Online Motion Estimation from Image and Inertial Measurements

D. Strelow and S. Singh

Carnegie Mellon University
Pittsburgh, PA 15213
{dstrelow, ssingh}@cs.cmu.edu

Abstract

We present two algorithms for estimating sensor motion from image and inertial measurements, which are suitable for use with inexpensive inertial sensors and in environments without known fiducials. The first algorithm is a batch method, which produces optimal estimates of the sensor motion, scene structure, and other parameters using measurements from the entire observation sequence simultaneously. The second algorithm recovers sensor motion, scene structure, and other parameters in an online manner, is suitable for use with long or "infinite" sequences, and handles sequences in which no feature is always visible.

We also describe initial results from running each algorithm on a sequence for which ground truth is available. We show that while image measurements alone are not sufficient for accurate motion estimation from this sequence, both batch and online estimation from image and inertial measurements produce accurate estimates of the sensors' motion.

1 Introduction

Cameras and inertial sensors are each good candidates for autonomous vehicle navigation because they do not project any detectable energy into the environment, estimate six degree of freedom motion, are not subject to outages or jamming, and are not limited in range. 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 acceleration, gravity, and bias in accelerometer measurements. Conversely, 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 present two algorithms for estimating sensor motion and scene structure from image and inertial measurements. The first is a batch algorithm that generates optimal estimates of the sensor motion, scene structure, and other parameters by considering all of the measurements from a camera, gyro, and accelerometer simultaneously. In many applications, this optimal estimate is of interest in its own right. In others, the optimal 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.

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 an online 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, this method also includes a principled method for incorporating points that become visible after initialization. This capability is essential for operation on most real image sequences.

We also describe initial results from running each algorithm on a sequence for which ground truth is available. We show that while image measurements alone are not sufficient for accurate motion estimation from this sequence, both batch and online estimation from image and inertial measurements produce accurate estimates of the sensors' motion.

2 Related Work

Most existing methods for estimating motion from image and inertial measurements are online methods, and in this section we review those online methods most closely related to our own. For a discussion of existing batch methods for estimating motion from image and inertial measurements, see [16].

Huster and Rock[4][5] describe two filters for estimating the six degree 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. In this method, the emphasis is on exploiting inertial information to reduce the visual information required for motion estimation, since visual information is expensive to process and is minimal in many underwater scenarios. The state and propagation model used in our online method are similar to those described by Huster and Rock, but their use of a single point is problematic when no one point is visible throughout the entire image sequence, and places a higher demand on the accuracy of the inertial sensors than our method. The experimental results presented in these papers were preliminary in that they were either synthetic, or estimated only a subset of the state.

You and Neumann[17] 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 Huster and Rock's 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[14] 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, *et al.*[13] describe an extended Kalman filter (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, in that the scene is recovered, but this system makes the implicit assumption that the scene points are visible in every image of the sequence. In later work, Qian and Chellappa[12] also investigated motion estimation from image and gyro measurements within a sequential Monte Carlo framework. In this case, the authors showed that the inclusion of gyro measurements significantly reduced the number of samples required for accurate motion estimation.

The system described by Mukai and Ohnishi[10] 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 Mukai and Ohnishi's method, 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.

3 Method

3.1 Overview

In this section we describe both our batch and online algorithms for motion estimation from image, rate gyro, and accelerometer measurements. These two methods each estimate the camera 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 direction with respect to the initial camera coordinate system; and the gyro and accelerometer biases. In addition, the online algorithm also tracks the sensor rig's angular velocity and linear acceleration.

The batch method is described in section 3.2. This method uses Levenberg-Marquardt to find estimates of the motion and other parameters using all of the image and inertial measurements at once. The batch algorithm is related to the online algorithm in two ways:

- When applied to an entire sequence of observations, the estimates produced by the batch method provide a gold standard for the evaluation of the online algorithm.
- When applied to a prefix of a sequence of observations, the batch method provides the estimates required to initialize the online method.

The batch algorithm typically converges in just a few iterations even if the initial estimate is poor, and can produce good estimates even if the estimates produced from image or inertial measurements alone are poor. However, the time required by the batch algorithm is cubic in both the number of images and the number of points, so it is not applicable to long or "infinite" sequences of observations.

