Motion Estimation from Image and Inertial Measurements

Dennis Strelow and Sanjiv Singh

The International Journal of Robotics Research 2004; 23; 1157
DOI: 10.1177/0278364904045593

Dennis Strelow
Sanjiv Singh

Carnegie Mellon University
Pittsburgh, PA 15213, USA
dstrelow@cs.cmu.edu

Motion Estimation from Image and Inertial Measurements

Abstract

Cameras and inertial sensors are each good candidates for autonomous vehicle navigation, modeling from video, and other applications that require six-degrees-of-freedom motion estimation. However, these sensors are also good candidates to be deployed together, since each can be used to resolve the ambiguities in estimated motion that result from using the other modality alone. In this paper, we consider the specific problem of estimating sensor motion and other unknowns from image, gyro, and accelerometer measurements, in environments without known fiducials. This paper targets applications where external positions references such as global positioning are not available, and focuses on the use of small and inexpensive inertial sensors, for applications where weight and cost requirements preclude the use of precision inertial navigation systems.

We present two algorithms for estimating sensor motion from image and inertial measurements. The first algorithm is a batch method, which produces estimates of the sensor motion, scene structure, and other unknowns using measurements from the entire observation sequence simultaneously. The second algorithm recovers sensor motion, scene structure, and other parameters recursively, and is suitable for use with long or "infinite" sequences, in which no feature is always visible.

We evaluate the accuracy of the algorithms and their sensitivity to their estimation parameters using a sequence of four experiments. These experiments focus on cases where estimates from image or inertial measurements alone are poor, on the relative advantage of using inertial measurements and omnidirectional images, and on long sequences in which the percentage of the image sequence in which individual features are visible is low.

KEY WORDS—batch shape-from-motion, recursive shape-from-motion; inertial navigation, omnidirectional vision, sensor fusion, long-term motion estimation

1. Introduction

Cameras and inertial sensors are each good candidates for autonomous vehicle navigation, modeling from video, and other applications that require six-degrees-of-freedom motion estimation. In addition, cameras and inertial sensors are good candidates to be deployed together since, in addition to the obvious advantage of redundant measurements, each can be used to resolve the ambiguities in the estimated motion that result from using the other modality alone. For instance, image measurements can counteract the error that accumulates when integrating inertial readings, and can be used to distinguish between the effects of sensor orientation, acceleration, gravity, and bias in accelerometer measurements. On the other hand, inertial data can resolve the ambiguities in motion estimated by a camera that sees a degenerate scene, such as one containing too few features, features infinitely far away, or features in an accidental geometric configuration; to remove the discontinuities in estimated motion that can result from features entering or leaving the camera's field of view; to establish the global scale; and to make motion estimation more robust to mistracked image features.

In this paper, we consider the specific problem of estimating sensor motion and other unknowns from image, gyro, and accelerometer measurements, in environments without known fiducials and without external positioning systems such as GPS. Accurate motion estimation under these circumstances has a number of potential applications in autonomous navigation underwater, indoors, in rubble, in the urban canyon, or on Mars, which all preclude the use of global positioning; and in modeling from video. Some important potential applications preclude the use of precise inertial navigation systems, such as micro air vehicle navigation, where weight and cost are limiting factors. So, we specifically focus on the use of lightweight and inexpensive inertial sensors.

We present two algorithms for estimating sensor motion and scene structure from image and inertial measurements. The first is a batch algorithm that generates estimates of the sensor motion, scene structure, and other parameters by considering all of the image and inertial measurements simultaneously. In many applications, this batch estimate is of interest in its own right. In others, the batch estimate is important in understanding the best quality we can expect given a particular sensor configuration, vehicle motion, environment, and set of observations, and in measuring the inherent sensitivity of the estimate with respect to random observation errors.

The International Journal of Robotics Research
Vol. 23, No. 12, December 2004, pp. 1157-1195,
DOI: 10.1177/0278364904045593

Because the batch method uses all of the measurements from an observation sequence simultaneously, it requires that all of the observations be available before computation begins. The second algorithm is a recursive method that estimates sensor motion, scene structure, and other parameters from image, gyro, and accelerometer measurements as they become available, and is therefore suitable for long or "infinite" image sequences. This algorithm is a multirate method, meaning that image measurements and inertial measurements are processed by separate update steps, which allows the higher rate of inertial measurements to be exploited. Unlike many methods for motion estimation that use tracked image point features as measurements, our recursive method also includes a mechanism for incorporating points that become visible after initialization while maintaining an accurate state covariance estimate. This capability is essential for operation on most real image sequences.

We give a detailed description of the results of estimating motion from image and inertial data on a suite of four experiments. The first of these experiments, using a conventional camera, shows that the sensor motion can be accurately recovered by both the batch and recursive methods, even in a case when estimates from image or inertial data alone are poor. For this experiment we explore the sensitivity of the algorithms to their estimation parameters in detail. The second experiment uses a sensor motion similar to the first, but demonstrates the performance of the batch image-and-inertial, recursive image-and-inertial, and batch image-only algorithms using omnidirectional cameras. The purpose of this experiment is to evaluate the relative merits of inertial measurements and omnidirectional cameras for motion estimation. The last two experiments explore the performance of the recursive image-and-inertial algorithm on long observation sequences that would be difficult for image-only motion estimation because of the low percentage of images in which each scene point is visible.

The paper is organized as follows. Section 2 gives an overview of the previous work in estimating motion from image and inertial data that is most closely related to our own. An overview of our algorithms for estimating motion and other unknowns from image and inertial data is given in Section 3, and the details of our batch and recursive algorithms are described in Sections 4 and 5, respectively. Section 6 gives a high-level overview of the experiments, followed by detailed descriptions of each experiment in Sections 7, 8, 9, and 10. We conclude with a discussion of the results and promising future directions in Section 11.

2. Related Work

2.1. Overview

In this section, we briefly review the existing methods for estimating sensor motion from image and inertial measurements that are most closely related to our own. While the majority of these are recursive methods, a few relevant batch methods do exist, and we review both batch and recursive methods for this problem, in Sections 2.2 and 2.3, respectively. In Section 2.4, we briefly review the relationship between our own work and work in simultaneous localization and mapping (SLAM) and shape-from-motion (SFM). In Section 2.5, we briefly discuss data association as it is commonly used in radar tracking applications and SLAM, and its relevance to image feature tracking for our application.

2.2. Batch Methods

Deans and Hebert (2001) describe a batch method for bearings-only localization and mapping that uses Levenberg–Marquardt to estimate the planar motion of a vehicle and the two-dimensional location of landmarks observed by the vehicle's omnidirectional camera, from the landmarks' vehicle coordinate system bearings (one-dimensional projections) and the vehicle's odometry. The image-and-odometry error function of Deans and Herbert is closely related to our own error function, which utilizes image, gyro, and accelerometer measurements and is described in Section 4. However, estimating six-degrees-of-freedom motion from image measurements and inertial sensors introduces some difficulties that do not arise in estimating planar motion from bearings and odometry. In particular, using image measurements for six-degrees-of-freedom motion normally requires careful modeling and calibration of the camera, especially in the omnidirectional case, whereas this modeling is not required in two dimensions. In addition, the use of accelerometer observations for six-degrees-of-freedom motion requires estimation of the vehicle's velocity and orientation relative to gravity, which odometry does not require. In subsequent work, Deans (2002) also considered iteratively reweighted least squares (IRLS) for robust estimation within the batch framework, to improve the quality of estimates in the presence of image feature mistracking and other gross observation errors.

A second batch method is described by Jung and Taylor (2001). This method applies SFM to a set of widely spaced keyframes from an omnidirectional image sequence, then interpolates the keyframe positions by a spline that best matches the inertial observations. The resulting algorithm provides a continuous estimate of the sensor motion, and only requires that feature correspondences be established between the keyframes, rather than between every image in the sequence. Since the image and inertial measurements are not used simultaneously, however, the interpolation phase will propagate rather than fix errors in the motion estimated in the SFM phase.

2.3. Recursive Methods

Huster and Rock (2001a, 2001b, 2003) and Huster, Frew, and Rock (2002) detail the development of a recursive algorithm for estimating the six-degrees-of-freedom motion of an autonomous underwater vehicle (AUV) using gyro measurements, accelerometer measurements, and the image measurements of a single point in the vehicle's environment. Huster and Rock (2001a) develop an extended Kalman filter (EKF) and a two-step algorithm for a simplified, two-dimensional version of the motion estimation problem. The difficulties with linearizing the measurement equations about uncertain estimates in the EKF are sidestepped in the two-step filter, which chooses the state so that the image and inertial measurement equations become linear, and avoids linearization in the state time propagation using the unscented filter. In Huster, Frew, and Rock (2002), a full three-dimensional version of the two-step filter is constructed, and integrated with a controller and a method for choosing an endpoint trajectory that optimizes the quality of the motion estimates. The authors' experiments show that the resulting system is able to reliably perform a grasping task on a manipulator. In the measurements employed and in the quantities estimated, this system has many similarities to our own iterated extended Kalman filter (IEKF) for motion estimation, described in Section 5, but is more sophisticated in its handling of nonlinearities. However, the use of a single image point feature, which is motivated by the potentially poor quality of underwater images, precludes the use of the Huster and Rock method for long distance traverses.

You and Neumann (2001) describe an augmented reality system for estimating a user's view relative to known fiducials, using gyro and image measurements. This method is simpler than that of Huster and Rock in that it does not employ an accelerometer, which is a more difficult instrument to incorporate than a rate gyro, but expands the scene from a single point to a set of known points. Rehbinder and Ghosh (2001) also describe a system for estimating motion relative to a known scene, in this case containing three-dimensional lines rather than point features. Rehbinder and Ghosh incorporate accelerometer measurements as well as gyro and image measurements.

Qian, Chellappa, and Zhang (2001) describe an EKF for simultaneously estimating the motion of a sensor rig and the sparse structure of the environment in which the rig moves, from gyro and image measurements. The authors show motion estimation benefits from the addition of gyro measurements in several scenarios, including sequences with mistracking and "mixed domain" sequences containing both sensor translation and pure rotation. This system is more general than that described by Huster and Rock, You and Neumann, or Rehbinder and Ghosh, in that the sparse scene structure is estimated in addition to the motion rather than known, but this system makes the implicit assumption that each scene point is visible in every image of the sequence. In other work, Qian and Chellappa (2001) also investigated motion estimation from image and gyro measurements within a sequential Monte Carlo framework. In this case, they showed that the inclusion of gyro measurements significantly reduced the number of samples required for accurate motion estimation.

Chai, Hoff, and Vincent (2002) describe a system for simultaneously estimating the motion of a sensor rig and the sparse structure of the environment in which the rig moves, from gyro, accelerometer, and image measurements. This system estimates nearly the same unknowns as our own, but divides the estimation task between a motion filter, which estimates motion by assuming that the scene structure is known, and a structure filter, which estimates the scene structure by assuming the motion is known. The two filters are combined into a system for simultaneous estimation of motion and scene structure by supplying the estimates from each filter as the known inputs to the other. While this arrangement improves efficiency, it will result in artificially low covariances on the estimated motion and structure, particularly for long-term problems where, due to drift, the motion and the scene structure are likely to contain large but coupled errors. The authors do not explicitly consider the problem of estimating motion in sequences where points enter and leave the image sequence. They also consider the relative benefits of using one and two cameras in synthetic tests, and conclude that the use of two cameras can produce estimates with significantly lower errors.

The system described by Mukai and Ohnishi (1999) also simultaneously estimates the motion of a sensor rig and the sparse structure of the environment in which the rig moves using gyro and image measurements. In the method of Mukai and Ohnishi, the motion between pairs of images is estimated up to a scale factor, and the estimated motion is used to determine the structure of the points seen in both images. These pairwise estimates are then merged sequentially by applying the scaled rigid transformation that best aligns the recovered structures. This method handles sequences where points do not appear in every image, but both the pairwise motion recovery and merging steps of this method are ad hoc. For instance, this method does not maintain any measure of the error in the resulting motion estimates.

Foxlin (2002) describes a general framework for recursive simultaneous localization, mapping, and sensor calibration. The system consists of a decentralized filter that integrates the results of three complementary error filters for localization only, localization and mapping, and localization and sensor calibration. This architecture reduces the computation relative to including environmental object locations and sensor calibration parameters in the same state vector, and allows mapping and sensor calibration to be easily removed from the estimation once the positions of environmental objects or sensor bias values have been estimated with sufficient accuracy. Foxlin and Naimark (2003) describe one instance of this architecture, which uses gyro measurements, accelerometer measurements, and the image measurements of fiducials whose appearance but not (in general) location are known. In the case of auto-mapping, where the locations of fiducials are not known, the system pose is first determined relative to four fiducials whose x, y, and z positions are known, and the positions of subsequently acquired fiducials are entered into the filter using the image location and distance to the fiducial estimated from the fiducial's image size. In our own system, the initial position is initialized without a priori knowledge of any feature locations using a batch algorithm, and the positions of natural features are initialized using triangulation from multiple image positions, which is more appropriate for features whose appearance is not known a priori. While Foxlin and Naimark report real-time performance for an initial implementation of automapping, they do not report on the accuracy of their system for automapping.

In addition to the batch algorithm described in Section 2.2, Deans (2002) describes a hybrid batch-recursive method that estimates the planar motion of a vehicle and the two-dimensional location of landmarks observed by the vehicle's omnidirectional camera from the landmarks' vehicle coordinate system bearings (one-dimensional projections) and the vehicle's odometry. This method adapts the variable state dimension filter (VSDF), originally described by McLauchlan (1999) for SFM, to recursively minimize the same image-and-odometry error function minimized by the batch method of Deans and Hebert (2001). This approach naturally handles cases where points enter and leave the image sequence, and delays the linearization of measurements until the estimates used in the linearization are more certain, reducing bias in the state estimate prior. A similar adaptation of our batch algorithm for estimating motion from image and inertial measurements using the VSDF would be a natural extension of the work we describe in this paper.

2.4. Simultaneous Localization and Mapping and Shape-From-Motion

SLAM and SFM are two broad areas related to the problem of estimating motion from image and inertial measurements. In this section, we briefly describe the relation between these problems and our own, without describing specific methods for SLAM and SFM. Deans (2002) gives a good overview of specific methods for these problems.

Algorithms for SLAM typically estimate planar vehicle motion and the two-dimensional positions of landmarks in the vehicle's environment using observations from the vehicle's odometry and from a device that provides both the range and bearing to the landmarks. Since both range and bearing are available, the device provides noisy two-dimensional landmark positions in the device's coordinate system, and the major technical problem becomes the correct incorporation of these vehicle system measurements into the global coordinate system mean and covariance estimates. Recent work on SLAM focuses on mapping large areas, in which each landmark may only be visible from a small portion of the vehicle path, in a recursive fashion; and on "closing the loop", which exploits a landmark that has been lost and reacquired to maintain the topological consistency of the reconstructed motion and landmark positions.

Algorithms for SFM typically estimate the six-degrees-of-freedom motion of a camera and the three-dimensional position of points, from point features tracked in the camera's image sequence. Here, the observations the camera provides are the two-dimensional projections of landmark positions in the camera's coordinate system, rather than the three-dimensional camera system position, so in SLAM terminology, we would say that the camera provides bearings but not range. So, SFM typically estimates more unknowns than SLAM from less generous data. However, very little work has been done on automatically mapping large areas or on closing the loop using recursive SFM.

Our own work estimates six-degrees-of-freedom motion, sparse scene structure, and other unknowns from image, gyro, and accelerometer measurements. So, it falls somewhere between SLAM and SFM in terms of the observations and recovered unknowns. The technical approach that we describe in the following sections is more akin to existing algorithms for SFM than to approaches for SLAM. In particular, extending our approach to map large areas and close the loop, which are commonly addressed in SLAM algorithms, is future work.

In Sections 9 and 10, we have considered experiments in which the average fraction of the image sequence in which each point feature appears, or "fill fraction", is 4.3% and 0.62%, respectively. These are low fill fractions by SFM standards, and the SFM literature shows that the estimation of motion from such sequences is difficult. For instance, Weng, Cui, and Ahuja (1997) derive an analytical result describing the accuracy of optimal shape and motion estimates from image measurements as the fill fraction varies. They show that for a scenario including some reasonable simplifying assumptions, the error variance in both the estimated shape and motion increases as $O(n/(f^2))$, where n is the number of images and f is the average number of images that each point was visible in. That is, the error variances increase rapidly as the fill fraction decreases. Poelman (1995) considered the convergence of his own batch SFM algorithm given data sets with varying fill fractions and image observation noise levels. He found that his method performed well for fill fractions above some threshold, typically 50% for data sets with 2.0 pixel image observation noise and 30% for data sets with 0.1 pixel image observation noise, and failed "catastrophically" for data sets with lower fill fractions.

2.5. Tracking and Data Association

Our algorithms rely on accurate image feature tracking. In many scenarios where multiple targets are tracked over time using radar, sonar, or similar sensors, the correspondence between the targets and the measurements they generate at any one time is required for tracking the targets, but is not given explicitly. The problem of matching measurements with the tracked targets that generated them is called data association. Because the measurements alone may provide no information about which target generated them, methods for data association typically associate measurements with a target based on the likelihood of the measurements given the target's estimated kinematics. So, the problems of target tracking and data association become coupled. In scenarios where the targets may cross each other, may maneuver erratically, and may not produce a measurement at each time, solutions for the combined problems of tracking and data association may become quite elaborate. A good review of these issues and the corresponding approaches for tracking and data association is given by Bar-Shalom and Li (1995).

The data association paradigm has been widely adopted in simultaneous localization and mapping for associating sonar or laser returns with the landmarks in the environment that generated them (see, for example, Montemerlo and Thrun 2003). In the computer vision community, the data association paradigm has been investigated for image feature tracking by Rasmussen and Hager (2001). In this case, the authors applied two baseline data association methods, the probabilistic data association filter (PDAF) and joint probabilistic data association filter (JPDAF), to the problems of tracking homogeneous colored regions, textured regions, and non-rigid image contours.

Our own application requires tracking sparse, textured image features as the camera moves in a rigid environment, and we are concerned with two variations on this problem. In the first, we want to track a large number of features, typically 50–200, between adjacent images in the video sequence. For this problem, correlation tracking techniques, such as that of Lucas and Kanade (1981), have often been used in the SFM literature, and we have used Lucas–Kanade with manual correction in the experiments we describe in Sections 7, 8, and 9. Such techniques exploit the image texture information in the neighborhood of each feature rather than the kinematics of the feature, which are typically used in data association applications. In addition, two images of point features in a rigid scene are related by strong geometric constraints that are not applicable in typical data association applications. In Section 10 we have used a new tracking algorithm of our own design that combines correlation tracking with these constraints, and can produce highly reliable tracking through long image sequences, eliminating the need for manual correction. Because the focus of this paper is estimation rather than tracking, we have omitted a detailed description of this algorithm.

In the second variation, we are interested in reacquiring feature points whose three-dimensional positions we have previously estimated, possibly in the distant past, e.g. after the vehicle and camera close a loop. For reliability and efficiency, it becomes important in this variation to the compare currently visible features to only a small number of previously visible candidate features. Using the estimated camera and point positions for this purpose is promising, and couples the estimation and matching just as tracking and data association are coupled in other applications. As described in Section 11.3, this is future work and we expect to address this issue by exploiting recent work on image feature invariants (e.g. by Lowe 1999) and data association approaches.

3. Algorithms Overview

As mentioned in the introduction, we are working toward an algorithm for estimating six-degrees-of-freedom sensor and vehicle motion in scenarios where external position references, such as GPS, and a priori information on the sensor's environment are not available. Cameras and inertial sensors can each be used to estimate sensor motion in these scenarios, but the use of either modality alone is problematic. In this section, we review the specific difficulties in estimating sensor motion from image or inertial measurements alone, give an overview of our algorithms for estimating motion from both image and inertial measurements, and briefly describe the difficulties that remain for future work.

Inertial sensor outputs alone can be integrated to provide six-degrees-of-freedom position estimates, but noise in the inertial sensor outputs causes these estimates to drift with time. In addition, the outputs of inexpensive inertial sensors depend on biases that change with time and, in the case of accelerometers, the sensor's orientation with respect to gravity. So, some complementary modality must be employed to reduce drift and to estimate the biases and other variables that determine the sensor output.

Simultaneous estimation of camera motion and sparse scene structure from images alone, like that performed by SFM algorithms, can be used to estimate six-degrees-of-freedom sensor positions. However, this approach is brittle in practice. The estimates from these algorithms can be sensitive to random observation errors and errors in calibration, particularly in problem instances where the viewed scene is degenerate, such as one containing too few features, features infinitely far away, or features in an accidental geometric configuration. In image sequences where features enter and leave the sequence at a high rate, estimates from SFM can be sensitive to the number of tracked features and the rate of feature reextraction. Furthermore, motion estimates in this scenario will drift, just as estimates from inertial sensors will drift, because no one feature serves as a global reference of the position. In addition, image observations alone provide no mechanism for determining the global scale of the estimate.

To address these problems, we have developed two algorithms that estimate sensor motion, sparse scene structure, and other unknowns from both image and inertial measurements. The first is a batch algorithm that estimates the sensor position at the time of each image, the three-dimensional position of all point features visible in the image sequence, and other unknowns using all of the observations at once, as shown in

Figure 1(a). This algorithm is useful as a diagnostic, in understanding the best quality we can expect given a particular sensor configuration, vehicle motion, environment, and set of observations, and in measuring the inherent sensitivity of the estimate with respect to random observation errors. In Section 4 we give a detailed description of this algorithm.

