* $30ms$*. The platform was at rest for a few seconds, then it was moved with a sufficiently exciting sequence lasting several seconds. "Pseudo-ground truth" (dashed line), has been determined in a separate calibration experiment as the average of the final estimate of multiple trials with a larger number of features and a known gravity vector. The small variability of the final estimates of translation is due to slight errors in gravity.*

with their time-stamp and the average delay, where they are sorted before processing. However, because of the fluctuation of the delay, it is possible for data to be processed in the wrong order, and indeed this happens in our experiments, and can occasionally create ripples or small loops in the estimated path if an uncompensated delay causes a negative time-step. In order to completely avoid these artifacts, the differing latencies must be estimated using dedicated hardware, and compensated by re-ordering them before they are presented to the filter.

3.2 Autocalibration mode and Normal Operation

As we have pointed out in the previous section, the calibration between the camera and the IMU is observable only under a sufficiently exciting motion, which may not be achieved during scenarios of interest. For example, a ground vehicle typically rotates around only a single axis, so translational calibration along that axis is not identifiable.

Given that calibration does not change significantly over time, we find it useful to determine the calibration parameters using a sufficiently complex sequence, which we call an "auto-calibration sequence," processed by the full filter. The calibration parameters can then be fixed, along with gravity, for ordinary operation during which the biases are allowed to change. In contrast with (Mirzaei and Roumeliotis, 2007) the structure of the environment and the motion do not need to be known, and the calibration can be initialized with rather coarse estimates. Once calibration is determined in this manner, it is simply removed from the state and fixed (or the covariance of the model error of the corresponding states is assigned to a very small number, typically 10^{-12}).

While it would be highly desirable to have automatic procedures to switch from normal model to autocalibration mode when a sufficiently exciting motion is occurring, this test would require an external processing pathway, since a sufficiently exciting sequence cannot be detected from statistics computed on the innovation process. In practice, however, we find that a manual switch from autocalibration to normal mode is adequate for most applications.

In figure 1 we show typical results of autocalibration sequences. The calibration parameters are coarsely initialized using a ruler, and gravity is initialized to $\gamma(0) = [0,\ 0,\ -9.8]^T$. The figure shows the estimates of the translational calibration parameters (position of the camera relative to the IMU in meters) and rotational calibration parameters (exponential coordinates of the rotation of the camera relative to the IMU in radians), for repeated trials. The platform is turned on and picked up after a few seconds, moved around with a sufficiently exciting motion, and then placed on the ground and the filter is run for a few more seconds. The entire experiment lasts about a minute and a half. Since mechanically measured ground-truth is actually less repeatable than the results of our autocalibration sequences, we have used as "pseudo-ground truth" the average of multiple trials obtained by processing the data off-line with a larger number of features (up to 500, rather than up to 100). To test convergence to actual ground truth we have performed extensive simulation experiments that confirm the results reported in (Jones et al., 2007).

3.3 Handling of Missing and New Data

The general form of the filter includes the full three dimensional position of each tracked feature, or three states for each, represented by y_0^i and ρ^i. We adopt the model of (Favaro, 1998), and assume that the initial detection of the feature *defines* the location being tracked. It is therefore, by definition, noise free, which eliminates the need for the two degrees of freedom y_0^i in the image plane, leaving only one state, ρ^i, per feature in the filter as in (Azarbayejani and Pentland, 1995). However, because of violation of the planar-Lambertian assumption implicit in most feature tracking schemes, what is being tracked changes over time, and the drift consists of two additional degrees of freedom per point. Adding the unknown drift to the state takes us back to representing each point with three degrees of freedom as in (Chiuso et al., 2002).

However, because the average lifetime of a point feature, intended as the number of frames in which the feature is visible and successfully tracked, typically ranges between 10 and 30 frames, the tracking drift is often small, so we implement a tradeoff suggested by (Favaro, 1998), whereby features are selected in *groups* of, say, $N(t)$ points, and instead of associating $N(t)$ states to the group, thus exposing the filter to the effects of tracking bias, or $3N(t)$, thus increasing the computational complexity of the scheme, we represent each group with $N(t) + 6$ states, the first $N(t)$ representing the log-depth of each point, and the remaining 6 states representing the position and orientation of the group relative to the best estimate of ego-motion at time t. This approach allows some slack to reduce bias by adjusting the group as a rigid ensemble, rather than allowing each point to drift independently of the others in the group. While the implementation of this approach is significantly more laborious than either the filter with $N(t)$ or $3N(t)$ states, it provides a desirable tradeoff between drift and computational complexity.

When adding features to the filter, the initial value of their depth has significant impact on performance. Vision-only (monocular) systems use various approaches to initialization, such as spawning a sub-filter in which an arbitrarily-initialized feature's depth is estimated (Chiuso et al., 2002). The advantage of our system over (monocular) vision-only approaches, however, is that we expect to have a reasonable local estimate for motion, even in the absence of useful features. Therefore, we can bootstrap our feature depth estimates using the motion estimate of the IMU. By triangulating each new feature over time before it is added to the state, we obtain good estimates for the features and eliminate the bias of a fixed initialization.

In addition to features that appear and disappear due to occlusions, violation of the model etc., we also have mismatches due to repeated structures, "sliding" of features on surfaces due to specularities and other violations of the Lambertian assumption, or features corresponding to T-junctions that are not physically attached to a surface in space. In this case, the feature tracker successfully accomplished its job of determining the translational or affine *image motion* of a salient point, but its motion is not compatible with the model (12). Therefore, in addition to estimating the state of the filter, we have to test the hypothesis that any given datum is compatible with the model. This falls in the realm of robust statistics and a variety of methods are available, mostly based on heuristics that have been successfully employed in practice. In our real-time implementation we use a robust filter, as in (Chiuso et al., 2002), that is significantly simpler than acceptance or rejection sampling, such as (Vedaldi et al., 2005), and proves sufficient for our purpose. Feature detection and tracking is performed using a standard multi-scale implementation of (Lucas and Kanade, 1981). Intrinsic calibration parameters, including radial distortion, are computed using standard

tools (Tsai, 1987).

In order to make this more precise, we call $y^i_{\tau_j}$ the coordinates on the image plane of the position of feature i that is part of group j that appeared at time instant τ_j, and ρ^i the log-depth of the same point relative to the time instant where it first appeared. Then we have that $g(\tau_j) = g_{sb}(\tau_j)$ is the transformation mapping the body frame (the moving reference frame attached to the IMU) to the spatial frame (the IMU reference frame at time $t = 0$). We call this transformation $g^j_{ref} = (R^j_{ref}, T^j_{ref}) = (e^{\widehat{\Omega}^j_{ref}}, T^j_{ref}) = g(\tau_j)$, and include its local coordinate representation $T^j_{ref}, \Omega^j_{ref}$ in the state of the filter, together with the log-depths ρ^i. As we have anticipated, we do *not* include in the state of the filter the image coordinates $y^i_{\tau_j}$, that are used in the vision measurement equation, that now becomes somewhat more complicated in that we have to take the point in the reference frame at time τ_j when it first appears, $X^i_{\tau_j} = y^i_{\tau_j} e^{\rho^i}$, then transform it to the body reference frame $g^{-1}_{cb} X^i_{\tau_j}$, then bring it to the body reference frame at the initial time, $g^j_{ref} g^{-1}_{cb} X^i_{\tau_j}$, then move it to the body frame at time t, $g(t)^{-1} g^j_{ref} g^{-1}_{cb} X^i_{\tau_j}$, finally bringing it back to the camera frame to obtain $X^i_t = g_{cb} g(t)^{-1} g^j_{ref} g^{-1}_{cb} X^i_{\tau_j}$, which is the point that is projected onto the current image plane to obtain the vision measurement $y^i(t)$.

In the initialization phase, the reference for the initial group is $T(0) = 0$ and $\Omega(0) = 0$. As soon as all features in the initial group are lost, it is necessary to choose another group as a reference, and to fix its reference frame, via $T^j_{ref} = T(\tau_j)$ and $\Omega^j_{ref} = \Omega(\tau_j)$. Failure to do so makes the global reference frame unobservable, and the filter can drift.

To ease the notation, we define the total transformation between the camera reference frame when the point first appears and the current camera reference frame to be

$$g_{tot} = g_{cb} g(t)^{-1} g^j_{ref} g^{-1}_{cb} \tag{13}$$

that in coordinates can be written as

$$R_{tot}(t) = R_{cb} R^T(t) R^j_{ref} R^T_{cb} = e^{\widehat{\Omega}_{cb}} e^{-\widehat{\Omega}(t)} e^{\widehat{\Omega}^j_{ref}} e^{-\widehat{\Omega}_{cb}} \tag{14}$$

$$T_{tot}(t) = -R_{tot}(t) T_{cb} + R_{cb} R^T(t) \left(T^j_{ref} - T(t) \right) + T_{cb}. \tag{15}$$

