J Intell Robot Syst (2009) 55:203–221
DOI 10.1007/s10846-008-9294-6

UNMANNED SYSTEMS PAPER

Inertially Aided Visual Odometry for Miniature Air Vehicles in GPS-denied Environments

Bryce B. Ready · Clark N. Taylor

Received: 14 April 2008 / Accepted: 11 November 2008 / Published online: 4 December 2008

Abstract Unmanned miniature air vehicles (MAVs) have recently become a focus of much research, due to their potential utility in a number of information gathering applications. MAVs currently carry inertial sensor packages that allow them to perform basic flight maneuvers reliably in a completely autonomous manner. However, MAV navigation requires knowledge of location that is currently available only through GPS sensors, which depend on an external infrastructure and are thus prone to reliability issues. Vision-based methods such as Visual Odometry (VO) have been developed that are capable of estimating MAV pose purely from vision, and thus have the potential to provide an autonomous alternative to GPS for MAV navigation. Because VO estimates pose by combining relative pose estimates, constraining relative pose error is the key element of any Visual Odometry system. In this paper, we present a system that fuses measurements from an MAV inertial navigation system (INS) with a novel VO framework based on direct image registration. We use the inertial sensors in the measurement step of the Extended Kalman Filter to determine the direction of gravity, and hence provide error-bounded measurements of certain portions of the aircraft pose. Because of the relative nature of VO measurements, we use VO in the EKF prediction step. To allow VO to be used as a prediction, we develop a novel linear approximation to the direct image registration procedure that allows us to propagate the covariance matrix at each time step. We present offline results obtained from our pose estimation system using actual MAV flight data. We show that fusion of VO and INS measurements greatly improves the accuracy of pose estimation and reduces the drift compared to unaided VO during medium-length (tens of seconds) periods of GPS dropout.

A preliminary version of this paper was presented at the 2007 American Controls Conference, under the title “Improving Accuracy of MAV Pose Estimation using Visual Odometry”.

B. B. Ready (✉) · C. N. Taylor
Department of Electrical and Computer Engineering, Brigham Young University,
Provo, UT 84602, USA
e-mail: bready@byu.edu

C. N. Taylor
e-mail: taylor@ee.byu.edu

Keywords Visual odometry · Visually aided inertial navigation · Pose estimation · Kalman filter · Sensor fusion · Inertially aided visual odometry · GPS-denied navigation · Miniature unmanned air vehicles

1 Introduction and Motivation

Unmanned Air Vehicles in the mini- to micro- size ranges (<6 ft wingspan, referred to as MAVs) have been a focus of much research activity in the last few years. MAVs provide an attractive platform for use in various applications, including remote surveillance, mapping, and target tracking. By virtue of being unmanned, they can be used to collect data in dangerous or inconvenient situations, while their small size, light weight, and inexpensive construction and maintenance costs allow them to be used in many applications where UAV technology was previously not feasible.

Many of the advantages associated with fixed-wing MAVs stem from their ability to operate autonomously at two levels. On a basic level, MAVs must be able to autonomously perform low-level flight tasks such as taking off, flying straight & level, climbing, descending, banking, etc. On a higher level, MAVs must be able to combine these maneuvers in order to fly to specific locations, follow specific trajectories, and otherwise *navigate* autonomously. In order to perform these tasks, MAVs must be able to estimate their own pose, which consists of location (t_x, t_y, t_z) and attitude, which is commonly expressed using Euler angles for yaw(ψ), pitch(θ), and roll(ϕ). We can divide these six pose parameters into two sets of three parameters. The first group, which we will refer to as *aviation parameters* (pitch, roll, and altitude), are required to perform basic autonomous maneuvers, while the second, which we term *navigation parameters* (x, y, yaw), are additionally required for autonomous navigation. Current MAV systems [1–4] carry a simple Inertial Navigation System (INS) consisting of accelerometers, rate gyros, and pressure sensors. These sensors can provide only relative information about navigation parameters [1]; thus INS based estimates of navigation parameters will drift without bound over time. Because MAVs use low-cost, lightweight MEMS-based inertial sensors, errors in navigation parameter estimates from the INS alone typically increase extremely rapidly, often becoming unacceptably large within a few seconds. For this reason, current MAV systems rely on the Global Positioning System (GPS) to provide estimates of the navigation parameters. While GPS does provide bounded-error estimates of geo-location (x,y) and heading, it makes the operation of the MAV dependent on external infrastructure—the network of orbiting GPS satellites. Signals from these satellites can be blocked, both by environmental obstacles (eg. urban terrain) and by delibrate or unintentional jamming [5]. Much effort has therefore been directed at finding ways to reduce the dependence of MAV platforms on GPS.

Vision-based pose estimation techniques are a promising way to estimate pose in GPS-denied environments, and thus reduce dependence on GPS. Vision sensors are typically already available on MAV platforms, and provide a rich source of information about the environment. There are two main methods of performing vision-based pose estimation reported in the literature. Visual Simultaneous Localization And Mapping, or Visual SLAM (e.g. [6–10]) is perhaps the most elegant and complete method. SLAM algorithms in general estimate both robot state and the location of

[1] As we shall see, INS sensors can provide absolute measurements of aviation parameters. Exploitation of this fact is a critical component of this paper.

landmarks in the environment simultaneously. If perfected, a solution to the Visual SLAM problem would allow a robot to function in a truly autonomous manner, using vision and other sensors to navigate an unfamiliar environment as a human being can, without relying on fiducial markings, GPS signals, or other external infrastructure. However, there are still a number of problems with Visual SLAM which make its practical application challenging. SLAM algorithms in general have non-constant computation time as more and more landmarks are observed, and managing and reducing this computational load is still a focus of ongoing research. Current Visual SLAM systems can use either a video camera alone (e.g. [10]) or a video camera with low-quality MEMS-based inertial sensors (e.g. [9]), and can provide impressive navigational accuracy and stability. Unfortunately, they rely on the assumption that the environment is bounded and relatively small, so that an excessive and growing collection of landmarks does not slow down the processor. This assumption is not valid for MAV platforms, which must navigate in extremely large unexplored environments. Furthermore, in navigation applications, the landmark locations are typically not of interest, meaning that much of this computational burden, while providing greater accuracy, does not contribute directly to the desired result.

In this work, our proposed scenario is a system in which GPS is typically available, but may drop out for an intermediate length of time (several tens of seconds). Our goal, then, is not to produce estimates that do not accumulate error, but estimates that accumulate error slowly enough that they are still accurate within this time window. Visual Odometry (VO) methods are a viable means of performing vision-based pose estimation under this scenario. Named in analogy to wheeled-robot odometry, VO methods use computer vision algorithms to estimate the relative orientation between image frames. With any VO method, then, the key goal is to somehow decrease the amount of error introduced at each step, thereby slowing the growth of error sufficiently that pose estimates are valid within a desired time window.

In this work, we employ two strategies to improve the error characteristics of VO. First, we utilize a novel VO system based upon prior work by Dellaert et al. [11, 12]. Our VO system uses a direct image registration algorithm, estimating a single parametric transformation mapping pixels in an image to ground locations. This is in contrast to more standard feature-based image registration methods, which track a series of feature points and use their motion to infer the relative pose between frames. Direct registration is generally able to produce more accurate results than such feature-based methods; this increased registration accuracy helps to slow the accumulation of pose error.

