Proceedings of the 38th
Conference on Decision & Control
Phoenix, Arizona USA • December 1999

Vision Guided Landing of an Unmanned Air Vehicle

Omid Shakernia[†1] Yi Ma[†] T. John Koo[†] João Hespanha[‡] S. Shankar Sastry[†]

[†]Department of EECS
University of California at Berkeley
Berkeley, CA 94720-1774
{omids,mayi,koo,sastry}@eecs.berkeley.edu

[‡]Department of EE-Systems
University of Southern California
Los Angeles, CA 90089-2563
hespanha@usc.edu

Abstract

In this paper, we study the problem of using computer vision as a sensor to control the landing of an Unmanned Air Vehicle (UAV). The vision problem we address is a special case of the general ego-motion estimation problem due to the fact that all feature points lie on a plane. We propose a new geometric estimation scheme for solving the differential version of the planar ego-motion estimation problem. The algorithm is computationally inexpensive and amenable for real-time implementation. We present a performance evaluation of the algorithm under different levels of image measurement noise and camera motions relative to the landing pad. We also present a full dynamic model of a UAV, discuss a nonlinear controller based on differential flatness, and show through simulation that the vision guided UAV performs stable landing maneuvers even under large levels of image measurement noise.

Keywords: unmanned air vehicle, autonomous landing, structure from motion

1 Introduction

Unmanned air vehicles (UAVs) have generated considerable interest in the control community due to the fact that the design of UAVs brings to light research questions falling in some of the most exciting new directions for control. One of these directions in the use of computer vision as a sensor in the feedback control loop. The task of autonomous aircraft landing is well suited to vision-based control, especially in cases where the landing pad is in an unknown location and is moving, such as the deck of a ship.

In this paper, we present a computer vision algorithm to estimate the motion of a UAV relative to a landing pad. The algorithm is computationally inexpensive and amenable to real-time implementation. We present a performance evaluation of the algorithm under different levels of image measurement noise and camera motions relative to the landing pad. We also present a full dynamic model of a UAV, discuss nonlinear controller based on differential flatness, and show through simulation that the vision guided UAV performs stable landing maneuvers even under large levels of image measurement noise.

The organization of the paper is as follows. In section 2, we establish the notation for the camera motion and imaging models. In section 3 we formulate the problem of ego-motion estimation from image measurements of a planar scene. In section 4, we present simulation results of the motion estimation algorithm. In section 5 we present a dynamic model of a UAV, discuss a controller based on differential flatness, and present simulation results of a vision guided landing. We end the paper in section 6 with concluding remarks and directions for future research.

2 Motion and Imaging Models

In this section we give the mathematical model of the UAV motion and the imaging model of the on-board camera. We adhere to the following convention: We denote the coordinates of a point in the inertial frame with a tilde, for example $\tilde{q} \in \mathbb{R}^3$, and denote the coordinates of the point in the camera frame using the same letter, but without a tilde. We assume a monocular camera is fixed to the UAV and the optical axis of the camera coincides with the vertical axis of the UAV body frame.

We assume the motion of the UAV is described by a smooth curve in the special Euclidean group $SE(3)$. Let $(p(t), R(t)) \in SE(3)$ denote the position and orientation of the camera with respect to the inertial frame at time t. The coordinates of a fixed point in the inertial frame and its coordinates in the camera frame at time t are related by [8]:

$$\tilde{q} \quad = \quad R(t)q(t) + p(t) \qquad (1)$$

Given body angular and linear velocities $\omega, v \in \mathbb{R}^3$, the

[1] This research was supported by ONR under grant N00014-97-1-0946, and ARO under grant DAAH04-96-1-0341.