In 3.3 we describe our online method, which uses an iterated extended Kalman filter (IEKF) to estimate the motion and other parameters. Unlike the batch algorithm, the online algorithm requires only constant time for each new image or inertial measurement, so it is applicable to long sequences. However, the online method is less robust to image feature mistracking and erratic sensor motion than the batch method.

3.2 Batch estimation

Our batch algorithm for estimating sensor motion 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 [11] for a discussion of Levenberg-Marquardt.

The error function is:

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

The image error term E_{image} is:

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

E_{image} specifies an image reprojection error given the six degree 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, and can be either conventional (e.g., orthographic or perspective) or omnidirectional.

All of the individual distance functions D are Mahalanobis distances. Common choices for the covariances defining the distances are uniform isotropic covariances (e.g., with $\sigma = 2$ pixels), or directional covariances determined using image texture in the vicinity of each feature[2][7].

The inertial error term is:

$$\begin{aligned} &E_{\text{inertial}} = \\ &\sum_{i=1}^{f-1} D\left(\rho_i, I_\rho(\tau_{i-1}, \tau_i, \rho_{i-1}, b_\omega)\right) + \\ &\sum_{i=1}^{f-1} D\left(v_i, I_v(\tau_{i-1}, \tau_i, \rho_{i-1}, b_\omega, v_{i-1}, g, b_\alpha)\right) + \\ &\sum_{i=1}^{f-1} D\left(t_i, I_t(\tau_{i-1}, \tau_i, \rho_{i-1}, b_\omega, v_{i-1}, g, b_\alpha, t_{i-1})\right) \end{aligned} \tag{3}$$

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_α 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. Over an interval $[\tau, \tau']$ where the camera coordinate system angular velocity is assumed constant, e.g., between the two inertial readings or between an inertial reading and an image time, I_ρ is defined as follows:

$$I_\rho(\tau_{i-1}, \tau_i, \rho, b_\omega) = r(\Theta(\rho) \cdot \Delta\Theta(\tau_i - \tau_{i-1}, b_\omega)) \tag{4}$$

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:

$$\Delta\Theta(\Delta t, b_\omega) = \exp\left(\Delta t \ \text{skew}(\omega)\right) \tag{5}$$

Here,

$$\text{skew}(\omega) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{6}$$

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

$$\omega = \omega' + b_\omega \tag{7}$$

and ω' is the biased camera coordinate system angular velocity given by the gyro. Over an interval $[\tau, \tau']$ when the world coordinate system linear acceleration is assumed constant, I_v and I_t are given by the familiar equations:

$$I_v(\tau_{i-1}, \tau_i, \ldots, b_\alpha) = v + a(\tau_i - \tau_{i-1}) \tag{8}$$

and

$$I_t(\tau_{i-1}, \tau_i, \ldots, t) = t + v(\tau_i - \tau_{i-1}) + \frac{1}{2}a(\tau_i - \tau_{i-1})^2 \tag{9}$$

where a is the world coordinate system acceleration

$$a = \Theta(\rho) \cdot (a' + b_\alpha) + g \tag{10}$$

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

The bias prior term E_{prior} is:

$$E_{\text{prior}} = f \cdot b_\alpha^T C_b^{-1} b_\alpha \tag{11}$$

The bias prior error term (11) is small if the accelerometer bias is near zero, and reflects our expectation that the accelerometer voltage corresponding to zero acceleration is close to the precalibrated value. The bias prior is most useful in those cases where the sensors undergo little change in rotation, in which case, the effects of gravity and accelerometer bias cannot be reliably distinguished from the observations, as equation (10) shows. As above, f is the number of images and b_α is the accelerometer bias. C_b is the accelerometer bias prior covariance, which we take to be isotropic with standard deviations 0.5 m/s^2.

As mentioned in section 3.1, the combined error function is minimized with respect to the six degree 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 direction with respect to the world coordinate system g; and the gyro and accelerometer biases b_ω and b_α.

Because the algorithm uses Levenberg-Marquardt, an initial estimate is required, but the algorithm converges from a wide variety of initial estimates. The online algorithm, described in the next section, could be used to generate a suitable initial estimate. For the experiments described in section 4, we have chosen the following initial estimate:

- All camera positions are placed at the origin
- The point positions are initialized by backprojecting the image position at which they first appear from the origin to a fixed distance in space.
- The velocities, gravity, and bias are all initialized to zero.