The second method we use to slow VO error growth is to fuse VO measurements with INS data in an EKF framework. In the literature, a technique known as vision-assisted inertial navigation [13–18, references therein] is the usual method of doing this. Current techniques use the INS to provide relative measurements of the pose parameters, and these relative measurements are integrated in the EKF time update step to provide absolute pose. The relative pose measurements provided by VO are used in the EKF measurement update step to correct drift in this INS pose estimate. SLAM methods which incorporate inertial sensors similarly use inertial measurements in the time update (e.g. [6, 9]). In this work, we fuse INS and vision in a different way: rather than performing vision-assisted inertial navigation, we propose to perform inertially-aided visual odometry. The distinction is somewhat subtle, but nevertheless significant. One key contribution of this work is that we interpret INS measurements in a different way than the standard vision-aided inertial navigation literature, allowing us to make use of a low-quality MEMS-based INS without inducing INS integration error. This is in contrast to most

vision-assisted inertial navigation literature, where the assumption is usually made that a relatively high fidelity INS system is available, or else that another sensor can slow or stop the drift induced by a low quality INS. Another interpretation is possible, however: the three-axis accelerometers in the INS measure the acceleration of the aircraft in each dimension, which for a fixed wing aircraft, will primarily measure the direction of the gravity vector. This information allows us to directly compute the pitch and roll of the aircraft [4]. INS data has been interpreted in this way in the computer vision community to estimate the pose of a camera [19, 20] as well as to perform such tasks as scene reconstruction and camera calibration [21–23]. Combining these pitch and roll estimates with the altitude estimate obtained from the INS pressure sensors, we have a complete estimate of the aviation pose parameters of the aircraft. Many current MAV systems interpret INS data in this way [1–3]: however, to our knowledge ours is the first work to apply this information to MAVs in a VO setting. Because we interpret the INS data as providing absolute measurements of the aviation pose parameters of the aircraft, we use INS data in the EKF measurement update, rather than as a time update. Since VO measures the relative pose between frames, we use VO measurements in the time-update step. Thus, VO is the main means of estimating aircraft pose, and its accuracy is improved by incorporating measurements of the aviation parameters from the INS. Use of VO as the time-update step in an EKF framework requires that we be able to propagate both the MAV pose and its covariance matrix. The standard EKF method for doing this (linearizing the time update function) will not work with our proposed VO system, as it is neither differentiable nor available in closed form. The second major contribution of this paper is thus a novel means of estimating uncertainty associated with our VO estimates.

In the remainder of this paper, we first give a general overview of our pose estimation system (Section 2), and then describe our VO method and the underlying direct image registration method (Section 3), giving further background on existing VO methods. We then develop our proposed method of uncertainty propagation through the VO system (Section 4). Finally, we will present MAV flight results obtained using our method (Section 5), and offer some concluding remarks (Section 6).

2 Proposed System Overview

The operation of our pose estimation framework is summarized in Fig. 1. The goal of this system is to estimate the pose of the aircraft, which we represent with a 6-vector χ:

$$\chi = \begin{bmatrix} t_x & t_y & t_z & \psi & \theta & \phi \end{bmatrix}^T,$$

composed of a 3-D location state (t_x, t_y, t_z) and three Euler angles (ψ, θ, ϕ) representing attitude (yaw [2], pitch, and roll). Video frames Y_n and Y_{n-1} from an MAV camera

[2] Yaw is typically defined as the compass direction in which the nose of the aircraft is pointing, while heading is the direction in which the aircraft is moving. If the aircraft is flying with a 'crab angle' (i.e. the nose is not pointing exactly in the direction of flight) due to wind conditions, these two quantities will not be identical. Pose estimates obtained using VO provide yaw information, while heading estimates can be provided directly by GPS. Heading can also be estimated by using the difference in location estimates.

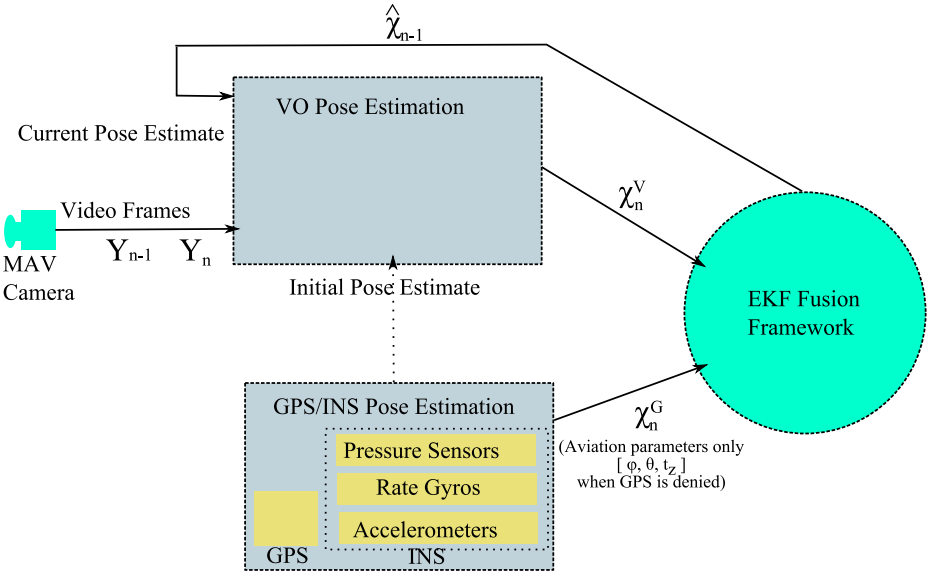

Fig. 1 Layout of our GPS/INS/VO pose estimation system

are fed into the VO system, along with current pose estimates. The VO system then produces an estimated pose χ_n^V for each new video frame Y_n, using the image data Y_{n-1} and estimated pose $\hat{\chi}_{n-1}$ of the previous frame.

Independently, information from other on-board sensors are fed into the GPS/INS pose estimation system, which separately estimates a pose χ_n^G of the aircraft at the time each frame was taken. If GPS is unavailable, this estimate includes only the aviation parameters, which can be obtained from the INS alone. In our system, altitude is computed directly from the autopilot pressure sensors, while pitch and roll are computed by combining accelerometer and rate gyro measurements in a complimentary filter. Pitch/roll states are propagated forward in time using rate measurements from the gyroscopes. Accelerometer measurements are used to give bounded error estimates of pitch and roll, computed according to the following formula:

$$\begin{aligned} \left[\, A_x \;\; A_y \;\; A_z \,\right] &\equiv \text{accelerometer readings} \\ \phi_{acc} &= \tan^{-1}\left(\frac{A_y}{-A_z}\right) \\ \theta_{acc} &= \tan^{-1}\left(\frac{A_x}{-A_y \sin(\phi) - A_z \cos(\phi)}\right). \end{aligned}$$

The weighted sum of the accelerometer based measurements and predicted states become the new pitch/roll estimates. If the aircraft is turning, the accelerometers will measure not only the lift force opposing gravity, but also extra acceleration from the d'Alembert force due to centripetal acceleration. To help account for this fact, the relative weight of the accelerometer-based measurements is reduced as the turn rate of the MAV increases, causing the system to rely more heavily on the propagated

values. In our experience, this method produces sufficiently accurate estimates to enable MAV navigation (see [4]).