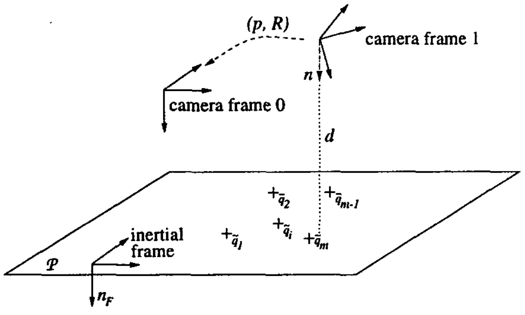


Figure 1: Geometry of camera frame relative to plane

coordinates of a fixed point in the inertial frame satisfy

$$\dot{q} = -\hat{\omega}q - v \tag{2}$$

where the skew symmetric matrix $\hat{\omega}$ is defined by $\hat{\omega}q \triangleq \omega \times q \ \forall q \in \mathbb{R}^3$. The body angular and linear velocities are given by $\hat{\omega} = R^T\dot{R}$, $v = R^T\dot{p}$.

The imaging model of a calibrated camera is given by *perspective projection*

$$\pi : (q_x, q_y, q_z)^T \quad \mapsto \quad \left(\frac{q_x}{q_z}, \frac{q_y}{q_z}, 1\right)^T \tag{3}$$

If $\mathbf{x}$ is the image of a point q, *i.e.* $\mathbf{x} = \pi(q)$, then we write $\lambda\mathbf{x} = q$ where $\lambda = q_z \in \mathbb{R}$ encodes the depth of q from the camera along the optical axis. We denote the optical axis of the camera by $e_3 = (0, 0, 1)^T$.

3 Visual Ego-Motion Estimation

In this section, we give a formulation of the so-called "visual ego-motion estimation" problem. The goal is to recover the motion of the camera using image measurements of fixed points in the environment. In the general case when the features in the images correspond to points in a general configuration in 3D, the ego-motion estimation problem from a pair of images can be solved by the well-known "8-point algorithm" [4]. Recently, Ma *et al* [7] derived a counterpart for the 8-point algorithm for the "differential case" where the measurements are image velocities. Our ego-motion estimation problem for the purpose of landing a UAV is a special case of the general one: All the image points correspond to coplanar points on the landing pad. It is well known that the case where all features points are coplanar is a degenerate case that makes the 8-point algorithm ill-conditioned, giving poor estimation results. Hence one needs algorithms specific for the planar case.

3.1 Discrete Case

The "discrete" version of the planar visual ego-motion estimation problem has been studied extensively in the literature [5, 1, 12]. Here we recall the well known results.

Suppose we have a set of m fixed coplanar points $\{\tilde{q}_i\}_{i=1}^m \subset \mathcal{P}$, where $\mathcal{P}$ denotes the plane. Without loss of generality, we take the origin of the inertial frame to be in $\mathcal{P}$. The geometry is depicted in Figure 1.

Proposition 1 *Suppose the camera undergoes a rigid motion $(p, R) \in SE(3)$ between "camera frame 0" and "camera frame 1", where without loss of generality, "camera frame 0" differs from the inertial frame by a pure translation. Then the coordinates $\{q_i^0\}_{i=1}^m, \{q_i^1\}_{i=1}^m$ of the fixed coplanar points $\{\tilde{q}_i\}_{i=1}^m \subset \mathcal{P}$ in the two camera frames are related by $q_i^0 = (R + \frac{1}{d}pn^T)q_i^1$ for $i = 1, \ldots, m$.*

The proposition is direct to prove using the identity $n^T q_i^1 \equiv d$ for $i = 1, \ldots, m$, where d is the distance of camera frame 1 to the plane $\mathcal{P}$ and $n \in S^2$ is the surface normal to $\mathcal{P}$ relative camera frame 1. We call the matrix

$$A \triangleq \left(R + \frac{1}{d}pn^T\right) \in \mathbb{R}^{3\times3} \tag{4}$$

the "planar essential matrix," since it contains all the motion and structure parameters that we need to recover. The vision task is to recover the matrix A based on image measurements, then decompose it into its motion and structure parameters. We say that a set of coplanar points $\{\tilde{q}_i\}_{i=1}^m$ are in *general configuration* if there is a group of 4 points such that no three are collinear. It turns out that the A matrix can be uniquely estimated from image point correspondences if and only if the points $\{\tilde{q}_i\}_{i=1}^m$ are in general configuration.. The A matrix may then be decomposed up 2 physically possible solutions for the motion and structure parameters [12].