3.3 Online estimation

Our online method is an iterated extended Kalman filter (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 motion estimates 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 the inertial and image measurements are taken at the same rate.

The state vector is:

$$x(\tau) = \begin{bmatrix} \rho(\tau) \\ t(\tau) \\ X_0 \\ \vdots \\ X_{p-1} \\ v(\tau) \\ b_\omega \\ g \\ b_\alpha \\ \omega(\tau) \\ a(\tau) \end{bmatrix} \tag{12}$$

The components of the state vector are the same or similar to those described for the batch method in section 3.2. $\rho(\tau)$ and $t(\tau)$ are the Euler angles and translation specifying the camera-to-world transformation at time τ; $X_0, \ldots, X_{p-1}$ are the three-dimensional locations of the tracked points visible in the most recent image; $v(\tau)$ and $a(\tau)$ are the linear velocity and acceleration at time τ expressed in the world coordinate system; g, b_ω, b_α are the gravity vector direction with respect to the first camera coordinate system, the gyro bias, and the accelerometer bias, respectively; and $\omega(\tau)$ is the camera coordinate system angular velocity at time τ.

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

$$\dot{x}(\tau) = f(x(\tau)) + w \tag{13}$$

where w is a zero mean Gaussian noise vector with covariance Q. The nonzero components of f are $dt/d\tau = v$, $dv/d\tau = a$, and

$$\frac{d\rho}{d\tau} = \frac{d\rho}{d\Theta(\rho)} \frac{d\Theta(\rho)}{dt} \tag{14}$$

As in section 3.2, $\Theta(\rho)$ is the camera-to-world rotation matrix specified by ρ. $d\rho/d\Theta(\rho)$ is a 3×9 matrix that can be computed from the definition of $\Theta(\rho)$, and $d\Theta(\rho)/dt$ is a 9×1, flattened version of

$$\Theta(\rho)\,\mathrm{skew}(\omega) \tag{15}$$

where skew(ω) is given by equation (6). The noise covariance matrix Q is zero except for the 3×3 submatrices corresponding to ω and α, which are assumed to be isotropic.

Assuming that the true state propagates according to (13), a state estimate mean $\hat{x}(\tau)$ can be propagated using

$$\dot{\hat{x}}(\tau) = f(\hat{x}(\tau)) \tag{16}$$

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

$$\dot{P}(\tau) = F(\hat{x}(\tau))P(\tau) + P(\tau)F^T(\hat{x}(\tau)) + Q \tag{17}$$

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 nonzero blocks of F are $\partial^2\rho/\partial\tau\,\partial\rho$, which we compute numerically, and

$$\frac{\partial^2 t}{\partial\tau\,\partial v} = I_3 \tag{18}$$

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

$$\frac{\partial^2 \rho}{\partial\tau\,\partial\omega} = \frac{d\rho}{d\Theta(\rho)} \frac{d\mathrm{skew}(\omega)}{d\omega} \tag{20}$$

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

When image or inertial measurements are received, the state estimate mean and covariance are propagated from the previous measurement update time using (16) and (17), and then updated using the IEKF measurement update. For brevity, we concentrate here on the image and inertial measurement equations, and refer the reader to [3] 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:

$$\begin{bmatrix} x_{0,u} \\ x_{0,v} \\ x_{1,u} \\ x_{1,v} \\ \vdots \\ x_{p-1,u} \\ x_{p-1,v} \end{bmatrix} = \begin{bmatrix} \pi_u(C_{\rho,t}(X_0)) \\ \pi_v(C_{\rho,t}(X_0)) \\ \pi_u(C_{\rho,t}(X_1)) \\ \pi_v(C_{\rho,t}(X_1)) \\ \vdots \\ \pi_u(C_{\rho,t}(X_{p-1})) \\ \pi_v(C_{\rho,t}(X_{p-1})) \end{bmatrix} + n_v \tag{21}$$