Because these INS based pose measurements have bounded error as we have discussed, we model this estimate as the true pose of the aircraft corrupted with zero-mean Gaussian noise (ν):

$$\text{GPS unavailable: } \chi_n^G = \begin{bmatrix} t_z & \theta & \phi \end{bmatrix}^T + \nu. \tag{1}$$

We desire to fuse these partial measurements of aircraft pose with the pose information from the VO system using an EKF framework. The standard EKF framework estimates the state of a system given knowledge of the system dynamics and measurements that are functions of the state:

$$\chi_n = \mathbf{f}(\chi_{n-1},\, u_n) + \eta \tag{2}$$

$$y_n = \mathbf{h}(\chi_n) + \nu \tag{3}$$

where η and ν are zero-mean, Gaussian random vectors with covariance matrices Q and R respectively. At each time step, we estimate the new state of the system and its covariance from the previous state:

$$\hat{\chi}_n^- = \mathbf{f}\left(\hat{\chi}_{n-1},\, u_n\right) \tag{4}$$

$$\hat{P}_n^- = F\hat{P}_{n-1}F^T + Q \tag{5}$$

where the matrix F is the jacobian of the system dynamics function:

$$F = \left.\frac{\partial \mathbf{f}}{\partial \chi}\right|_{\chi=\hat{\chi}_{n-1}, u=u_n}. \tag{6}$$

When a measurement becomes available, we incorporate the information it provides about the state, performing what is known as a measurement update step:

$$\hat{\chi}_n = \hat{\chi}_n^- + K\left(y_n - \mathbf{h}\left(\hat{\chi}_n^-\right)\right) \tag{7}$$

$$\hat{P}_n = (I - KH)\,\hat{P}_n^- \tag{8}$$

where H is the jacobian of the measurement function (analogous to F):

$$H = \left.\frac{\partial \mathbf{h}}{\partial \chi}\right|_{\chi=\hat{\chi}^-} \tag{9}$$

As we have discussed, we use the aviation parameters available from the INS as our measurement function. This means that our $\mathbf{h}$ function is in fact just a linear operator, and we can find H directly:

$$\begin{aligned} y = \begin{bmatrix} \theta \\ \phi \\ t_z \end{bmatrix} &= \mathbf{h}(\chi) \\ &= \underbrace{\begin{bmatrix} 0\,0\,0\,0\,1\,0 \\ 0\,0\,0\,0\,0\,1 \\ 0\,0\,1\,0\,0\,0 \end{bmatrix}}_{H} \chi. \end{aligned} \tag{10}$$

By contrast, the time update in our EKF will come from VO, meaning that our **f** function represents the VO system as follows:

$$\begin{aligned} \chi_n^- &= \mathbf{f}(\chi_{n-1}) \\ &= \text{vo\_system}(\chi_{n-1}, Y_{n-1}, Y_n). \end{aligned} \tag{11}$$

In the next section, we will describe the operation of our VO pose estimation system, which is represented by the function $\mathbf{f}(\cdot)$. We will then proceed in Section 4 to approximate F, enabling us to implement Eq. 5.

3 Visual Odometry System

Several VO frameworks are delineated in the literature. Most commonly used existing methods function by detecting and tracking feature points between frames in a video sequence, and using the motion of these points to estimate the relative pose between frames. This is done by using feature point motion to estimate either the essential matrix [24–28] or a homography [29, 30] relating pairs or sets of frames, and then decomposing these matrices [31, 32] to find the relative pose.

All VO methods share a common implementation challenge that must be addressed to allow absolute pose to be estimated. This problem is that of determining the scale factor of the estimated relative pose. Because a video camera is a bearing-only sensor and provides no depth information, it is impossible to distinguish whether a pair of frames are widely separated and observing large, distant objects or closely spaced and observing small, nearby objects. If care is not taken to ensure that relative pose estimates are expressed in the same scale then gross errors in absolute pose estimates can be accumulated very rapidly. This problem is typically addressed in the literature by triangulating the 3-D location of feature points common between two frame pairs.

In this work, we propose a novel VO strategy, based upon prior work by Dellaert et al. [11, 12]. Rather than infer the inter-frame relative pose by using the motion of extracted feature locations, we directly compute the absolute pose of the second frame from the absolute pose of the first frame by means of an iterative image registration approach. This approach works by projecting the first observed image onto the terrain and rendering a view of this projected data from the currently estimated pose of the second frame. The estimated pose of the second frame is iteratively adjusted by means of image jacobians to make the second frame and the re-projected first frame match as closely as possible. Thus, this method directly estimates a single parametric transform using the captured image as a whole, rather than the estimated motion of selected feature points. This fact typically allows direct image registration methods to provide greater accuracy in image registration. Furthermore, since the absolute pose of each frame is estimated, the scale factor problem is handled implicitly.

Direct image registration such as we are performing depends upon three main assumptions: (1) that the scene being imaged is planar, (2) that all image motion is due to camera motion, meaning that motion due to independently moving objects is negligible, (3) there is sufficient texture in the imaged scene to allow the iterative descent registration algorithm to avoid converging to a local minimum.

While the planarity assumption is not appropriate for ground-based robots, it is generally a reasonable assumption in other applications. Unmanned Underwater Vehicle applications [33–35] commonly make this assumption. As MAVs are typically relatively distant from the terrain they observe, this assumption is often workable in many fixed-wing MAV situations. In addition, we have found that our direct registration algorithm is robust to small amounts of non-planarities (e.g. trees, small structures) in the images. This is a reasonable scenario for many applications; low flights in complicated (e.g. urban) terrain will violate this assumption, however.

The assumption that objects do not move independently is not problematic in most environments. At typical MAV altitudes, any moving objects will occupy only a tiny fraction of a captured image, and thus will have little effect on image registration. This assumption will fail in some cases (i.e. viewing a highway with heavy, fast-moving traffic) but will be a good assumption in many others.

Sufficient image texture is also usually a good assumption: most UAV flights are daytime flights, and most real flight environments contain significant visual texture. Certain flight environments could of course cause the direct image registration to not converge. In general, however, our practical experience suggests that direct registration is more robust than feature-based methods in low-texture video.

Our VO algorithm for estimating the current pose of the MAV is illustrated in Fig. 2. To estimate the pose of video frame Y_2 (or rather, the pose of the aircraft when this frame was captured), we assume that a previous frame Y_1 is available with associated pose information $\hat{\chi}_1$ for that frame. We also assume that we have a coarse estimate of the pose from which Y_2 was captured $\left(\chi_2^E\right)$. This coarse estimate could be obtained from the current GPS/INS estimate χ_2^G, the pose of the previous frame $\hat{\chi}_1$, or the result of a quick feature-based motion estimation algorithm. When registering sequential video frames (30 fps frame rate), simply using the pose of the previous frame as the initial pose estimate $\left(\text{i.e. let } \chi_2^E = \hat{\chi}_1\right)$ was found to produce the most

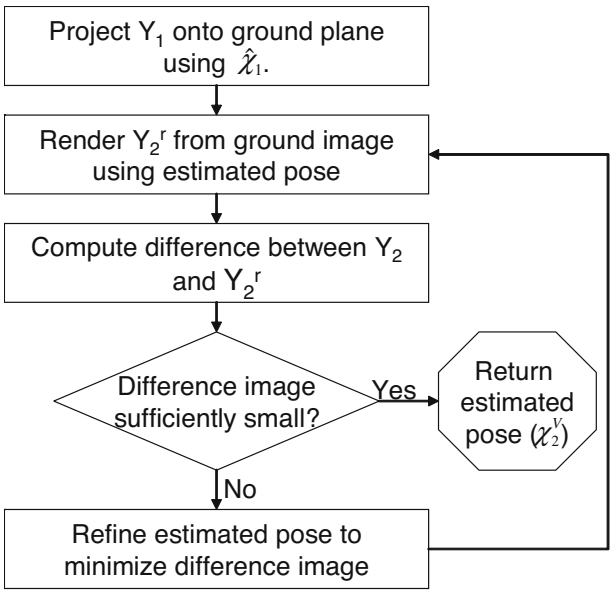

Fig. 2 Our method for computing the pose of the MAV when image 2 was captured assuming image 1's pose is perfectly known

rapid convergence, as the aircraft typically did not move far enough in one frame interval to make this a bad initial guess. The goal of our algorithm is to compute a more refined estimate χ_2^V of the MAV pose when frame Y_2 was captured.