3.2 Differential Case

Our contribution is to the "differential" version of the problem, that is, the task of recovering the linear and angular velocity of the camera given image velocities of fixed points in a plane. The differential version is important for the control of a dynamic mobile robot such as a UAV, since velocity estimates are necessary for the computation of control inputs. The differential structure from motion problem for a planar scene has also been studied in [2, 10]. We propose a new geometrical estimation scheme for the motion and structure parameters.

Proposition 2 *Suppose the camera undergoes a rigid motion with body linear and angular velocities v, ω. Then the coordinates of coplanar points $\{\tilde{q}_i\}_{i=1}^m$ in the instantaneous camera frame satisfy:*

$$\dot{q}_i = -(\hat{\omega} + \tfrac{1}{d}vn^T)q_i \quad \textit{for } i = 1, \ldots, m.$$

We call the matrix

$$B \triangleq \left(\hat{\omega} + \frac{1}{d} v n^T\right) \in \mathbb{R}^{3\times 3} \tag{5}$$

the "planar differential essential matrix", since it contains all the (differential) motion and structure parameters that we need to recover. We first show how to recover the matrix B based on image measurements, then decompose it into its motion and structure parameters.

Proposition 3 *The B matrix given in equation (5) satisfies the constraint:*

$$\dot{\mathbf{x}}_i = -(I - \mathbf{x}_i e_3^T) B \mathbf{x}_i, \quad i = 1, \dots, m \tag{6}$$

where $\{\mathbf{x}_i(t), \dot{\mathbf{x}}_i(t)\}_{i=1}^m$ are image points and velocities of fixed points $\{\tilde{q}_i\}_{i=1}^m$ in the plane.

Proof: Differentiating $\lambda \mathbf{x} = q$ and substituting $\dot{q} = -Bq$ gives $\dot{\lambda}\mathbf{x} + \lambda\dot{\mathbf{x}} = -\lambda B\mathbf{x}$. Differentiating $\lambda = e_3^T q$ gives $\dot{\lambda} = -\lambda e_3^T B\mathbf{x}$. Using these relations and eliminating λ gives the result. ■

Equation (6) is the planar differential epipolar constraint. Since the constraint is linear in B, by stacking the entries of B as $b = (b_{11}, b_{12}, b_{13}, b_{21}, \dots, b_{33})^T \in \mathbb{R}^9$, we may re-write (6) as $\dot{\mathbf{x}}_i = \mathbf{g}_i^T b$, where $\mathbf{g}_i \in \mathbb{R}^{9\times 3}$ is a function of $\mathbf{x}_i$. However, since the third row of equation (6) contains only zeros, each image point velocity only has two constraints on the matrix B. Given a set of m image point and velocity pairs $\{\mathbf{x}_i, \dot{\mathbf{x}}_i\}_{i=1}^m$ of fixed points in the plane, we may stack each equation $\dot{\mathbf{x}}_i = \mathbf{g}_i^T b$ into a single equation:

$$\dot{\mathbf{X}} = \mathbf{G} b \tag{7}$$

where $\dot{\mathbf{X}} = (\dot{\mathbf{x}}_1^T, \dots, \dot{\mathbf{x}}_m^T)^T \in \mathbb{R}^{3m}$ and $\mathbf{G} = (\mathbf{g}_1, \cdots, \mathbf{g}_m)^T \in \mathbb{R}^{3m\times 9}$.

Proposition 4 $\mathrm{rank}(\mathbf{G}) = 8$ *if and only if the points $\{\tilde{q}_i\}_{i=1}^m$ are in general configuration.*

For the proof, please refer to [9]. If $\mathrm{rank}(\mathbf{G}) = 8$ then by linear least squares techniques, equation (7) can be used to recover b up to one dimension. That is, we can recover $B = B_L + \xi B_K$ where B_L corresponds to the minimum norm least squares estimate of B, B_K corresponds to the vector in $\ker(\mathbf{G})$ and $\xi \in \mathbb{R}$ is an unknown scale. By inspection, of equation (6) one can see that $B_K = I$. Thus we only need to recover the unknown ξ in order to uniquely estimate B. Next we give a constraint imposed by the structure of B which will be used to estimate B.