Here, $(x_{0,u}, x_{0,v}), (x_{1,u}, x_{1,v}), \ldots, (x_{p-1,u}, x_{p-1,v})$ are the projections visible in the current image. As in section 3.2, π is the projection from a three-dimensional, camera coordinate system point onto the image; $C_{\rho,t}$ is the world-to-camera transformation specified by the Euler angles ρ and translation t; and X_j is the three-dimensional, world coordinate system position of point j. 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:

$$\begin{bmatrix} \omega' \\ a' \end{bmatrix} = \begin{bmatrix} \omega - b_\omega \\ \Theta(\rho)^T(a - g) - b_\alpha \end{bmatrix} + n_i \tag{22}$$

The top and bottom component equations of (22) are equivalent to (7) and (10), 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. $\Theta(\rho)$ is the camera-to-world rotation specified by the Euler angles ρ. ρ, ω, b_ω, g, and b_α, and a are the same members of the state that we encountered in (12). n_i is a vector of Gaussian noise, and in our experiments we have assumed that the gyro and accelerometer measurements are isotropic with $\sigma = 0.1$ radians/s and $\sigma = 0.1$ m/s^2, respectively.

To generate an initial mean and covariance for the state estimate, we use the batch method described in 3.2. 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 has been performed, we have adapted the stochastic map approach that Smith, *et al.*[15] describe for simultaneous localization and mapping (SLAM) from range data. We briefly describe our use of this approach in the remainder of this section, where we adopt a notation similar to that used in [15].

A newly visible point feature that has been extracted in the image sequence and tracked through a small, predetermined number of frames is a candidate for incorporation into the state estimate. We first estimate a mean and covariance for that point, relative to the most recent camera coordinate system. Assume for the moment that the camera-to-world estimates corresponding to the images in which the point has been tracked, which have been produced by the filter's recent image measurement update steps, are correct with respect to each other. Then, a least squares algorithm can be used to compute an estimate for the point's three-dimensional position z_c and covariance $C(z_c)$, relative to the current camera coordinate system, from the camera-to-world estimates, the observed image projections, and the assumed projection error covariances. The resulting covariance accounts for the noise in the image observations, but not for the uncertainty in the current camera-to-world estimate or for any relative error in the recent camera-to-world estimates.

We then use the following heuristic to determine if the point should be incorporated into the state estimate. The length l of the longest axis of the covariance ellipsoid described by $C(z_c)$ is computed, and the ratio $r = l/b$ is computed. Here b is the longest baseline, or translation between camera centers, of any two of the fixed camera positions used in the initialization. If r is less than some threshold (e.g., 0.5), the point is incorporated into the state estimate distribution, as described in the following paragraphs. If not, the point will remain a candidate, and r will be recomputed, on subsequent steps in which the point is visible. This criterion is invariant to the global scale of the estimate, and is nonincreasing as the number of cameras used in the initialization increases.

Now 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 camera 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, Smith *et al.*'s method transforms the camera 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

$$C(z_w) = G_x C(x) G_x^T + G_{z_c} C(z_c) G_{z_c}^T \tag{23}$$

$$C(x, z_w) = G_x C(x) \tag{24}$$

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 $C(x, z_w)$.

This method accounts for the noise in the image observations and for the uncertainty in the current camera-to-world estimate, but not for any relative error between the recent camera-to-world estimates.

In cases where relative error between the camera-to-world estimates is negligible, this method accurately initializes new points. However, in those cases where there is a significant relative error between the recent camera-to-world 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. To address this problem, we are investigating the application of the variable state dimension filter (VSDF)[9], which is capable of modeling the relative error between successive camera-to-world transformations, as an alternative to the IEKF for estimating motion from image, gyro, and accelerometer measurements.

4 Results

4.1 Overview

This section describes the results of running our algorithm on a perspective dataset obtained by mounting the sensor rig on a preprogrammed robotic arm. We compare the results from the batch image-and-inertial method, the online image-and-inertial method, and a batch image-only method.

4.2 Configuration

The sensor rig consists of a Sony XC-55 industrial vision camera paired with a 6 mm lens, 3 orthogonally mounted CRS04 rate gyros from Silicon Sensing Systems, and a Crossbow CXL04LP3 3 degree of freedom accelerometer. The gyros and accelerometer measure motions of up to 150 degrees per second and 4 g, respectively. The camera exposure time is set to 1/200 second to reduce motion blur.