In addition, the biases α_{bias} and ω_{bias} have to be estimated on-line and therefore are inserted into the state of the filter. Recall that $y^i_{\tau_j} \doteq y^i(\tau_j) \quad i = 1, \ldots, N_j(t)$ denotes the points selected at time t_j as part of the

group j:

$$\begin{cases} \rho^i(t+dt) = \rho^i(t) + n^i_\rho(t)dt \quad \text{initialized by triangulation from IMU inter} - \text{frame motion estimates} \\ T(t+dt) = T(t) + v(t)dt, \quad T(0) = 0 \\ \Omega(t+dt) = Log_{SO(3)}(\exp(\widehat{\Omega}(t)) \exp(\widehat{\omega}(t)dt), \quad \Omega(0) = 0 \quad v(t+dt) = v(t) + \alpha(t)dt \\ \omega(t+dt) = \omega(t) + w(t)dt \\ \alpha(t+dt) = \alpha(t) + \xi(t)dt \\ \xi(t+dt) = \xi(t) + n_\xi(t)dt \\ w(t+dt) = w(t) + n_w(t)dt \\ \gamma(t+dt) = \gamma(t) + n_\gamma(t); \quad \gamma(0) = \gamma_0 \quad \text{from calibration} \\ T_{cb}(t+dt) = T_{cb}(t) + n_{T_{cb}}(t)dt, \quad T_{cb}(0) = T_{cb} \quad \text{from calibration} \\ \Omega_{cb}(t+dt) = \Omega_{cb}(t) + n_{\Omega_{cb}}(t)dt, \quad \Omega_{cb}(0) = \Omega_{cb} \quad \text{from calibration} \\ T^j_{ref}(t+dt) = T^j_{ref}(t) + n_{T^j_{ref}}(t)dt, \quad T^j_{ref}(\tau_j) = T(\tau_j) \quad j = 1, \dots, M(t) \\ \Omega^j_{ref}(t+dt) = \Omega^j_{ref}(t) + n_{\Omega^j_{ref}}(t)dt, \quad \Omega^j_{ref}(\tau_j) = \Omega(\tau_j) \\ \omega_{bias}(t+dt) = \omega_{bias}(t) + n_{\omega_{bias}}(t)dt \\ \alpha_{bias}(t+dt) = \alpha_{bias}(t) + n_{\alpha_{bias}}(t)dt \\ y^i(t) = \pi \left(R^j_{tot}(t) y^i_{\tau_j} e^{\rho^i(t)} + T^j_{tot}(t) \right) + n^i(t) \\ y_{imu}(t) = \begin{bmatrix} \omega(t) + \omega_{bias} \\ e^{-\widehat{\Omega}(t)}(\alpha(t) - \gamma(t)) + \alpha_{bias} \end{bmatrix} + n_{imu}(t) \\ \frac{9.8^2}{2} = \frac{1}{2}\|\gamma\|^2 \end{cases} \tag{16}$$

This is the model we use in our implementation that is described in detail in Appendix B of (Jones and Soatto, 2009). In the next section we describe the performance of the ensuing filter.

4 Open-loop Navigation Experiments

Evaluating a vision-based navigation system is a challenge because performance is affected by a large number of factors that are beyond the control of the user. These include the nature and amount of motion and whether it is "sufficiently exciting" or whether it is close to singular configurations; the three-dimensional layout of the scene, whether it is close to singular configurations such as planarity, or whether salient feature points are far relative to the inter-frame motion (parallax); visibility effects, affecting the average lifetime of the features, also in relation to the effective field of view of the sensor; reflectance properties of the scene, such as the presence of specular or translucent surfaces, or the amount and temporal stability of the light source (e.g. fast-moving clouds); the nature of the camera, whether it has auto-gain control, its resolution, field of view, optical aberrations; the complexity of the scene, including its occlusion structure (e.g. a forest) and the presence of multiple moving objects (e.g. a crowded indoor environment), just to mention a few.

It is therefore impossible to provide precise guarantees on the performance of the filter under anything but trivial laboratory scenarios. Our approach to this problem is to test our algorithm on challenging sequences, so as to elicit as many failure modes as possible. Below we present results on indoor sequences up to several hundred meters, and outdoor sequences up to several kilometers. It is important to stress that the system, including the tuning of the filter, is entirely identical in each case, so there is no ad-hoc tuning to the particular kind of motion (hand-held, on a wheeled base, or on a passenger vehicle), the nature of illumination, re-calibration etc.

Our optimized implementation of the filter recovers ego-motion from inertial measurements and video on a modern CPU faster than the sensor data arrives (30Hz for images with 768×640 resolution and 100Hz for inertial). The sensor platform is shown in Fig. 2 and includes various cameras (omni-directional, binocular and trinocular), of which only a monocular one is used, a BEI Systems IMU, a LIDAR, and GPS. Processing

Figure 2: ***Embedded platform.*** A view of our embedded platform. Only a monocular camera and an IMU are used and the filter is implemented on a custom battery-operated computer. Other sensors, including stereo cameras, GPS, and LIDAR, are used only for verification.

is done on a custom computer, and power is drawn from battery packs. The platform can be carried with shoulder straps, or mounted on a vehicle or wheeled base.

To test our method indoors, we captured a 10 minute long sequence while carrying the sensor platform through the hallways of a rectangular building. The rectangle was traversed twice, and the total length of the path was 570 meters. A scaled blueprint of the building is available, and our results are shown overlaid on the blueprint in figure 3. The total horizontal and vertical drifts were 1.07 meters and 0.15 meters respectively, less than 0.2% of distance, or 6 meters per hour. We compare our result to the state of the art in table 4. While the results shown are not all on the same data, we have selected a challenging sequence in which all features are lost five times. The results we show for monocular vision and inertial only are on the same sequence, but are slanted in favor of those methods. In the case of vision, scale has been manually adjusted and the segments between losses of all features have been manually stitched together. The drift figure for inertial only is quoted for our IMU over a short time scale (one minute); over longer time periods, the effects of drift compound and grow more rapidly.

Our outdoor reconstruction results are similarly favorable. For the first sequence, shown in figure 4, we drove a 7.9Km path in 16.5 minutes, revisiting a position $5,808$m after the first visit. This sequence is particularly challenging: it was captured during rush hour in a dense urban area with many moving pedestrians and vehicles; there are also optical artifacts due to the low evening sun. Our result is shown in figure 4. The total horizontal and vertical drifts were 15.2 meters and 4.9 meters respectively, or 0.27% of the distance driven. This result is slightly better than that reported in (Mourikis and Roumeliotis, 2007) (0.31%) for a shorter driving sequence, though our sequence includes more moving obstacles, and we do not perform a bundle adjustment step, or any other post-mortem refinement.

We also drove a much longer loop: 30.3Km in 62 minutes, under similarly challenging traffic and lighting conditions. At this scale of time and distance, the rotation and curvature of the earth result in non-negligible errors. These could be substantially reduced by adding terms to the model accounting for the earth's rotation, as is done in (Mourikis and Roumeliotis, 2007) and most inertial-based navigation algorithms. However, this would come at the cost of the local nature of our filter and map-building approach: global latitude and true heading must be known in order to properly compensate for the Earth's rotation. While this is quite reasonable in many circumstances and easy to introduce to our model, we present here the results of the more general model that does not compensate for earth curvature and rotation. Our horizontal and vertical drift for this sequence are 146 meters and 5 meters respectively, or 0.5% of the distance traveled. This result is obtained on significantly greater distance than previous vision or vision-inertial systems have reported.

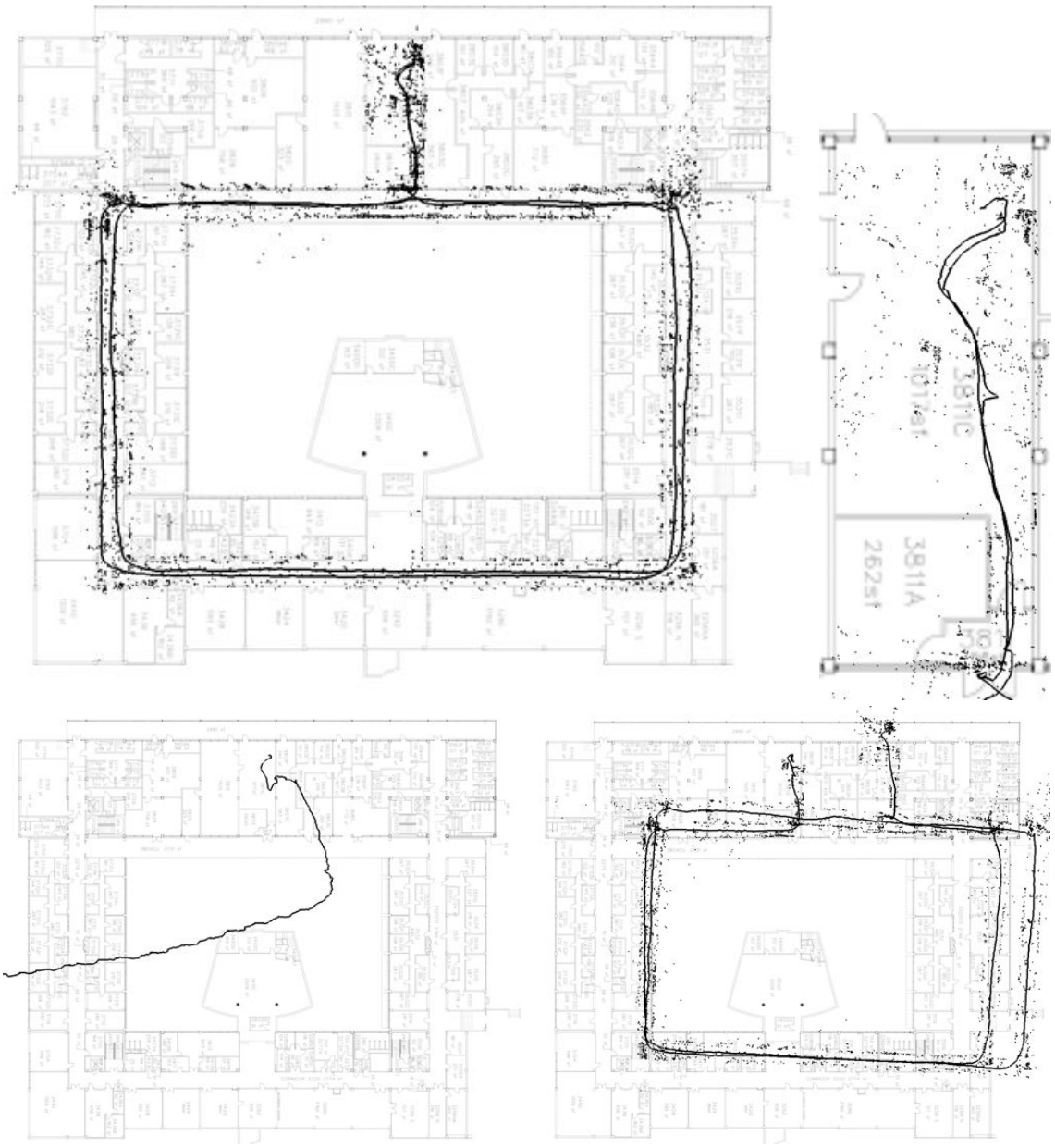

Figure 3: ***Comparison of indoor reconstruction.*** *Three approaches to reconstructing a sequence captured while walking through the corridors of a rectangular building. Top left: Blueprint of the building with our result overlaid. Error is less than 0.2%. Top right: Detail of beginning and end of sequence. Bottom left: Inertial only. Bottom right: Vision only (with manual scale correction and stitching of segments between locations where all features are lost).*