Lemma 1 *Suppose $u, v \in \mathbb{R}^3$, and $\|u\|^2 = \|v\|^2 = \alpha$. If $u \neq v$, the matrix $D = uv^T + vu^T \in \mathbb{R}^{3\times 3}$ has eigenvalues $\{\lambda_1, 0, \lambda_3\}$, where $\lambda_1 > 0$, and $\lambda_3 < 0$. If $u = \pm v$, the matrix D has eigenvalues $\{\pm 2\alpha, 0, 0\}$.*

Proof: Let $\beta = u^T v$. If $u \neq \pm v$, we have $-\alpha < \beta < \alpha$. We solve the eigenvectors of D to be $\{u+v, u\times v, u-v\}$ with the corresponding eigenvalues $\{\beta+\alpha, 0, \beta-\alpha\}$. Clearly $\lambda_1 = (\beta + \alpha) > 0$. and $\lambda_3 = \beta - \alpha < 0$. It is direct to check the case where $u = \pm v$. ■

Theorem 1 *The matrix B can be uniquely estimated from the image measurements if and only if the points $\{\tilde{q}_i\}_{i=1}^m$ are in general configuration.*

Proof: We use sorted eigenvalues, that is, given eigenvalues $\{\lambda_1, \lambda_2, \lambda_3\}$ then $\lambda_1 \geq \lambda_2 \geq \lambda_3$. If the points are not in general configuration, then by Proposition 4, $\mathrm{rank}(\mathbf{G}) < 8$, and the problem is underconstrained. If the points are in general configuration, then by least squares we may recover $B = B_L + \xi I$ for some unknown $\xi \in \mathbb{R}$. By lemma 1, we have that $B + B^T = \frac{1}{d} v n^T + \frac{1}{d} n v^T$ has eigenvalues $\{\lambda_1, \lambda_2, \lambda_3\}$ where $\lambda_1 \geq 0$, $\lambda_2 \equiv 0$, and $\lambda_3 \leq 0$. Compute the eigenvalues of $B_L + B_L^T$ to be $\{\gamma_1, \gamma_2, \gamma_3\}$. Then $\lambda_i = \gamma_i + 2\xi$, for $i = 1, 2, 3$. Since we must have $\lambda_2 = 0$, we have $\xi = -\frac{1}{2}\gamma_2$, and set $B = B_L - \frac{1}{2}\gamma_2 I$. ■

The following constructive proof gives a new technique for the recovering the motion and structure parameters from the B matrix.

Theorem 2 *Given a matrix $B \in \mathbb{R}^{3\times 3}$ in the form $B = \hat{\omega} + \frac{1}{d} v n^T$, one can recover the parameters $\{\hat{\omega}, \frac{v}{d}, n\}$ up to at most 2 physically possible solutions. There is a unique solution if $v = 0$, $v \times n = 0$ or $e_3^T v = 0$, where $e_3 = (0, 0, 1)^T$ is the optical axis.*

Proof: Compute the eigenvalue/eigenvector pairs of $B + B^T$ to be $\{\lambda_i, u_i\}$, $i = 1, 2, 3$. If $\lambda_i = 0$ for $i = 1, 2, 3$, then we have $v = 0$ and $\hat{\omega} = B$. In this case we can not recover the normal of the plane n. Otherwise, if $\lambda_1 > 0$, and $\lambda_3 < 0$, then we have $v \times n \neq 0$. Let $\alpha \triangleq \|v/d\| > 0$, let $\tilde{v} = v/\sqrt{\alpha}$ and $\tilde{n} = \sqrt{\alpha} n$, and let $\beta \triangleq \tilde{v}^T \tilde{n}$. According to Lemma 1, the eigenvalue/eigenvector pairs of $B + B^T$ are given by

$$\left.\begin{array}{ll} \lambda_1 = \beta + \alpha > 0, & u_1 = \frac{1}{\|\tilde{v}+\tilde{n}\|}(\tilde{v} + \tilde{n}) \\ \lambda_3 = \beta - \alpha < 0, & u_3 = \frac{1}{\|\tilde{v}-\tilde{n}\|}(\tilde{v} - \tilde{n}) \end{array}\right\} \tag{8}$$