Images were captured at 30 Hertz on a PC using a conventional frame grabber. To remove the effects of interlacing, only one field was used from each image, producing 640 × 240 pixel images. Voltages from the gyros and the accelerometer were simultaneously captured on the same PC at 200 Hertz with two separate Crossbow CXLDK analog-to-digital acquisition boards.

The camera intrinsic parameters (e.g., focal length and radial distortion) were calibrated using the method in [6]. This calibration also accounts for the reduced geometry of our one-field images. 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 individual gyro voltage-to-rate calibrations were determined using a turntable with a known rotational rate. The fixed gyro-to-camera and accelerometer-to-camera rotations were assumed known from the mechanical specifications of the mount.

4.3 Observations

To perform experiments with known and repeatable motions, the rig was mounted on a Yaskawa Perfomer-MK3 robotic arm, which has a maximum speed of 3.33 meters per second and a payload of 2 kilograms. The programmed motion translates the camera x, y, and z through seven pre-specified points, for a total distance traveled of about two meters. 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). The camera rotates through an angle of 270 degrees about the camera's optical axis during the course of the motion.

The observation sequence consists of 152 images, approximately 860 gyro readings, and approximately 860 accelerometer readings. 23 features were tracked through the image sequence, but only 5 or 6 appear in any one image. Points were tracked using the Lucas-Kanade algorithm[8][1], but because the sequence contains repetitive texture and large interframe motions, mistracking was common and was corrected manually. Example images from the sequence, with tracked points overlaid, are shown in Figure 1.

4.4 Estimates

We generated five estimates for the motion and other parameters:

- An optimal estimate from image and inertial measurements, using the batch algorithm described in 3.2
- Three estimates from the online algorithm described in section 3.3, using 15, 30, and 45 images in the batch initialization
- A batch estimate using image observations only

In this section we briefly compare the estimated motions against ground truth.

The error statistics for the five motion estimates are summarized in Figure 1. For the four estimates from image and inertial data, the estimates are accurate, with the optimal estimates closest to ground truth, and with the exception of the rotation error from the online result initialized with 30 images, the online estimates improving as the number of images used in the initialization increases. The x, y translations generated by the optimal image-and-inertial algorithm and by the online algorithm using 15 images for initialization are shown in Figure 2(a), as the dashed and solid lines, respectively. The x, y positions of the seven known ground truths points are shown in this figure as squares. In this table, the reported rotation errors are the average scalar angles from the angle-axis representation of the rotations that map the estimated rotations to the corresponding ground truth rotations.

On the other hand, the optimal motion estimate from image measurements only is clearly wrong, and the errors versus ground truth are much higher. This estimate was found by applying a batch image-only