The first step shown in Fig. 2 is to project Y_1 onto a ground image. The projection process assumes that the terrain over which the MAV is flying is planar and horizontal and uses the estimated pose $\hat{\chi}_1$ with respect to this ground plane to produce an ortho-rectified image of the region of ground observed by Y_1. This projection is computed by perspective warping: i.e. the ground image is a perspectively warped version of the captured image. Rather than interpolating between pixel values, a gaussian point spread function is assumed to act on each pixel value, and the perspective projection of this point spread function determines the amount by which each pixel in Y_1 affects each ground image pixel. Further explanation and details of this warping process are given by Dellaert et al. [11, 12].

Once we have inferred the appearance of the ground plane using image Y_1, we desire to iteratively refine our initial pose estimate for Y_2. At the kth iteration, we produce a rendered image $Y_2^r(k)$ of this ground image using the current pose estimate $\chi_2^V(k)$ for image Y_2. This rendering process is simply the inverse of the projection process, and is performed using a projectively distorted gaussian point spread function to determine the amount by which each ground pixel affects a given pixel in $Y_2^r(k)$. The rendered image $Y_2^r(k)$ represents the visual information in Y_1 as it would appear in Y_2, assuming that the poses $\hat{\chi}_1$ and $\chi_2^V(k)$ used for projection and rendering were accurate. The difference or residual image $\left(Y_2^r(k) - Y_2\right)$ provides information about the error in the current pose estimate $\chi_2^V(k)$. To determine an update $\Delta\chi_2^V(k)$ to the current pose estimate, we use a variant of the popular Lucas-Kanade image registration method [36], based on Gauss-Newton gradient descent. We attempt to choose $\Delta\chi_2^V(k)$ to minimize the pixel for pixel squared magnitude of the residual image:

$$J = \sum_{p \in \mathbb{P}} \left(Y_{2,p} - Y_{2,p}^r\right)^2 \tag{12}$$

where $\mathbb{P}$ is the set of all pixels in image Y_2, $Y_{2,p}^r$ are the pixels in the rendered image and $Y_{2,p}$ are the pixels in Image 2. A Gauss-Newton iteration essentially consists of computing partial derivatives of the residual image with respect to all of the pose parameters. Each of these partial derivative images approximates the change in the residual image caused by a differential change in the associated pose parameter. The goal at each iteration is to express the residual image as a weighted sum of the different Jacobian images, after which the pose is changed according to these weights. A graphical example of a single iteration is shown in Fig. 3.

As discussed in [11, 12], the partial derivatives or "Jacobian Images" of the residual can be approximated using the chain rule as follows:

$$\underbrace{\frac{\partial Y_2^r}{\partial \square}}_{\text{Jacobian Image}} = \underbrace{\frac{\partial Y_2^r}{\partial x}}_{\nabla_x} \frac{\partial x}{\partial \square} + \underbrace{\frac{\partial Y_2^r}{\partial y}}_{\nabla_y} \frac{\partial y}{\partial \square} \tag{13}$$

Each of the terms in this equation represents an "image" or matrix of values, one for each pixel location. The symbol $\square$ represents one of the six pose parameters $\left[t_x, t_y, t_z, \psi, \theta, \phi\right]$, and the terms labeled ∇_x and ∇_y are gradients of the rendered

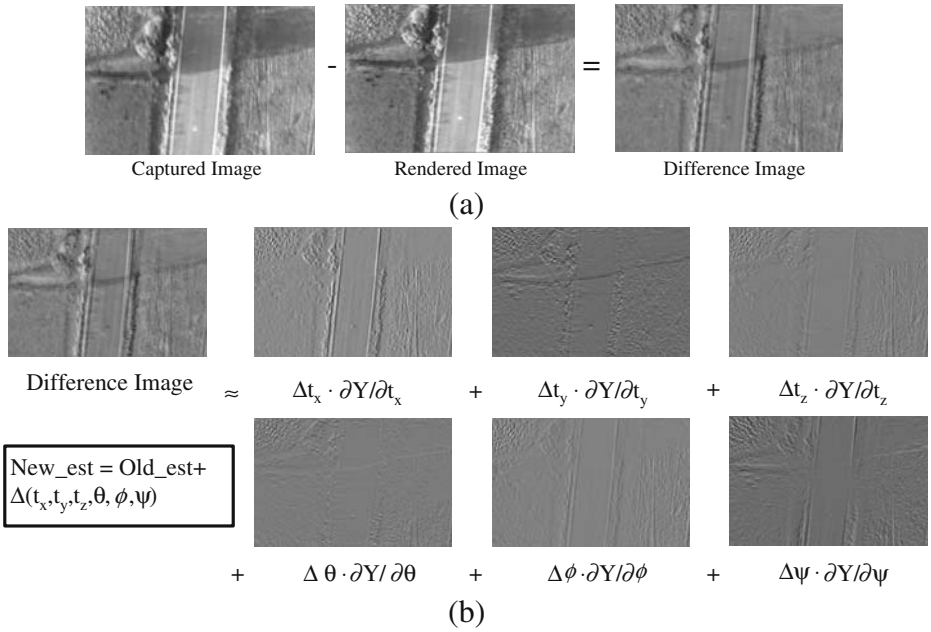

Fig. 3 An example iteration of the image registration process using our Gauss-Newton registration method. In subfigure **a**, a difference image is created to evaluate how accurate the current pose estimate is. In subfigure **b**, the new pose (New_est) is computed using the difference image and the Jacobian images

image, i.e. partial derivatives of the luminance function in the vertical and horizontal image directions. The $\frac{\partial y}{\partial \square}$ and $\frac{\partial x}{\partial \square}$ terms represent the differential location change of the image of the preimage of each image point. That is, each pixel location (x, y) is

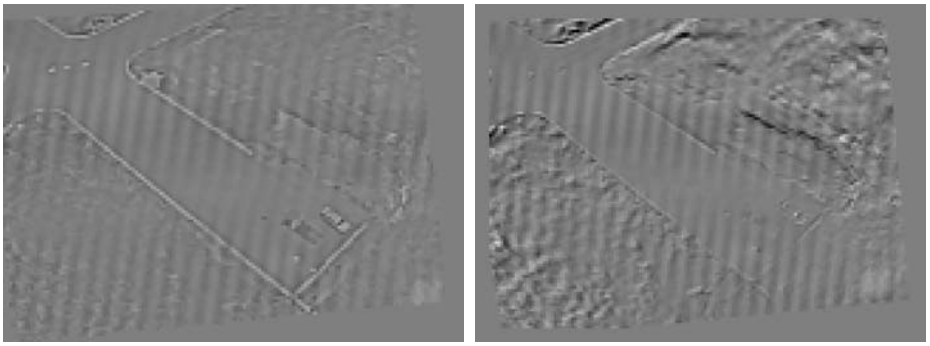

Fig. 4 Residual image differences produced by our direct registration method (*left image*) and a standard feature-based registration method (*right image*). Feature-based registration was performed on a 640 × 480 video sequence, tracking Harris corners between frames using the OpenCV™ toolbox, using RANSAC to estimate a homography matrix relating each frame pair, and warping the first frame to align it with the second. Direct registration was performed using the method indicated in this section on a 4× downsampled version of the same video. Direct registration is able to consistently reduce the minimum mean squared pixel error compared to feature based registration

imaging a particular world point P, and a differential change in any pose parameter ($\square$) will cause a differential change in the (x, y) image coordinates of the projection of P. The $\frac{\partial y}{\partial \square}$ and $\frac{\partial x}{\partial \square}$ terms thus represent the way in which a feature observed at any point in the image will appear to move due to a differential change in the pose parameter $\square$. After multiple iterations like the one shown in Fig. 3, the estimated change $(\Delta \chi_2^V(k))$ in the pose estimate will become very small. At this point, the current pose estimate $\chi_2^V(k)$ becomes the final χ_2^V returned from our VO algorithm.

