Motion Estimation of a Miniature Helicopter Using a Single Onboard Camera

Anoop Cherian, Jon Andersh, Vassilios Morellas, Bernard Mettler, and Nikos Papanikolopoulos

Abstract— **This paper describes a technique for the estimation of the translational and rotational velocities of a miniature helicopter using the video signals from a single onboard camera. For every two consecutive frames from the camera, point correspondences are identified and Epipolar Geometry based algorithms are used to find the likely estimates of the absolute rotations and relative displacements. Images from onboard camera are often corrupted with various types of noises; SIFT descriptors were found to be the best feature descriptors to be used for point correspondences. To speed up the processing, we introduce a new representation of these descriptors based on compressive sensing formalisms. To estimate the absolute displacement of the helicopter between frames, we use the measurements from a simulated IR sensor to find the true change in altitude of the body frame, scaling other translational dimensions accordingly, and later estimating the velocities. Experiments conducted using data from a real helicopter in an indoor environment demonstrate promising results.**

I. INTRODUCTION

Today's interest in unmanned aerial vehicles (UAVs) has expanded to smaller vehicles that can operate indoors. Highly miniaturized UAVs have a number of useful applications that include surveillance, search and rescue, environmental monitoring, and exploration [1]. For this paper, the UAV under investigation is a miniature helicopter that is used in an indoor environment.

Miniature rotorcraft provide a robotic platform with a number of benefits. They have a broad flight envelope, ranging from a stationary hover to an airplane-like cruise and their inherent agility gives them a unique position among the variety of UAVs currently in use today [2]. However, miniature aerial vehicles like rotor craft present unique challenges. These arise from the type of environments and tasks that can be performed but also from the control challenges inherent to rotor craft flight. In addition, the stringent payload limitations restrict the computational power and sensors that can be located onboard the vehicle. As a result, building control systems for these vehicles is a difficult problem for an indoor application. The helicopter has to integrate the sensing and electronics into a payload that is limited to approximately 50g. Additional problems arise due to the UAV operating in an indoor environment where positioning sensors such as GPS or magnetometers are unavailable.

In order to improve the closed-loop control performance of the vehicle, we need to be able to estimate the body-frame velocities of the vehicle. Using an onboard MEMS based IMU, combined with GPS and magnetometer, is a typical approach and allows for estimation of the vehicle states during flight. Unfortunately, state estimation from the IMU without GPS aiding proves to be too inaccurate for feedback stabilization. The main problems with the IMU are due to vibrational noise as well as difficulty in separating the gravity component out of the signals from the accelerometers. In order to improve the accuracy of the state estimation, additional sensor information is thus sought. In this paper, we attempt to obtain this information from a video camera mounted onboard the vehicle and extract the state from captured image sequence. The goal of this research is to gather information on the vehicles rotational and translational velocities from the video feed with the plan of using this information to improve the state estimation performance.

We approach the problem from the perspective of Epipolar Geometry and try to estimate the motion of the helicopter from two consecutive video frames captured by the onboard camera. The first step in motion estimation using this approach is to find the point correspondences between the two frames. It was found that this is a challenging problem, since the images are very often corrupted with noise from vibrations of the vehicle and radio interference. Also, other problems like dramatic changes in the illumination of the environment due to uneven indoor lighting, white noise from the camera sensor, and extreme motion blur are typical. Thus Scale Invariant Feature Transform (SIFT) based feature descriptors were found to provide the best results for the point correspondences. Since computational time is a critical factor for the real-time implementation of our algorithm, we propose a novel idea of using a sparse representation of the SIFT descriptors, which we claim to be a faster way for finding matches across frames.

Once the correspondence problem is solved, we use Epipolar Geometry to find the fundamental matrix from the positions of the feature matches between the frames and later find the rotation matrix and translations from the fundamental matrix. Note, the translation vector estimated from the fundamental matrix will be normalized to unity and thus we use information from a simulated IR sensor (the data

This work has been supported in part by the National Science Foundation through grants #CNS-0324864, #CNS-0420836, #IIP-0443945, #IIP-0726109, #CNS-0708344, and #IIP-0934327.