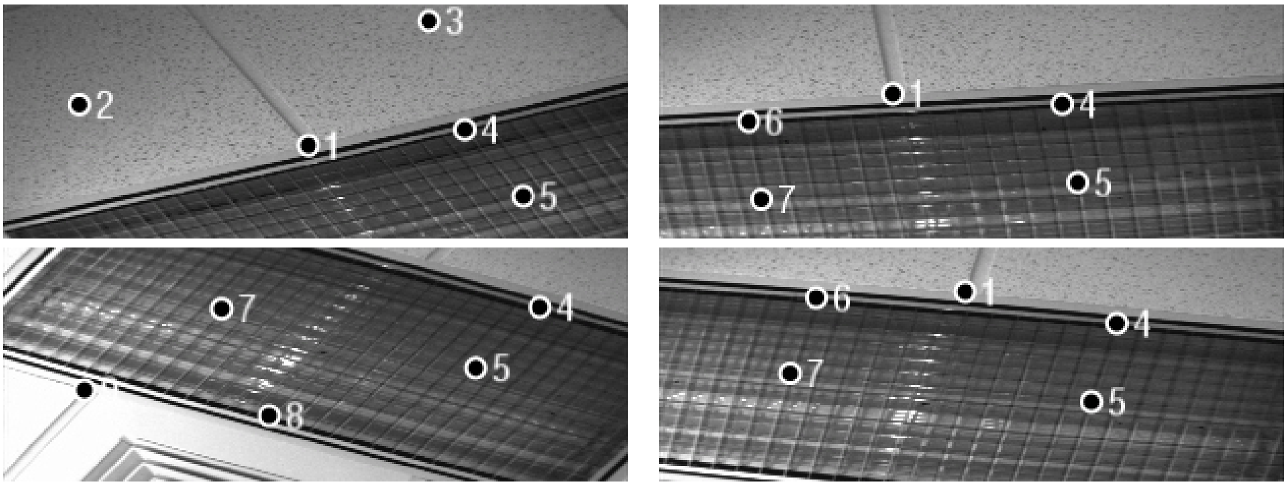


Figure 1: *Images 16, 26, 36, and 46 from the 152 image sequence, with the tracked features overlaid, are shown clockwise from the upper left. As described in section 4.2, the images are one field of an interlaced image, so their height is half that of the full image.*

algorithm to the same image measurements used to find the optimal image-and-inertial estimate, using the optimal image-and-inertial estimate as the initial estimate. Applying this algorithm reduces the image error versus the optimal image-and-inertial estimate, but at the cost of introducing a large error in the motion estimate.

When synthetic errorless image measurements are generated from the optimal image-and-inertial camera and three-dimensional point estimates, and the batch image-only algorithm is applied to a perturbed version of the optimal image-and-inertial estimate, the optimal image-and-inertial estimate is recovered from the image data. It follows that, while the shape and motion in this example are not strictly degenerate, estimating the motion from image observations only for this dataset is highly sensitive to the small measurement errors in the image observations. Adding inertial data allows the correct motion to be recovered.

The x, y translations from the batch, image-only estimate are shown as the erratic solid line in Figure 2(b). The smooth dashed line in Figure 2(b) shows the x, y translations from the optimal image-and-inertial estimate, and is the same as shown in Figure 2(a). As in Figure 2(a), the x, y locations of the seven known ground truth points are shown as squares.

5 Conclusion

Image and inertial sensing are highly complimentary modalities, and we have described both batch and online algorithms that exploit this complimentary nature. Our initial experiment shows that even in a case where optimal batch estimation from image measurements alone is not sufficient to recover accurate motion, both batch and online estimation can recover accurate motion when both image and inertial measurements are employed.

Our upcoming work will evaulate our online algorithm with omnidirectional data, and begin the development and evaluation of a hybrid batch-online algorithm for estimating motion from image and inertial measurements. This method will be based on the batch method described in section 3.2 and the variable state dimension filter (VSDF)[9], and should provide a more accurate mechanism for initializing newly acquired image point features.

References

[1] S. Birchfield. KLT: An implementation of the Kanade-Lucas-Tomasi feature tracker. `http://vision.stanford.edu/~birch/klt/`.

[2] M. J. Brooks, W. Chojnacki, D. Gawley, and A. van den Hengel. What value covariance information in estimating vision parameters? In *Eighth IEEE International Conference on Computer Vision (ICCV 2001)*, Vancouver, Canada, 2001.

[3] A. Gelb, editor. *Applied Optimal Estimation.* MIT Press, Cambridge, Massachusetts, 1974.