Figure 4 demonstrates the potential improvement in registration given by direct registration methods versus more standard feature-based image registration. Since more accurate pixel registration implies more accurate relative pose estimates, improved accuracy can lead to slower growth in VO pose estimates, leading to more accurate overall pose estimates.

4 VO Covariance Estimation

As discussed in Section 2, to fuse INS and VO measurements in an EKF framework, we need to be able to find the matrix

$$F = \left.\frac{\partial \mathbf{f}}{\partial \chi}\right|_{\chi=\hat{\chi}_{n-1}}$$

that linearly approximates the state transition function. The state transition function $\mathbf{f}$ in our EKF is the VO system described in the previous section, which produces the current aircraft pose χ_2^V given the estimated previous pose $\hat{\chi}_1$. Unfortunately, this $\mathbf{f}$ function is not differentiable, and is expressed only as an iterative algorithm, not in mathematical closed form. In this section, we will make some simplifying assumptions that allow us to approximate the jacobian F of this algorithm. This will allow us to propagate covariance in pose $\chi_1$to covariance on pose χ_2^V, enabling the fusion of VO and INS measurements.

The final χ_2^V produced by our VO system is a function of both the pose $\hat{\chi}_1$ and the two images Y_1 and Y_2. The interplay of these two images in the iterative image registration algorithm leads to a non-differentiable $\mathbf{f}$. Stated differently, changes in χ_2^V can be due either to changes in the quality of image registration between Y_1 and Y_2 or due to changes in the original pose $\hat{\chi}_1$: this fact makes it impossible to differentiate $\mathbf{f}$ directly. Even if we could parametrize and precisely describe "the quality of image registration" in a meaningful way, differentiating $\mathbf{f}$ with respect to this parametrization would still involve differentiation of an iterative procedure. To rectify this situation, we will first assume in Section 4.1 that the image registration process is able to perfectly register Y_1 with Y_2: this is the same as assuming that all our uncertainty about the final pose χ_2^V is due to propagated uncertainty in $\hat{\chi}_1$. We will find the desired matrix F using this assumption. We will then discuss error due to mis-registration in Section 4.2.

4.1 Propagating Errors in Image 1 Pose to Image 2

In order to find the component of uncertainty on χ_2^V (due to propagation of uncertainty from $\hat{\chi}_1$), we need to be able to characterize the function $\mathbf{f}(\cdot)$ such that $\chi_2^V = \mathbf{f}(\hat{\chi}_1)$. We seek to linearize this function so that we can perform a standard linear covariance update $P_2^V = F\hat{P}_1 F^T$. To begin, we note that once the iterative

image registration process has converged, $\hat{\chi}_1$ and χ_2^V can each be used to compute the homography matrices H_{G1} and H_{G2}, which map pixel locations in images Y_1 and Y_2 respectively to locations on the ground image. These homographies were used respectively in the projection and rendering processes of the VO system (see Section 3), and are thus available from the VO process. H_{G1} and H_{G2} individually have uncertainty associated with them, because each of their components is a function of $\hat{\chi}_1$ and χ_2^V, which are imperfectly known and thus have associated covariances. Because homography matrices can be composed by matrix multiplication and are invertible [32, 37], we can combine these two homography matrices into a single homography that maps pixel locations in Y_1 to pixel locations in Y_2 as:

$$H_{12} = H_{G2} H_{G1}^{-1} = H_{G2} H_{1G} \tag{14}$$

If, as we have assumed, the registration between Y_1 and Y_2 is perfectly accurate, then this homography H_{12} is perfectly known. Thus, although both H_{G2} and H_{1G} (the inverse of H_{G1}) have associated uncertainty, their product H_{12} does not. This observation is a direct consequence of the fact that VO methods fundamentally measure relative poses: while χ_1 and χ_2 may both be incorrect, the accuracy of the relationship between them is constrained only by the accuracy of image registration.

The relationship in Eq. 14 forms the basis of our desired $\mathbf{f}(\cdot)$. We first post-multiply by H_{G1}:

$$H_{G2} = H_{12} H_{G1} \tag{15}$$

Our insight about H_{G1}, H_{G2}, and H_{12} allows us to write:

$$H_{G2}(\chi_2) = H_{12} H_{G1}(\chi_1)$$

In order to isolate χ_2 as a function of χ_1 from this last equation, it would be desirable if we could invert the function $H_{G2}(\chi_2)$; we would like to be able to deduce the pose of a camera given a homography mapping its image points to the ground. We will refer to this inverse function as ξ instead of H_{G2}^{-1}, to emphasize the fact that the inverse we would like is not the matrix inverse of the matrix H_{G2}, but rather a function mapping a homography matrix to a pose. If we could find such a function, we would have our desired formula for $\mathbf{f}$:

$$\xi\left(H_{G2}\left(\chi_2^V\right)\right) = \xi\left(H_{12} H_{G1}\left(\hat{\chi}_1\right)\right) \tag{16}$$

$$\chi_2^V = \xi\left(H_{12} H_{G1}\left(\hat{\chi}_1\right)\right) \tag{17}$$

$$\chi_2^V = \mathbf{f}\left(\hat{\chi}_1\right) \tag{18}$$

We could then compute the desired derivative F by the chain rule:

$$\frac{\partial \mathbf{f}}{\partial \hat{\chi}_1} = \frac{\partial \chi_2^V}{\partial \hat{\chi}_1} = \frac{\partial \xi}{\partial H_{G2}} \frac{\partial H_{G2}}{\partial H_{G1}} \frac{\partial H_{G1}}{\partial \hat{\chi}_1} \tag{19}$$

The function $H_{G1}\left(\hat{\chi}_1\right)$ can already be computed in closed form using standard computer vision techniques, and its derivative, the term $\frac{\partial H_{G1}}{\partial \hat{\chi}_1}$ in Eq. 19 is a 9×6 matrix of partial derivatives that can be computed from this closed form expression in a straightforward manner. The term $\frac{\partial H_{G2}}{\partial H_{G1}}$ is a 9×9 matrix of partial derivatives: these partials are also straightforward to compute, as they are simply elements of H_{12} (see Eq. 15).