Then $\alpha = \frac{1}{2}(\lambda_1 - \lambda_3)$. It is direct to check that

$$\|\tilde{v} + \tilde{n}\|^2 = 2\lambda_1, \qquad \|\tilde{v} - \tilde{n}\|^2 = -2\lambda_3 \tag{9}$$

Then (8) and (9) give a solution:

$$\left.\begin{array}{lcl} \tilde{v}_1 & \triangleq & \frac{1}{2}(\sqrt{2\lambda_1}\, u_1 + \sqrt{-2\lambda_3}\, u_3) \\ \tilde{n}_1 & \triangleq & \frac{1}{2}(\sqrt{2\lambda_1}\, u_1 - \sqrt{-2\lambda_3}\, u_3) \\ \hat{\omega}_1 & \triangleq & \frac{1}{2}((B - \tilde{v}_1 \tilde{n}_1^T) - (B - \tilde{v}_1 \tilde{n}_1^T)^T) \end{array}\right\} \tag{10}$$

The estimate of $\hat{\omega}_1$ is computed as above because due to measurement noise, in general $B - \tilde{v}_1\tilde{n}_1^T \notin so(3)$, so we take the projection of $B - \tilde{v}_1\tilde{n}_1^T$ onto $so(3)$.

However, the eigen-decomposition $\{\lambda_i, u_i\}$ is not unique – there is a sign ambiguity in the eigenvectors u_1 and u_3. This sign ambiguity leads to a total of 4 possible solutions for $\tilde{v}$ and $\tilde{n}$ computed according to (10). In order to reduce the number of physically possible solutions, we impose the "positive depth constraint" – since the camera can only see points that are in front of it, we must have $n^T e_3 > 0$. The positive depth constraint reduces the ambiguity of solutions to the true solution $\{v, n, \omega\}$ plus the ambiguous solution $\{v^+, n^+, \omega^+\}$, where $v^+ = \|v/d\|\, n$, $n^+ = \frac{1}{\|v/d\|} v/d$, $\hat{\omega}^+ = \hat{\omega} - nv^T/d + vn^T/d$. If $v^T e_3 = 0$, the ambiguous solution may be eliminated by the positive depth constraint, and if $v \times n = 0$, the two solutions are equivalent. ∎

3.3 Disambiguation of Vision Estimates

We assume that we have stored in memory the images $\{x_i^0\}_{i=1}^m$ of features points on the landing pad taken from the desired configuration. These features could for example be the corners of the typical "H" pattern found on most helicopter landing pads. The corners of the "H" pattern satisfy the general configuration condition of proposition 4, and hence the vision based motion estimation problem is well conditioned with respect to this pattern.

Let $(p_0, I) \in SE(3)$ be the configuration of the desired camera frame above the landing pad, and let $d_0 = -n_F^T p_0 > 0$ be the desired distance of the camera to the landing plane with known surface normal $n_F \in \mathbb{R}^3$.

Proposition 5 *If $d_0, d > 0$ are the distances from the plane to camera frames 0 and 1, respectively, and $A = (R + \frac{1}{d}pn^T) \in \mathbb{R}^{3\times3}$ is the planar essential matrix, relating the two camera frames, then the distance from camera frame 1 to the plane is given by $d = d_0/\det(A)$.*

The knowledge of d_0 allows to compute d, which solves the scale ambiguity p/d in the discrete algorithm and v/d in the differential algorithm. The knowledge of n_F allows us to disambiguate the pair of solutions by taking the one that minimizes $\|n_{\text{est}} - R_{\text{est}}^T n_F\|$, where n_{est} is the vision estimated surface normal, and R_{est} is the estimated rotation matrix according to the discrete algorithm.

3.4 Real-time Implementation Considerations

The most computationally intensive task in the vision algorithm is the least squares estimation of the B matrix, which involves the singular value decomposition (SVD) of the matrix $\mathbf{G} \in \mathbb{R}^{3m\times9}$ where m is the number of tracked feature points. The cost of the SVD of a matrix $M \in \mathbb{R}^{n\times m}$ for $n \leq m$ is $O(n^2m)$ flops. Then, as the number of feature points m increases, the cost of the vision algorithms grows as $O(m)$. We have implemented the above algorithm using the MATHLIB C/C++ library in Matlab, and have found that on a 450 MHz Pentium II running Linux, the vision algorithms can perform motion estimation based on 25 tracked feature points at a rate of over 150 Hz.