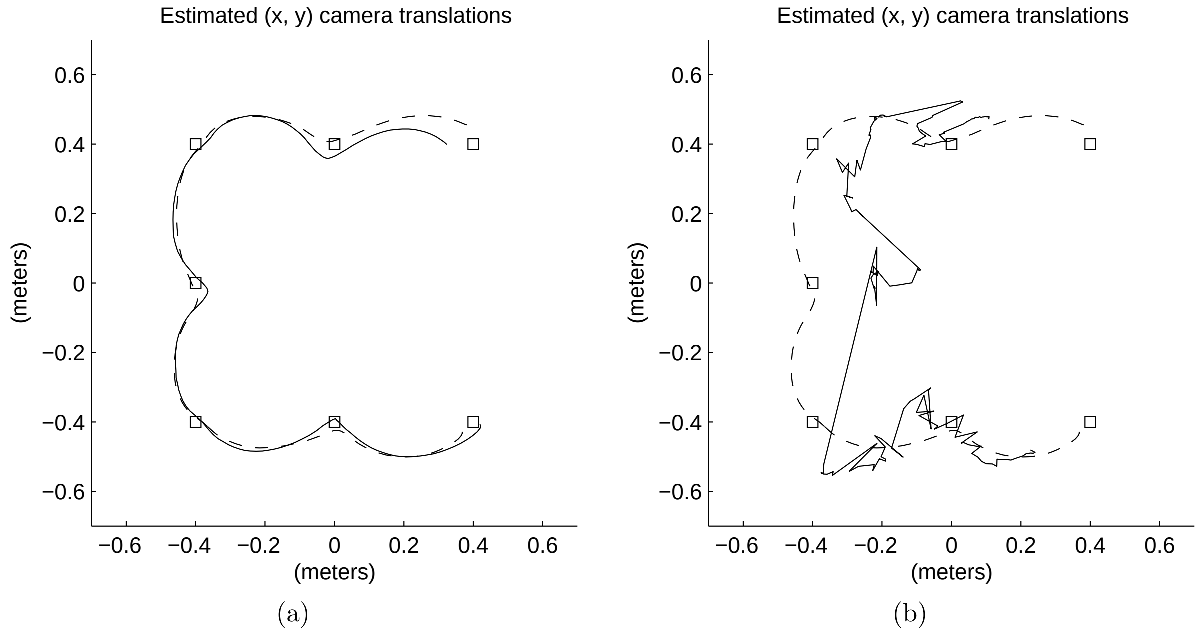
[4] A. Huster and S. M. Rock. Relative position estimation for intervention-capable AUVs by fusing vision and inertial measurements. In *Twelfth International Symposium on Unmanned Untethered Submersible Technology*, Durham, New Hampshire, August 2001.

[5] A. Huster and S. M. Rock. Relative position estimation for manipulation tasks by fusing vision and inertial measurements. In *Oceans 2001 Conference*, volume 2, pages 1025–1031, Honolulu, November 2001.

[6] Intel corporation open source computer vision library. `http://www.intel.com/research/mrl/research/opencv/`.

[7] Y. Kanazawa and K. Kanatani. Do we really have to consider covariance matrices for image features? In *Eighth IEEE International Conference on Computer Vision (ICCV 2001)*, Vancouver, Canada, July 2001.

[8] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Seventh International Joint Conference on*



(a)

(b)

Figure 2: *Left: The x, y translation estimates generated by the batch image-and-inertial method and by the online method with a 15 image initialization are shown as the dashed and solid lines, respectively. Right: The x, y translation estimates generated by the batch image-and-inertial method and by the batch image-only method are shown as the dashed 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.*

	rotation error (radians)	translation error (centimeters)
batch, image and inertial	0.15 / 0.23	3.4 / 5.4
online, 45 image initialization	0.26 / 0.32	4.3 / 7.1
online, 30 image initialization	0.36 / 0.41	5.6 / 7.3
online, 15 image initialization	0.28 / 0.38	6.4 / 8.7
batch, image only	0.49 / 0.62	19.0 / 31.1

Table 1: *Errors versus ground truth for the four estimates. Each entry gives the average error before the slash and the maximum error after the slash.*

Artificial Intelligence, volume 2, pages 674–679, Vancouver, Canada, August 1981.

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

[10] T. Mukai and N. Ohnishi. The recovery of object shape and camera motion using a sensing system with a video camera and a gyro sensor. In *Seventh IEEE International Conference on Computer Vision (ICCV 1999)*, pages 411–417, Corfu, Greece, September 1999.

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

[12] G. Qian and R. Chellappa. Structure from motion using sequential Monte Carlo methods. In *Eighth IEEE International Conference on Computer Vision (ICCV 2001)*, pages 614–621, Vancouver, Canada, July 2001.

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

[14] H. Rehbinder and B. K. Ghosh. Rigid body state estimation using dynamic vision and inertial sensors. In *Fortieth IEEE Conference on Decision and Control (CDC 2001)*, pages 2398–2403, Orlando, Florida, December 2001.

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

[16] D. Strelow and S. Singh. Optimal motion estimation from visual and inertial measurements. In *IEEE Workshop on Applications of Computer Vision (WACV 2002)*, Orlando, Florida, December 2002.

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