Method	Drift: distance (%)	Drift: time (meters/min)
IMU only	N/A	43
Monocular vision only	3.5	N/A
Stereo vision only (Nister et al., 2006)	1.1	N/A
Stereo and IMU* (Konolige and Agrawal, 2008)	0.3	N/A
(Konolige and Agrawal, 2008) with global bundle adj.*	0.1	N/A
(Mei et al., 2009) with stereo	1.1	N/A
(Mourikis and Roumeliotis, 2006)	0.5	N/A
Ours	0.2	0.1

Table 1: Comparison of drift statistics for indoor reconstruction techniques. Comparison with (Konolige and Agrawal, 2008), has been performed by taking their best result of 8.5m/5km RMS error with IMU and no bundle adjustment, and 3m/5km RMS error with global bundle adjustment (in addition to stereo and IMU) and projecting it to an end-point error after 5km assuming linear drift (figure 11 of (Konolige and Agrawal, 2008)), yielding an end-point error $\sqrt{3}RMS$, or 14.7m/5km and 5.2m/5km, corresponding to 0.3% and 0.1% imputed drift. It should be noted, however, that methods using stereo have the advantage of knowledge of scale, and the disadvantage of requiring accurate calibration in order to compute disparity. (Mei et al., 2009) use stereo and estimate $15 - 25$m over 2.26km based on repeated runs of the experiment but without ground truth.

Figure 4: ***Sample frames from first outdoor sequence.***

Figure 5: ***Outdoor reconstruction.*** *Top: Our reconstruction of a 7.9 km long driving sequence, overlaid on an aerial view. Error is less than 0.27%. Bottom: Detail of area in which the vehicle returned to a previously visited point. The initial point was at the center of the image, and two loops around the UCLA campus are shown, with the second loop terminating to the south-east on the bottom-right corner of the image.*

Figure 6: ***Sample frames from long outdoor sequence.***

We have also tested our algorithm on the data provided to us by (Mourikis and Roumeliotis, 2007). The result is shown in figure 4 superimposed to an aerial photograph. The overall drift is similar to the previous experiment, below 0.3% of distance traveled.

We also present an illustrative example in figure 4 comparing our results to the output of a high-quality inertial-aided GPS system. The experiment starts on top of a parking structure with a clear view of the sky. However, the path immediately descends two stories, blocking GPS (as well as providing very dark images with relatively few features). Upon exiting the garage at ground level, both approaches have accumulated drift, but the GPS drift and resulting state jump is clearly much worse.

In the next section we address the problem of recognizing a previously visited location, and building a coherent map from independent segments reconstructed by possibly different users.

5 Map Building and Localization

The filter described in the previous section produces an on-line estimate of the trajectory of the platform relative to the position and orientation when it was first switched on, together with the spatial position of a number of point features. When feature tracks are lost, they are removed from the state and stored in memory together with their position in the global reference frame[6] and with a descriptor of the local photometry around the tracked feature. This representation constitutes a *map* that can be used to localize the platform using standard tools of multiple-view geometry (Ma et al., 2003). Because hundreds of features are tracked, and lost, at each time instant, the complexity of the map grows rapidly, and therefore visual features must be organized efficiently in order to enable rapid localization. Many have addressed this issue in the simultaneous localization and mapping (SLAM) community, for instance (Bosse et al., 2004, Eade and Drummond, 2007a, Guivant and Nebot, 2001, Klein and Murray, 2007, Konolige and Agrawal, 2008, Mouragnon et al., 2006, Nebot and Durrant-Whyte, 1999, Kelly and Sukhatme, 2009, Chum et al., 2009) just to mention a few. In section 5.2 we describe our own topological representation that is based on the notion of "locations" defined by co-visibility.

While a variety of algorithms have been proposed to search the map, at the lowest level they all entail the comparison of local feature descriptors. Because the same portion of the scene can be seen from different vantage points and under different illuminations, such *nuisance variables* must be either marginalized during the matching process or canonized in the representation. Marginalization entails searching or averaging with respect to all possible nuisance variables (e.g. all possible locations, scales, orientations etc.), a computationally prohibitive proposition when we need to perform hundreds of thousands of comparison each second. For this reason, canonization is the preferred approach in the literature, whereby a *descriptor*[7] is designed to be ideally invariant, but at least insensitive, to viewpoint and illumination variability. For instance, the ubiquitous SIFT descriptor (Lowe, 2004) is designed to be invariant to contrast and planar similarity

[6]We refer to the reference frame when the platform is first turned on as the "global" reference frame, even though it is not geo-referenced, because that frame is maintained throughout the sequence.

[7]A descriptor is a statistic, i.e. a deterministic function of the image.

Figure 7: ***Long Outdoor reconstruction.*** *Left: Our reconstruction of a 30 km long driving sequence, overlaid on an aerial view (UCLA Campus at the top, Santa Monica's Ocean Avenue and the beach at the bottom). Error is less than 0.5%. Right: Detail of area showing the position of point features and the motion reconstruction, overlaid to an orthographic aerial image.*

Figure 8: ***Outdoor reconstruction*** *for the data of (Mourikis and Roumeliotis, 2007).*

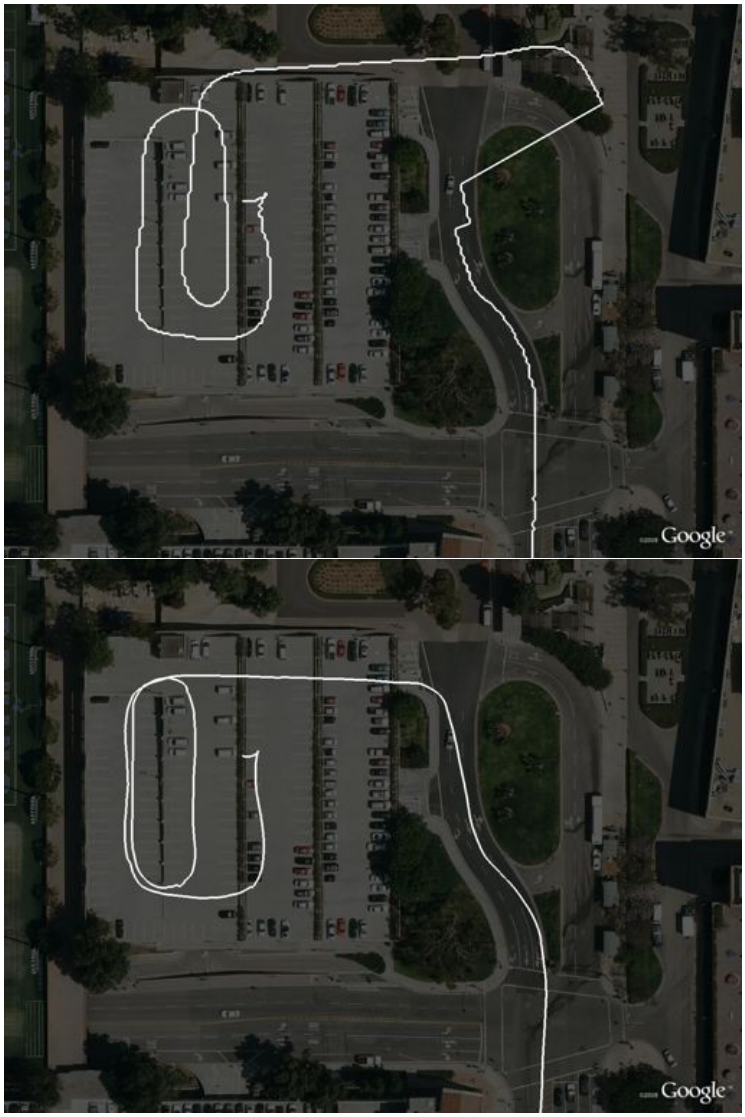

Figure 9: ***Reconstruction of driving through a parking garage.*** *Top: GPS with inertial (note the large jump in state estimate upon exiting the garage and reacquiring satellite coverage). Bottom: Our result.*

transformations (by referring the image to a similarity co-variant frame), and insensitive to local viewpoint changes (by employing a coarse binning of gradient orientation).[8] Unfortunately, any image-based canonization procedure reduces the discriminative power of the representation, as it has been shown in (Vedaldi and Soatto, 2005) and illustrated in figure 11. Therefore, one is faced with the choice of either using local invariant features with limited discriminative power, or perform a costly comparison to marginalize nuisance variables such as scaling and rotation. The advantage of our integrated approach is that we can bypass this choice. In fact, the filter provides a global planar orientation reference via the current estimate of the gravity vector, and a global reference for scale, based on the distance of the feature from the viewer when it was stored in the map. Therefore, we do not need to canonize rotation and scale – which reduces the discriminative power of the representation – and we do not need to marginalize them, because we can perform the comparison in a common similarity reference frame. We take full advantage of this benefit in section 5.1, where we introduce the low-level descriptor that is used as a building block of our visual map.[9]

In Figure 10 we compare our location recognition approach to a purely image-based approach such as FabMap Cummins and Newman (2008). Comparison is not exactly straightforward, or fair, since FabMap uses a more sophisticated processing pipeline to increase the discriminative power of features, whereas our approach can benefit from the global orientation and scale reference available from the filter. In our approach, as illustrated in Figure 11, the features are more discriminative because they can exploit a global orientation and scale reference from the filter. Therefore, a simpler (and faster) processing pipeline is sufficient. Of course, our approach could benefit from any improvement in the classifier downstream, although we do not exploit this in our current implementation. In this experiment we have considered locations (adjacent nodes in the graph linked by co-visibility) and tested whether at least one loop closure was detected for each location. Note that a full "kidnapped robot" scenario would not enable full exploitation of our framework, so for that scenario a generic image-based location recognition approach is more appropriate.