The only remaining obstacle is the first term in Eq. 19, determining the derivative of the ξ function. As discussed in [32], there are well documented methods of decomposing a homography matrix to determine a relative pose; these methods, however, in general produce four solutions among which we must choose based on the cheirality constraint. This fact makes the ξ function non-differentiable. Instead of attempting to differentiate ξ directly, we notice that we can approximate changes in the homography matrix H_{G2} due to changes in χ_2^V using the partial derivative matrix $\frac{\partial H_{G2}}{\partial \chi_2}$, just as we do for H_{G1}. The inverse of this linear mapping, if it existed, would give changes in χ_2^V due to changes in H_{G2} as desired. The inverse does not exist, as the 9×6 matrix $\frac{\partial H_{G2}}{\partial \chi_2^V}$ represents an over-determined system. We can, however, find the pseudo-inverse of the matrix $\frac{\partial H_{G2}}{\partial \chi_2}$, which still maps changes in H_{G2} to changes in χ_2, minimizing error in the elements of H_{G2} . We use this pseudo-inverse to approximate the derivative $\frac{\partial \xi}{\partial H_{G2}}$.

Combining these three terms, we compute the 6×6 matrix of partial derivatives $\frac{\partial \chi_2^V}{\partial \hat{\chi}_1}$ as:

$$\frac{\partial \chi_2^V}{\partial \hat{\chi}_1} = \left(Q^T Q\right)^{-1} Q^T \frac{\partial H_{G2}}{\partial H_{G1}} \frac{\partial H_{G1}}{\partial \hat{\chi}_1} \tag{20}$$

where

$$Q = \frac{\partial H_{G2}}{\partial \chi_2^V} \tag{21}$$

We now have the desired partial derivative matrix of our VO method, which we can use to approximate covariance propagation.

(a) MAV (b) Kestrel™ Autopilot

Fig. 5 The MAV platform used in this work. The MAV is a flying wing aircraft constructed of EPP foam with a 6 ft wingspan, controlled by a Kestrel™ autopilot (**a**, **b**)

4.2 Determining Covariance with Imperfect Registration

Naturally, the process of determining H_{12} (i.e. registration) is not, as we assumed in the previous subsection, without error. This registration error will cause the difference image to have residual structure (which is not due to parallax) after the registration process completes. We address this source of error by computing the ratio of the residual cost function J of Eq. 12 with an empirically determined cost value, and boosting the diagonal values of the covariance matrix P_2^V proportionally. Doing this is similar to the "Q-boosting" technique common in EKF practice: we simply increase our estimated uncertainty on the VO pose such that the overall filter yields desirable results. In practice, this extra boost in the diagonal elements of P_2^V was not found to be necessary to yield good results, and was not used in generating result data.

5 Results and Analysis

In order to evaluate our pose estimation framework, we estimate MAV pose without using GPS information, and show that fusion of VO and INS information allows MAV location to be estimated with accuracy similar to that of GPS for a reasonable period of time.

All results presented in this work are collected using an inexpensive, hand-launchable MAV platform, shown in Fig. 5. The aircraft is a flying wing design, with a 6-foot wingspan, constructed of EPP foam. The on-board Kestrel™ autopilot and associated Virtual Cockpit™ software platform allow the aircraft to autonomously aviate and navigate. The Kestrel™ autopilot comprises a small microcontroller and a collection of sensors that includes three-axis accelerometers, rate gyroscopes, and differential pressure sensors. Pitch, roll, and altitude are estimated on-board from these sensor readings, while 2-D location and heading are estimated using a small on-board GPS receiver [1, 4]. This pose estimate and other telemetry data is transmitted to a ground station at a rate of about 4 Hz. A separate camera/transmitter system collects video footage during flight and transmits this video to the ground station. This video stream is synchronized on the ground station with the stream of pose estimates from autopilot telemetry. VO and sensor fusion are performed off-line using this data.

5.1 Pose Estimation During GPS Dropout

A key goal of this work is to explore the use of VO for MAV localization during GPS dropout. In order to evaluate the accuracy of our fused VO/INS pose estimates, we compare the estimated (t_x, t_y, t_z) location of the aircraft with baseline location measurements produced by the current MAV autopilot pose estimation method, which uses a GPS receiver to measure t_x and t_y, and a differential pressure sensor to measure t_z. We will hereafter refer to these baseline pose estimates as `gpsins` pose estimates. These baseline location estimates are here compared with location estimates from:

1. Our unaided VO system (referred to as `voonly`) shown in Fig. 6
2. Our fusion system, incorporating only VO and INS measurements (referred to as `voins`) shown in Fig. 7.

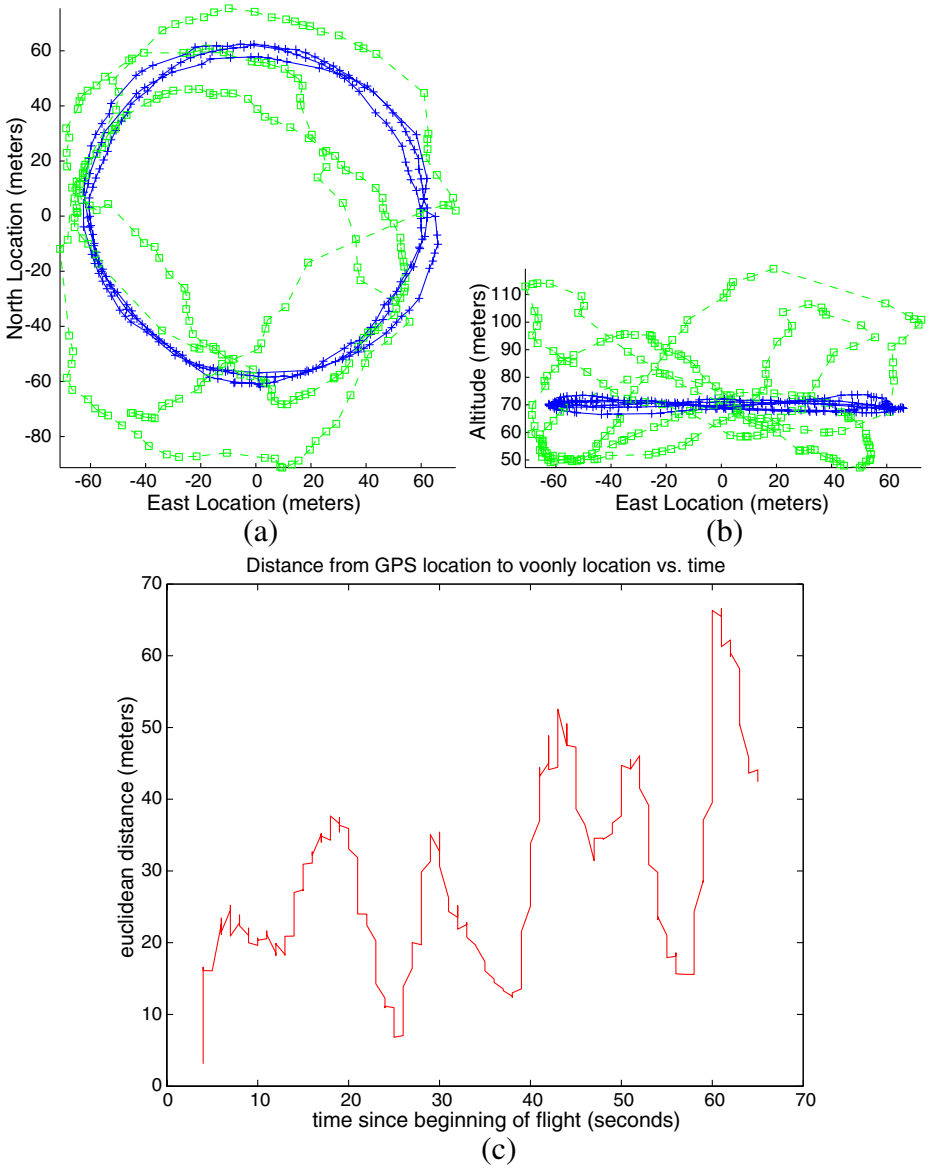

Fig. 6 (**a**, **b**) `gpsins` location estimates (*blue* +) and `voonly` location estimates (*green* □) in a circular flight path. (**c**) Euclidean distance between `gpsins` and `voonly` location estimates at each point in time. Note rapid error growth