The batch method requires that all of the observations be available before computation begins, and requires $O((f + p)^3)$ computation time in the number of images f and the number of tracked points p, so the batch method is not appropriate for online operation or for use with long observation sequences. Our recursive method, shown in Figure 1(b), incorporates image and inertial measurements as they arrive, and is applicable to long or "infinite" sequences. A detailed description of the recursive algorithm is given in Section 5.

As mentioned in the introduction, both the batch and recursive algorithms are suitable for motion estimation in environments where there are no known fiducials, which are objects whose appearance and/or position are known. An environment with fiducials whose appearance and position are both known is shown in Figure 2. As an aside, while our algorithms do not require known fiducials, the current implementation of our batch algorithm can exploit fiducials whose positions are known if they are present.

As we show in Section 7, the batch and recursive algorithms can each produce accurate estimates of the sensor motion and other unknowns by using both image and inertial measurements, even in some problem instances where the estimates produced using image or inertial measurements alone are grossly wrong. However, there are some difficulties in estimating motion from these sensors that we have not addressed. Both the batch and recursive algorithms can be affected by gross image feature tracking errors, despite the incorporation of both modalities. In addition, the recursive method inherits the difficulties associated with any use of the IEKF, particularly the a priori expectation of motion smoothness, approximation of the state estimate distribution by a Gaussian, and the linearization of measurements around uncertain state estimates. For these reasons, our recursive method is necessarily less robust to erratic sensor motion and measurement noise than the batch method. The recursive method as we describe it is also unable to "close the loop" when an image feature is lost and later reacquired, so the motion and scene structure estimates that it produces may not be topologically consistent in these situations. Our plans for addressing these issues are described in Section 11.2.4.

4. Batch Algorithm

4.1. Overview

Our batch algorithm for motion estimation from image and inertial measurements finds estimates of the sensor motion and other unknowns using the entire observation sequence from a camera, rate gyro, and accelerometer simultaneously. The algorithm estimates the sensor rig rotation, translation, and linear velocity at the time of each image; the three-dimensional position of each point observed in the image sequence; the gravity vector with respect to the world coordinate system; and the gyro and accelerometer biases.

More specifically, the algorithm uses Levenberg–Marquardt to minimize a combined image and inertial error function. Since Levenberg–Marquardt is widely used, we concentrate on the error function, and refer the reader to Press et al. (1992) for a discussion of Levenberg–Marquardt. The error function is

$$E_{combined} = E_{image} + E_{inertial} + E_{prior}. \tag{1}$$

The three components E_{image}, $E_{inertial}$, and E_{prior} of this error are described in the following subsections.

4.2. Image Error

The image error term E_{image} is

$$E_{image} = \sum_{i,j} D(\pi(C_{\rho_i,t_i}(X_j)), x_{ij}). \tag{2}$$

E_{image} specifies an image reprojection error given the six-degrees-of-freedom camera positions and three-dimensional point positions. In this error, the sum is over i and j, such that point j was observed in image i. x_{ij} is the observed projection of point j in image i. ρ_i and t_i are the camera-to-world rotation Euler angles and camera-to-world translation, respectively, at the time of image i, and C_{ρ_i,t_i} is the world-to-camera transformation specified by ρ_i and t_i. X_j is the world coordinate system location of point j, so that $C_{\rho_i,t_i}(X_j)$ is location of point j in camera coordinate system i.

π gives the image projection of a three-dimensional point specified in the camera coordinate system. For perspective images, we use the standard normalized perspective projection model:

$$\pi_{perspective}\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) = \begin{bmatrix} x/z \\ y/z \end{bmatrix}. \tag{3}$$

Equation (3) assumes that tracked image feature locations and their associated observation covariances have been converted from image coordinates to normalized coordinates prior to the minimization using the camera's known intrinsics, including the focal length, image center, and radial distortion. In our experiments, we have assumed the intrinsics model described by Heikkilä and Silvén (1997), and estimated the parameters of this model using a publicly available calibration program for each camera–lens combination.

All of the individual distance functions D are Mahalanobis distances. Reasonable choices for the associated observation covariances are uniform isotropic covariances or directional

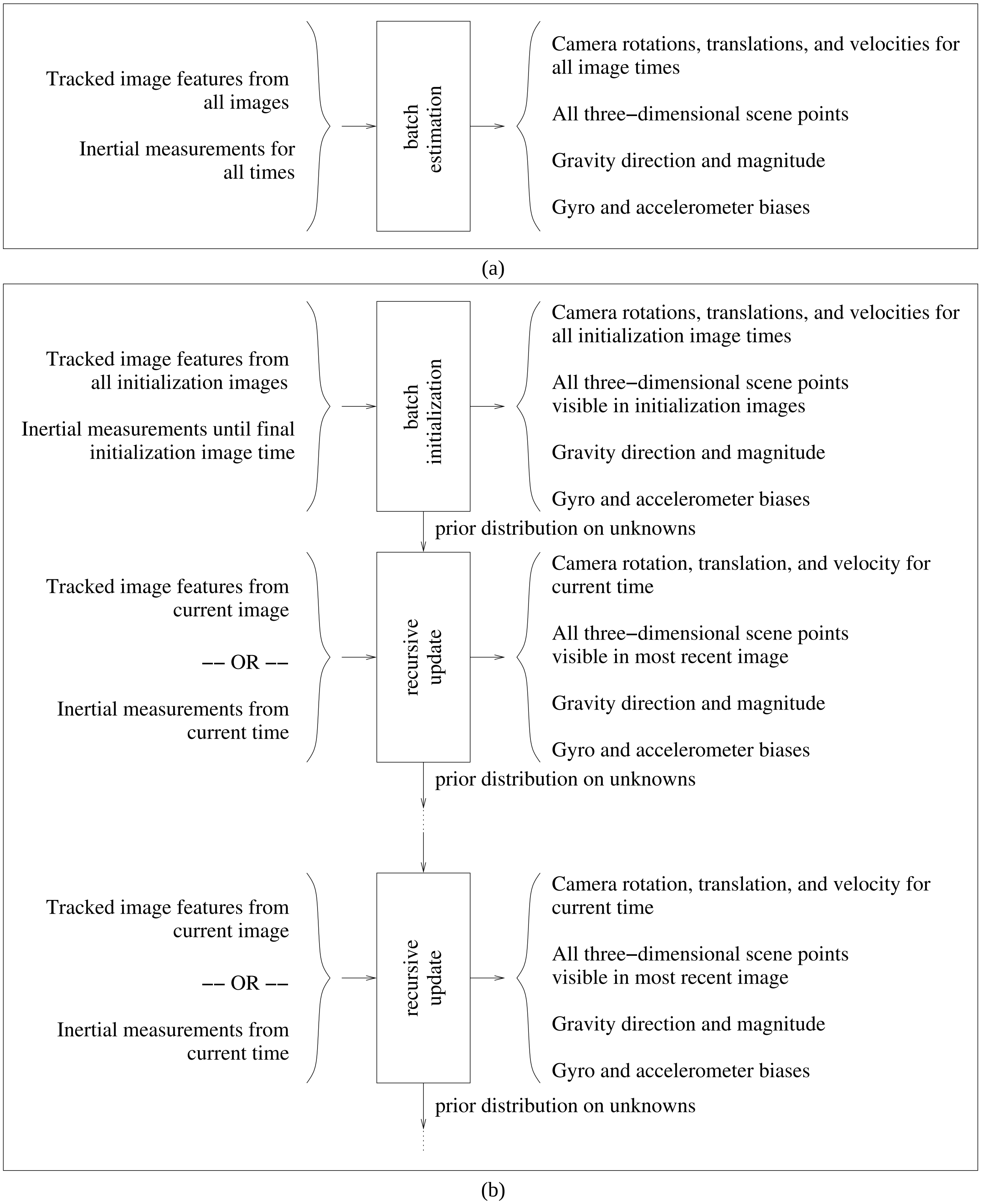

Tracked image features from all images
Inertial measurements for all times
batch estimation
Camera rotations, translations, and velocities for all image times
All three−dimensional scene points
Gravity direction and magnitude
Gyro and accelerometer biases
(a)
Tracked image features from all initialization images
Inertial measurements until final initialization image time
batch initialization
Camera rotations, translations, and velocities for all initialization image times
All three−dimensional scene points visible in initialization images
Gravity direction and magnitude
Gyro and accelerometer biases
prior distribution on unknowns
Tracked image features from current image
–– OR ––
Inertial measurements from current time
recursive update
Camera rotation, translation, and velocity for current time
All three−dimensional scene points visible in most recent image
Gravity direction and magnitude
Gyro and accelerometer biases
prior distribution on unknowns
Tracked image features from current image
–– OR ––
Inertial measurements from current time
recursive update
Camera rotation, translation, and velocity for current time
All three−dimensional scene points visible in most recent image
Gravity direction and magnitude
Gyro and accelerometer biases
prior distribution on unknowns
(b)

Fig. 1. The batch algorithm for estimating motion and other unknowns from image and inertial measurements, shown in (a), uses the entire observation history at once. The recursive algorithm, shown in (b), uses the batch algorithm on a short prefix of the observations to initialize the estimates and their prior distribution, and recursively updates the estimates when subsequent observations arrive.

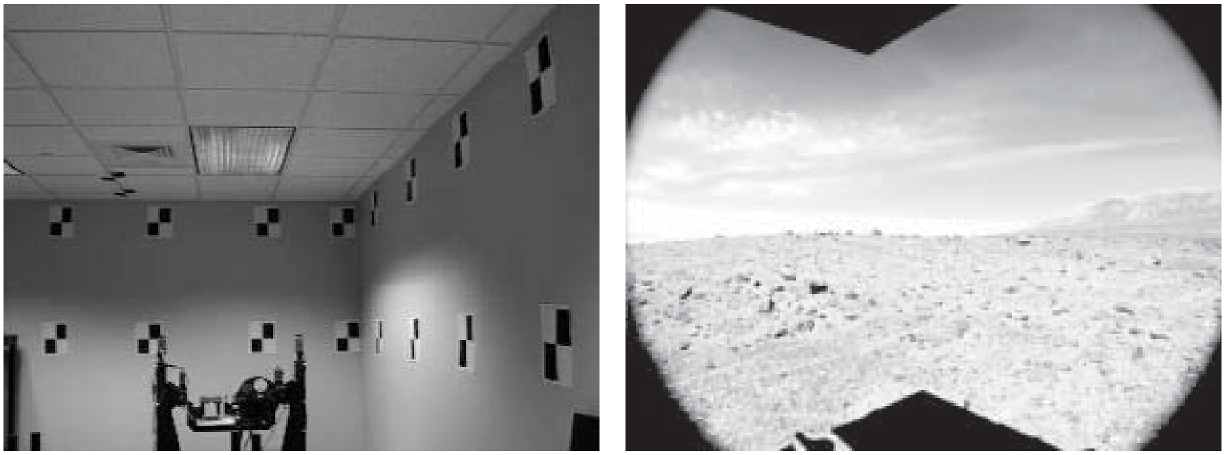


Fig. 2. Left: an environment containing fiducials whose appearance and positions are known. Right: an environment without known fiducials. Motion can be estimated in the latter case using automatic image feature tracking and the simultaneous estimation of sensor motion and sparse scene structure.

covariances determined using image texture in the vicinity of each feature (Brooks et al. 2001; Kanazawa and Kanatani 2001). In our experiments we have assumed isotropic covariances with $\sigma = 2$ pixels.

4.3. Inertial Error

The inertial error term is

$$\begin{aligned} E_{inertial} = & \\ & \sum_{i=1}^{f-1} D\left(\rho_i, I_\rho(\tau_{i-1}, \tau_i, \rho_{i-1}, b_\omega)\right) + \\ & \sum_{i=1}^{f-1} D\left(v_i, I_v(\tau_{i-1}, \tau_i, \rho_{i-1}, b_\omega, v_{i-1}, g, b_a)\right) + \\ & \sum_{i=1}^{f-1} D\left(t_i, I_t(\tau_{i-1}, \tau_i, \rho_{i-1}, b_\omega, v_{i-1}, g, b_a, t_{i-1})\right). \end{aligned} \tag{4}$$

$E_{inertial}$ gives an error between the estimated positions and velocities and the incremental positions and velocities predicted by the inertial data. Here, f is the number of images and τ_i is the time image i was captured. ρ_i and t_i are the camera rotation and translation at time τ_i, just as in the equation for E_{image} above. v_i gives the camera's linear velocity at time τ_i. g, b_ω, and b_a are the world coordinate system gravity vector, gyro bias, and accelerometer bias, respectively.

I_ρ, I_v, and I_t integrate the inertial observations to produce estimates of ρ_i, v_i, and t_i from initial values ρ_{i-1}, v_{i-1}, and t_{i-1}, respectively. We assume that the camera coordinate system angular velocity and world coordinate system acceleration remain constant over the intervals between consecutive measurements, whether those measurements are image or inertial measurements. Over such an interval $[\tau, \tau']$, I_ρ is defined as follows

$$I_\rho(\tau, \tau', \rho, b_\omega) = r(\Theta(\rho) \cdot \Delta\Theta(\tau' - \tau, b_\omega)), \tag{5}$$

where $r(\Theta)$ gives the Euler angles corresponding to the rotation matrix Θ, $\Theta(\rho)$ gives the rotation matrix corresponding to the Euler angles ρ, and $\Delta\Theta(\Delta t)$ gives an incremental rotation matrix:

$$\Delta\Theta(\Delta t, b_\omega) = \exp\left(\Delta t \ \mathrm{skew}(\omega)\right). \tag{6}$$

Here,

$$\mathrm{skew}(\omega) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{7}$$

where $\omega = (\omega_x, \omega_y, \omega_z)$ is the camera coordinate system angular velocity

$$\omega = \omega' + b_\omega \tag{8}$$

and ω' is the biased camera coordinate system angular velocity given by the gyro. Over $[\tau, \tau']$ I_v and I_t are given by the familiar equations

$$I_v(\tau, \tau', \ldots, b_a) = v + a(\tau' - \tau) \tag{9}$$

and

$$I_t(\tau, \tau', \ldots, t) = t + v(\tau' - \tau) + \frac{1}{2}a(\tau' - \tau)^2, \tag{10}$$

where a is the world coordinate system acceleration

$$a = \Theta(\rho) \cdot (a' + b_a) + g \tag{11}$$

and a' is the biased, camera coordinate system apparent acceleration given by the accelerometer.

Because inertial measurements are received at a higher rate than image measurements, the intervals $[\tau_{i-1}, \tau_i]$ between images span several inertial readings. To integrate the rotation, velocity, and translation over these larger intervals where the angular velocity and linear acceleration are not assumed constant, I_ρ, I_v, and I_t divide $[\tau_{i-1}, \tau_i]$ into the subintervals $[\tau, \tau']$ demarcated by the measurement times and sequentially apply eqs. (5), (9), and (10) over each subinterval.

The distances D in $E_{inertial}$ are Mahalanobis distances chosen to specify the relative importance of the image error terms in E_{image} and the rotation, linear velocity, and linear acceleration error terms in $E_{inertial}$. In the experiments in Section 7, we investigate the sensitivity of the batch algorithm to the choice of variances that define these distances.

Two additional comments about $E_{inertial}$ are in order. First, note that $E_{inertial}$ does not make any assumptions restricting the relative timing between image and inertial readings. In particular, image and inertial readings need not arrive at the same time, and can arrive at different rates, as long as each measurement is accurately timestamped. Secondly, a possible alternative formulation for $E_{inertial}$ is to use discrete differences to approximate the first and second derivatives of the estimated motion, and then require these derivatives to be consistent with the observed inertial measurements. However, this formulation requires that the durations between image times be small relative to the rate at which the derivatives change. Our formulation makes no such assumption, so our error function is suitable for cases where the duration between images is long.

4.4. Accelerometer Bias Prior Error

The gyro and accelerometer biases b_ω and b_a represent the angular rates and accelerations reported by the inertial sensors that correspond to zero angular velocity and zero acceleration. These values are not always zero, and are estimated as part of the minimization because the voltages produced by the sensors for zero angular rate and zero acceleration can differ with temperature and across sensor powerups.

Given observation sequences where the sensors undergo little change in orientation, gravity and accelerometer bias cannot be reliably distinguished from the observations, as eq. (11) shows. Therefore, we include an accelerometer bias term in our combined error, which reflects our expectation that the accelerometer voltage corresponding to zero acceleration is close to the precalibrated value. The bias prior term E_{prior} is

$$E_{prior} = f \cdot b_a^{\mathrm{T}} C_b^{-1} b_a. \tag{12}$$

As above, f is the number of images and b_a is the accelerometer bias. C_b is the accelerometer bias prior covariance.

In our experiments we have taken the accelerometer bias prior covariance to be isotropic with standard deviations of 0.5 m s^{-2}. This is roughly 1% of the 4 g range of our accelerometer.

4.5. Omnidirectional Projections

Recent omnidirectional cameras combine a conventional camera with a convex mirror that greatly expands the camera's field of view, typically to 360° in azimuth and 90–140° in elevation. Omnidirectional cameras are likely to produce better motion estimates than conventional cameras, because of the wide field of view, and because tracked image features are likely to be visible throughout a larger portion of the image sequence.

To exploit omnidirectional cameras in our framework, we set the projection operator π in Sections 4 and 5 to an omnidirectional projection model. Specifically, we use a projection model, described in Strelow et al. (2001a), that can accommodate camera–mirror combinations that are not single viewpoint (Chahl and Srinivasan 1997; Ollis, Herman, and Singh 1999) as well as the more common single viewpoint designs (Nayar 1997; Baker and Nayar 2001). In Section 8 we describe a motion estimation experiment using two omnidirectional cameras from the family described by Ollis , Herman, and Singh (1999). These cameras, along with example images produced by them, are shown in Figures 3 and 4.

Our projection model can also accommodate camera–mirror combinations where there is a known six-degrees-of-freedom misalignment between the mirror and camera, and Strelow et al. (2001b) describe a calibration procedure for determining the six-degrees-of-freedom transformation between the camera and mirror from one image of known three-dimensional calibration targets. In the experiments described in this paper we have not incorporated this precise mirror–camera calibration.

4.6. Minimization

As described in Section 4.1, the combined error function is minimized with respect to the six-degrees-of-freedom camera position ρ_i, t_i at the time of each image; the camera linear velocity v_i at the time of each image; the three-dimensional point positions of each tracked points X_j; the gravity vector with respect to the world coordinate system g; and the gyro and accelerometer biases b_ω and b_a.

Since the batch algorithm uses iterative minimization, an initial estimate is required, but the algorithm converges from a wide variety of initial estimates. For many sequences, an initial estimate that works well and requires no a priori knowledge of the motion or other unknowns is as follows.

- All camera positions (rotations and translations) are coincident with the world coordinate system.
- The point positions are initialized by backprojecting

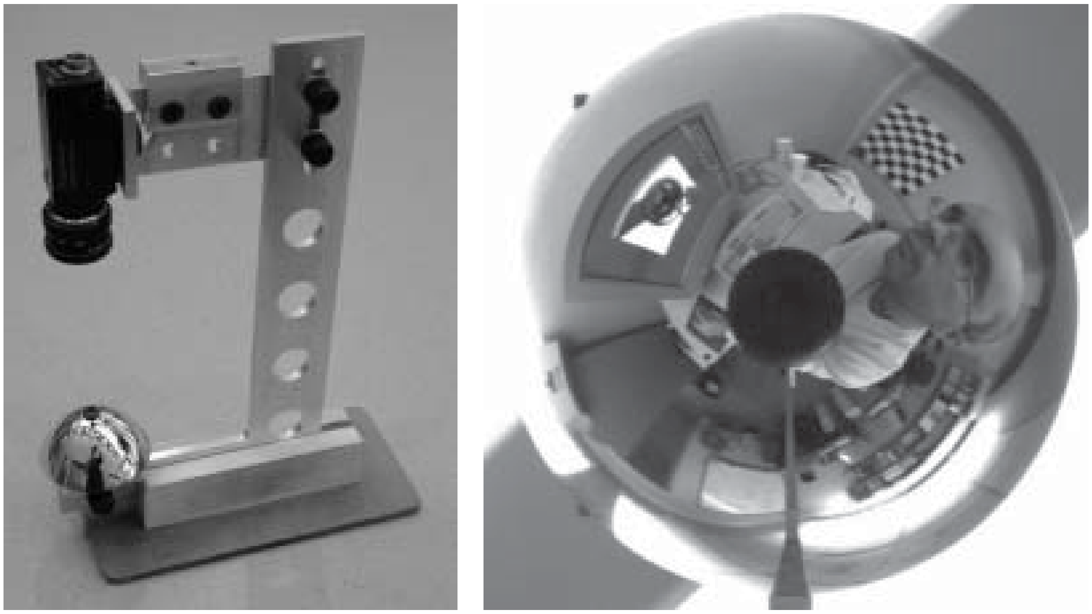


Fig. 3. Left: an omnidirectional camera consisting of a CCD camera attached to an equiangular mirror by a rig that allows the relative rotation and translation between the mirror and camera to be adjusted. Right: an omnidirectional image produced by this design.