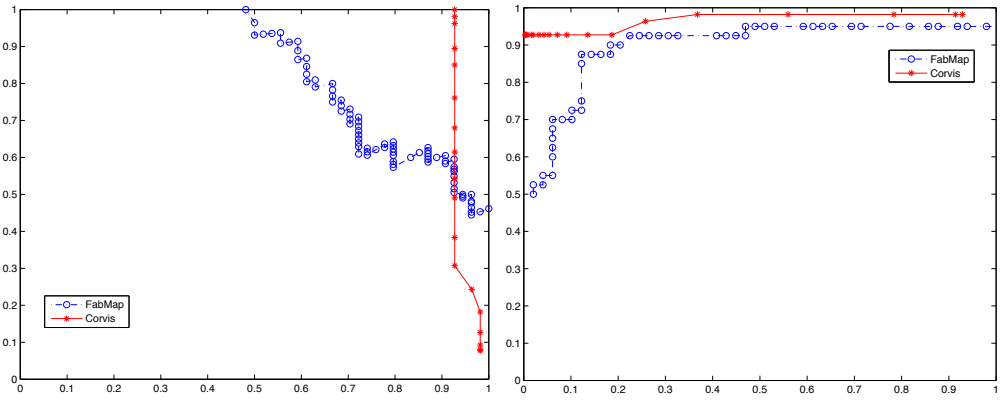

Figure 10: *Comparison with FabMap. Precision vs. recall (left) is shown for Corvis (red starred line) compared to a stock implementation of a general image-based location recognition system (Cummins and Newman (2008), blue circled line). Receiver operator characteristics (ROC) Are shown on the right. Despite using a simpler classifier, Corvis is competitive because the features are more discriminative. This is due to the fact that orientation and scale are relative to the global reference provided by the estimate of gravity and depth in the filter. Our approach can be further improved by adopting more sophisticated classifiers, for instance the one used in FabMap, or any other improvement in generic location recognition schemes. Note that our approach has a geometric validation gate as well as aggregation of nodes into locations.*

The novelty of our approach is not in the particular search algorithm we use for location recognition, and

[8] Similarity and affine transformations model small changes of viewpoint for planar patches that are far enough away from the viewer. However, it has been shown in (Sundaramoorthi et al., 2009) that one can construct contrast- and viewpoint-invariant descriptors for scenes of *any* shape.

[9] The reader can consult the recent reference (Soatto, 2010) for bounds on the discriminative power of local descriptors.

Figure 11: ***Loss of discriminative power*** *due to orientation canonization. If orientation is canonized using local gradient statistics, as in SIFT, all patches would collapse into the same descriptor, and therefore would (incorrectly) match each other in a standard similarity-invariant bag-of-features. However, the top right patch has a physically different scale (due to greater depth) than the other two, and the lower right patch is physically rotated by* 90^0 *relative to the others. Our approach performs canonization relative to an affine reference determined by the current estimate of gravity and the depth of the feature, thus resolving the three patches as three distinct features.*

indeed any improvement in large-scale search algorithms can be readily transferred to our approach. It is in the representation we employ, that uses a planar similarity reference frame *provided by the filter,* rather than one computed from the image.

5.1 Feature Representation

In typical location recognition applications, a local feature detector is responsible for determining a covariant reference frame relative to the desired invariance group. For instance, the SIFT detector determines a location, a scale and a direction in the image that provide a planar similarity reference frame. In our case, the feature position in the global frame, and gravity, provide this similarity reference frame, and therefore we do away with a feature detector altogether, and focus instead on the feature descriptor. Already the image, in the local frame, is by construction invariant to similarity transformations. To achieve invariance to contrast, we replace the image with the gradient direction at each point – since the gradient direction is dual to the geometry of the level lines which is a maximal contrast invariant statistic (Soatto, 2009). However, instead of coarsely binning the descriptor to achieve some kind of insensitivity to viewpoint changes beyond similarities, as in SIFT and HOG (Dalal and Triggs, 2005), we have the luxury of tracking, which gives us samples of the image in the local frame under a distribution of viewpoint changes. Therefore, we simply average gradient orientations over time, instead of coarsely binning them in space. This descriptor has been introduced by (Lee and Soatto, 2010), where it is shown to provide the smallest expected probability of matching error under a nearest-neighbor classifier rule. It is also much faster than SIFT or HOG to compute (see (Lee and Soatto, 2010) for performance specifications and for a real-time implementation on an iPhone).

Note that other researchers have bypassed rotational canonization, for instance (Zhang et al., 2006, Lazebnik et al., 2006). However, this assumes that the ordinate axis of the image is the reference orientation,

which is a safe assumption for human-captured images uploaded from the web, but it is definitely not wise for robot-captured images, especially for unmanned aerial vehicles (UAVs). An approach similar in spirit to ours was also presented in (Goedeme et al., 2004). A similar approach applies to scale canonization, which (monocular) vision-only techniques do not have the option of eliminating without even stronger assumptions than in the orientation case. Because scale is observable and consistent in the vision-inertial system, we can use the depth $Z = e^{\rho}$ of a particular feature to determine its descriptor's scale σ:

$$\sigma = \frac{\sigma_0}{Z}, \tag{17}$$

where σ_0 represents the size of the descriptor's support region at a depth of 1 meter. This is obviously sensitive to the (widely-varying) depth of features. For example, a reasonable patch size of 32 pixels at 1 meter for an indoor scene would degenerate to 1 pixel for a feature 32 meters away in an outdoor scene. Therefore, we select an appropriate σ_0 based on the depth of the feature, effectively dividing our descriptors into depth bands, as (Mei et al., 2009) have also advocated. This prevents some potential matches at widely varying viewpoints, but quantization errors would make such descriptors unlikely to match regardless.

Recall that features are added to the filter in groups. This allows us to substantially reduce the number of frames which are processed for recognition (a very expensive task). For each video frame in which a group is added to the filter, the recognition module preprocesses the gradient orientation and magnitude for descriptor generation. As each feature is removed from the graph, we determine the final estimate of its scale and orientation in the first frame, and compute its descriptor using the last few tracked frames. These descriptors are then quantized using a small dictionary, and both the descriptor and the dictionary label, or "visual word", are sent to the mapping module, described in the next section, and in more detail in Appendix B of (Jones and Soatto, 2009).

This representation enables faster performance, in the sense that with equal number of comparisons, the descriptor is more discriminative than a canonized invariant, and therefore fewer matches have to be performed in order to return a positive location recognition. Conversely, at equal discriminative power, we do not need to marginalize rotation and scale at decision time. Direct comparison with other localization schemes that do not provide a gravity and scale reference is somewhat unfair (Fig. 10), as the representation we have described is independent of what search mechanism one employs, so we can benefit from any of the most recent large-scale search algorithms proposed in the literature.

Using the results of the filter tracks in the representation has the added advantage that, instead of storing features independently detected in each image, we store one descriptor for every track, which greatly reduces redundancy in the map. Thus our maps scale neither by time, as common for most maps built from features independently detected in single images, nor by space, as common for more sophisticated maps that take into account odometry. Instead, they scale by *visibility*, as we describe in the next section.