In each of these figures, we display the path of the MAV as estimated by `gpsins` and by one of the vision-aided fusion schemes, giving both a horizontal and vertical view (subfigures a and b, respectively), and plot the time-varying distance between these two paths (subfigure c). Clearly the errors in `voonly` location are both much larger than those of `voins`, and increase dramatically over the course of the flight (∼70 seconds). This demonstrates that fusion of VO and INS data can

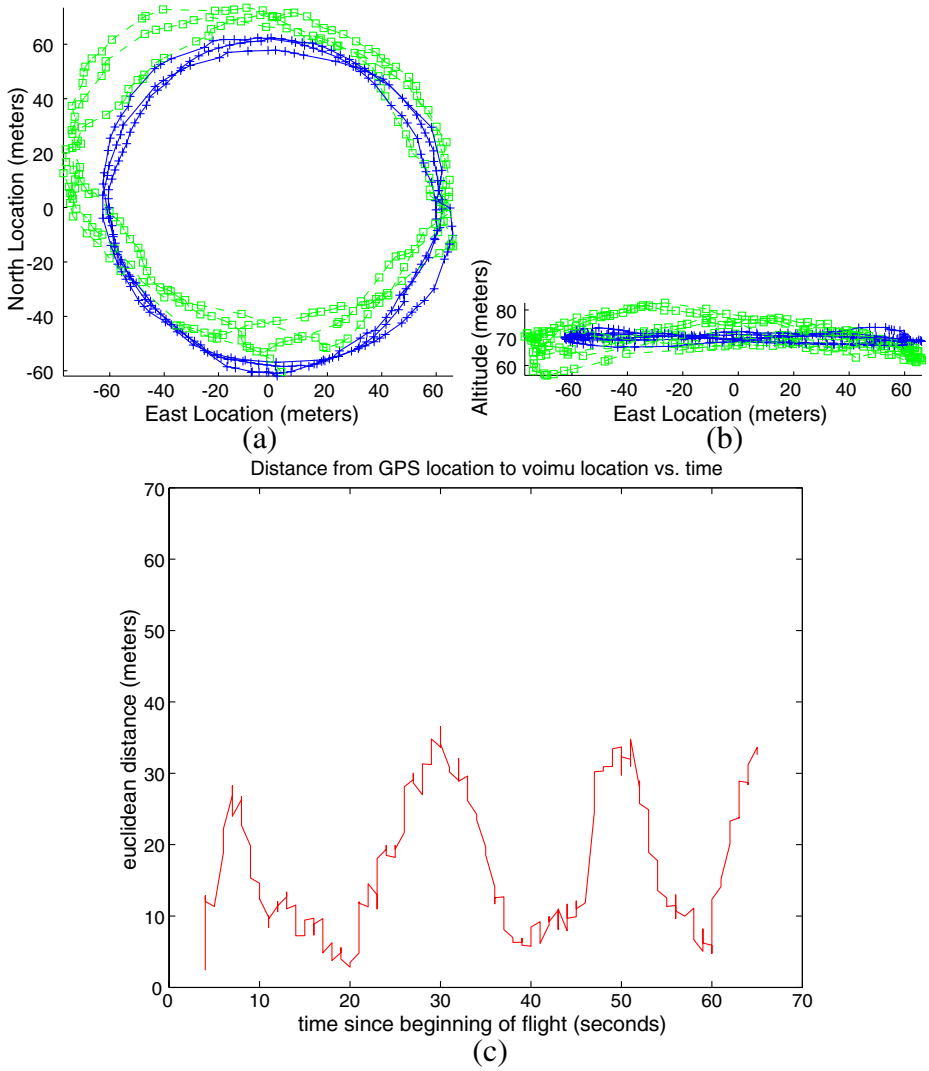

Fig. 7 (**a**, **b**) `gpsins` location estimates (*blue* +) and `voins` location estimates (*green* □) in a circular flight path. (**c**) Euclidean distance between `gpsins` and `voins` location estimates at each point in time. Notice the slow growth of error as compared to Fig. 6c

significantly reduce the drift in location estimates that is inherently part of VO systems, as well as dramatically reducing (~40%) the worst-case location error. It is also meaningful to realize that the cyclic pattern of errors present in Figs. 6c and 7c is likely a consequence of inaccuracies in camera mounting, wind estimation, and temporal data association. It is perhaps remarkable that position can be estimated to within <40 m of GPS estimates in the presence of these inaccuracies. As the size, cost, and field use constraints on MAV platforms necessarily make them prone to these errors, the ability to function in their presence is an important benefit for MAV platforms. It should be noted that this error is comparable to the error

obtained by some SLAM-based visual/inertial navigation solutions, such as that of Kim and Sukkarieh [6] and Bryson and Sukkarieh [7] (whose flight path is similar to ours). Several other SLAM-based solutions give much better accuracy in simulation (e.g. [38]) or in limited environments with relatively rapid loop closure (e.g. [10]).

5.2 Computational Complexity

The results given here are presented as a proof of concept. The navigational results shown were obtained using real flight data, but processing was performed offline, in a framework that was not optimized for speed. A number of further optimizations are possible, which we feel would allow our estimation framework to run at a frame rate of about 5 Hz or better on computing hardware compatible with MAV weight & power constraints (~1.5 GHz standard laptop-style CPU). These optimizations could include the following:

- The simulation environment is currently coded in MATLAB, and no MEX functions are utilized. A functioning system would need a C/C++ implementation, which in itself would bring performance significantly closer to realtime.
- We currently perform forward additive image registration: that is, the second image is repeatedly warped (requiring expensive recomputation of image jacobians) until it matches the first image. Baker et al. [36] present several ways of improving on this, notably the inverse compositional method, which needs to compute jacobians only once and applies the inverse of the computed warping to the first image. This could also significantly reduce computational load.
- Not all pixels need actually be compared during the direct registration process: only a sampling of pixels would be needed. This could significantly reduce computation costs.

Memory requirements associated with this method are both fixed and reasonable, as only the previous video frame and the vehicle state estimates need to be stored.

6 Conclusion

We have presented a Visual Odometry system based on direct image registration, and an EKF framework by which these VO pose estimates can be fused with INS data to improve pose estimation performance. As a key to utilizing an EKF framework, we have developed a method to linearly approximate covariance through the VO system. We have demonstrated the ability of this system to estimate pose without using GPS information in a flight of $\sim 70\,\mathrm{s}$ ($> 500\,\mathrm{m}$), with much slower error growth and with a 40% increase in worst-case accuracy compared to unaided VO.