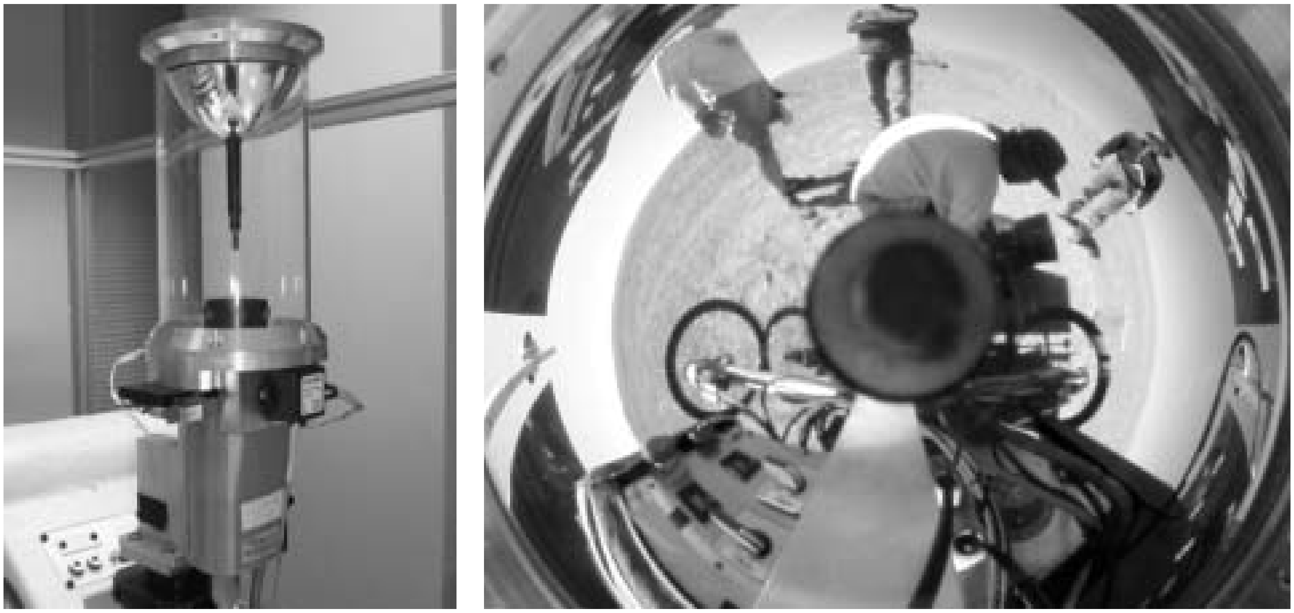


Fig. 4. Left: an omnidirectional camera consisting of an IEEE 1394 camera attached to an equiangular mirror by a rigid cylinder. The accelerometer and one rate gyro are visible, attached to the rig below the cylinder. Right: an omnidirectional image produced by this design.

the image position at which they first appear from the origin to a fixed distance in space.

- The velocities, gravity, and biases are initialized to zero.

In subsequent sections we have referred to this initial estimate as the "blind" initial estimate, and we show that in some cases the algorithm can converge from the blind estimate, even if the true motion is quite different from the blind estimate. For longer sequences, the recursive algorithm described in the next section can be used to generate an initial estimate close to the final estimate, making convergence more likely or reducing the number of iterations required for convergence.

As shown in Figure 1 and described further in Section 5.2, we initialize the state estimate distribution in our recursive algorithm by applying the batch image-and-inertial algorithm to a prefix of the observation sequence. However, if a recursive estimate is recommended as the initial estimate for the batch algorithm, how can the batch algorithm be used to initialize the recursive method? When the batch algorithm is used to initialize the recursive algorithm, we have used the blind estimate for the prefix as the initial estimate for the batch initialization. For the short observation sequences used in the batch initialization, the prefix motion is typically benign in that the camera positions throughout the prefix are close to the initial camera positions, and most of the points visible in the first image are visible throughout the prefix. In all of the experiments we have performed, including the experiments described in this paper, the batch algorithm has converged reliably from the blind estimate for prefixes including 20, 30, 40, 50, or 60 images. Of course, if the motion during the prefix is particularly erratic, the batch method may have difficulties converging and some different initialization should be substituted.

4.7. Image-Only Estimation

In perspective arm and omnidirectional arm experiments, described below in Sections 7 and 8, respectively, we report batch image-only motion estimates in addition to the estimates produced by the batch image-and-inertial algorithm described in this section. Our image-only motion estimates algorithm uses Levenberg–Marquardt to minimize E_{image} alone, with respect to the camera-to-world transformations ρ_i, t_i and the three-dimensional point positions X_j. This algorithm is the same as bundle adjustment (Wolf 1983) in photogrammetry or nonlinear SFM (Szeliski and Kang 1994) in computer vision.

4.8. Discussion

Both the batch algorithm described in this section and the recursive algorithm described in the Section 5 represent the three-dimensional camera orientation at the time of each image using Z–Y–X Euler angles (Craig 1989). We have adopted this representation because every choice of Euler angles represents a valid orientation, and therefore the Euler angles resulting from an unconstrained minimization step or Kalman measurement update always represent valid orientations. It is well known, however, that each orientation including a 90° rotation about Y is not represented by a unique choice of Z–Y–X Euler angles, but by a range of Z–Y–X Euler angles. This lack of a unique representation can lead to numerical problems in problem instances that include such camera orientations.

For these problem instances, quaternions would be a better choice for representing orientations. Since only unit quaternions correspond to valid orientations, a normalization or some other modification is required to enforce the unit constraint after an unconstrained minimization step or Kalman measurement update. However, unlike Euler angle representations, unit quaternions do not introduce numerical problems for special orientations. A few relevant, representative methods from the literature that use quaternions for representing orientation include the batch algorithm of Szeliski and Kang (1994) for estimating shape and motion from image measurements and the method of Huster and Rock (2001a, 2001b) for estimating motion relative to a single point from image and inertial measurements. A concise discussion on the relative merits of Euler angles and quaternions for motion estimation from images is given by McLauchlan (1999), who also considers a local representation that pairs a reference rotation with a small incremental rotation expressed using a minimal representation.

5. Recursive Algorithm

5.1. Overview

Our recursive method, described in this section, is an IEKF in which the image and inertial measurements are incorporated as soon as they arrive, in separate measurement update steps. This approach exploits the higher acquisition rate of inertial data to provide state estimate updates at the higher rate, and is more principled than possible alternatives, which include queuing the inertial data until the next image measurements are available, or assuming that image and inertial measurements are taken at the same time. Our application of the IEKF to this problem is described in Section 5.2, which gives our state vector, in Section 5.3, which describes our state estimate time propagation, and in Section 5.4, which gives our image and inertial measurement update steps.

Efficiency requires that the recursive method maintain estimates of, and joint error covariances between, the three-dimensional positions of only those points visible in the current image. This limitation requires a policy for removing tracked points that have been lost from the estimated state distribution, and for adding tracked points that have been newly acquired to the estimated state distribution. Our policies for incorporating new points and for removing lost points are described in Sections 5.5 and 5.6, respectively.

5.2. State Vector and Initialization

The state vector is

$$x(\tau) = \begin{bmatrix} \rho(\tau) \\ t(\tau) \\ X_{i,0} \\ \vdots \\ X_{i,p-1} \\ v(\tau) \\ b_\omega \\ g \\ b_a \\ \omega(\tau) \\ a(\tau) \end{bmatrix}. \tag{13}$$

Most of the components of this state vector are the same as those described for the batch method in Section 4. $\rho(\tau)$ and $t(\tau)$ are the Euler angles and translation specifying the camera-to-world transformation at time τ; $X_{i,0}, \ldots, X_{i,p-1}$ are the three-dimensional locations of the tracked points visible in the most recent image; $v(\tau)$ is the linear velocity at time τ expressed in the world coordinate system; and g, b_ω, and b_a are the gravity vector with respect to the world coordinate system, the gyro bias, and the accelerometer bias, respectively. The two components of the state not estimated by the batch algorithm, $\omega(\tau)$ and $a(\tau)$, are the camera coordinate system angular velocity and world coordinate system linear acceleration at time τ.

Those components of the state estimate distribution that can be estimated using the batch algorithm are initialized by applying the batch algorithm to a prefix of the observation sequence. The angular velocity and linear acceleration, which are not estimated by the batch algorithm, are initialized using the first gyro and accelerometer readings that follow the observation prefix used in the batch initialization.

5.3. State Propagation

We assume that the state $x(\tau)$ propagates according to

$$\dot{x}(\tau) = f(x(\tau)) + w(\tau) \tag{14}$$

where w is a zero mean Gaussian noise vector with covariance Q. The non-zero components of f are $\mathrm{d}t/\mathrm{d}\tau = v$, $\mathrm{d}v/\mathrm{d}\tau = a$, and

$$\frac{\mathrm{d}\rho}{\mathrm{d}\tau} = \frac{\mathrm{d}\rho}{\mathrm{d}\Theta(\rho)} \frac{\mathrm{d}\Theta(\rho)}{\mathrm{d}\tau}. \tag{15}$$

As in Section 4, $\Theta(\rho)$ is the camera-to-world rotation matrix specified by ρ. $\mathrm{d}\rho/\mathrm{d}\Theta(\rho)$ is a 3×9 matrix that can be computed from the definition of $\Theta(\rho)$, and $\mathrm{d}\Theta(\rho)/\mathrm{d}\tau$ is a 9×1, flattened version of

$$\Theta(\rho)\ \mathrm{skew}(\omega) \tag{16}$$

where skew(ω) is given by eq. (7). The noise covariance matrix Q is zero except for the 3×3 submatrices corresponding to ω and a, which are assumed to be isotropic.

Assuming that the true state propagates according to eq. (14), a state estimate mean $\hat{x}(\tau)$ can be propagated using

$$\dot{\hat{x}}(\tau) = f(\hat{x}(\tau)) \tag{17}$$

and a state estimate covariance $P(\tau)$ propagated using

$$\dot{P}(\tau) = F(\hat{x}(\tau))P(\tau) + P(\tau)F^{\mathrm{T}}(\hat{x}(\tau)) + Q, \tag{18}$$

where P is the error covariance estimate, F is the derivative of $f(\hat{x}(\tau))$ with respect to the state estimate $\hat{x}$, and Q is the noise covariance matrix. The non-zero blocks of F are $\partial^2\rho/\partial\tau\,\partial\rho$, which we compute numerically, and

$$\frac{\partial^2 t}{\partial\tau\,\partial v} = I_3 \tag{19}$$

$$\frac{\partial^2 v}{\partial\tau\,\partial a} = I_3 \tag{20}$$

$$\frac{\partial^2 \rho}{\partial\tau\,\partial\omega} = \frac{\mathrm{d}\rho}{\mathrm{d}\Theta(\rho)} \frac{\mathrm{dskew}(\omega)}{\mathrm{d}\omega}. \tag{21}$$

Here, $\mathrm{d}\rho/\mathrm{d}\Theta(\rho)$ and $\mathrm{dskew}(\omega)/\mathrm{d}\omega$ are flattened, 3×9 and 9×3 versions of the derivatives.

5.4. Measurement Updates

When image or inertial measurements are received, the state estimate mean and covariance are propagated from the previous measurement update time using eqs. (17) and (18), and then updated using an IEKF measurement update. For brevity, we concentrate here on the image and inertial measurement equations, and refer the reader to Gelb (1974) for a discussion of the IEKF measurement update.

The image measurement equation combines the projection equations for all of the points visible in the current image i:

$$\begin{bmatrix} x_{0,u} \\ x_{0,v} \\ x_{1,u} \\ x_{1,v} \\ \vdots \\ x_{p-1,u} \\ x_{p-1,v} \end{bmatrix} = \begin{bmatrix} \pi_u(C_{\rho(\tau),t(\tau)}(X_{i,0})) \\ \pi_v(C_{\rho(\tau),t(\tau)}(X_{i,0})) \\ \pi_u(C_{\rho(\tau),t(\tau)}(X_{i,1})) \\ \pi_v(C_{\rho(\tau),t(\tau)}(X_{i,1})) \\ \vdots \\ \pi_u(C_{\rho(\tau),t(\tau)}(X_{i,p-1})) \\ \pi_v(C_{\rho(\tau),t(\tau)}(X_{i,p-1})) \end{bmatrix} + n_v. \tag{22}$$

Here, $(x_{0,u}, x_{0,v})$, $(x_{1,u}, x_{1,v})$, …, $(x_{p-1,u}, x_{p-1,v})$ are the projections visible in the current image i. As in Section 4, π is the projection from a three-dimensional, camera coordinate system point onto the image; $C_{\rho(\tau),t(\tau)}$ is the world-to-camera transformation specified by the Euler angles $\rho(\tau)$ and translation $t(\tau)$; and $X_{i,j}$ is the three-dimensional, world coordinate system position of the jth point in the current image. n_v is a vector of zero mean noise, which we normally take to be isotropic with $\sigma = 2$ pixels.

The inertial measurement equation is

$$\begin{bmatrix} \omega' \\ a' \end{bmatrix} = \begin{bmatrix} \omega - b_{\omega} \\ \Theta(\rho)^{\mathrm{T}}(a - g) - b_{\alpha} \end{bmatrix} + n_i. \qquad (23)$$

The top and bottom component equations of eq. (23) are equivalent to eqs. (8) and (11), rearranged to given the biased angular velocity and biased linear acceleration. As before, ω' and a' are the camera system measurements from the rate gyro and accelerometer, respectively, and $\Theta(\rho)$ is the camera-to-world rotation specified by the Euler angles ρ. ρ, ω, b_ω, g, and b_a, and a are the same members of the state that we encountered in eq. (13). n_i is a vector of Gaussian noise.

5.5. Newly Acquired Points

To generate an initial mean and covariance for the state estimate, we use the batch method described in Section 4. This method properly incorporates points that are seen at the beginning of the observation sequence into the estimate mean and covariance. To incorporate points that become visible after the batch initialization into the state estimate, we have to address three issues:

1. determining which new points to incorporate into the state estimate;

2. finding a mean for new points that are to be incorporated into the state estimate;

3. finding a variance for new points that are to be incorporated into the state estimate, and the covariances that relate them to the estimates of the other state components and to the other new points.

In this subsection we describe our approach to each of these issues, with emphasis on the most difficult issue, correct computation of the variances and covariances.

To address the first issue, we delay the incorporation of newly visible points into the state estimate until the point's three-dimensional position can be computed accurately relative to the camera positions from which it was visible, i.e., until its variance relative to the most recent camera position is below some threshold. This is to prevent numerical problems and reduce bias in the filter. The number of images required to achieve a sufficiently low variance in the point's position depends on the translation between the images.

Finding a mean for a point that has become visible after the batch initialization is straightforward. We use the camera positions estimated by the filter for the images in which the point was visible to triangulate the point's position.

Determining the new state covariance matrix is more complex, and must consider the following interrelated requirements.

1. The mean and covariance estimates produced by the recursive estimation should approximate those produced by a batch estimation as closely as allowed by the choice of the filter state and the filter's Gaussian prior assumption. This requirement is a fundamental design goal of recursive estimation algorithms, and requirements 2, 3, and 4 below are satisfied if this requirement is satisfied.

2. The state estimate distribution (i.e., mean and covariance) that results from the point initialization should not be too optimistic, which would weight the observations used in the point initialization more than those used in subsequent measurement updates, or too conservative, which would weight the observations used in the point initialization less than those used in subsequent measurement updates. This is important to ensure that the resulting estimates will be consistent with all of the observations, insofar as the observations allow. This also ensures that the available observations are best utilized, which is important in sequences where each point is only visible in a short subsequence of the image sequence.

3. The resulting filter should accurately estimate the covariances of the camera position, camera velocity, and point positions in the world coordinate system so that these can be monitored in critical applications. This is particularly important for long sequences, in which these variances will necessarily increase with time as points enter and leave the camera's field of view.

4. The variance of points relative to the current camera position, which is generally lower than the variance of the points relative to the world coordinate system, should be accurately estimated. This is important if the output of the filter will be used to generate image coordinate system priors for tracking the points in subsequent images.

To satisfy these requirements as closely as possible, we have adapted the method described by Smith, Self, and Cheeseman (1990) for integrating newly visible landmarks into EKFs in SLAM applications. The motivation of the method of Smith, Self, and Cheeseman is to properly incorporate sensor coordinate system measurements and covariances into global coordinate system estimates and covariances given the covariance associated with the sensor position, and to allow the sensor coordinate system measurement and covariance to be reextracted from the global coordinate system measurement and covariance. As described in Section 2.4, SLAM applications typically use active range sensors, which provide measurements of both the range and bearing (i.e., the point's full three-dimensional location) in the sensor's coordinate system. In our application, which uses cameras, only the bearing is available from any one sensor measurement. In the rest of this subsection we briefly describe the method of Smith, Self, and Cheeseman, describe how we have applied it to the bearings-only case, and discuss the relative advantages and disadvantages of this approach versus some possible alternative approaches.

The method of Smith, Self, and Cheeseman operates as follows. Suppose that:

- x is the filter's state estimate before the incorporation of the new point estimate, and that $C(x)$ is the covariance estimate for x produced by the most recent measurement update step;
- z_w is the world coordinate system point corresponding to the sensor coordinate system point z_c;
- $g(x, z_c)$ is the rigid transformation that maps z_c to z_w;
- G_x and G_{z_c} are the derivatives of g with respect to x and z_c, respectively, evaluated at the current estimates of x and z_c.

Then, the method of Smith, Self, and Cheeseman transforms the sensor coordinate system covariance $C(z_c)$ into a world coordinate system covariance $C(z_w)$, and establishes a cross-covariance $C(x, z_w)$ between x and z_w, using

$$C(z_w) = G_x C(x) G_x^{\mathrm{T}} + G_{z_c} C(z_c) G_{z_c}^{\mathrm{T}} \tag{24}$$

$$C(x, z_w) = G_x C(x). \tag{25}$$

The new point can then be incorporated into the state estimate by augmenting x with z_w, and $C(x)$ with $C(z_w)$ and $C(x, z_w)$.

To adapt this method to the bearings only case, we assume that the camera position estimates corresponding to the initialization images in which the point was visible are correct with respect to each other. Mean and covariance estimates for the point's position relative to the most recent camera position can be computed under this assumption, effectively producing a "virtual range device" that can be used in the method of Smith, Self, and Cheeseman. The mean and variance of this virtual device's position with respect to the world coordinate system are that of the most recent camera position.

This method correctly accounts for the variances in the image observations used in the point initialization and the estimated uncertainty in the current camera estimate, but not for any relative error between the recent estimated camera positions. As a result, we have found that, in cases where relative error between the camera estimates is small, this method produces covariances that are quite close to those produced by a batch estimation, satisfying requirement 1 and requirements 2, 3, and 4, which follow from it. However, in those cases where there is a significant relative error between the recent camera estimates (e.g. because of point feature mistracking), the new point will be initialized with an overly optimistic covariance, and subsequent state estimates will be contaminated by this error.

This approach is superior to two problematic strategies that we have tried. In the first, camera system variances for the newly visible points are computed using the error variances associated with the points' image feature observations, and rotated into the world coordinate system using the current camera-to-world rotation estimate. This strategy correctly reflects the accuracy of the points' estimated positions relative to the current camera position (requirement 4), but does not capture the potentially high error in the points' estimated positions relative to the world coordinate system (requirement 3). This potentially large error in the points' world coordinate system position estimates is inevitable, since the error in the camera positions used to estimate the points' positions will grow large for long image sequences. After these inadvertently low variances with respect to the world coordinate system have been assigned to the points, all of the filter's error variances will become corrupted, since subsequent camera positions will be estimated with respect to points that appear to have a low world coordinate system variances, and will therefore be assigned low variances in turn (violating requirement 1).

In the second, fixed variances, the variances described in the previous paragraph, or some other reasonable camera coordinate system variances are adopted, but inflated in an attempt to reflect the potential inaccuracy in the points' estimated positions relative to the world coordinate system. This strategy is poor for the following reasons. First, this strategy does not reflect the accuracy of the points' estimated positions relative to the current camera position, which may be useful information, e.g. for tracking the point's position in subsequent images (requirement 4). Secondly, for any fixed inflation factor, this choice also may not reflect the point's inaccuracy in the world coordinate system, which may grow arbitrarily large given long sequences (requirement 3). Thirdly, this method effectively jettisons all of the information about the points' positions provided by the points' initial image feature observation variances, so the estimated three-dimensional point positions may become inconsistent with the initial views of the point (requirement 2).

As mentioned above, our strategy cannot model the relative errors between the successive camera position estimates that comprise the "virtual range device". An alternative approach that models the relative errors between successive camera positions is the VSDF described by McLauchlan (1999). The VSDF is a hybrid batch-recursive approach rather than a Kalman filter, so this modeling comes at the cost of increased computation. The batch method we describe in Section 4 could be adapted for hybrid batch-recursive operation within the VSDF framework. For use within the Kalman filter framework, however, we recommend the adaptation of the method of Smith, Self, and Cheeseman that we have described in this subsection.

5.6. Lost Points

To limit the filter state dimension as new points become visible and are added to the state, our recursive method removes the three-dimensional positions of points lost by the tracker from the state by removing the corresponding entries of the mean and covariance estimates, and leaving all other mean and covariance values unchanged. This is equivalent to marginalizing over the removed points' state components.

5.7. Image-Only Estimation

In Section 10 we describe both recursive image-and-inertial estimates and recursive image-only estimates. Our recursive image-only algorithm is similar to the recursive image-and-inertial algorithm described in this section, with the following differences.