5.2 Locations as topological entities defined by visibility

We now describe our mapping module, which constructs a graph based on the output of the structure from motion filter. We refer the reader to figure 12 for a high-level illustration of how the map is built for a trivial example. Like other mapping and localization algorithms such as (Zhu et al., 2007, Mei et al., 2009, Konolige and Agrawal, 2008, Mourikis and Roumeliotis, 2008, Zhang and Kosecka, 2006, Wang et al., 2005, Williams et al., 2007, Goedeme et al., 2007, Košecká et al., 2005), we exploit structural constraints provided by the map for location matching. Our topological map construction is inspired by some previous approaches to topological map building, such as (Eade and Drummond, 2007a). In contrast to their approach, we do not use the graph for global optimization (at least not online). Instead, we point out that a topological map can be useful *without* global optimization. The approaches of (Goedeme et al., 2007, Mei et al., 2009, Cummins and Newman, 2008) also scale favorably and produce a topological map, also considering loop closure. In (Schindler et al., 2007), scalable search is achieved using a vocabulary tree. However, the techniques they present are generic, treating the localization task like any other recognition problem. Each "place" is simply a single image, and their recognition task seeks to find a matching image within a few meters of their query image. They also treat each query as a "kidnapped robot" problem and do not exploit temporal or spatial

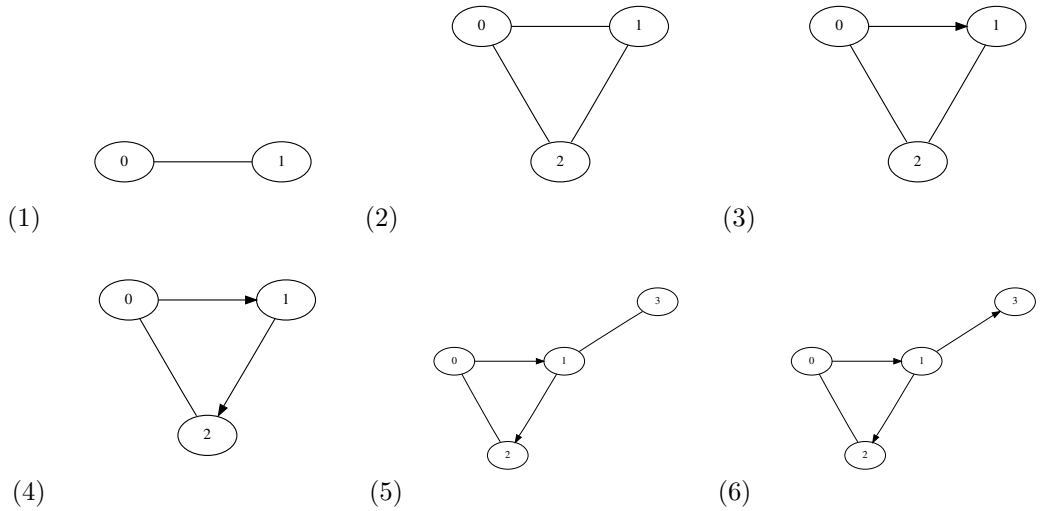

Figure 12: ***Building a Map*** *(1) Two groups are acquired at the same time, and are connected by an edge because they were seen together. (2) A new group is acquired, covisible with both of the current groups. (3) Group 0 is lost, and group 1 become the reference group. The transformation between groups 0 and 1 becomes fixed. (4) Group 2 is lost, fixing the transformation between groups 1 and 2; group 1 remains the reference group. (5) Group 3 is acquired. (6) The sequence ends; the transformation between groups 1 and 3 is fixed.*

consistency. Such an approach is somewhat orthogonal to ours, in that the improvements we present for descriptor generation and database construction could be combined with the improvements they present for vocabulary trees. The approach in (Goedeme et al., 2007) defines each "place" in terms of loop-closing hypotheses decided using Dempster-Shafer inference tools, which are considerably more laborious than what we propose here, and that we can afford given computational constraints for real-time operation.

Although it is possible to recognize a place from just one image, images are not a natural unit for recognition of locations. It is more useful to combine the visual information and geometric relationships acquired while moving through space. Therefore, we propose a definition of location based on a graph built from the covisibility relationships and track lifetimes of features in our inertial structure from motion filter. As the mapping module receives information from the filter and the recognition module, it adds a node to the graph for each new group. When a node is added, undirected edges are added to the graph connecting nodes for each of the groups currently in the filter to the newly added node. When the number of features in a group declines below the minimum, no more edges are added to that group's node. Thus, the edges encode the covisibility relationships between various sets of features in the filter. In particular, the set of all features associated with one node and its immediate neighbors forms a superset of all the features that one might expect to see in an image when revisiting the same area. It is in this way that we define our concept of location. An example of the covisibility graph for a short sequence is shown in Fig. 13.

While the topology of the covisibility graph contains a great deal of information, we also need geometric information in order to address many problems of interest. To this end, we add directed edges to the graph corresponding to the *constraining* geometric relationships between nodes. Our filter yields an estimate of the geometric relationship between any two nodes in the graph, together with its uncertainty. However, certain of those relationships are special because the gauge ambiguity is fixed by constraining the reference position T_{ref} and orientation Ω_{ref} of one group at a time in the filter. Here Ω_{ref} denotes the exponential coordinates of $R_{ref} = e^{\widehat{\Omega}_{ref}}$. Thus, each time a group is removed from the filter, the final estimate of its position and orientation (and their respective variances) are estimated *relative to the reference group at the time*. These

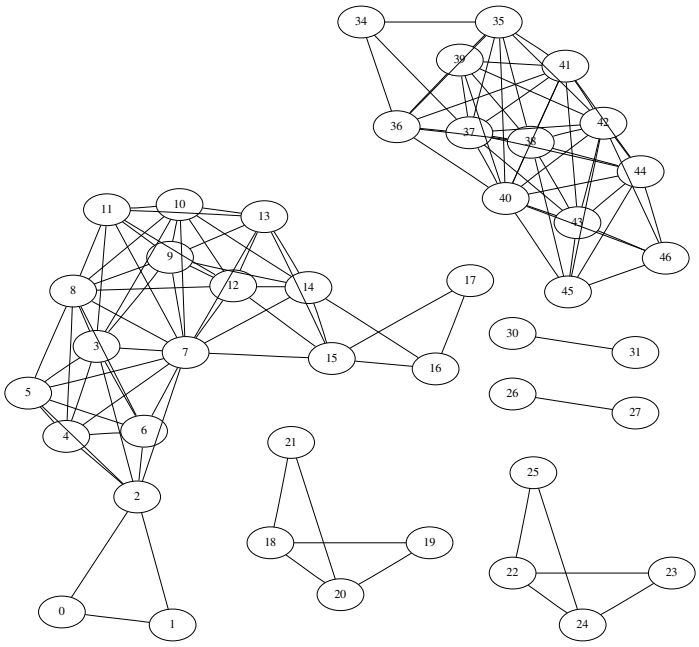

Figure 13: ***Covisibility graph*** *for a short laboratory sequence. Each node corresponds to one group in the filter, and each edge connects two nodes that were visible at the same time. Unconnected regions of the graph correspond to loss of all features in the filter due to fast turns (which would cause vision-only SFM to fail).*

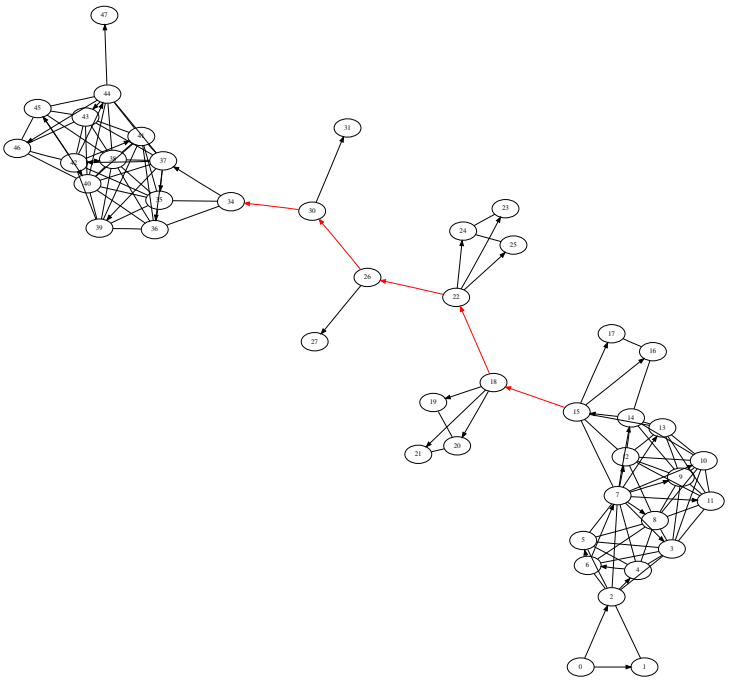

Figure 14: ***Geometric constraints*** *for a short laboratory sequence. Each node corresponds to one group in the filter, and each undirected edge connects two nodes that were visible at the same time. Directed edges connect two groups that have a constraining geometric relationship. Nodes highlighted in red are geometric relationships added based solely on inertial information (that is, where all features were lost).*

are the geometric relationships which we describe as constraining, and are the necessary and sufficient set for the reconstruction of every other geometric relationship in the map. In figure 14, we show the addition of these constraints to our previous example.

There are two special cases which require some (minimal) care in handling. First, when a group becomes the reference group, its T_{ref} and Ω_{ref} become fixed; their final estimate is relative to the previous reference group which was just dropped. Second, when all groups are lost, the covariance increases as the IMU integrates relative to the last reference group. So, although the previous reference group and the next added group do not have a covisibility relationship, they do have a constraining geometric relationship.

Each directed edge in the graph is associated with a transformation (T, R) describing the position and orientation of the child node relative to the orientation. Thus, the relative transformation between any two nodes in the graph can be determined by traversing the directed edges between them, accumulating transformations as they are encountered. An edge which is traversed opposite its direction is inverted before being accumulated.

In addition to the geometric relationships between nodes, each node is associated with the non-outlier features in the relevant group. Each feature is stored with its position relative to the group's reference frame, as well as the descriptor and visual words described previously. The lists of features are stored sorted by their visual word.

5.3 Loop Closure

As the map is being built, every new location is checked against the entire map, or a subset depending on the application, to determine if a geometric relationship can be determined between it and another location, thus

creating a loop closure mechanism. Because we check every location, and we want our system to operate in real time while scaling to thousands of locations, we must be able to identify candidate matches and verify them quite rapidly, in linear or sublinear time. The first step of our approach identifies candidate matches by visual similarity. It is inspired by the text search literature, and is similar in spirit to other large-scale visual recognition systems (Sivic and Zisserman, 2003, Nister and H.Stewenius, 2006, Ho and Newman, 2007, Chum et al., 2009), but adapted to make use of the map. The second stage verifies matches geometrically, and generates the relative transformation between the two locations.