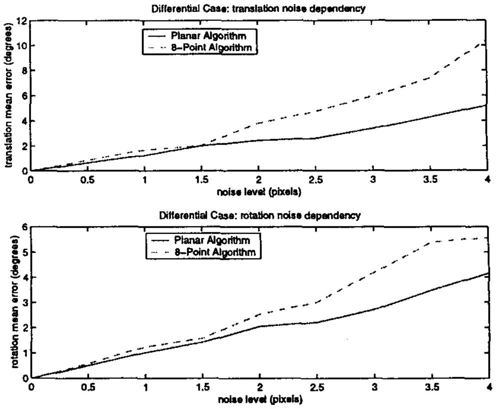


Figure 2: Noise Sensitivity

4 Vision Performance Evaluation

Of utmost consideration is the performance of the vision sensor in the presence of noise in the image measurements. Another important criteria is how the estimation errors depend on different camera motions with respect to the observed plane. By way of comparison, for each simulation we also show the performance of the differential "8-point algorithm" described in [7]. For more detailed simulation results and analysis, see refer to [9].

For all simulations, we generated 50 random points uniformly distributed within a 60° camera field of view. The image correspondences and the optical flow measurements were corrupted by additive white Gaussian noise. For evaluating the 8-point algorithm, we randomly scattered the depths of these points uniformly between distance of `zmin=100` and `zmax=400` focal lengths. For evaluating the planar algorithm, we placed the points on the fronto-parallel plane at a distance of `(zmax+zmin)/2`.

In Figure 2 we show the performance of the algorithm as a function of noise in the image velocity measurements. Notice that the planar algorithm is more robust to noise than the 8-point algorithm.

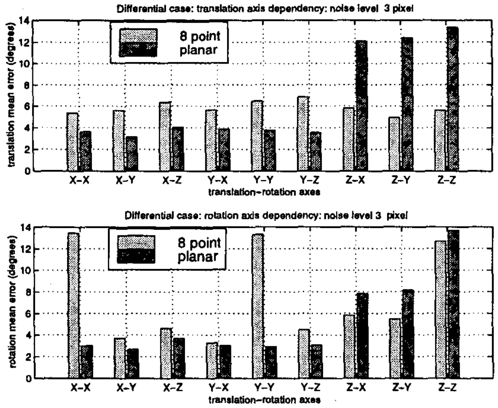


Figure 3: Sensitivity to translation-rotation axes

In Figure 3 we show the noise sensitivity of the algorithm with respect to different motions relative to the plane. We ran the algorithm for each different translation-rotation axis pair for a noise level of 3 pixels standard deviation. In general, the planar algorithm performs better than the 8-point algorithm except when the translation axis is parallel to the optical axis (and hence the surface normal of the plane). The higher noise sensitivity in this case can be seen as an overall numerical sensitivity to perturbations in the algebraic eigenvalue/eigenvector problem when there are repeated eigenvalues. The situation of having repeated eigenvalues occurs in the planar differential algorithm when the translational motion is parallel to the surface normal of the plane and in the 8-point algorithm when the translation and rotation axes are parallel [7].

5 UAV Control Design

In this section, we give the dynamic model of a UAV helicopter, and discuss a control design based on differential flatness.

We parameterize R by the "roll, pitch, yaw" Euler angles $\Theta = (\phi, \theta, \psi)^T$, away from the representation singularity $\theta = -\pi/2$. The Euler angle velocities and body angular velocity are related through $\dot{\Theta} = \Psi\omega$. Refer to [8] for details on Euler angles.