- Only the camera rotation, the camera translation, and the three-dimensional positions of the currently visible points are included in the state.
- The state propagation assumes slowly changing camera rotation and translation rather than slowly changing angular velocity and linear acceleration.
- The state estimate is initialized using the batch image-only algorithm rather than the batch image-and-inertial algorithm.
- There is no inertial measurement update.

The image measurement update step, the method for incorporating newly visible points, and the policy for dropping lost points from the state are the same as those for the recursive image-and-inertial algorithm.

5.8. Discussion

As we mentioned in Section 5.5, a common goal of recursive algorithm design is to incorporate measurements as they become available, while approximating the state mean and covariance estimates that would be produced by a batch algorithm given all of the measurements at once. For instance, the recursive image-only algorithm described in Section 5.7 above produces estimates that closely match those of the batch image-only algorithm described in Section 4.7. This is particularly true as the time propagation variances, which specify how close we expect camera positions at adjacent times to be, are allowed to grow, easing the assumption of motion smoothness, which the batch method does not incorporate.

Our batch and recursive image-and-inertial algorithms are not as closely related in this way. For instance, the batch algorithm estimates the sensor position only at the time of each image, while the recursive algorithm estimates the sensor position at the time of each image and each inertial measurement.

More importantly, the multirate design of the recursive algorithm implicitly requires the assumption of smoothness in the angular velocity and linear acceleration across measurement times to exploit inertial and image measurements acquired at different times. Consider, for example, an inertial measurement followed by an image measurement. If the angular velocity and linear acceleration time propagation variances are high, as required in scenarios with erratic motion, then the state prior covariance that results from the time propagation between the inertial and image measurement times grows quickly and only loosely constrains the image measurement step estimate. So, the filter is free to choose an estimate at the time of the image measurement that is quite different than the state estimate that resulted from the inertial measurement step. On the other hand, motion smoothness is not a critical assumption for combining measurements taken at different image times, as it is in the recursive image-only filter, because these estimates are related through the three-dimensional positions of the points observed at both times.

Our batch algorithm, including the integration functions I_ρ, I_v, and I_t that integrate inertial measurement between image times, neither requires nor exploits such an assumption, so the batch algorithm is stronger than the recursive algorithm in the presence of erratic motion. In our experience, robustness to erratic motion is more valuable in practice than the assumption of motion smoothness, and Tomasi and Kanade (1992) drew a similar conclusion in the context of SFM. Increasing the robustness of the recursive image-and-inertial algorithm to erratic motion is a promising direction for future work.

6. Experiments Overview

6.1. Overview

In the remainder of the paper we present the results from a suite of experiments in which we have estimated sensor motion and other unknowns from image and inertial measurements. The camera configuration, observations, and results for each of these experiments are described in detail in Sections 7, 8, 9, and 10. In this section, we give a high-level overview of the experiments (Section 6.2), a description of the inertial sensor configuration (Section 6.3), and the motion error metrics used to evaluate the accuracy of our estimates (Section 6.4). The inertial sensor configuration and the motion error metrics are the same across the experiments described in the subsequent sections.

6.2. Experiments

A high-level overview of the experiments is as follows.

6.2.1. Perspective Arm Experiment

The sensor rig includes a perspective camera and the inertial sensors and is mounted on a robotic arm that undergoes a known, preprogrammed motion. For each of the batch image-and-inertial, batch image-only, and recursive image-and-inertial algorithms, we explore the estimation accuracy as the estimation parameters, the initial estimates, and the speed of the motion vary.

6.2.2. Omnidirectional Arm Experiments

Three omnidirectional experiments were performing by mounting our two omnidirectional rigs and inertial sensors on the same arm used in the perspective arm experiment. Estimates were again generated using the batch image-and-inertial, batch image-only, and recursive image-and-inertial algorithms. In each experiment the motion is similar to that in the perspective arm experiment, so the relative merits of omnidirectional cameras and inertial sensors for motion estimation can be investigated.

6.2.3. Perspective Crane Experiment

The sensor rig includes a conventional camera and the inertial sensors, and is mounted on the platform of a robotic crane that can translate within a workspace of about $10 \times 10 \times 5$ m^3. An estimate was generated using the recursive image-and-inertial algorithm.

6.2.4. Perspective Rover Experiments

The sensor rig includes a conventional camera and the inertial sensors, and is mounted on a ground rover that traverses approximately 230 m. Estimates were generated using the recursive image-only algorithm and the recursive image-and-inertial algorithm.

Figure 5 shows the overall organization of the suite of experiments in more detail.

6.3. Inertial Sensors

As mentioned in the introduction, our approach to a robust system for motion estimation is to augment motion estimation from image measurements with data from small, low-cost inertial sensors. To this end, we have used the same configuration of inertial sensors in all of our experiments, consisting of the following.

- Three single-axis, orthogonally mounted CRS04 rate gyros from Silicon Sensing Systems, which measure up to ±150 degrees per second. Each of these gyros has approximate dimensions $1 \times 3 \times 3$ cm^3, weighs 12 g, and has a single unit cost of approximately $300.
- A Crossbow CXL04LP3 three-degrees-of-freedom accelerometer, which measures up to ±4 g. This accelerometer has approximate dimensions $2 \times 4.75 \times 2.5$ cm^3, weighs 46 g, and has a single unit cost of approximately $300.

We capture the voltages from the rate gyros and the accelerometer at 200 Hz using two separate Crossbow CXLDK boards.

The three gyro voltage-to-rate calibrations were determined using a turntable with a known rotational rate. The accelerometer voltage-to-acceleration calibration was performed using a field calibration that accounts for non-orthogonality between the individual x, y, and z accelerometers. The gyro and accelerometer measurement error variances were found by sampling the sensors while they were kept stationary. The fixed gyro-to-camera and accelerometer-to-camera rotations differed between experiments but in each case were assumed known from the mechanical specifications of the mounts.

6.4. Motion Error Metrics

The simultaneous estimation of camera motion and scene structure from images alone recovers estimates only up to a scaled rigid transformation. That is, applying the same scaling and rigid transformation to all of the camera and point estimates produces new estimates that explain the observed data as well as the original estimates. With the incorporation of accelerometer data, it is possible to recover the global scale and two components of the absolute rotation, but with our inexpensive accelerometer we expect the global scale to be only roughly correct.

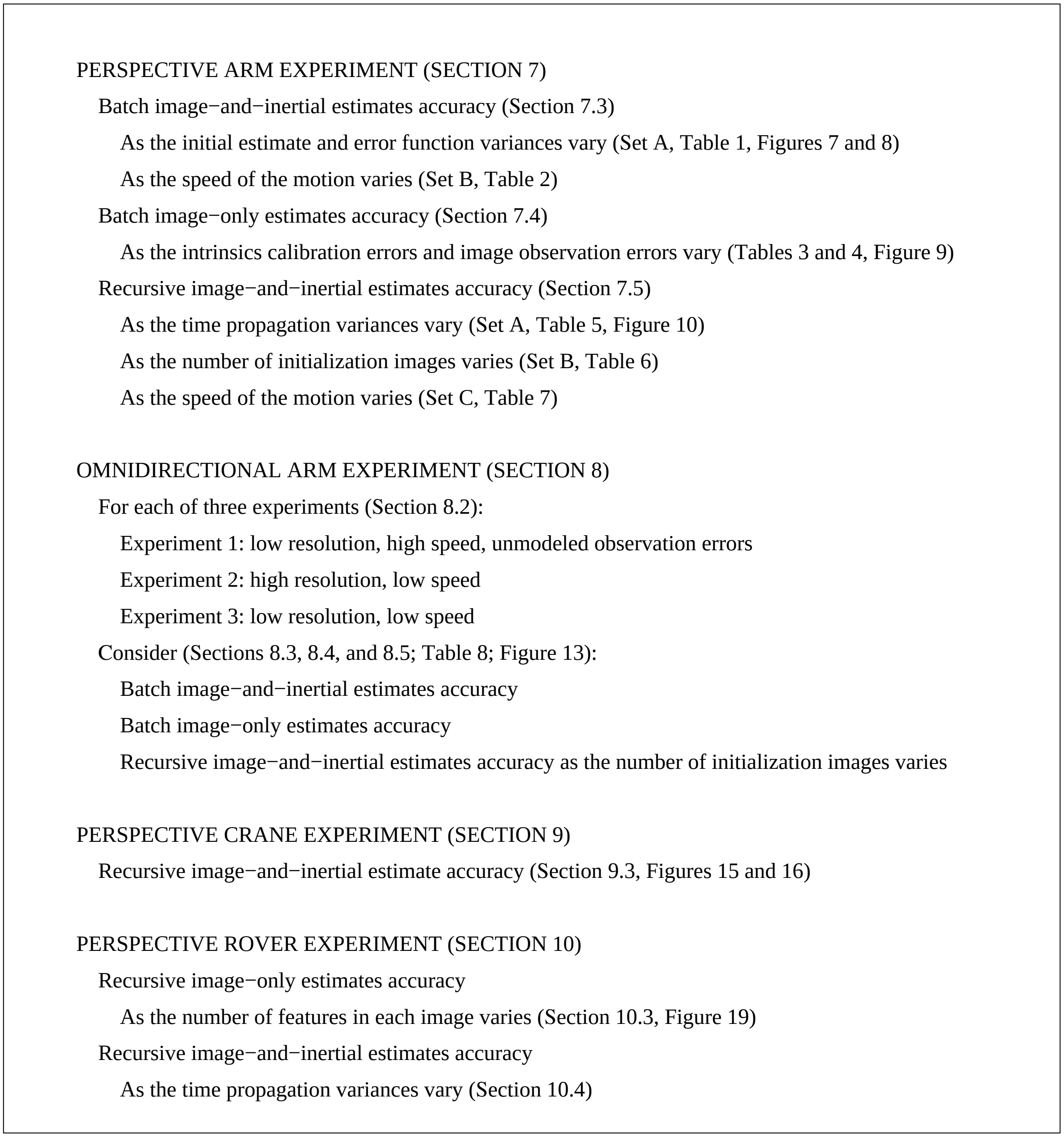
Therefore, we have transformed our recovered estimates into the ground truth coordinate system using a scaled rigid transformation before computing motion rotation and translation errors. This transformation has the form

$$t_g = sRt_e + t, \tag{26}$$

where t_g and t_e are camera translation in the ground truth and estimate coordinate systems, respectively, and s, R, and t are the scale, rotation, and translation of the scaled rigid transformation.

We have used the scaled rigid transformation that best aligns the estimated camera translations with the ground truth translations, which can be found in closed form using the absolute orientation method described by Horn (1987). An alternative is to align the estimate of the initial camera position with the ground truth initial camera position. However, this approach will not accurately capture the global accuracy of a motion estimate.

The rotation error that we report is the absolute rotation angle that aligns the estimated, transformed three-dimensional camera rotation with the ground truth rotation. This angle is the angle component of the angle-axis representation of



PERSPECTIVE ARM EXPERIMENT (SECTION 7)
Batch image−and−inertial estimates accuracy (Section 7.3)
As the initial estimate and error function variances vary (Set A, Table 1, Figures 7 and 8)
As the speed of the motion varies (Set B, Table 2)
Batch image−only estimates accuracy (Section 7.4)
As the intrinsics calibration errors and image observation errors vary (Tables 3 and 4, Figure 9)
Recursive image−and−inertial estimates accuracy (Section 7.5)
As the time propagation variances vary (Set A, Table 5, Figure 10)
As the number of initialization images varies (Set B, Table 6)
As the speed of the motion varies (Set C, Table 7)

OMNIDIRECTIONAL ARM EXPERIMENT (SECTION 8)
For each of three experiments (Section 8.2):
Experiment 1: low resolution, high speed, unmodeled observation errors
Experiment 2: high resolution, low speed
Experiment 3: low resolution, low speed
Consider (Sections 8.3, 8.4, and 8.5; Table 8; Figure 13):
Batch image−and−inertial estimates accuracy
Batch image−only estimates accuracy
Recursive image−and−inertial estimates accuracy as the number of initialization images varies

PERSPECTIVE CRANE EXPERIMENT (SECTION 9)
Recursive image−and−inertial estimate accuracy (Section 9.3, Figures 15 and 16)

PERSPECTIVE ROVER EXPERIMENT (SECTION 10)
Recursive image−only estimates accuracy
As the number of features in each image varies (Section 10.3, Figure 19)
Recursive image−and−inertial estimates accuracy
As the time propagation variances vary (Section 10.4)

Fig. 5. The organization of the experiments.

the relative rotation. The reported translation error is the distance in three-space between the estimated, transformed camera translation and the ground truth camera translations. For both rotation and translation we report average and maximum errors.

We also report the error in global scale between the estimated and ground truth motion. Specifically, we report, as a percentage

$$\text{scale error} = 1/s - 1 \tag{27}$$

where s is the transformation scale in eq. (26). So, the scale error is greater than zero if the scale of the estimated motion is larger than that of the ground truth motion, and less than zero otherwise.

7. Perspective Arm Experiment

In our first experiment we mounted a perspective camera and the inertial sensors on a robotic arm to produce image and

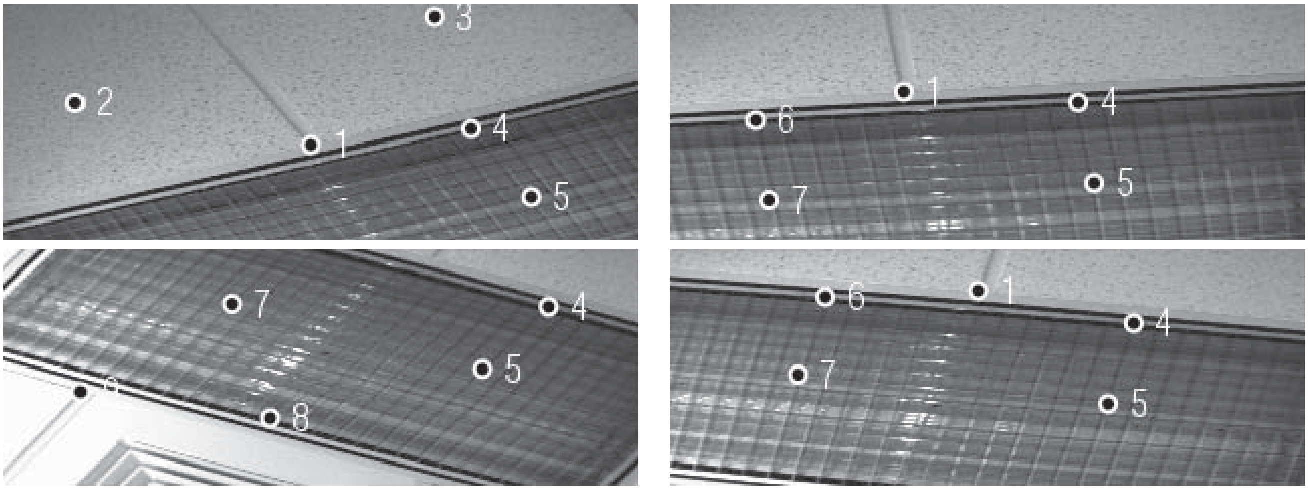


Fig. 6. Images 16, 26, 36, and 46 from the 152 image sequence in the perspective arm experiment, with the tracked features overlaid, are shown clockwise from the upper left. As described in Section 7.1, the images are one field of an interlaced image, so their height is half that of the full images.

inertial observations from a known motion. We used the resulting observation sequence and ground truth motion information to investigate the relative performance of the batch image-and-inertial, batch image-only, and recursive image-and-inertial algorithms, as the estimation parameters, initial estimates, and the speed of the motion varied.

In Sections 7.1 and 7.2 we have described the sensor configuration and observations, respectively, for this experiment. Sections 7.3, 7.4, and 7.5 describe the estimates produced by the batch image-and-inertial, batch image-only, and recursive image-and-inertial algorithms for these observations. Because these sections include a large number of estimates, we have included a concise summary of the key results in Section 7.6.

7.1. Camera Configuration

We used a conventional frame grabber to capture images at 30 Hz from a Sony XC-55 industrial vision camera paired with a 6 mm lens. The camera exposure time was set to 1/200 s to reduce motion blur. To remove the effects of interlacing, only one field was used from each image, producing 640 × 240 pixel images. As mentioned in Section 4, we have assumed the camera intrinsics model described by Heikkilä and Silvén (1997). This calibration naturally accounts for the reduced geometry of our one-field images.

7.2. Observations

To perform experiments with known and repeatable motions, the rig was mounted on a Yaskawa Performer-MK3 robotic arm, which has a maximum speed of 3.33 m s^{-1} and a payload of 2 kg. The programmed motion translates the camera x, y, and z through seven pre-specified control points, for a total distance traveled of about 3 m. Projected onto the (x, y) plane, these points are located on a square, and the camera moves on a curved path between points, producing a clover-like pattern in (x, y). In z, the camera alternates 0.3 m between high and low positions at consecutive control points. The camera rotates through an angle of 270° about the camera's optical axis during the course of the motion.

The observation sequence consists of 152 images and approximately 860 inertial readings, each consisting of both a gyro and accelerometer voltage. 23 features were tracked through the image sequence, but only five or six appear in any one image. Points were tracked using the Lucas–Kanade algorithm (Lucas and Kanade 1981), but because the sequence contains repetitive texture and large interframe motions, mistracking was common and was corrected manually. Four example images from the sequence, with tracked points overlaid, are shown in Figure 6. In this experiment, as in all the following experiments, we have assumed that the observation error associated with each tracked point position is isotropic with variance $(2.0\,\text{pixels})^2$.

7.3. Batch Image-and-Inertial Estimates

We computed two sets of estimates for the motion and other unknowns using the batch image-and-inertial algorithm:

- Set A explores the accuracy of the motion estimates versus ground truth as the quality of the initial estimate and the variances that specify the inertial error function $E_{inertial}$ vary;

- Set B explores the accuracy of the motion estimates versus ground truth as the speed of the motion and the variances that specify the inertial error function $E_{inertial}$ vary.

In this subsection we describe the convergence of the batch algorithm and the error in the resulting estimates versus ground truth for each of these estimates.

7.3.1. Set A

For the estimates in Set A, the rotation, translation, and velocity error variances that specify the Mahalanobis distances D in $E_{inertial}$ (described in Section 4.3) were always chosen to be equal to each other, and varied together from 10^{-8} to 10^{-3}. For each of these inertial error term variance choices, the batch algorithm was run starting from each of four different initial estimates, as follows.

- An "accurate" estimate produced by the recursive image-and-inertial algorithm.
- A "poor" (i.e., qualitatively correct but not otherwise accurate) estimate produced by the recursive image-and-inertial algorithm.
- A "failed" (i.e., not qualitatively correct) estimate produced by the recursive image-and-inertial algorithm.
- A "blind" estimate produced using the default initialization method described in Section 4.6. This initialization does not use the recursive method or any a priori knowledge of the unknowns.

The accurate, poor, and failed recursive estimates were generated using increasingly inappropriate choices of the recursive algorithm's angular velocity and linear acceleration propagation variances. We consider the effect of varying these propagation variances on the recursive algorithm in more detail in Section 7.5, which is devoted to the accuracy of the recursive algorithm for this data set.

Table 1 summarizes the results for Set A, including the number of iterations required for convergence and the error metrics for each combination of the $E_{inertial}$ error function variances and initial estimate quality. For each choice of the inertial error variances, the algorithm converged in a few iterations from the accurate and poor initial estimates, and failed to converge from the failed initial estimate. Figure 7 shows the x, y translation estimate from the batch image-and-inertial algorithm assuming inertial error variances of 10^{-5}, along with the accurate and poor initial estimates used in the minimization.

The algorithm also converged from the blind initial estimate for most choices of the inertial error variances, but required more iterations than convergence from the accurate or poor recursive estimates. In those cases where the algorithm converged from the blind initial estimate, the convergence was in spite of a large difference between the blind estimate and the final estimate. For example, the camera positions in the final estimate span 270° in rotation about z, whereas all of the camera positions in the blind initial estimate are zero. This is a much larger range of convergence than that found for batch image-only estimation by Szeliski and Kang (1994), who report failure to converge for spans in rotation above 50°.

The improved convergence above some inertial error variance threshold is an effect we see in many data sets, and presumably occurs because the inertial error function has a simpler topography than the image error function. In each case where the algorithm converged, it converged to the same estimates for a particular choice of the inertial error variances. In the table, this is reflected by the identical error metrics for each choice of the variances. Those cases where the method failed to converge from the blind estimate are indicated in the table by "N/A".

The rotation, translation, and scale errors versus ground truth were low across a wide range of the inertial error variances, and the trend in the translation errors as a function of the inertial error variances is shown in Figure 8. Variances between 10^{-8} and 10^{-4} all produced strong estimates. The estimates that result from the 10^{-3} variances, which weight the inertial measurements the least relative to the image measurements, are weak and show some of the problems inherent in estimating the motion from image measurements only. We discuss the problems with image-only estimation for this data set in Section 7.4.

7.3.2. Set B

As mentioned at the beginning of this subsection, we also considered a second set of estimates, Set B, which were produced by varying the speed of the motion and the inertial error term variances. More specifically, we artificially slowed the image and inertial observation sequences described in Section 7.2 by factors $f = 2$, $f = 4$, and $f = 8$ using the following four-step procedure.

1. Increase the image, gyro, and accelerometer timestamps by a factor of f.
2. Introduce $f - 1$ new interpolated inertial readings between each adjacent pair of original inertial measurements.
3. Reduce each of the observed gyro angular velocities by a factor of f and the observed accelerometer apparent accelerations by a factor of f^2.
4. Add Gaussian random noise to the scaled inertial measurements produced by step 3 to bring the measurement noise level back to that of the original observations.