The query is built simply by merging (in sorted order) the lists of visual words corresponding to all features in a particular location (a node and its first degree neighbors). The list (or histogram) is scored against the list of visual words at each node in the graph. The scoring function for nodes is based on the inverse document frequency, or IDF (Salton and Buckley, 1988):

$$S_i = \sum_{k=1}^{K} \frac{f_q^k f_n^k}{log(N/F^k)}, \tag{18}$$

where N is the number of nodes, K is the number of terms, f_q^k is the number of times term k appears in the query list, f_n^k is the number of times term k appears in node i's list, and F^k is the number of documents in which term k appears at least once. The idea behind this mechanism is to increase the score as the term matches the document better, but offset the increase as the term appears more frequently in the entire database (and is thus less discriminative). This score is computed in linear time thanks to the sorted feature lists.

We emphasize that a *node* in the graph is not the same as a *location* centered at that node, so this step is comparing locations to nodes rather than locations to locations. While it is obvious that the graph could be modified to store all descriptors associated with a particular location in one place, it would multiply the storage requirements and computational expense by the average degree of the graph. It is also made unnecessary by the following diffusion step. In order to upgrade our scores for nodes to scores for locations, we simply add the scores for all nodes corresponding to a particular candidate location, and divide by the total number of features which appear at that location (this corresponds to the term frequency, or TF). It is easy to verify that the result is the standard TF-IDF score computed on locations, and is identical to the score that would have been computed had we explicitly built term lists at each location and computed TF-IDF directly.

We describe this step as a "diffusion" step because it has the effect of spreading scores across the graph. If we consider the scores on the original nodes, we expect many false positives due to the rather weak matching constraints, but we also expect those to be spread out somewhat randomly across the graph. We also expect that due to the arbitrary divisions of features between the nodes that make up a particular location, the correct matches will be somewhat spread out amongst several nodes, but these nodes should be neighbors in the covisibility graph. Therefore, this diffusion process suppresses the false positives and reinforces the correct matches. This step yields a considerable improvement in performance.

Once we have generated a small number of candidate matches, we wish to determine which, if any, of the candidates truly match our query, and also determine the actual transformation between the two locations. The common approach to this problem is to use random-sample consensus to test consistency with an epipolar geometric model. This is usually done with RANSAC or one of its many variants Ma et al. (2003). The challenge with such an approach is that the number of candidate matches grows combinatorially as the number of features required to generate a hypothesis increases. Our the constraints provided by our filter can be exploited to simplify the hypothesis testing significantly, and even to forgo random sampling altogether. To see this, first consider that, because we match locations rather than images, we have at our disposal the position of each point. We therefore generate a transformation model directly rather than through standard epipolar geometry computation. This reduces the number of points required to generate a hypothesis from 5 or 8, depending on the formulation of RANSAC, to 3. Second, because gravity provides us with a consistent global reference, we can eliminate one more degree of freedom, requiring only two points to generate a hypothetical correspondence. Further, because we have a unique global scale, most candidate

matches for the second point can be immediately discarded based on the distance between the points in each location. On average, this means that the number of hypotheses to test is linear with the number of candidate matches. (For short baselines this is especially true, as drift in orientation can be effectively ignored.)

For these reasons we can forgo a sampling approach such as RANSAC, and exhaustively test each match directly. This is an important consequence of the tight integration between our filter and our location recognition module. For each candidate match of feature f_q in the query location and f_k in the candidate location, and each candidate match g_q and g_k, with coordinates in their respective local frames, we check if $|f_k - g_k|^2 = |f_q - g_q|^2 \pm \epsilon$, where epsilon is determined from the variances of the features (as provided by the filter). If this test is passed, the rotation about the Z axis is determined from the angle between the projections of the vectors $g - f$, and translation is determined as $f_k - f_q$. This transformation is applied to each set of matches and we count the number of inliers based on which features are as close to each other as required by their variances.

Given sufficient support, a match is declared as valid, and a new directed edge is added to the graph, with the transformation (refined using all inliers as support) attached to the edge. Additionally, to prevent spurious matches in future visits to the same location, the inlier features are removed from one of the sets of matching nodes, and new covisibility edges are added between their correspondents and their neighbors. Precisely, for each pair of nodes (i, j) which contain matching features, those features are removed from j and all of j's neighbors gain visibility edges connecting to node i. The graphical result of loop closure is shown in figure 15.

At this point, the objection may be raised that a geometric constraint has been added to the graph without checking its consistency with the existing constraints. In fact the loop just closed has a series of transformations along its edges; accumulating them while traversing the entire loop should yield the identity. However, whatever drift has been accumulated is *not* corrected by our approach, and the resulting transformation almost certainly will not be the identity.

While one could easily run a post-mortem bundle adjustment, our emphasis is on real-time causal processing. We also emphasize that the graph we generate is *not intended* to serve as a global metric map. In fact, by incorporating inconsistencies in the map, the graph actually represents many different mappings of the same locations, depending on what node is the reference. When using the graph for planning or navigation, the current node is selected as the origin, and the graph is traversed breadth-first. Each node's position is then generated relative to its parent node. Locally, the generated map will be consistent, and is every bit as useful as a globally consistent map. Inconsistencies are pushed to the far edges of the map, where they have little impact on any real task, even global planning.[10]

To illustrate loop closing, which by now is a standard component of any SLAM system, we report representative examples that illustrate the peculiarities of our approach using the indoor sequence from figure 3. To ease visualization, we have purposely detuned the filter, resulting in large drift. The map is made topologically consistent by the loop closing process, but geometric inconsistencies persist, and are pushed to the far edge of the graph. To gage the speed of our method, we ran it exhaustively: every node is compared against the entire map in detect loop closure in less than 10 seconds. During on-line operation we limit the test to locations that are within a prescribed radius so as to maintain real-time operation. With our current hardware we can achieve real-time operation, depending on the number of locations, on regions of diameter up to 4Km.

One of the benefits of a loop-closure mechanism is the ability to fuse multiple map segments even in the absence of a geo-referenced frame. in the experiment in figure 17, the first two maps were constructed from data captured 10 months before the third, illustrating the resilience of our method to changing conditions, lighting, and even construction. The combined map includes $3,743$ locations with $24,611$ features over 10Km of roads, and was generated in 2 minutes and 27 seconds, again using exhaustive search, using our current

[10] We reemphasize that even in the case where a globally consistent map is desired, our graph serves as input to global optimization. By maintaining inconsistencies in the map, and not doing online optimization, we delay decisions which may prove to have been suboptimal when new data is acquired. Global optimization would best be performed offline after all data has been acquired.

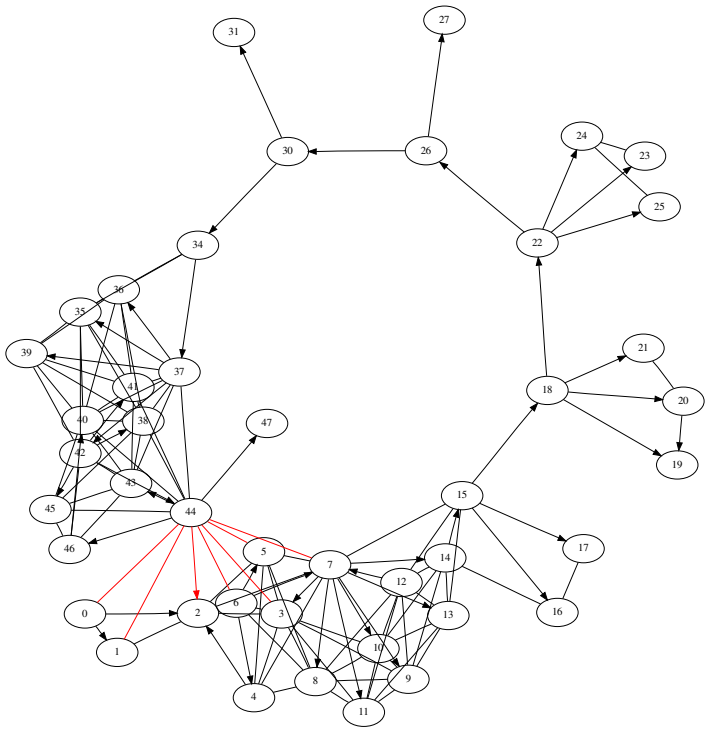

Figure 15: ***Closing the loop*** *for a short laboratory sequence. Each node corresponds to one group in the filter, and each undirected edge connects two nodes that were visible at the same time. Directed edges connect two groups that have a constraining geometric relationship. The edges added by the loop-closing process are highlighted in red.*