A dynamical model of a UAV helicopter based on rigid body dynamics and the force and moment generation process can be expressed as

$$\begin{aligned}
\ddot{p} &= \frac{1}{m} R \begin{bmatrix} -T_M \sin a_{1s} \\ T_M \sin b_{1s} - T_T \\ -T_M \cos a_{1s} \cos b_{1s} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} && (11) \\
\dot{\Theta} &= \Psi\omega && (12) \\
\dot{\omega} &= \mathcal{I}^{-1}(\tau^b - \omega \times \mathcal{I}\omega) && (13)
\end{aligned}$$

where m is the mass of the UAV, g is the gravitational constant, T_M and T_T are the thrusts generated by the main and tail rotors, a_{1s} and b_{1s} are the longitudinal and lateral tilt angles of the main rotor blades, the inertial matrix is $\mathcal{I} \in \mathbb{R}^{3\times3}$ and the body torque is $\tau^b \in \mathbb{R}^3$.

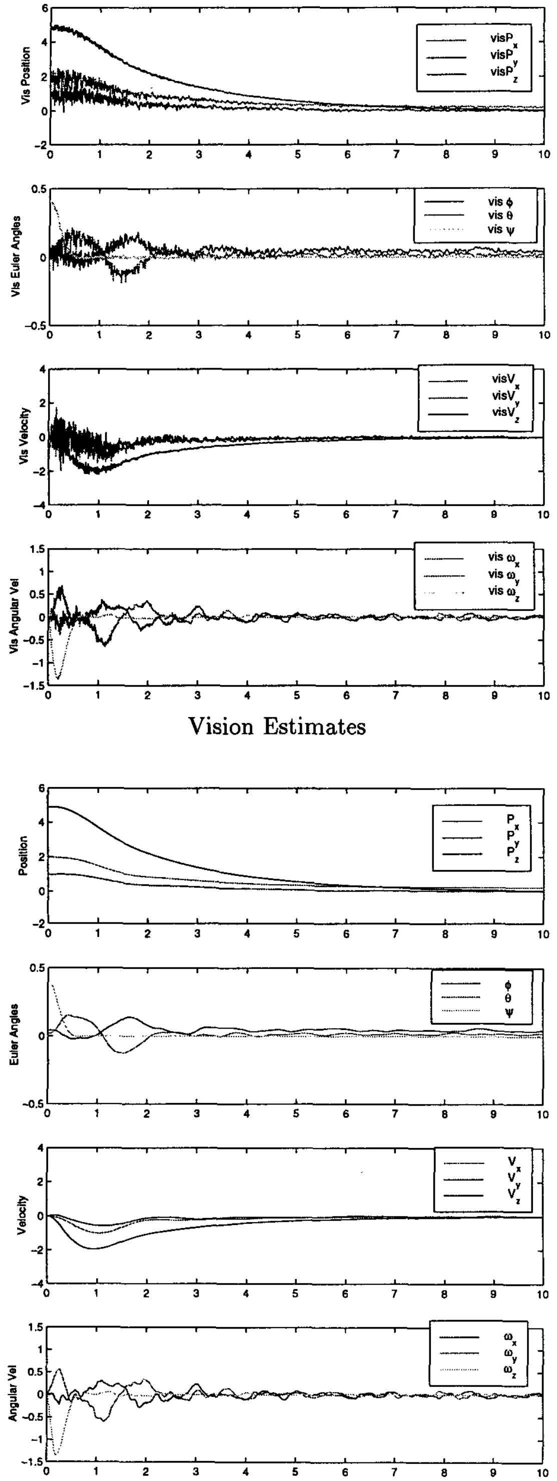
5.1 Differential Flatness

A system is said to be *differentially flat* [6] if there exist output functions, called flat outputs, such that all states and inputs can be expressed in terms of the flat outputs and their derivatives. Differential flatness has been applied on approximate models of aircraft for trajectory generation and control. Given an output trajectory, outer flatness [11] has been proposed and used for generating an inner trajectory for an inner system to track. In this scheme, one partitions a system into an "inner system" (*e.g.* the attitude dynamics) and an "outer system" (*e.g.* the position dynamics). The scheme applies to systems for which the outer system is flat.