This method works well for both the gyro and accelerometer measurements in this data set, subject to the following caveats.

- The frequency of the image observations is reduced by a factor f as well as the motion speed.

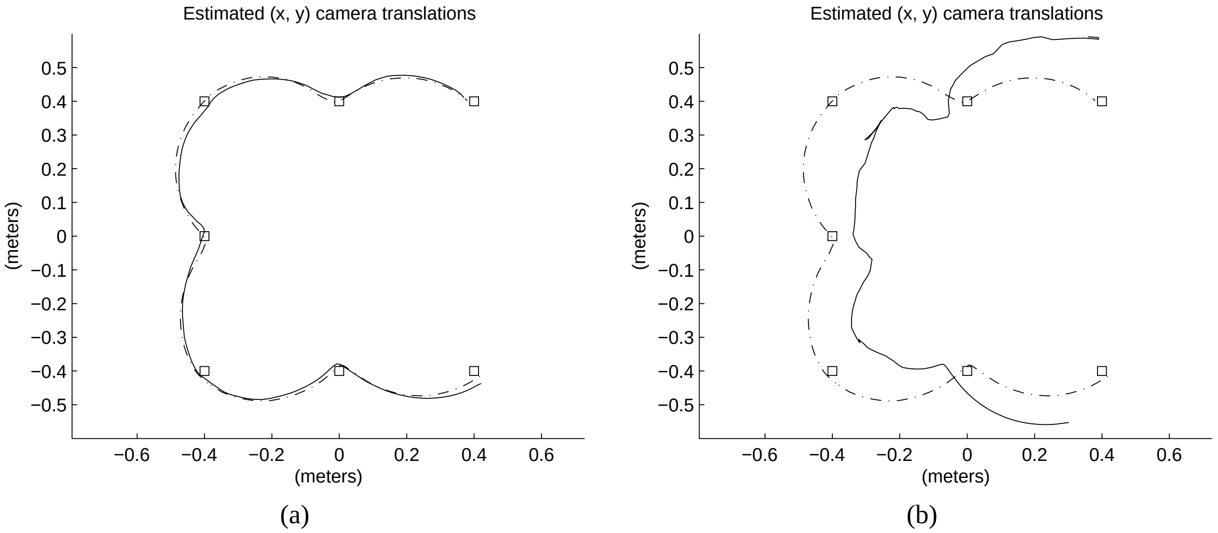


Fig. 7. The x, y translation estimates generated by the batch image-and-inertial algorithm for the perspective arm experiment, assuming inertial error variances of 10^{-5}. Left: the final estimate and the accurate initial estimate are shown as the dash-dotted and solid lines, respectively. The x, y locations of the seven known ground truth points are shown as squares. Right: the final estimate and the poor initial estimate are shown as the dash-dotted and solid lines, respectively. The known ground truth points are again shown as squares. The accurate and poor initial estimates were generated using the recursive image-and-inertial algorithm with different choices for the propogation variances.

- Since the effects of acceleration, accelerometer rotation, accelerometer bias, and gravity on the accelerometer measurements cannot be separated a priori, we have chosen to apply a simple scaling to both the gyro and accelerometer measurements in step 3 above. So, the effective bias and gravity in the resulting accelerometer measurements have magnitudes that are $1/f^2$ those of the original bias and gravity.

- The accelerometer measurements in this data set contain high-frequency vibrations with magnitudes larger than the accelerometer noise level. So, interpolating smoothly between two measurements that result from vibration does not completely reflect the nature of the original motion.

For each inertial error variance between 10^{-6} and 10^{-4} and for each slowdown factor $f = 2, 4, 8$, we ran the batch image-and-inertial algorithm from an initial estimate generated using the recursive image-and-inertial algorithm. The performance of the recursive image-and-inertial algorithm on these slowed observation sequences is described in Section 7.5. The error metrics and number of iterations required for convergence for each of these estimates are given in Table 2. The quality of the estimates degrades and becomes more sensitive to the inertial variance choices as the slowdown factor increases, and we would expect this trend to continue as the slowdown factor increases further. However, the rate of degradation for the best error variance choice is reasonable.

7.4. Batch Image-Only Estimates

To highlight the complementary nature of the image and inertial observations, we generated an estimate of the camera positions and three-dimensional point positions from image measurements only using the method described in Section 4.7. The initial estimate used was the batch image-and-inertial estimate marked ($\star$) in Table 1. The rotation and translation errors for the image-only estimate are given in the last row of Table 1. No scale error is reported for this estimate because, as mentioned in Section 6.4, no global scale is recovered by image-only motion estimation. The x, y translations estimated using the image-only method are shown with the image-and-inertial translation estimates in Figure 9(a).

The motion estimate from image measurements has large errors, and the errors versus ground truth are much higher than for the image-and-inertial estimate. The image-only algorithm reduces the image reprojection error versus the batch image-and-inertial estimate, but, in this case, reducing the image reprojection error degrades rather than improves the motion estimate.

What is the source of the large estimation errors produced by the batch image-only method? When synthetic, zero error image measurements are generated from the batch image-and-inertial camera and three-dimensional point estimates, and the batch image-only algorithm is applied to a perturbed version of the batch image-and-inertial estimate, the batch image-and-inertial estimate is correctly recovered from the image data alone. This is reflected in row 1 of Table 4. It follows that,

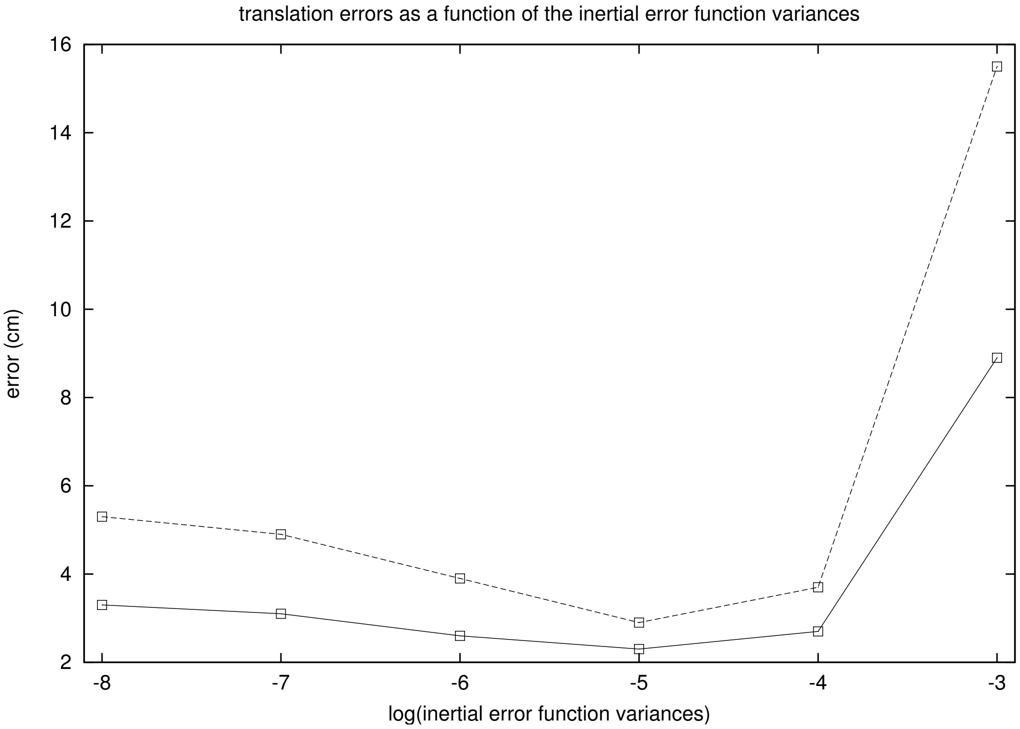
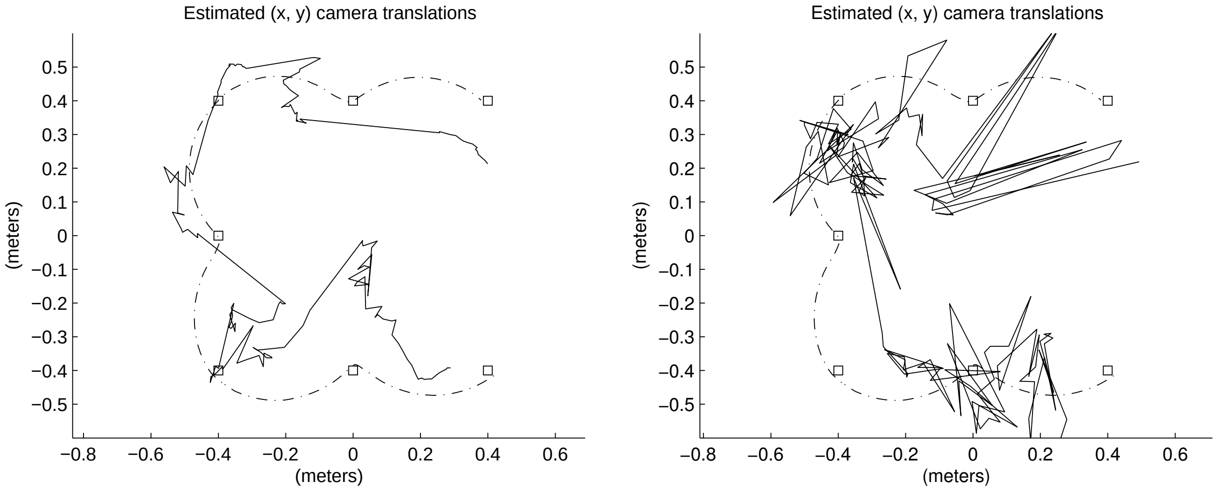


Fig. 8. The average (solid line) and maximum (dashed line) translation errors from Table 1, shown as a function of the inertial error function variances. The change in the errors as a function of the inertial error function variances is small over a wide range of variances, from 10^{-8} to 10^{-4}. The same is true for the rotation and scale errors.



(a) (b)

Fig. 9. Estimates for the perspective arm experiment generated using the batch image-only method. Left: the batch image-and-inertial and batch image-only x, y translation estimates are shown as the dash-dotted and solid lines, respectively. Right: the batch image-and-inertial estimate, and the batch image-only estimate generated from synthetic projections perturbed by isotropic, $(2.0\,\text{pixel})^2$ variance noise are shown as the dash-dotted and solid lines.

Table 1. The Error Metrics and Iterations Required for Convergence for the Set A Estimates from the Perspective Arm Experiment

ρ, t, v Variances	Initial Estimate	Rotation Error (rad)	Translation Error (cm)	Scale Error	Iterations for Convergence
10^{-8}	Accurate recursive estimate	0.13 / 0.20	3.3 / 5.3	−6.4%	3
	Poor recursive estimate	0.13 / 0.20	3.3 / 5.3	−6.4%	5
	Blind	0.13 / 0.20	3.3 / 5.3	−6.4%	201
10^{-7}	Accurate recursive estimate	0.10 / 0.16	3.1 / 4.9	−6.0%	3
	Poor recursive estimate	0.10 / 0.16	3.1 / 4.9	−6.0%	5
	Blind	0.10 / 0.16	3.1 / 4.9	−6.0%	163
10^{-6}	Accurate recursive estimate	0.09 / 0.14	2.6 / 3.9	−6.7%	3
	Poor recursive estimate	0.09 / 0.14	2.6 / 3.9	−6.7%	5
	Blind	0.09 / 0.14	2.6 / 3.9	−6.7%	69
10^{-5} ($\star$)	Accurate recursive estimate	0.09 / 0.14	2.3 / 2.9	−8.2%	4
	Poor recursive estimate	0.09 / 0.14	2.3 / 2.9	−8.2%	5
	Blind	0.09 / 0.14	2.3 / 2.9	−8.2%	209
10^{-4}	Accurate recursive estimate	0.09 / 0.12	2.7 / 3.7	−13.4%	5
	Poor recursive estimate	0.09 / 0.12	2.7 / 3.7	−13.4%	10
	Blind	N/A	N/A	N/A	N/A
10^{-3}	Accurate recursive estimate	0.12 / 0.17	8.9 / 15.5	−29.3%	23
	Poor recursive estimate	0.12 / 0.17	8.9 / 15.5	−29.3%	23
	Blind	N/A	N/A	N/A	N/A
Image only	Batch estimate ($\star$)	0.47 / 0.60	19.0 / 32.6	N/A	N/A

The rotation and translation error entries give the average error before the slash and the maximum error after the slash. The "N/A" entries indicate those cases where the algorithm failed to converge. The algorithm failed to converge from the failed recursive initial estimate for every choice of the inertial error variances, so we have excluded these entries from the table.

while the shape and motion in this example are not strictly degenerate, estimating the motion from image observations only for this data set is highly sensitive to errors in either the image observations or the camera intrinsics calibration.

To investigate the relative effects of image observation errors and camera intrinsics calibration errors, we have generated 10 additional estimates from the synthetic observations described above. In the first five, we used the synthetic, zero-error image observations, but randomly perturbed the camera intrinsics used in the estimation according to the standard deviations for these values reported by the camera intrinsics calibration estimation program. The estimated camera intrinsics, along with the reported standard deviations, are given in Table 3.

The results are shown in rows 2–6 of Table 4. The resulting errors in the estimates are of the order of, or less than, the errors that we observe in the batch image-and-inertial estimates from the real image measurements.

In the second five experiments, we used unperturbed camera intrinsics values, but added noise to the synthetic projections with a standard deviation of 2.0 pixels in each direction, which is the same observation error distribution we have assumed in our experiments. The resulting errors are shown in rows 7–11 of Table 4, and are an order of magnitude larger than the errors that result from perturbing the camera intrinsics. We conclude that the sensitivity to image observation errors is a major source of error in the image-only motion estimates for this data set. Furthermore, the image observation errors are a larger source of error in the estimated motion than error in the camera intrinsics calibration, unless the intrinsics calibration algorithm grossly underestimates the intrinsics error variances.

7.5. Recursive Image-and-Inertial Estimates

We computed three sets of estimates from the perspective arm observations using the recursive image-and-inertial algorithm:

- Set A explores the estimation accuracy as the recursive algorithm's angular velocity and linear acceleration propagation variances vary;
- Set B explores the estimation accuracy as the number of images used in the recursive algorithm's batch initialization varies;

Table 2. The Error Metrics and Iterations Required for Convergence for the Set B Estimates from the Perspective Arm Experiment

Slowdown Factor	ρ, t, v Variances	Rotation Error (rad)	Translation Error (cm)	Scale Error	Iterations for Convergence
1	10^{-6}	0.09 / 0.14	2.6 / 3.9	−6.7%	3
	10^{-5}	0.09 / 0.14	2.3 / 2.9	−8.2%	4
	10^{-4}	0.09 / 0.12	2.7 / 3.7	−13.4%	5
2	10^{-6}	0.09 / 0.12	2.4 / 3.3	−9.1%	4
	10^{-5}	0.11 / 0.17	2.3 / 3.0	−14.9%	4
	10^{-4}	0.13 / 0.18	3.4 / 5.4	−19.7%	6
4	10^{-6}	0.13 / 0.19	2.7 / 3.3	−14.1%	7
	10^{-5}	0.24 / 0.30	5.6 / 8.4	−29.0%	16
	10^{-4}	0.37 / 0.41	8.8 / 12.7	−37.0%	30
8	10^{-6}	0.17 / 0.25	4.5 / 6.7	−22.2%	59
	10^{-5}	0.24 / 0.33	7.4 / 11.3	−34.8%	8
	10^{-4}	0.43 / 0.47	10.7 / 14.6	−44.9%	116

As in Table 1, the rotation and translation error entries give the average error before the slash and maximum error after the slash.

Table 3. The Camera Intrinsics Estimates and Standard Deviations for the Perspective Arm Experiment, Which Were Used to Generate Perturbed Intrinsics for the Experiments in Section 7.4.

Camera Intrinsics Parameter	Estimated Value and Standard Deviation
x, y focal lengths in pixels	[834.18 419.88] ± [3.20 1.82]
x, y image center in pixels	[317.34 105.30] ± [9.46 4.34]
Second- and fourth-order radial distortion coefficients	[-0.290 0.557] ± [0.030 .210]
y, x tangential distortion coefficients	[-.00621 -0.00066] ± [0.0024 0.0015]

Table 4. The Sensitivity of Image-Only Batch Estimation to Camera Intrinsics Calibration Errors and to Image Observation Errors

Trial	Perturbed Instrinsics	Perturbed Projections	Rotation Error (rad)	Translation Error (cm)
1	No	No	3.4×10^{-6} / 1.1×10^{-5}	3.3×10^{-6} / 9.6×10^{-6}
2	Yes	No	0.032 / 0.056	1.0 / 2.5
3	Yes	No	0.030 / 0.037	1.3 / 2.7
4	Yes	No	0.094 / 0.130	3.0 / 8.0
5	Yes	No	0.046 / 0.070	1.5 / 4.0
6	Yes	No	0.028 / 0.043	1.3 / 2.8
7	No	Yes	0.27 / 0.60	22.9 / 60.1
8	No	Yes	0.19 / 0.57	21.0 / 53.2
9	No	Yes	0.42 / 1.07	34.4 / 70.3
10	No	Yes	0.57 / 0.90	29.8 / 59.2
11	No	Yes	0.32 / 0.66	17.7 / 61.5

The correct motion can be recovered from image measurements only given synthetic, zero noise observations and the intrinsics used to generate them (row 1). Estimating the motion from the synthetic image observations and a perturbed version of the camera intrinsics that generates the observations results in errors (rows 2–6) are much less than the errors that result from estimating the motion from the unperturbed camera intrinsics and noisy image observations (rows 7–11).

- Set C explores the estimation accuracy as the speed of the motion varies.

7.5.1. Set A

We computed 25 estimates using the recursive image-and-inertial algorithm, varying the angular velocity propagation variance from $(10^{-2}\,\text{rad s}^{-1})^2\text{s}^{-1}$ to $(10^{6}\,\text{rad s}^{-1})^2\text{s}^{-1}$ and the linear acceleration propagation variance from $(10^{-2}\,\text{m s}^{-2})^2\text{s}^{-1}$ to $(10^{6}\,\text{m s}^{-2})^2\text{s}^{-1}$. Guided by the batch algorithm results in Table 1, we have adopted rotation, translation, and velocity error variances of 10^{-5} for the batch initialization in each case, and the batch initialization uses 40 images in each case. The gyro and accelerometer measurements variances are from the sensor calibration described in Section 6.3.

The results are summarized in Table 5 and Figure 10. Generally, the estimates are good if the angular velocity variance is 10^2 or less and if the linear acceleration is 10^0 or greater. If the linear acceleration is less than 10^0, then due to the strong accelerations in the motion, the filter is unable to track the changes in linear acceleration. Of the variances tests, an angular velocity variance of 10^{-2} and a linear acceleration variance of 10^4 provide the best estimate, so we have adopted these propagation variances in Sets B and C below.

7.5.2. Set B

We computed five estimates using the recursive image-and-inertial algorithm varying the number of batch initialization images from 20 to 60. As in Set A above, we have used rotation, translation, and velocity error variances of 10^{-5} for the batch initialization. Based on the experiments in Set A above, we have adopted angular velocity and linear acceleration propagation variances of 10^{-2} and 10^4, respectively.

The results are summarized in Table 6. The quality of the estimates degrades as the number of initialization images goes down. In particular, initializing from only 20 images produces a poor recursive estimate for this data set.

7.5.3. Set C

To investigate the robustness of the recursive method to changes in the overall speed of the motion, we also computed three additional estimates using the recursive image-and-inertial algorithm on the artificially slowed observation sequences described in Section 7.3. As in Set A, the batch initialization uses 40 images and 10^{-5} for the inertial error variances. Based on the results from Set A above, we have again adopted 10^{-2} and 10^4 as the angular velocity and linear acceleration propagation variances, respectively.

The results are summarized in Table 7. As one might expect, the rotation and translation errors are slightly worse for the recursive algorithm than for the batch algorithm. However, their degradation as the overall motion speed and frequency of the image measurements decrease is reasonable, and comparable to that of the batch algorithm.

7.6. Summary

The experiments in this section indicate the following.

1. The batch image-and-inertial algorithm has a wide range of convergence. In particular, the algorithm can converge from poor initial estimates, including estimates that use no a priori knowledge of the unknowns. However, the algorithm can be made to fail by choosing a severely inaccurate initial estimate (Section 7.3, Set A).
2. The batch image-and-inertial algorithm converges from a wider range of initial estimates than batch image-only estimation (Section 7.3, Set A).
3. The accuracy of the batch image-and-inertial algorithm is largely insensitive to the choices of the variances that define the inertial error term (Section 7.3, Set A).
4. The overall scale of the motion and scene structure can be recovered to within a few percent from image and inertial measurements, even when an inexpensive accelerometer not intended for navigation applications is employed (Section 7.3, Sets A and B).
5. The accuracy and convergence of the batch image-and-inertial algorithm degrade gracefully as the overall speed of the sensor motion decreases and the time between image measurements increases (Section 7.3, Set B).
6. Batch image-only estimation can be highly sensitive to errors in both the image measurements and the camera intrinsics calibration. However, for the data set described in this section, the sensitivity to image measurements errors appears to be much higher (Section 7.4).
7. The recursive image-and-inertial algorithm can be highly sensitive to the angular velocity and linear acceleration propagation variances and to the number of initialization images, so these values need to be chosen in a way that accommodates the sensors' motion (Section 7.5, Sets A and B).
8. The sensitivity of the recursive algorithm to changes in the speed of the motion and the frequency of the image measurements is low, and is comparable to that of the batch algorithm (Section 7.5, Set C).