A. Cherian is a Computer Science graduate student at the University of Minnesota, Minneapolis, MN 55455, USA cherian@cs.umn.edu

J. Andersh is a Computer Science graduate student at the University of Minnesota, Minneapolis, MN 55455, USA jander@cs.umn.edu

V. Morellas is with the Computer Science Department, University of Minnesota, Minneapolis, MN 55455, USA morellas@cs.umn.edu

B. Mettler is with the Department of Aerospace Engineering and Mechanics, University of Minnesota, Minneapolis, MN 55455, USA mettler@aem.umn.edu

N. Papanikolopoulos is with the Computer Science Department, University of Minnesota, Minneapolis, MN 55455, USA npapas@cs.umn.edu

is actually derived from a motion tracking system for the experiments in the current paper, corrupted with Gaussian noise), to give us an estimate of the actual difference in the altitude of the helicopter.

To begin, the paper provides related literature about the problem in section II. The hardware and software that make up the experimental lab environment is discussed in section III. An overview of our algorithm and details of the individual components in it are provided in section IV and its subsections. Section V describes the computation of the absolute displacement and the velocities, section VI shows results of the the experiments. Section VII provides concluding remarks.

II. Related Work

Vision based techniques for robot motion estimation and control have been researched extensively over the past few decades. Survey on some of the well-known techniques can be seen in [3]. Estimation of the state and motion of aerial vehicles is a more challenging problem due to the extremely dynamic environment and more degrees of freedom (6-DOF). Monocular camera-based state estimation techniques for aerial vehicles fall under one of the following two major categories: i) optical flow based methods [4], [5], and ii) multi view geometric approaches. Flow based methods often suffer from noise in the environment, which deters their use here. Multi view geometry based approaches have been used extensively in the area of structure from motion. [6] provides a detailed survey of the well-known techniques with their applications. But their usage to state estimation for aerial vehicles has been more or less restricted to motion control for landing [7], where the landing site is assumed to be planar and thus the points of interest form a strict homography between consecutive images.

A closely related approach to ours is [8], with an application to space-craft motion estimation for landing. Though the approaches are similar, the paper considers images from a downward looking camera which poses a different set of challenges. The difference arises due to the fact that the structure of the scene is very similar from frame to frame (due to downward looking camera), while in the approach discussed in this paper, the camera looks forward and the scene it observes can change significantly in short periods of time.

III. Experimental Architecture

The Interactive Guidance and Control Lab at the University of Minnesota provides an indoor environment for conducting experimental work with miniature rotorcraft [9]. In the lab, small off-the-shelf mini helicopters are used for research into a number of areas that include modeling, control, state estimation, path planning, computer vision, and teleoperation. The rest of this section will discuss the helicopter platform, the camera, and the components that are used for experimentation in the lab environment.

A. Helicopter

The vehicle used in this work is based on the Blade CX2 helicopter from E-Flite. It is a coaxial helicopter with a rotor diameter of 34.5 cm, a height of 18.3 cm, and weighs approximately 220g (with battery). Fig. 1 shows the helicopter with a video camera mounted. The white markers are reflective tape used to help the Vicon tracking system locate the helicopter.

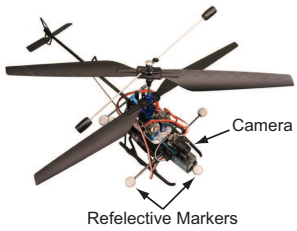


Fig. 1. Helicopter platform showing the onboard camera and reflective markers used for tracking.

As we move forward, we will need to test the algorithms on a platform that has additional sensing and processing. For this future work, a robotic platform has been developed that has the necessary capabilities. This platform can be outfitted to include various sensor configurations that can include an IMU, camera, ultrasonic range finder, and infrared proximity sensors. The onboard processing is done using a Gumstix Verdex board that allows for hardware interfacing and includes a wifi daughter card. For more details on the platform see [10].

B. Camera

The camera mounted onboard the helicopter is the Eyecam 2.4GHz Color Micro Wireless Video Camera System. The Eyecam is a color CMOS camera that generates NTSC video with 250K pixels. The camera transmits the video using a set frequency in the 2.4 GHz spectrum with the receiver being tunable to select the correct channel. Example images captured from the onboard camera are shown in Fig. 2.

Fig. 2. Images from the onboard camera.