References

1. Beard, R., Kingston, D., Quigley, M., Snyder, D., Christiansen, R., Johnson, W., McLain, T., Goodrich, M.: Autonomous vehicle technologies for small fixed wing UAVs. AIAA J. Aerosp. Comput. Inf. Commun. **2**(1), (2005)
2. Saunders, J.B., Call, B., Curtis, A., Beard, R.W., McLain, T.W.: Static and dynamic obstacle avoidance in miniature air vehicles. In: AIAA 5th Aviation, Technology, Integration, and Operations Conference, Arlington, 26–28 September 2005

3. Kingston, D.B., Beard, R.W.: Real-time attitude and position estimation for small UAV's using low-cost sensors. In: AIAA Unmanned Unlimited Systems Conference and Workshop, Chicago, September 2004
4. Christiansen, R.S.: Design of an autopilot for small unmanned aerial vehicles. Master's thesis, Brigham Young University (2004)
5. Volpe, J.: Vulnerability assessment of the transport infrastructure relying on the global positioning system. Technical report, Office of the Assistant Secretary for Transportation Policy, U.S. Department of Transportation, Aug. (2001)
6. Kim, J.-H., Sukkarieh, S.: Airborne simultaneous localisation and map building. In: Robotics and Automation, 2003. Proceedings. IEEE International Conference on ICRA '03, vol. 1, pp. 406–411. IEEE, Piscataway (2003)
7. Bryson, M., Sukkarieh, S.: Bearing-only SLAM for an airborne vehicle. In: Australasian Conference on Robotics and Automation, Sydney, 5–7 December 2005
8. Kim, J., Sukkarieh, S.: SLAM aided GPS/INS navigation in GPS denied and unknown environments. In: The 2004 International Symposium on GNSS/GPS, Sydney, 6–8 December 2004
9. Langelaan, J., Rock, S.: Passive GPS-free navigation for small UAVs. In: Proc. IEEE Aerospace Conference, pp. 1–9. IEEE, Piscataway (2005)
10. Davison, A.J., Reid, I.D., Molton, N.D., Stasse, O.: MonoSLAM: real-time single camera SLAM. J. Pattern Anal. Mach. Intell. **29**(6), 1052–1067 (2007)
11. Dellaert, F., Thrun, S., Thorpe, C.: Jacobian images of super-resolved texture maps for model-based motion estimation and tracking. In: 1998 IEEE Workshop on Applications of Computer Vision, pp. 2–7. IEEE, Piscataway (1998)
12. Dellaert, F., Thorpe, C., Thrun, S.: Super-resolved texture tracking of planar surface patches. In: 1998 IEEE/RSJ International Conference on Intelligent Robotic Systems, vol. 1, pp. 197–203. IEEE, Piscataway (1998)
13. Corke, P., Lobo, J., Dias, J.: An introduction to inertial and visual sensing. Int. J. Rob. Res. **26**(6), 519–535 (2007)
14. Roumeliotis, S.I., Johnson, A.E., Montgomery, J.F.: Augmenting inertial navigation with image-based motion estimation. In: 2002 IEEE International Conference on Robotics and Automation, vol. 4, pp. 4326–4333. IEEE, Piscataway (2002)
15. Diel, D.D.: Stochastic constraints for vision aided inertial navigation. Master's thesis, MIT (2005)
16. Bayard, D.S., Brugarolas, P.B.: An estimation algorithm for vision-based exploration of small bodies in space. In: 2005 American Control Conference, Portland, 8–10 June 2005
17. Mourikis, A.I., Roumeliotis, S.I.: A multi-state constraint Kalman filter for vision-aided intertial navigation. In: 2007 IEEE International Conference on Robotics and Automation, Roma, 10–14 April 2007
18. Armesto, L., Tornero, J., Vincze, M.: Fast ego-motion estimation with multi-rate fusion of inertial and vision. Int. J. Rob. Res. **26**(6), 577–589 (2007)
19. Viéville, T., Clergue, E., Facao, P.E.D.S.: Computation of ego-motion and structure from visual and inertial sensors using the vertical cue. In: International Conference on Computer Vision, pp. 591–598, Berlin, 11–14 May 1993
20. Domke, J., Aloimonos, Y.: Integration of visual and inertial information for egomotion: a stochastic approach. In: 2006 IEEE International Conference on Robotics and Automation, pp. 2053–2059. IEEE, Piscataway (2006)
21. Lobo, J., Dias, J.: Vision and inertial sensor cooperation using gravity as a vertical reference. IEEE Trans. Pattern Anal. Mach. Intell. **25**(12), 1597–1608 (2003)
22. Lobo, J., Dias, J.: Inertial Sensed ego-motion for 3D Vision. J. Robot. Syst. **21**(1), 3–12 (2004)
23. Lobo, J., Dias, J.: Relative pose calibration between visual and inertial sensors. Int. J. Rob. Res. **26**(6), 561–575 (2007)
24. Nister, D., Naroditsky, O., Bergen, J.: Visual odometry. In: 2004 IEEE Conference on Computer Vision and Pattern Recognition, vol. 1. IEEE, Piscataway (2004)
25. Nister, D., Naroditsky, O., Bergen, J.: Visual odometry for ground vehicle applications. J. Field Robot. **23**(1), 3–20 (2006)
26. Kanade, T., Amidi, O., Ke, Q.: Real-time and 3D vision for autonomous small and micro air vehicles. In: 2004 IEEE Conference on Decision and Control, vol. 2, pp. 1655–1662. IEEE, Piscataway (2004)
27. Kehoe, J.J., Causey, R.S., Arvai, A., Lind, R.: Partial aircraft state estimation from optical flow using non-model-based optimization. In: 2006 American Control Conference, p. 6, Minneapolis, 14–16 June 2006
28. Kehoe, J.J., Watkins, A.S., Causey, R.S., Lind, R.: State estimation using optical flow from parallax-weighted feature tracking. In: 2006 AIAA Guidance, Navigation, and Control Conference, Keystone, 21–24 August 2006

29. Kaiser, K., Gans, N., Dixon, W.: Position and orientation of an aerial vehicle through chained, vision-based pose reconstruction. In: 2006 AIAA Guidance, Navigation, and Control Conference, Keystone, 21–24 August 2006
30. Kaiser, K., Gans, N., Dixon, W.: Localization and control of an aerial vehicle through chained, vision-based pose reconstruction. In: 2007 American Control Conference, pp. 5934–5939, New York, 11–13 July 2007
31. Nister, D.: An efficient solution to the five-point relative pose problem. IEEE Trans. Pattern Anal. Mach. Intell. **26**(6), 756–770 (2004)
32. Ma, Y., Soatto, S., Kosecka, J., Shankar, S., Sastry: An Invitation to 3-D Vision: From Images to Geometric Models. Springer, Heidelberg (2004)
33. Eustice, R., Singh, H., Leonard, J., Walter, M., Ballard, R.: Visually navigating the RMS Titanic with SLAM information filters. In: Robotics: Science and Systems, Cambridge, June 2005
34. Richmond, K., Rock, S.: A real-time visual mosaicking and navigation system. Unmanned Untethered Submersible Technology (2005)
35. Fleischer, S.D.: Bounded-error vision-based navigation of autonomous underwater vehicles. Ph.D. Thesis, Stanford University (2000)
36. Baker, S., Matthews, I.: Lucas-Kanade 20 years on: a unifying framework. part 1: the quantity approximated, the warp update rule, and the gradient descent approximation. Int. J. Comput. Vis. **56**(3), 221–255 (2004)
37. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2004)
38. Kim, J., Sukkarieh, S.: Robust multi-loop airborne slam in unknown wind environments. In: Robotics and Automation, 2006, Proceedings 2006 IEEE International Conference on ICRA 2006, pp. 1536–1541. IEEE, Piscataway (2006)