8. Omnidirectional Arm Experiments

To compare the relative benefits of using inertial measurements versus omnidirectional images, we conducted three

Table 5. The Average Error Metrics for the Set A Estimates Generated Using the Recursive Image-and-Inertial Algorithm

ω Propagation Variance	a Propagation Variance	Rotation Error (rad)	Translation Error (cm)	Scale Error
10^{-2}	10^{-2}	0.25	8.4	−3.0%
	10^{0}	0.12	4.2	−3.1%
	10^{2}	0.11	4.0	−2.8%
	10^{4}	0.10	3.4	0.6%
	10^{6}	0.12	3.5	−0.3%
10^{0}	10^{-2}	0.25	8.3	−1.3%
	10^{0}	0.13	4.2	−2.9%
	10^{2}	0.11	3.9	−2.6%
	10^{4}	0.10	3.5	0.9%
	10^{6}	0.12	3.6	−0.1%
10^{2}	10^{-2}	0.4	14.0	16.7%
	10^{0}	0.14	3.8	−4.3%
	10^{2}	0.13	3.5	−4.7%
	10^{4}	0.10	3.5	−0.07%
	10^{6}	0.11	4.0	1.3%
10^{4}	10^{-2}	N/A	N/A	N/A
	10^{0}	0.22	5.0	−8.1%
	10^{2}	0.20	5.1	−9.2%
	10^{4}	0.13	3.6	−3.0%
	10^{6}	0.26	10.7	−24.0%
10^{6}	10^{-2}	0.30	13.0	1.4%
	10^{0}	0.32	14.3	8.7%
	10^{2}	0.18	5.8	−7.6%
	10^{4}	0.13	4.0	−2.5%
	10^{6}	0.54	13.4	14.9%

Table 6. The Error Metrics for the Set B Estimates Generated Using the Recursive Image-and-Inertial Algorithm

Initialization Images	Rotation Error (rad)	Translation Error (cm)	Scale Error
20	1.21/1.39	15.7/42.2	−47.8%
30	0.22/0.26	5.7/6.9	20.0%
40	0.10/0.15	3.4/4.5	0.6%
50	0.10/0.16	2.7/3.7	−4.2%
60	0.08/0.12	2.1/2.8	−3.2%

As in previous tables, the rotation and translation error entries give the average error before the slash and the maximum error after the slash.

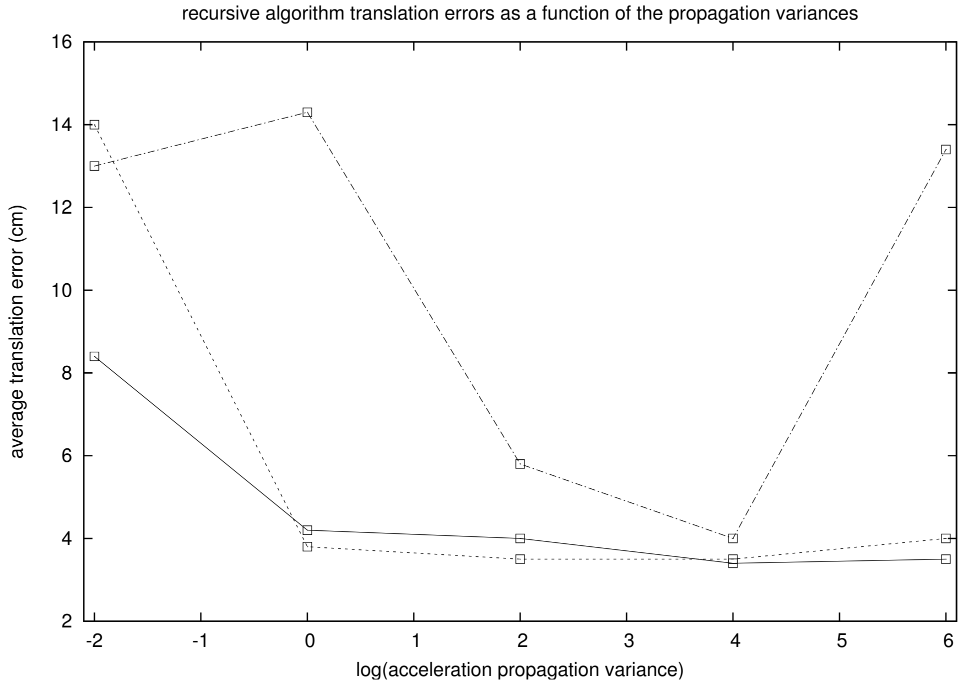


Fig. 10. The average translation errors given in Table 5 for the recursive image-and-inertial algorithm, as a function of the linear acceleration propagation variance, for angular velocity propagation variances of 10^{-2} (solid line), 10^{2} (dashed line), and 10^{6} (dash-dotted line). The accuracy of the recursive method is dependent on choosing propagation variances appropriate for the sensors' motion.

Table 7. The Error Metrics for the Set C Estimates Generated Using the Recursive Image-and-Inertial Algorithm

Slowdown Factor	Rotation Error (rad)	Translation Error (cm)	Scale Error
1	0.10/0.15	3.4/4.5	0.6%
2	0.12/0.16	3.7/4.8	0.7%
4	0.17/0.21	3.2/5.3	−10.9%
8	0.13/0.16	5.4/8.0	−9.3%

The rotation and translation error entries give the average error before the slash and the maximum error after the slash.

experiments with the omnidirectional cameras shown in Figures 3 and 4, using the same arm used for the perspective experiment described in Section 7, and a similar motion. The sensor rigs and observation sequences for these experiments are described in Sections 8.1 and 8.2, respectively. The batch image-and-inertial, batch image-only, and recursive image-and-inertial estimates are described in Sections 8.3, 8.4, and 8.5, respectively.

8.1. Sensor Configuration

8.1.1. First Omnidirectional Configuration

The first omnidirectional camera configuration is shown in Figure 3 and consists of a CCD camera attached to a convex mirror by a rig that allows the relative rotation and translation between the mirror and camera to be manually adjusted. As in the perspective arm experiments described in the previous section, the CCD camera is a Sony XC-55, captured by a conventional frame grabber at 30 Hz, and the 640 × 480 interlaced images are again reduced to 640 × 240 non-interlaced images by removing one field. The camera was paired with a 16 mm lens for use with the mirror. As mentioned in Section 4.5, the mirror is the equiangular mirror described by Ollis, Herman, and Singh (1999). The adjustable rig is convenient but, as we see in Sections 8.2 and 8.4, is somewhat flexible and allows some unmodeled vibration between the camera and mirror during erratic motion.

8.1.2. Second Omnidirectional Configuration

The second omnidirectional configuration is shown in Figure 4 and consists of an IEEE 1394 camera attached to a convex mirror by a clear, rigid cylinder. The color, 1042 × 768 images produced by the camera were captured at the camera's maximum acquisition rate of 15 Hz, and were reduced to gray-scale, 640 × 480 images before tracking. As with the first omnidirectional configuration, the camera was paired with a 16 mm lens. The mirror is again the equiangular mirror described by Ollis, Herman, and Singh (1999).

As mentioned in Section 4.5, our motion estimation algorithms can incorporate knowledge of the six-degrees-of-freedom misalignment between the camera and mirror if this information is available, and an accurate estimate of this transformation generally increases the accuracy of motion estimation from omnidirectional images. For the experiments described here, the precise mirror-to-camera transformations are not available, so we have used reasonable but necessarily imprecise values for this transformation in our experiments.

8.2. Observations

8.2.1. First Experiment

In the first experiment, the first omnidirectional rig was mounted on the Performer arm and the same motion used for the perspective arm experiment, described in Section 7.2, was executed. The resulting observation sequence contains 152 omnidirectional images and 862 inertial readings.

Six points were tracked through the sequence using Lucas–Kanade (Lucas and Kanade 1981) with manual correction. In most of the images, all six points were visible. A few example images from this sequence, with tracked features overlaid, are shown in Figure 11. As mentioned above in Section 8.1, the rig in the first configuration is somewhat flexible and allows some unmodeled vibration between the mirror and camera when the motion is erratic. So, some vibration of the mirror and of the tracked features, of the order of 10 pixels, is visible in the image sequence.

8.2.2. Second Experiment

In the second experiment, the second omnidirectional rig was placed on the Performer arm, and moved through a motion similar to that used in for the perspective arm experiment and for the first omnidirectional experiment described above. Since the camera was able to acquire images at only 15 Hz rather than 30 Hz, the arm velocity was reduced to half to produce an image sequence with the same number of images as the sequence used in the first experiment; inertial measurements were acquired at the same rate as in the previous experiments. The resulting observation sequence contains 152 omnidirectional images and 1853 inertial readings.

Eight points were tracked through the entire sequence using Lucas–Kanade. Mistracking was again common and corrected by hand. Four example images from the sequence, with the tracked features overlaid, are shown in Figure 12.

8.2.3. Third Experiment

For the third experiment, the image sequence from the second experiment was reduced to 640 × 240 to test the effects of image resolution on the omnidirectional motion estimates, and the eight points tracked in the second experiment were retracked through the entire sequence of reduced resolution images. The inertial measurements are the same as in the second experiment.

8.2.4. Comparisons With the Perspective Arm Data Set

The motion speed and image resolution in the first omnidirectional experiment, which uses the first omnidirectional configuration, are the same as those used to produce the unslowed perspective arm estimates (the Set A batch image-and-inertial estimates in Section 7.3, the batch image-only estimates in Section 7.4, and the Sets A and B recursive image-and-inertial estimates in Section 7.5). So, the batch image-and-inertial, batch image-only, and recursive image-and-inertial estimates produced from this omnidirectional data set can be compared against those described in Table 1 (rows labeled "⋆"), Table 1 (row labeled "image only"), and Tables 5 and 6, respectively. One caveat is that, as we see in Section 8.4, the unmodeled vibration in the first omnidirectional experiment introduces local motion errors in the image-only estimates. Since this unmodeled vibration is not an inherent flaw in omnidirectional cameras, this should be considered when comparing the image-only estimates from the perspective arm experiment and the first omnidirectional arm experiment.

The motion speed and image resolution in the third omnidirectional experiment, which uses the second omnidirectional configuration and vertically reduced images, are the same as those used to produce the slowed perspective arm estimates with the slowdown factor $f = 2$ (the Set B batch image-and-inertial estimates in Section 7.3 and the Set C recursive image-and-inertial estimates in Section 7.5). So, the batch image-and-inertial and recursive image-and-inertial estimates from this omnidirectional experiment can be compared against those described in Table 2 ($f = 2$) and Table 7 (line 2), respectively.

8.3. Batch Image-and-Inertial Estimates

For each of the three experiments, we generated a batch image-and-inertial estimate of the motion and other unknowns, using an estimate from the recursive image-and-inertial method as the initial estimate; these recursive estimates are described in Section 8.5. Based on the results in Section 7.3 exploring the accuracy of estimates as a function of the inertial error variances for the perspective sequence, we have chosen the

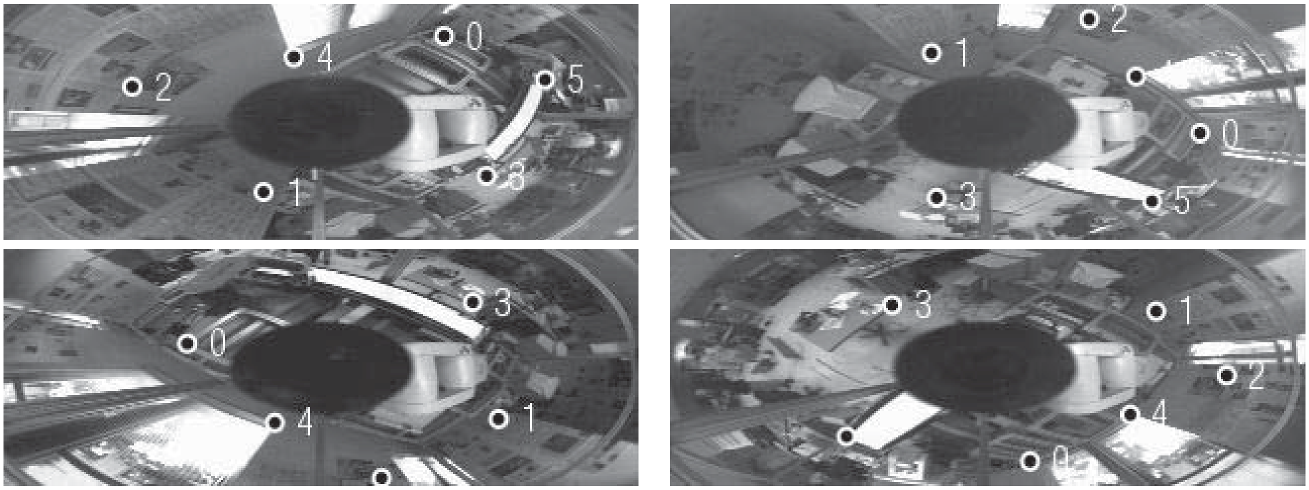


Fig. 11. Images 0, 50, 100, and 150 from the 152 image sequence acquired with the first omnidirectional camera configuration for the first experiment, shown clockwise from the upper left. The tracked features are overlaid. As with the images in Figure 6, the images are one field of an interlaced image, so their height is one-half the height of the full images.

inertial error variances to be 10^{-5} for each omnidirectional experiment.

The error statistics for each of the three batch image-and-inertial estimates are given in Table 8, and the x, y translation estimates from the first and second experiments are shown as the dash-dotted lines in Figure 13. In each experiment, the estimates suffer somewhat from the lack of a precise camera-to-mirror transformation calibration, but the results are comparable to those in the perspective arm experiment, despite the changes in motion speed, frame rate, and image resolution across the three experiments.

8.4. Batch Image-Only Estimates

For each of the three experiments we also generated a batch image-only estimate. For consistency with the perspective experiments described in Section 7.4, we used the batch image-and-inertial estimate as the initial estimate for the image-only initialization. The error statistics for each of the image-only estimates are also given in Table 8, and the x, y translation estimates for the first and second experiments are shown as the solid lines in Figure 13. Unlike the perspective image-only estimate, each of the three omnidirectional estimates is globally correct, which reflects the advantage of using omnidirectional cameras over perspective cameras for motion estimation. However, as mentioned in Section 8.1, the first omnidirectional camera configuration, which is used in the first experiment, suffers from an unmodeled vibration between the camera and mirror when the camera's motion is erratic. The effects of this vibration appear in the batch image-only estimate on the left of Figure 13 as local perturbations in the motion. Interestingly, the use of inertial measurements in the batch image-and-inertial estimate eliminates these perturbations.

8.5. Recursive Image-and-Inertial Estimates

We also generated a sequence of estimates using the recursive image-and-inertial algorithm, varying the number of images used in the batch initialization. Based on the experiments described in Sections 7.3 and 7.5, we have chosen to use inertial error variances of 10^{-5} for the batch initialization, 40 images in the batch initialization, and angular velocity and linear acceleration propagation variances of 10^{-2} and 10^{4} for recursive operation.

The error metrics for the recursive estimates are also given in Table 8. In the first experiment, which uses the adjustable omnidirectional camera rig, the recursive algorithm is unable to overcome the unmodeled effects of vibration between the camera and mirror caused by the erratic motion of the arm, whereas the batch image-and-inertial method is robust to this problem. The recursive method produces accurate estimates for the second and third experiments, which use the more rigid second omnidirectional camera rig, and for those experiments the estimates degrade gracefully as the number of initialization images is decreased.

8.6. Summary

The experiments in this section indicate the following.

1. Image-only motion estimation from omnidirectional images is less sensitive to random image measurement errors than image-only motion estimation from perspective images, so the advantage of incorporating inertial measurements is less for omnidirectional images than for perspective images (Sections 8.3 and 8.4).
2. The incorporation of inertial measurements in the batch image-and-inertial algorithm provides some robustness

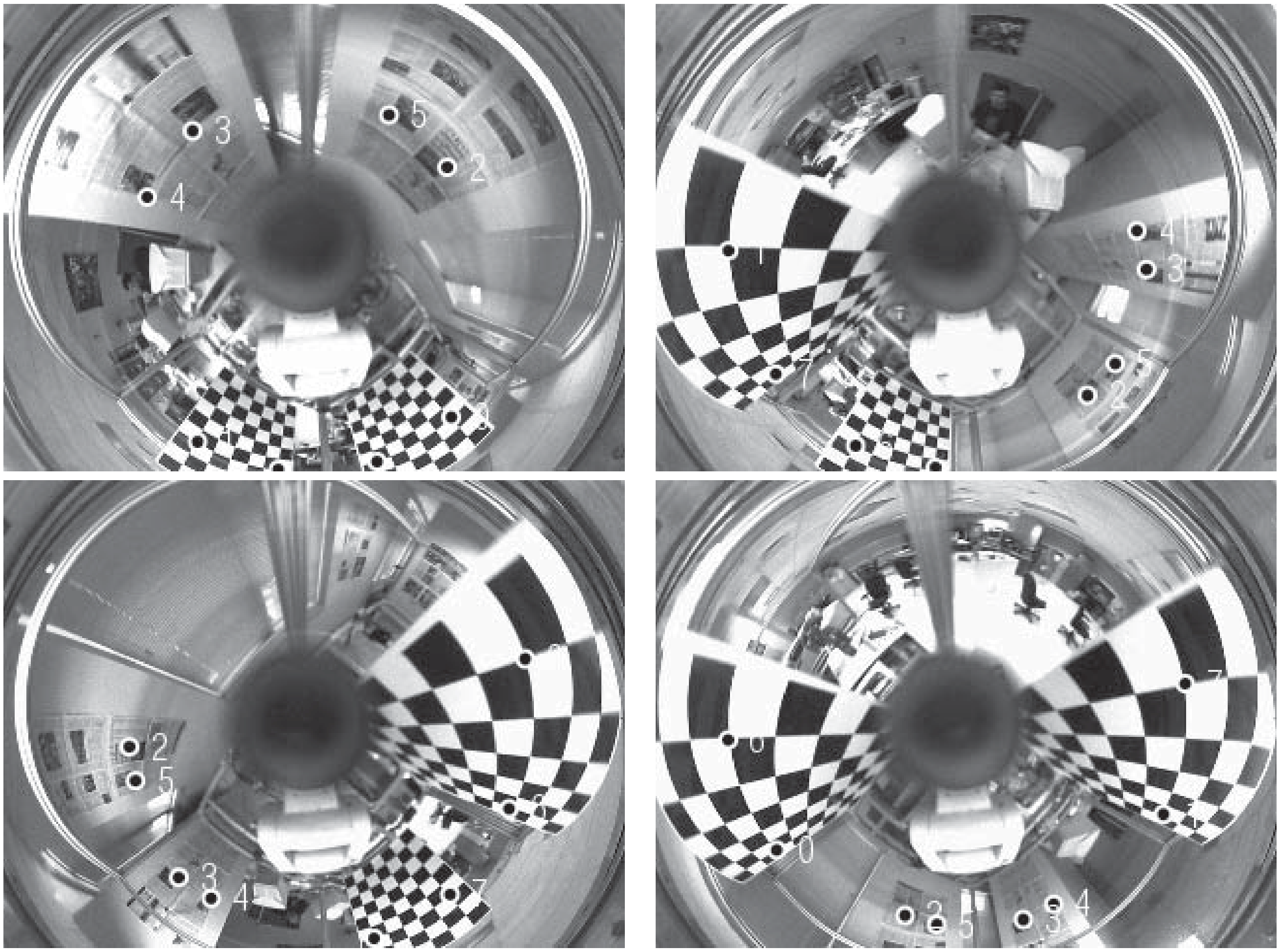


Fig. 12. Images 0, 50, 100, and 150 from the 152 image sequence acquired with the second omnidirectional camera configuration for the second experiment, shown clockwise from the upper left. The tracked features are overlaid. In this experiment, the camera produces full-resolution, non-interlaced images.

to moderate, non-random image measurement errors, in this case unmodeled vibration between the mirror and camera in the first experiment (Sections 8.3 and 8.4).

3. The recursive image-and-inertial algorithm produces poor estimates in the presence of non-random image measurement errors in the first experiment, despite the incorporation of inertial measurements (Section 8.5).

9. Perspective Crane Experiment

9.1. Camera Configuration

The camera configuration for the perspective crane experiment is the same as that for the perspective arm experiment, except that to gain a wider field of view a 3.5 mm lens was used in place of the 6 mm lens.

9.2. Observations

The rig was mounted on the platform of a specially constructed crane that translates the platform within a 10 × 10 × 5 m^3 area. For the experiment, the crane was programmed to move between 11 pre-specified control points. These points take the platform from the center of the work area to one corner of a 6 m wide square aligned with the crane coordinate system (x, y), around the square, and then back to the center of the work area. In z, the platform translates 1 m between adjacent control points, alternating between high and low positions.