Based on the natural time scale separation between position and attitude dynamics of a helicopter, the outer and inner systems are defined since attitudes can be treated as inputs to the position dynamics. It has be shown in [3] that the helicopter rigid body dynamics are approximately differentially flat, with the position and heading $\{p, \psi\}$ as the flat outputs. The approximation is based on the assumption that the coupling terms $a_{1s}, b_{1s}, T_T/T_M$ are small and can be neglected from the model. With this assumption, the outer system dynamics are simply equation (11) with $a_{1s} = b_{1s} = T_T = 0$, while the inner system dynamics are equations (12, 13). In [3], it is shown that the approximation for the outer system is differentially flat, and a stabilizing controller is designed based on this approximation. The interested reader is referred to [3] for detailed treatment of the control design.

5.2 Closed Loop Simulation Results

We present the simulation results of the "vision in the control loop" landing scheme. In this simulation, the initial conditions of the UAV are $p = (2, 1, 5)^T$ meters away from the desired landing configuration above the landing pad (the origin), the initial heading is $\psi = 0.4$ radians. The additive noise on the image correspondences and optical flow is 2 pixels standard deviation. The top of Figure 4 shows the vision based estimates of position, orientation, linear and angular velocity, while the bottom of the figure shows the trajectory followed by the UAV using the noisy state estimates for the control computations. The simulation results attest to the feasibility of using computer vision in the feedback control loop under relatively large noise levels in the feature-tracking estimates. Please refer to [9] for more detailed discussion and simulations of the vision sensor in the control loop.



Vision Estimates

Actual Trajectory

Figure 4: Vision in the loop landing, 2 pixel noise std

6 Conclusions and Future Work

In this paper, we presented the problem of using computer vision to control the landing of an Unmanned Air Vehicle. We derived a new geometric method of estimating the camera angular and linear velocity relative to a planar scene, and a presented performance evaluation of the algorithm. The vision sensor was put into the feedback loop of a UAV controller based on differential flatness. Through simulation results, the vision guided UAV was shown to perform stable landing maneuvers for large levels of image measurement noise.

We are currently implementing the above vision algorithm and controller on a model helicopter as part of the UC Berkeley BEAR (BErkeley Aerial Robot) project. Our UAV is a Yamaha R-50 model helicopter, on which we have mounted computers, inertial navigation sensors, GPS, and a vision system, consisting of a camera, a real-time feature tracker board, and a Pentium II running Linux.

References

[1] O.D. Faugeras and F. Lustman. Motion and structure from motion in a piecewise planar environment. *Int. J. of Pat. Recog. and AI*, 2(3):485–508, 1988.

[2] K. Kanatani. Detecting the motion of a planar surface by line & surface integrals. In *Computer Vision, Graphics, and Image Processing*, volume 29, pages 13–22, 1985.

[3] T.J. Koo and S.S. Sastry. Differential flatness based full authority helicopter control design. In *IEEE Conference on Decision & Control*, 1999.

[4] H.C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. In *Nature*, volume 293, pages 133–135, 1981.

[5] H.C. Longuet-Higgins. The reconstruction of a plane surface from two perspective projections. In *Proc. of Royal Society of London*, volume 227 of *B*, pages 399–410, 1986.

[6] M. Fliess, J. Lévine, Ph. Martin,and P. Rouchon. Flatness and defect of nonlinear systems: introductory theory and applications. *Int. J. of Control*, 61(6), 1995.

[7] Y. Ma, J. Košecká, and S.S. Sastry. Linear differential algorithm for motion recovery: A geometric approach. *Int. Journal on Computer Vision*, to appear.

[8] R.M. Murray, Z. Li, and S.S. Sastry. *A Mathematical Introduction to Robotic Manipulation.* CRC Press, 1993.

[9] O. Shakernia, Y. Ma, J. Koo, and S.S. Sastry. Vision guided landing of an unmanned air vehicle. *Asian Journal of Control*, to appear.

[10] M. Subbarao and A.M. Waxman. On the uniqueness of image flow solutions for planar surfaces in motion. In *Third IEEE Workshop on Computer Vision: Representation and Control*, pages 129–140, 1985.

[11] M. J. van Nieuwstadt and R. M. Murray. Outer flatness: Trajectory generation for a model helicopter. In *Proc. of European Control Conference*, 1997.

[12] J. Weng, T.S. Huang, and N. Ahuja. *Motion and Structure from Image Sequences.* Springer-Verlag, 1993.