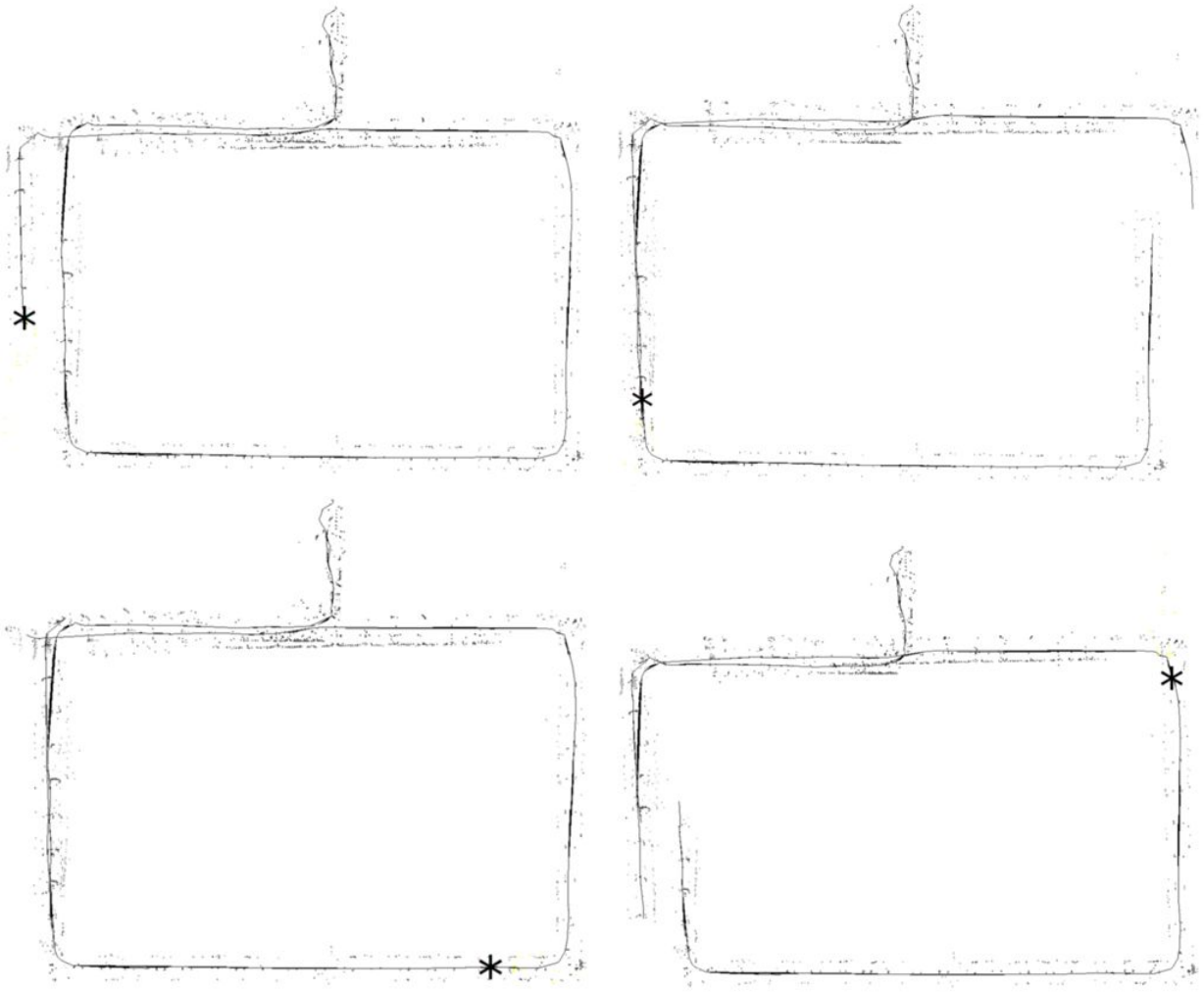

Figure 16: ***Loop closing.*** *Each of the four maps are generated by traversing the graph breadth-first, starting at the node marked with *. Top left: Map just before loop closure. Top right: Map just after loop closure. Notice that the local area has become consistent, and the inconsistency has been moved to the other side of the map. Bottom: The same graph rendered from two other reference nodes. In all cases the local area remains consistent and the inconsistency is pushed far away.*

hardware. Loop closing runs as a separate thread in our implementation, with lower priority than the filter updates.

6 Discussion

We have presented a system for estimating the motion of a sensing platform, building a 3-D map of the environment and recognizing previously seen locations in the map in real-time in a scalable fashion. We have analyzed the conditions under which this system can operate, that reveal that sensitive parameters such as gravity and the camera-to-inertial calibration can be estimated on-line, through an "auto-calibration" procedure. We have implemented our system both in simulation and in a custom embedded platform that we have tested extensively, indoor and outdoors, achieving results that exceed the state of the art without requiring knowledge of global position and orientation or lengthy initialization procedures for bias convergence.

Despite these advances, there are still operational and development challenges that must be overcome before having an integrated "turn-key" solution for vision-aided navigation. Complex visual environments with multiple moving objects (e.g. cars, people), with significant deviation from Lambertian reflection, camera miscalibration, photometric saturation, motion blur in low-light situation, aberrations due to rain, fog, smoke and other visual disturbances make it difficult to provide performance guarantees, or even strict empirical characterization, in a way that is possible for simpler sensory modalities such as inertial or GPS. Therefore, it is important that the complementary modalities can compensate for the failure modes of vision, which we have shown to be possible, but that has to be engineered systematically before vision-aided navigation can be employed at the push of a button.

In our experiments we have for the most part been able to maintain successful operation through very long sequences, without need for re-initialization. The failure modes that we have experienced have been due to hardware glitches causing incorrect ordering of the data. Our current implementation does not discard data, so if an interrupt causes an image to be significantly delayed, this may cause temporal inconsistencies. In all the experiments we have conducted, we have been able to establish loop closure, even when significant changes had occurred, for instance in the map-stitching experiment where loop closure was established at 10-month intervals. We have been able to use the platform in a closed-loop experiment on a wheeled platform moving rather slowly (up to 1m/s). We do not currently handle situations that generate complete inconsistencies between the two modalities, such as elevators.

Acknowledgment

Research supported by AFOSR FA9550-09-1-0427 and ONR N00014-08-1-0414. We wish to thank S. Roumeliotis and A. Mourikis for their advice, suggestions, and access to data. We also thank J. Meltzer, J. Dong and B. Taylor for assistance with the

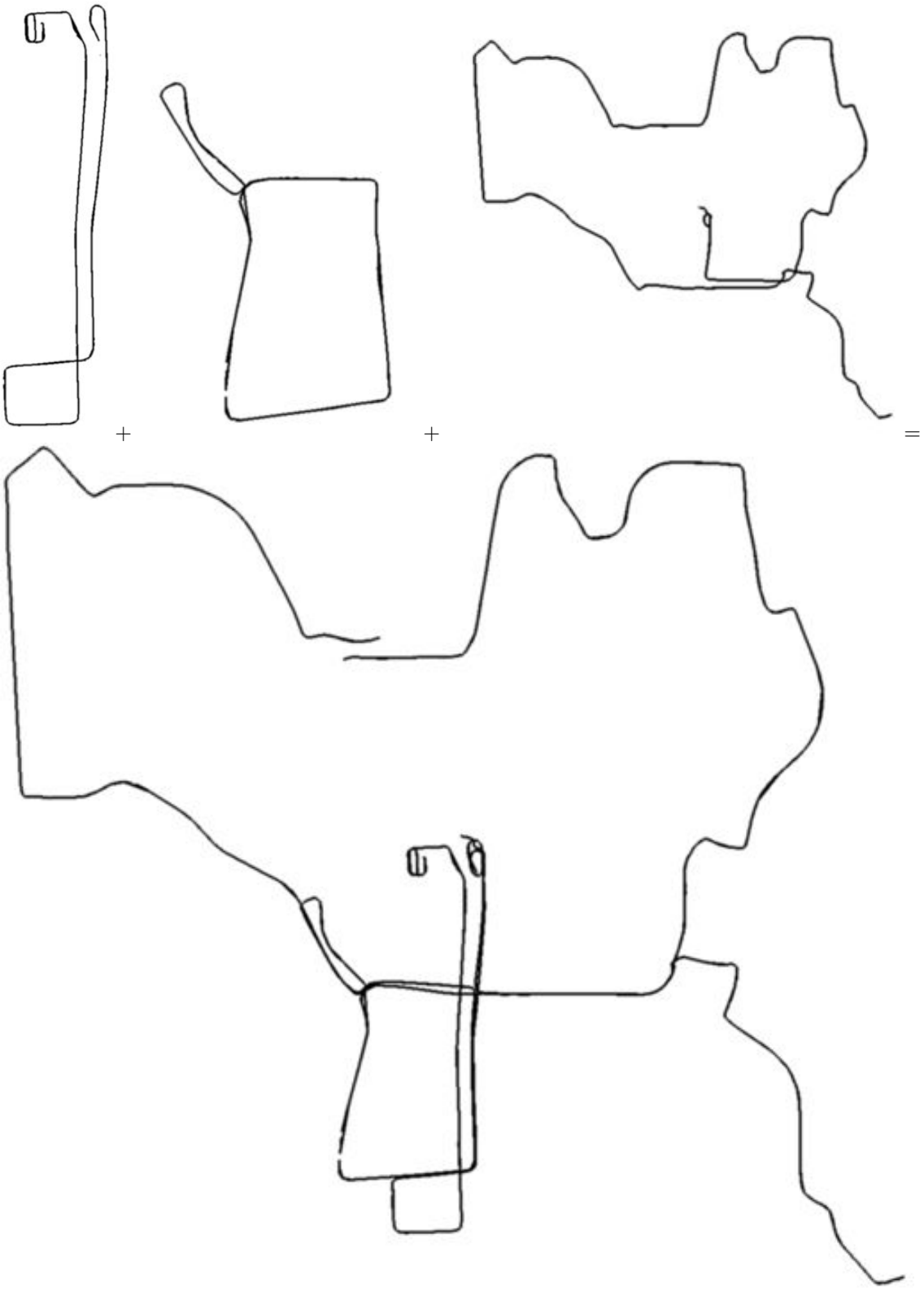

Figure 17: ***Map Fusion.*** *The three maps in the top row (captured nearly a year apart) were combined to generate the bottom map.*

References

D. L. Alspach and H. W. Sorenson. Nonlinear bayesian estimation using gaussian sum approximation. *IEEE Trans. Aut. Contr.*, 17(4):439–448, 1972.