The observation sequence taken during this circuit consists of 1430 images and 51,553 inertial readings spanning 4 min 46 s; eight example images from the sequence are shown in Figure 14. 100 points were extracted in the first image of the sequence, and tracked through the sequence using Lucas–Kanade. In subsequent images in which the number of tracked points fell below 80 due to features leaving the

Table 8. The Error Metrics for the First, Second, and Third Omnidirectional Arm Experiments

		Rotation	Translation	Scale
Experiment	Estimate	Error (rad)	Error (cm)	Error
First	Batch image-and-inertial	0.12 / 0.15	3.4 / 4.5	5.2%
	Batch image-only	0.12 / 0.15	5.1 / 7.5	N/A
	Recursive, 20 image initialization	0.14 / 0.18	7.4 / 12.3	−33.8%
	Recursive, 30 image initialization	0.12 / 0.15	7.5 / 14.2	−48.9%
	Recursive, 40 image initialization	0.12 / 0.20	9.6 / 15.6	−6.1%
	Recursive, 50 image initialization	0.11 / 0.15	6.4 / 9.2	−5.8%
	Recursive, 60 image initialization	0.11 / 0.15	6.1 / 8.1	2.8%
Second	Batch image-and-inertial	0.11 / 0.13	4.1 / 6.3	6.7%
	Batch image-only	0.10 / 0.12	4.0 / 6.1	N/A
	Recursive, 20 image initialization	0.14 / 0.22	6.2 / 9.5	113.5%
	Recursive, 30 image initialization	0.14 / 0.18	4.7 / 7.0	−11.6%
	Recursive, 40 image initialization	0.13 / 0.17	4.1 / 5.2	−13.3%
	Recursive, 50 image initialization	0.13 / 0.17	3.9 / 5.2	−9.2%
	Recursive, 60 image initialization	0.11 / 0.15	3.9 / 4.9	8.5%
Third	Batch image-and-inertial	0.11 / 0.14	4.0 / 5.9	8.0%
	Batch image-only	0.11 / 0.14	4.0 / 5.9	N/A
	Recursive, 20 image initialization	0.15 / 0.20	4.9 / 8.3	41.5%
	Recursive, 30 image initialization	0.15 / 0.19	4.6 / 7.1	−14.6%
	Recursive, 40 image initialization	0.14 / 0.17	4.3 / 5.5	−13.9%
	Recursive, 50 image initialization	0.14 / 0.17	4.2 / 5.4	−10.5%
	Recursive, 60 image initialization	0.12 / 0.15	4.0 / 4.8	9.4%

The estimates are largely insensitive to the changes in the speed and image resolution across the three experiments, and the sensitivity of the batch image-only estimates apparent in the perspective arm experiment is eliminated by use of the omnidirectional camera.

field of view, becoming occluded, or changing appearance, re-extraction was used to bring the number of tracked points back to 100. Many mistracked points were identified by applying RANSAC in conjunction with batch, image-only shape and motion estimation to each subsequence of 30 images in the sequence. 166 additional point features that were mistracked but not detected by the RANSAC procedure were pruned by hand.

These tracking and pruning stages resulted in a total of 1243 tracked points. Each image contains an average of 53.8 points, and each point is visible in an average of 61.9 images, so that on average each point was visible in 4.3% of the image sequence. As described in Section 2.4, this is a very low percentage by the standards of image-only motion estimation.

9.3. Estimate

We generated an estimate of the motion and other unknowns with the recursive image-and-inertial method described in Section 5, using 40 images in the batch initialization. In this experiment, the average (maximum) rotation and translation

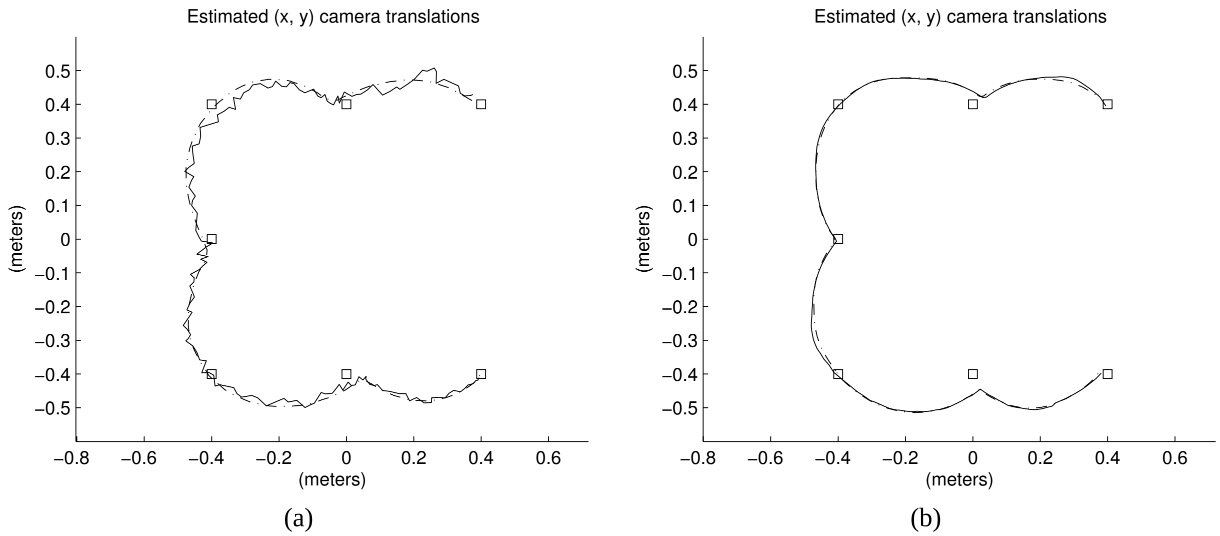


Fig. 13. The x, y translation estimates generated by the batch image-and-inertial and batch image-only algorithms for the first and second omnidirectional arm experiments. Left: the estimates generated by the batch image-and-inertial algorithm and by the batch image-only algorithm for the first omnidirectional experiment are shown as the dash-dotted and solid lines, respectively. Right: the estimates generated by the batch image-and-inertial algorithm and by the batch image-only algorithm for the second omnidirectional experiment are shown as the dash-dotted and solid lines, respectively. On both the left and right, the x, y locations of the seven known ground truth points are shown as squares.

errors are 13.6 (0.17) rad and 31.5 (61.9) cm, respectively. This average translation error is approximately 0.9% of the total 34.1 m traveled. The scale error in this experiment is -3.4%.

The resulting x, y translation estimates are shown from above in Figure 15, and the resulting z translation estimates are shown in Figure 16. In each figure the squares are the known ground control points.

10. Perspective Rover Experiment

The Hyperion rover is a test bed for candidate technologies for a long-range, robotic search for life on Mars. During April 2003, Hyperion performed the first of three scheduled field tests in Chile's Atacama Desert, and, on eight of those days, Hyperion carried our camera, gyro, and accelerometer. In this section we describe the observations and estimates for one of the resulting observation sequences.

10.1. Camera Configuration

The sensor rig consisted of an IEEE 1394 camera coupled with a 3.5 mm lens, the gyro, and the accelerometer.

10.2. Observations

The sensor rig was mounted horizontally about 1 m from the ground at the rear of the rover, and rotated 45° about the vertical axis to look out to the rear and right of the rover. The rover executed three large, roughly overlapping loops during a 15 min 36 s traverse, covering a total distance of roughly 230 m. During this time, a sequence of 1401 images was acquired at approximately 1.5 Hz, and 112,917 inertial readings were acquired at an average frequency of approximately 121 Hz.

The custom tracker mentioned in Section 2.5 was used to generate two point feature data sets from this image sequence. In the first, 16,007 point features were tracked through the sequence, with each image containing an average of 201.5 points, and each point appearing in an average of 17.5 images. That is, each point appears in an average of 1.3% of the image sequence. The density of these features in one image is shown in Figure 17(a), and a few features from this data set are shown over time in Figure 18. In the second, we considered a subset of the first point feature data set containing 5611 point features, where each image contained an average of 34.9 points and each tracked point appears in an average of 8.7 images. This is 0.62% of the image sequence. The relative density of these features in one image is shown in Figure 17(b). Some point feature tracks in these data sets exhibit drift, with a maximum cumulative drift of the order of five pixels, particularly for features in the distance that can be tracked through many images. However, we are aware of no gross tracking errors in this data set of the kind that Lucas–Kanade often produces.

10.3. Recursive Image-Only Estimates

We used the recursive image-only algorithm described in Section 5.7 to generate one motion estimate from each of the two

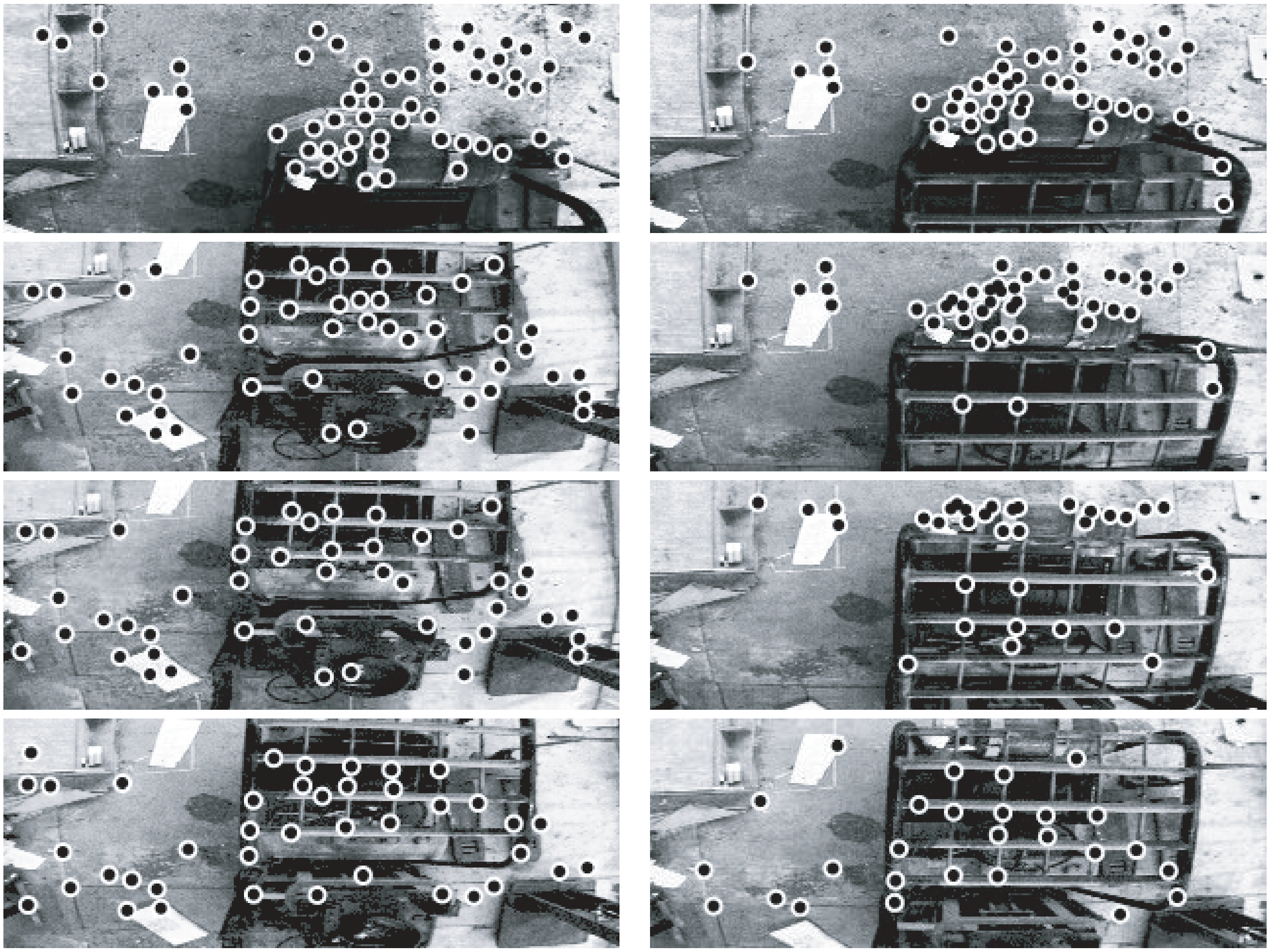


Fig. 14. Images 480, 490, 500, . . . , 550 from the 1430 image sequence in experiment 3 are shown clockwise from the upper left, with the tracked image features overlaid. As with the images from experiment 1 shown in Figure 6, the images are one field of an interlaced image, so their height is half that of a full image.

feature data sets. The x, y translation estimates for the 201.5 features per image data set and for the 34.9 features per image data set are shown in Figures 19(a) and 19(b), respectively. The x, y translation estimates from the rover's odometry and from the rover's GPS receiver are shown in Figures 19(c) and 19(d), respectively. In each diagram, a 1 m wide "x" marks the start position of the rover. The easiest way to understand the rover's qualitative motion is to first examine the estimates in Figures 19(a) and 19(b), which are smooth everywhere, and then to examine the odometry and GPS estimates in Figures 19(c) and 19(d), which contain local errors.

Each of these four estimates contains errors of different kinds. The estimates generated by our algorithm, shown in Figures 19(a) and 19(b), are locally correct everywhere, but exhibit slow drift over time. For the estimate generated using 201.5 features per image, this drift is primarily in x, y translation, while the estimate generated using 34.9 features per image contains drift in both x, y translation and rotation about z. In each case, the drift components in z translation and rotation about x and y, which are not indicated in the figure, are quite small. The estimates from GPS exhibit serious errors, which were common in the GPS data Hyperion acquired and are due to changes in GPS satellite acquisition during the traverse.

The estimates from odometry contain some local errors, visible as small juts of the order of 1 m in Figure 19. These correspond to times in the image sequence where the rover's wheels became caught on a large rock in the rover's path. However, the estimates from odometry are, globally, the best of the four. For instance, the odometry estimate indicates that at the end of the second loop, the rover comes very close to the location it visited at the end of the first loop, as shown in Figure 19(c) near (310, 752) where the two loops touch; visual inspection of the image sequence confirms that the rover's

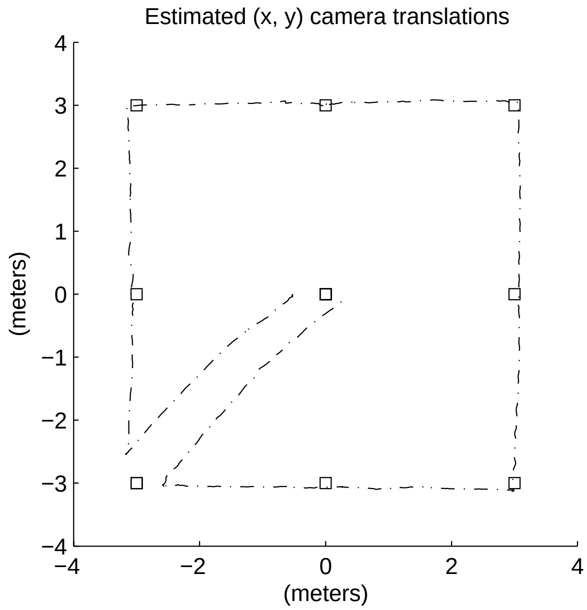


Fig. 15. The x, y translation estimate for the perspective crane experiment generated by the recursive image-and-inertial method is shown by the line. The x, y locations of the 11 known ground truth points are shown as squares. Ground truth points 0 and 1 are identical to points 10 and 9, respectively, so that only nine squares are visible in the figure.

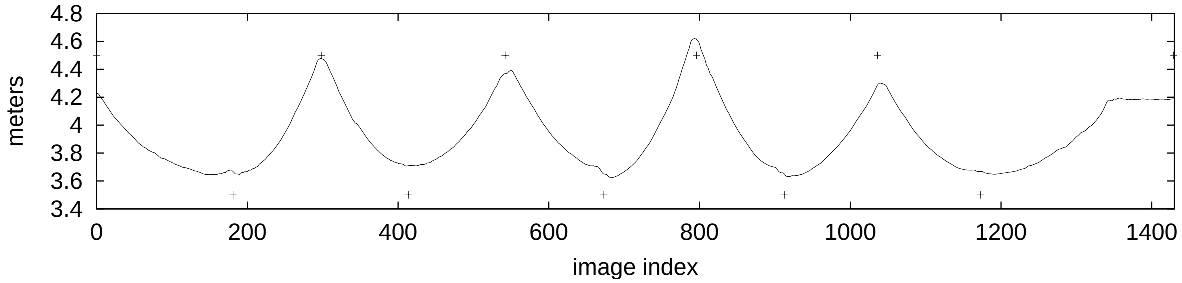


Fig. 16. The z translation estimates for perspective crane experiment generated by the recursive image-and-inertial method are shown by the line. As in Figure 15, the known points are shown as squares.

position was almost identical during these two times. Similarly, the estimate from odometry indicates that at the end of the third loop, the rover's position is significantly behind the rover's initial position; this also appears to be confirmed by the image sequence. So, we have adopted the odometry estimate as our ground truth estimate.

Compared with the odometry estimate, the average and maximum translation errors in the 201.5 features per image estimate, after the scaled rigid transformation, are 1.74 and 5.14 m, respectively. These are approximately 0.8% and 2.2% of the total distance traversed by the rover. The rotation error appears to be negligible in this case. The average and maximum translation errors in the 34.9 features per image estimate are 2.35 and 5.27 m, respectively. These are approximately 1.0% and 2.3% of the total distance traversed by the rover. In this case, an accumulated error in the z rotation is noticeable.

10.4. Recursive Image-and-Inertial Algorithm

We also computed a collection of estimates from the 34.9 features per image data set using the recursive image-and-inertial algorithm. In the collection, we allowed the angular velocity and linear acceleration propagation variances to vary from 10^{-4} to 10^{5} and from 10^{-2} to 10^{4}, respectively. As detailed in

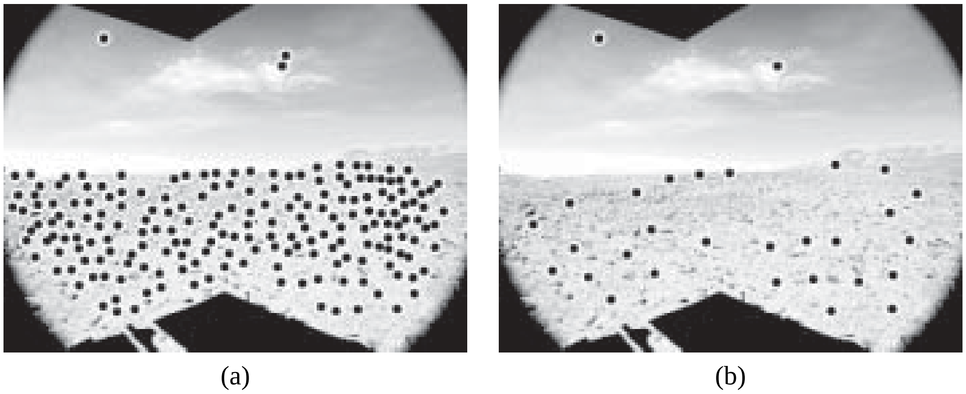


Fig. 17. An image from the perspective rover sequence with features from the 201.5 features per image data set (a) and features from the 34.9 features per image data set (b) overlaid.

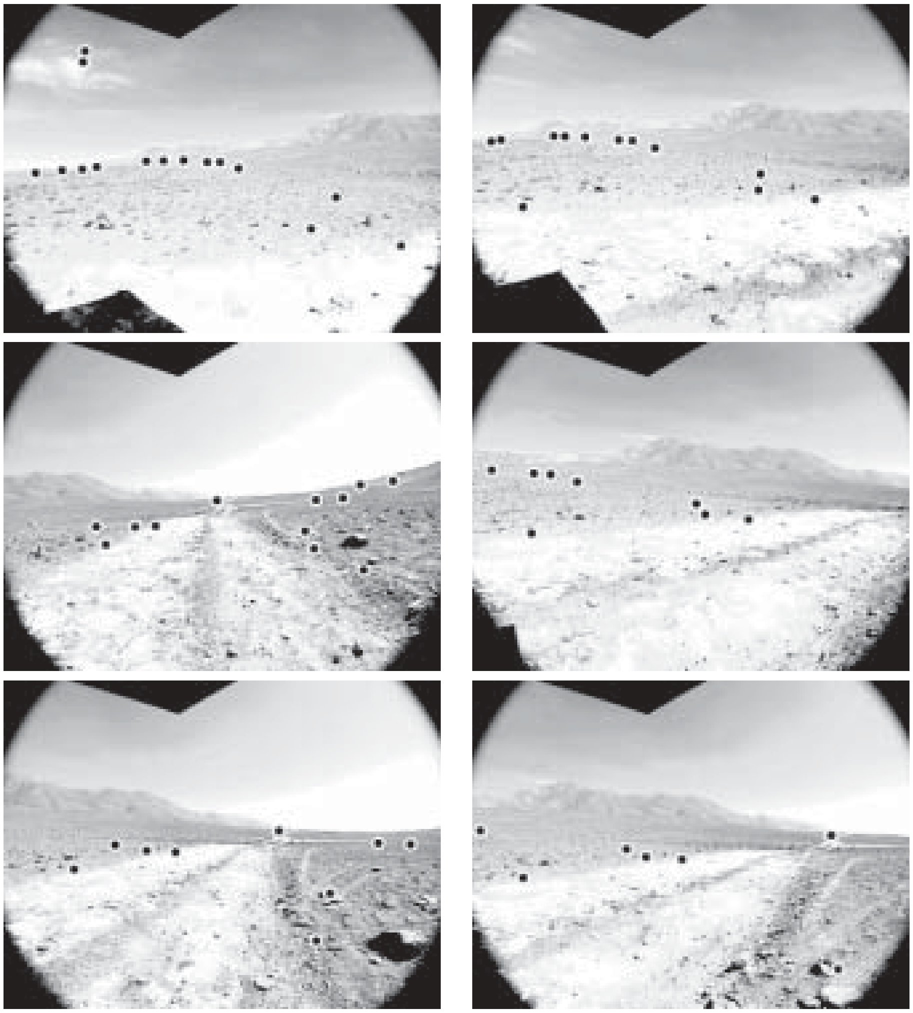
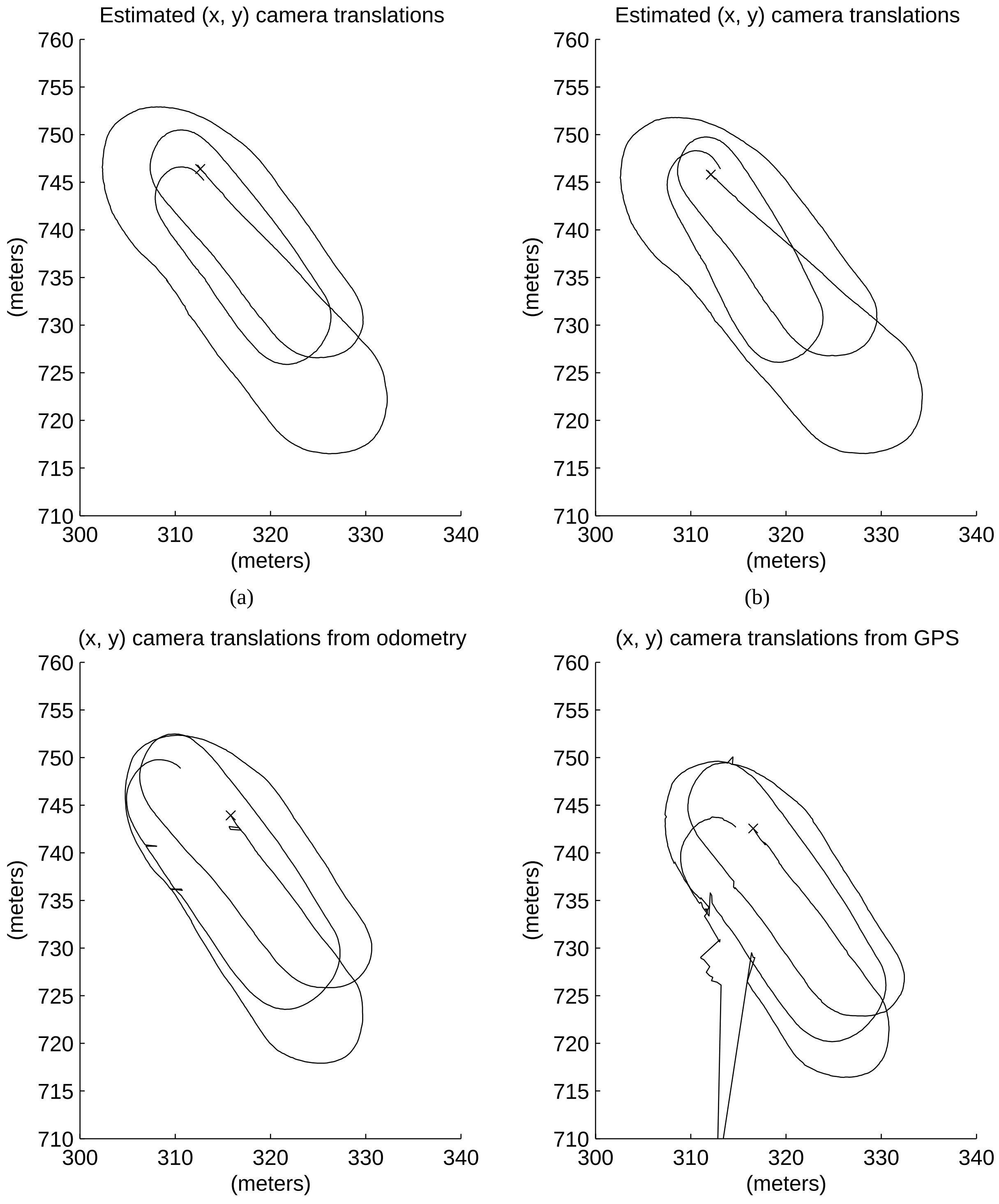


Fig. 18. Images 150, 160, . . . , 200 from the perspective rover experiment are shown clockwise from the upper left, with tracking data from the 201.5 features per image data set. For clarity, only a number of those tracked features that were visible in a relatively long subsequence of the images are shown.



(a)

(b)

(c)

(d)

Fig. 19. The x, y translation estimates for the perspective rover data set, generated using the recursive, image-only algorithm with 201.5 features per image (a), the recursive image-only algorithm with 34.9 features per image (b), the rover's odometry (c), and GPS (d).

the paragraphs below and in Section 10.5, the contribution of the inertial sensors is limited in this example by the erratic and long-term motion in the sequence.

We find that the rover's sudden accelerations and decelerations are incompatible with linear acceleration propagation variances of 10^2 or below, and, for these tests, the distributions that result from the inertial measurement update steps eventually become incompatible with the subsequent image observations. As a result, most of the image observations are eventually rejected as outliers when these variances are used, and the algorithm fails.

For linear acceleration propagation variances above 10^2 and angular velocity propagation variances of 10^4 or below, the resulting estimates are qualitatively similar to the results shown in Figures 19(a) and 19(b). However, the image-and-inertial estimates show noticeably more drift in rotation in azimuth, and less drift in global scale.

As described in Section 5.8, weakening the assumption of continuity in the angular velocity and linear acceleration necessarily weakens the contribution of the inertial measurements in the recursive image-and-inertial algorithm. This is reflected in this experiment: for the most generous propagation variances, i.e., linear acceleration propagation variances above 10^2 an angular velocity variance of 10^5, respectively, the effect of the inertial measurements becomes negligible, and the estimates are almost identical to the 34.9 features per image, recursive image-only estimate shown in Figure 19(b).

10.5. Discussion

In this experiment, the image-only motion estimates benefit from dense, high-quality feature tracking data, and are locally correct everywhere as a result. However, due to the small number of images in which each point feature is visible, the motion is not being estimated with respect to any one external point, and so necessarily drifts with time.

We consider long-term drift in the image-only estimates to be a type of ambiguity, in the sense that the true, globally correct motion and a large space of drifting estimates will all produce small reprojection errors. In fact, given that the image observations and the camera calibration both contain small errors, the estimate that best minimizes the reprojection error will almost never be a globally correct estimate, but will contain some drift. This is in contrast to cases such as that described in Section 7, where only sparse, noisy tracking data are available, and where even the local motion cannot be uniquely identified from image measurements alone.

Can the incorporation of measurements from inexpensive inertial sensors remove this ambiguity in the long-term motion, if the image-only estimates are locally correct everywhere? Since the sensors' absolute orientation with respect to gravity can be reliably found by separating the effects of gravity and the other unknowns in the accelerometer readings, inexpensive inertial sensors can eliminate long-term drift in the estimated rotation about the two axes perpendicular to the gravity direction. (In the image-only estimates described in Section 10.3, the drift in these two components of the rotation was not significant when compared to the drift in the rotation about the gravity direction and in the x, y translation, so orientation information from the accelerometer was not needed.) However, inexpensive inertial sensors are unlikely to resolve long-term drift in the translation or in the rotation about the gravity direction, because the rate of drift in the inertial sensor estimates is likely to be much faster than that of the image-only estimates.

On the other hand, can the incorporation of measurements from inexpensive inertial sensors improve long-term motion estimates if the image-only estimates are locally ambiguous at some points? Of course, in this case the inertial sensors can improve the long-term motion by resolving these ambiguities, which may turn a qualitatively wrong estimate into an accurate estimate. However, our intuition is that estimates will drift less slowly if the image-only estimates can be made locally correct (e.g. by obtaining dense feature tracking data), than if the image-only estimates are locally ambiguous (e.g. generated using sparse image data) and inertial measurements are incorporated to resolve the local ambiguities.

11. Conclusion

11.1. Summary

In this paper, we have presented algorithms for motion estimation from image and inertial measurements. This work targets applications that require six-degrees-of-freedom motion without external position references such as GPS, and focuses on the use of small and inexpensive inertial sensors, for applications where weight and cost requirements preclude the use of precision inertial navigation systems.

11.2. Contributions

11.2.1. Batch Algorithm

We have presented a batch algorithm for estimating sensor motion, scene structure, and other unknowns from image, gyro, and accelerometer measurements. This algorithm requires no general restrictions on the motion, such as smoothness. The batch algorithm converges in just a few iterations from estimates that are near the optimum, it can often converge from a poor initial estimate that incorporates no a priori knowledge of the motion or other unknowns, and it appears to have a significantly wider range of convergence than image-only batch estimation. In our experiments, we have shown that this algorithm can produce highly accurate estimates for the data sets for which it is applicable, which include those containing up to a few hundred images. The batch algorithm is a valuable diagnostic tool for studying the best quality that we can expect given a particular sensor configuration, vehicle motion, environment, and set of observations. For instance, in this paper we have used it to show that inertial measurements can resolve the ambiguity that results from noisy image observations, and that the global scale of the motion can be recovered using the measurements from an inexpensive accelerometer, among other results. This batch algorithm is also a useful tool in the context of recursive estimation: it allows the recursive algorithm, described next, to be initialized without a priori knowledge of the state, and it is a good candidate for conversion into a recursive method using the VSDF, which is a promising direction for future research.

11.2.2. Recursive Algorithm

We have presented a recursive algorithm for estimating sensor motion, scene structure, and other unknowns from image, gyro, and accelerometer measurements, based on the IEKF. This algorithm is a multirate filter, meaning that image and inertial measurements arriving at different rates and times can each be exploited as they arrive. As mentioned in the previous paragraph, this method can be initialized without any heuristics or a priori knowledge of the unknowns using the batch algorithm. As described in Section 5.5, an important aspect of this filter is that it can incorporate newly visible points into the IEKF state covariance estimate in a way that closely approximates the covariances produced by a batch estimation, best exploits the available observations, and correctly reflects the uncertainty in the new points' position estimates relative to both the world and camera coordinate systems.

11.2.3. Experiments

We have performed a large set of experiments, both (1) to investigate some fundamental questions about the use of images and inexpensive inertial sensors for motion estimation, and (2) to evaluate the accuracy and performance of our algorithms. In the first category, a few of the relevant results are as follows.

1. Motion estimation from image and inertial measurements can produce accurate estimates even when the estimates from image or inertial measurements alone are poor.

2. Measurements from an inexpensive accelerometer are often sufficient to establish the global scale of the motion and scene structure to within a few percent.

3. The incorporation of inertial measurements is less advantageous with omnidirectional images than with conventional images.

4. Dense image feature tracking data appear to be more valuable in reducing long-term drift than inexpensive inertial sensors.

In the second category, a few of the relevant results are as follows.

1. The batch algorithm can produce accurate estimates across a wide range of initial estimates and a wide range of the error variances that define the error function.

2. Given accurate image tracking data, the recursive algorithm can produce accurate estimates for image sequences with extremely low fill fractions.

3. The recursive algorithm is more sensitive to the estimation parameters than the batch algorithm. In particular, the recursive algorithm can be sensitive to the choice of the angular velocity and linear acceleration propagation variances, especially for sequences with erratic motion.

11.3. Future Directions

As we described in Section 3, there are two remaining difficulties in deploying the algorithms we have described. First, our recursive algorithm is subject to difficulties inherent in the EKF framework, including the a priori expectation of motion smoothness, approximation of the state estimate distribution by a Gaussian, and the linearization of measurements around uncertain state estimates. The VSDF (McLauchlan 1999), originally designed for SFM and similar applications, is a hybrid batch-recursive method and addresses each of these issues. Adapting our batch method for use in the VSDF framework is a promising direction for future research.

Secondly, our recursive estimation framework should be extended to minimize drift in the estimated motion by "closing the loop" when image features can be reacquired after being lost during a traverse. This requires two additions to the current recursive algorithm, as follows.

- Visual features must be recognized when they are revisited. Recent methods for matching features based on image feature invariants, such as Lowe (1999), are good candidates for this task. In addition, the mean and covariance estimates on the six-degrees-of-freedom camera motion and three-dimensional point positions produced by the recursive algorithm might be exploited to improve this reacquisition, just as they might be used for tracking between temporally adjacent images, as suggested in Section 5.5.

- The recursive algorithm must be modified to maintain some estimate of the relative covariance between the current camera position and currently visible points, and the positions of points that are no longer visible but might be reacquired. Because efficiency requires limiting the dimension of the state vector and covariance matrix, an approach that represents full covariances only between features in local maps, and can generate covariances between features in different local maps using covariances between the transformations associated with the local maps, is promising for this problem (Chong and Kleeman 1999; Guivant and Nebot 2001).

Appendix: Index to Multimedia Extensions

The multimedia extension page is found at http://www.ijrr.org.

Table of Multimedia Extensions

Extension	Type	Description
1	Video	Perspective arm – Tracked features
2	VRML	– Estimated shape and motion
3	Data	– Input and output data
4	Video	First omni arm – Tracked features
5	VRML	– Estimated shape and motion
6	Data	– Input and output data
7	Video	Second omni arm – Tracked features
8	VRML	– Estimated shape and motion
9	Data	– Input and output data
10	Video	Perspective crane – Tracked features
11	VRML	– Estimated shape and motion
12	Data	– Input and output data
13	Video	Perspective rover – Tracked features
14	VRML	– Estimated shape and motion
15	Data	– Input and output data

Acknowledgments

We would like to thank Jeffrey Mishler, Henele Adams, and Tom Minka for their help and advice, and Chris Urmson, Michael Wagner, and David Wettergreen for acquiring the perspective rover data described in Section 10.

References

Baker, S., and Nayar, S.K. 2001. Single viewpoint catadioptric cameras. *Panoramic Vision: Sensors, Theory, and Applications*. Springer-Verlag, New York, pp. 39–71.

Bar-Shalom, Y., and Li, X.-R. 1995. *Multitarget–Multisensor Tracking: Principles and Techniques*. YBS Publishing, Storrs, CT.

Brooks, M.J., Chojnacki, W., Gawley, D., and van den Hengel, A. 2001. What value covariance information in estimating vision parameters? *Proceedings of the 8th IEEE International Conference on Computer Vision (ICCV 2001)*, Vancouver, Canada.

Chahl, J.S., and Srinivasan, M.V. 1997. Reflective surfaces for panoramic imaging. *Applied Optics* 36(31):8275–8285.

Chai, L., Hoff, W.A., and Vincent, T. 2002. Three-dimensional motion and structure estimation using inertial sensors and computer vision for augmented reality. *Presence* 11(5):474–491.

Chong, K.S., and Kleeman, L. 1999. Feature-based mapping in real, large scale environments using an ultrasonic array. *International Journal of Robotics Research* 18(1):3–19.

Craig, J.J. 1989. *Introduction to Robotics: Mechanics and Control*. Addison-Wesley, Reading, MA.

Deans, M.C. 2002. *Bearings-Only Localization and Mapping*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA.

Deans, M., and Hebert, M. 2001. Experimental comparison of techniques for localization and mapping using a bearing-only sensor. *Experimental Robotics VII*, D. Rus and S. Singh, editors. Springer-Verlag, Berlin, pp. 395–404.

Foxlin, E.M. 2002. Generalized architecture for simultaneous localization, auto-calibration, and map-building. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2002)*, EPFL, Lausanne, Switzerland, September 30–October 4.

Foxlin, E., and Naimark, L. 2003. VIS-Tracker: a wearable vision-inertial self-tracker. *IEEE Virtual Reality Conference (VR 2003)*, Los Angeles, CA, March.

Gelb, A., editor. 1974. *Applied Optimal Estimation*. MIT Press, Cambridge, MA.

Guivant, J., and Nebot, E. 2001. Compressed filter for real time implementation of simultaneous localization and mapping. *International Conference on Field and Service Robotics (FSR 2001)*, Otaniemi, Finland, June, Vol. 1, pp. 309–314.

Heikkilä, J., and Silvén, O. 1997. A four-step camra calibration procedure with implicit image correction. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 1997)*, San Juan, Puerto Rico, pp. 1106–1112.

Horn, B.K.P. 1987. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A* 4(4):629–642.

Huster, A., and Rock, S.M. 2001a. Relative position estimation for intervention-capable AUVs by fusing vision and inertial measurements. *Proceedings of the 12th International Symposium on Unmanned Untethered Submersible Technology*, Durham, NH, August.

Huster, A., and Rock, S.M. 2001b. Relative position estimation for manipulation tasks by fusing vision and inertial measurements. *Oceans 2001 Conference*, Honolulu, HI, November, Vol. 2, pp. 1025–1031.

Huster, A., and Rock, S.M. 2003. Relative position sensing by fusing monocular vision and inertial rate sensors.

Proceedings of the 11th International Conference on Advanced Robotics (ICAR 2003), Coimbra, Portugal, July, Vol. 3, pp. 1562–1567.

Huster, A., Frew, E.W., and Rock, S.M. 2002. Relative position estimation for AUVs by fusing bearing and inertial rate sensor measurements. *Oceans 2002 Conference*, Biloxi, MS, October, pp. 1857–1864.

Jung, S.-H., and Taylor, C.J. 2001. Camera trajectory estimation using inertial sensor measurements and structure from motion results. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, Kauai, HI, December, Vol. 2, pp. 732–737.

Kanazawa, Y., and Kanatani, K. 2001. Do we really have to consider covariance matrices for image features? *Proceedings of the 8th IEEE International Conference on Computer Vision (ICCV 2001)*, Vancouver, Canada, July.

Lowe, D.G. 1999. Object recognition and local scale-invariant features. *Proceedings of the 7th International Conference on Computer Vision (ICCV 1999)*, Corfu, Greece, September, pp. 1150–1157.

Lucas, B.D., and Kanade, T. 1981. An iterative image registration technique with an application to stereo vision. *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, Vancouver, Canada, August, Vol. 2, pp. 674–679.

McLauchlan, P.F. 1999. The variable state dimension filter applied to surface-based structure from motion. Technical Report VSSP-TR-4/99, University of Surrey, Guildford, UK.

Montemerlo, M., and Thrun, S. 2003. Simultaneous localization and mapping with unknown data association using FastSLAM. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2003)*, Taipei, Taiwan.

Mukai, T., and Ohnishi, N. 1999. The recovery of object shape and camera motion using a sensing system with a video camera and a gyro sensor. *Proceedings of the 7th IEEE International Conference on Computer Vision (ICCV 1999)*, Corfu, Greece, September, pp. 411–417.

Nayar, S.K. 1997. Catadioptric omnidirectional camera. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 1997)*, San Juan, Puerto Rico, June, pp. 482–488.

Ollis, M., Herman, H., and Singh, S. 1999. Analysis and design of panoramic stereo vision using equi-angular pixel cameras. Technical Report CMU-RI-TR-99-04, Carnegie Mellon University, Pittsburgh, PA.

Poelman, C.J. 1995. *The Paraperspective and Projective Factorization Methods for Recovering Shape and Motion*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA.

Press, W.H., Teukolsky, S.A., Vetterling, W.T., and Flannery, B.P. 1992. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, Cambridge.

Qian, G., and Chellappa, R. 2001. Structure from motion using sequential Monte Carlo methods. *Proceedings of the 8th IEEE International Conference on Computer Vision (ICCV 2001)*, Vancouver, Canada, July, pp. 614–621.

Qian, G., Chellappa, R., and Zhang, Q. 2001. Robust structure from motion estimation using inertial data. *Journal of the Optical Society of America A* 18(12):2982–2997.

Rasmussen, C., and Hager, G.D. 2001. Probabilistic data association method for tracking complex visual objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23(6):560–576.

Rehbinder, H., and Ghosh, B.K. 2001. Rigid body state estimation using dynamic vision and inertial sensors. *Proceedings of the 40th IEEE Conference on Decision and Control (CDC 2001)*, Orlando, FL, December, pp. 2398–2403.

Smith, R., Self, M., and Cheeseman, P. 1990. Estimating uncertain spatial relationships in robotics. *Autonomous Robot Vehicles*, I.J. Cox and G.T. Wilfong, editors. Springer-Verlag, New York, pp. 167–193.

Strelow, D., Mishler, J., Singh, S., and Herman, H. 2001a. Extending shape-from-motion to non-central omnidirectional cameras. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2001)*, Wailea, HI, October.

Strelow, D., Mishler, J., Koes, D., and Singh, S. 2001b. Precise omnidirectional camera calibration. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, Kauai, HI, December, Vol. 1, pp. 689–694.

Szeliski, R., and Kang, S.B. 1994. Recovering 3D shape and motion from image streams using nonlinear least squares. *Journal of Visual Communication and Image Representation* 5(1):10–28.

Tomasi, C., and Kanade, T. 1992. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision* 9(2):137–154.

Weng, J., Cui, Y., and Ahuja, N. 1997. Transitory image sequences, asymptotic properties, and estimation of motion and structure. *IEEE Transactions on Image Analysis and Machine Intelligence* 19(5):451–464.

Wolf, P.R. 1983. *Elements of Photogrammetry*. McGraw-Hill, New York.

You, S., and Neumann, U. 2001. Fusion of vision and gyro tracking for robust augmented reality registration. *Proceedings of the IEEE Virtual Reality Conference (VR 2001)*, Yokohama, Japan, March, pp. 71–78.