A. Azarbayejani and A.P. Pentland. Recursive estimation of motion, structure, and focal length. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 17(6):562–575, June 1995.

G. Baldwin, R. Mahony, J. Trumpf, T. Hamel, and T. Cheviron. Complementary filter design on the Special Euclidean group SE (3). *A, A*, 1:2, 2007.

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

D. Brigo, B. Hanzon, and F. LeGland. A differential geometric approach to nonlinear filtering: the projection filter. *IEEE Trans. on Automatic Control*, 68:181–188, 1998.

A. Chiuso, P. Favaro, H. Jin, and S. Soatto. Motion and structure causally integrated over time. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24 (4):523–535, 2002.

O. Chum, M. Perdoch, and J. Matas. Geometric min-hashing: Finding a (thick) needle in a haystack. 2009.

M. Cummins and P. Newman. Accelerated appearance-only SLAM. In *Proc. IEEE International Conference on Robotics and Automation (ICRA'08)*, Pasadena, California, April 2008.

N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. IEEE Conf. on Comp. Vision and Pattern Recogn.*, 2005.

A. Davison. Real-time simulataneous localisation and mapping with single camera. In *Proc. 9^{th} Int. Conf. on Computer Vision*, 2003.

E. D. Dickmanns and B. D. Mysliwetz. Recursive 3-D road and relative ego-state estimation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 14(2):199–213, February 1992.

E. Eade and T. Drummond. Monocular SLAM as a graph of coalesced observations. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, Rio de Janeiro, Brazil,, October 2007a.

M. Euston, P. Coote, R. Mahony, J. Kim, and T. Hamel. A complementary filter for attitude estimation of a fixed-wing uav. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008. IROS 2008*, pages 340–345, 2008.

P. Favaro. Stima del moto e della struttura della scena tramite visione dinamica. Laurea thesis, University of Padova, 1998.

P. Favaro, H. Jin, and S. Soatto. A semidirect approach to structure from motion. In *IEEE Intl. Conf. on Image Analysis and Processing*, pages 250–255, 2001.

P. Favaro, H. Jin, and S. Soatto. A semi-direct approach to structure from motion. *The Visual Computer*, 19:1–18, 2003.

T. Goedeme, T. Tuytelaars, and L. Van Gool. Fast wide baseline matching for visual navigation. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, June/July 2004.

T. Goedeme, M. Nuttin, T. Tuytelaars, and L. VanGool. Omnidirectional vision based topological navigation. *Int. J. of Computer Vision*, 74(3):219–236, 2007.

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

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

A. Isidori. *Nonlinear Control Systems*. Springer Verlag, 1989.

A.H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, 1970.

H. Jin, P. Favaro, and S. Soatto. Real-time 3-d motion and structure from point features: a front-end system for vision-based control and interaction. In *Computer Vision and Pattern Recognition*, pages 778–779, 2000.

E. Jones and S. Soatto. Visual-inertial navigation, localization and mapping: A scalable real-time large-scale approach. *Technical Report UCLA-CSD-100010*, August 27, 2009 (revised May 10, 2010).

E. S. Jones, A. Vedaldi, and S. Soatto. Inertial structure from motion and autocalibration. In *Workshop on Dynamical Vision*, October 2007.

S. J. Julier and J. K. Uhlmann. A new extension of the kalman filter to nonlinear systems. In *Int. Symp. o Aerospace/Defense Sensing, Simulation and Control*, 1997.

T. Kailath. *Linear Systems*. Prentice Hall, 1980.

M. Kayton and W.R. Fried. *Avionics Navigation Systems*. Wiley and Sons, 1996. ISBN 0-471-54795-6.

J. Kelly and G. Sukhatme. Fast Relative Pose Calibration for Visual and Inertial Sensors. In *Experimental Robotics*, pages 515–524, 2009.

G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 1–10. IEEE Computer Society, 2007.

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

K. Konolige, M. Agrawal, and J. Sola. Large scale visual odometry for rough terrain. In *Proc. International Symposium on Robotics Research*, 2007.

J. Košecká, F. Li, and X. Yang. Global localization and relative positioning based on scale-invariant keypoints. *Robotics and Autonomous Systems*, 52(1):27–38, 2005.

S. Lazebnik, C. Schmid, and J. Ponce. Beyond bag of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. IEEE Conf. on Comp. Vision and Pattern Recogn.*, 2006.

T. Lee and S. Soatto. An end-to-end visual recognition system. *Technical Report UCLA-CSD-100008*, 2010 (available on-line after May, 2011).

D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2(60):91–110, 2004.

B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Image Understanding Workshop*, pages 121–130, 1981.

Y. Ma, S. Soatto, J. Kosecka, and S. Sastry. *An invitation to 3D vision, from images to geometric models*. Springer Verlag, 2003. ISBN 0-387-00893-4.

P. F. McLauchlan. Gauge invariance in projective 3d reconstruction. In *IEEE Workshop on Multi-View Modeling and Analysis of Visual Scenes, Fort Collins, CO, June 1999*.

C. Mei, G. Sibley, M. Cummins, P. Newman, and I. Reid. A constant time efficient stereo slam system. In *BMVC*, 2009.

F.M. Mirzaei and S.I. Roumeliotis. A Kalman filter-based algorithm for IMU-camera calibration. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007.

A.I. Morikis and S.I. Roumeliotis. A multi-state constraint kalman filter for vision-aided inertial navigation. Technical report, Dept. of Computer Science and Engineering, University of Minnesota, 2006. www.cs.umn.edu/∼mourikis/tech_reports/TR_MSCKF.pdf.

E. Mouragnon, F. Dekeyser, P. Sayd, M. Lhuillier, and M. Dhome. Real time localization and 3d reconstruction. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, 2006.

A. I. Mourikis and S. I. Roumeliotis. A multi-state constraint Kalman filter for vision-aided inertial navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3565–3572, Rome, Italy, April 10-14 2007.

A. I. Mourikis and S. I. Roumeliotis. A dual-layer estimator architecture for long-term localization. In *Proceedings of the Workshop on Visual Localization for Mobile Platforms*, pages 1–8, Anchorage, AK, June 2008.

R. M. Murray, Z. Li, and S. S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

E. Nebot and H. Durrant-Whyte. Initial calibration and alignment of low-cost inertial navigation units for land vehicle applications. *Journal of Robotic Systems*, 16(2):81–92, 1999.

D. Nister. Preemptive ransac for live structure and motion estimation. In *Proc. 9^{th} Int. Conf. on Computer Vision*, 2003.

D. Nister and H.Stewenius. Scalable recognition with a vocabulary tree. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2006.

D. Nister, O. Naroditsky, and J. Bergen. Visual odometry for ground vehicle applications. *Journal of Fields Robotics*, 2006.

G. Qian, R. Chellappa, and Q. Zheng. Robust structure from motion estimation using inertial data. In *Journal of the Optical Society of America A*, 2001.

S. I. Roumeliotis, A. E. Johnson, and J. F. Montgomery. Augmenting inertial navigation with image-based motion estimation. In *IEEE Intl. Conf. on Robotics and Automation*, 2002.

G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 1988.

G. Schindler, M. Brown, and R. Szeliski. City-scale location recognition. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–7, June 2007.

J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proc. 9^{th} Int. Conf. on Computer Vision*, 2003.

S. Soatto. Actionable information in vision. In *Proc. of the Intl. Conf. on Comp. Vision*, October 2009.

S. Soatto. Steps Towards a Theory of Visual Information. NIPS Tutorial Lecture Notes, (also course lectures for UCLA CS269, W10), 2010.

G. Sundaramoorthi, P. Petersen, V. S. Varadarajan, and S. Soatto. On the set of images modulo viewpoint and contrast changes. In *Proc. IEEE Conf. on Comp. Vision and Pattern Recogn.*, June 2009.

R. Tsai. A versatile camera calibration technique for high accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE J. Robotics Automat.*, RA-3(4):323–344, 1987.

A. Vedaldi and S. Soatto. Features for recognition: viewpoint invariance for non-planar scenes. In *Proc. of the Intl. Conf. of Comp. Vision*, pages 1474–1481, October 2005.

A. Vedaldi, H. Jin, P. Favaro, and S. Soatto. Kalmansac: Causal inference of dynamical processes in the presence of outliers. In *Proc. of the Intl. Conf. on Comp. Vision*, October 2005.

M. Veth and J. Raquet. Fusion of low-cost imaging and inertial sensors for navigation. In *Proc. of the ION meeting on Global Navigation Satellite Systems*, 2006.

M. Veth, J. Raquet, and M. Pachter. Stochastic constraints for efficient image correspondence search. *IEEE Transactions on Aerospace and Electronic Systems*, 2006.

J. Wang, R. Cipolla, and H. Zha. Vision-based global localization using a visual vocabulary. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 4230–4235, April 2005.

B. Williams, G. Klein, and I. Reid. Real-time SLAM relocalisation. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, Rio de Janeiro, Brazil,, October 2007.

R. Yang and M. Pollefeys. Multi-resolution real-time stereo on commodity graphics hardware. In *Proc. CVPR'03 (IEEE Conf. on Computer Vision and Pattern Recognition)*, pages 211–218, 2003.

J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *Intl. J. of Comp. Vis.*, 2006.

W. Zhang and J. Kosecka. Image based localization in urban environments. *3dpvt*, 0:33–40, 2006.

Z. Zhu, T. Oskiper, S. Samarasekera, R. Kumar, and H. Sawhney. Ten-fold improvement in visual odometry using landmark matching. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, Rio de Janeiro, Brazil,, October 2007.