C. Lab Systems

To determine the position and orientation of the helicopter during test flights in the lab, a Vicon motion tracking system is used. The Vicon system and video from the onboard camera are integrated into lab's software architecture. The software includes a Linux based system for running real time experiments, logging data, and interfacing the input and output from RC controllers. As part of the architecture, software modules have been written to read Vicon data, output control signals to the helicopter, read manual control

inputs, read data from onboard sensors, capture video, run control loops, and log data. To identify the helicopter, retro-reflective spherical markers are placed on the vehicle to form a pattern that can be uniquely identified. Once the ViconIQ software has identified the helicopter as an object, the position and orientation information can be streamed over a network connection to control or monitoring systems. The system is capable of providing the vehicle's 6 DOF motion in real time with a tracking accuracy of approximately 0.02 pixels which corresponds to sub-millimiter accuracy in the experimental space. For the experiments in this paper, data sets were collected at 50 Hz and contained information on the helicopters position, orientation, time stamp, and operator control signals. In addition, images from the onboard camera were captured at 5 Hz and stored in files with the time stamp embedded in the file name.

D. Estimation and Control

In the lab, closed-loop control experiments are currently conducted using readings from the Vicon system as the feedback source. The dynamic control is used to modify the helicopter flying qualities such as stability or control sensitivity in order to make the task easier for the human operator. Through experimentation we have found that doing state estimation using only an onboard IMU does not provide reliable results for closed loop control. To help compensate for the short comings of the IMU, we envision a scenario similar to the one shown in Fig. 3. By using both a camera and IMU as part of the state estimation process we hope to achieve results that are better than either sensor could provide individually.

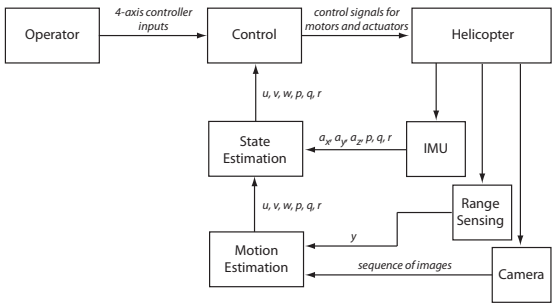


Fig. 3. Block diagram showing state estimation using an IMU, camera, and range sensor.

IV. Approach

In this section, we consider the problem of estimating the motion parameters of the UAV from images captured using the onboard camera. Fig. 4 shows an outline of the procedure.

First, we build the point correspondences between two consecutive frames captured by the onboard camera. The point correspondences are then used to find the fundamental matrix using the epipolar geometric constraint. The fundamental matrix is decomposed into the translation vector and rotation matrix. Finally the readings from the altimeter is used to scale the translation vector appropriately. In the following sections, each of these steps is explained in detail.

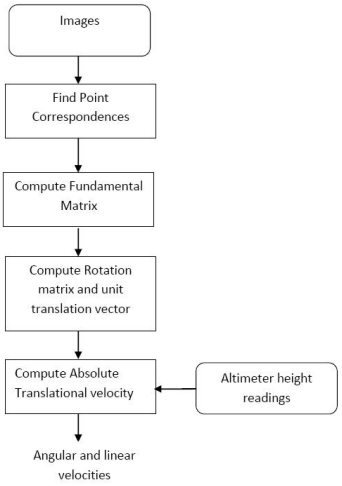


Fig. 4. Motion estimation algorithm outline.

A. Feature Vectors

The first step in any algorithm using two view geometry is the establishment of feature correspondences between the images. The correspondences can be established using a variety of features, like Harris-Corner descriptors [11], Affine invariant descriptors [12], etc. Considering the extremely noisy scenario at our disposal and the high accuracy requirement for the correspondences as discussed in the previous section, we chose to use the SIFT descriptors [13]. SIFT descriptors are 128 dimensional vectors and have been proven to be robust to affine transformations, noise, minor changes in viewpoint and illumination in the images. On the negative side, they are computationally expensive in both generating the descriptors and matching them. Thus they have rarely been used in real time applications. In this paper, we introduce a novel representation of these descriptors in a sparse basis using compressive sampling formalisms, which make them more tractable in matching across images . In the next few subsections, we introduce the compressive sensing ideas, and the unique representation we propose for these descriptors.

B. Sparse SIFT Representation

The basic motivation for the sparse representation of SIFT comes from compressed sensing. Compressive sensing suggests that if a given signal is dense in a basis, then there should exist an incoherent over-complete basis where the signal will be sparse [14]. For example, a given signal that is dense in the time domain, will have a sparse representation in the frequency domain. It was seen from the analysis of the structure of the SIFT descriptors that it is almost always dense when seen as a time domain signal of 128 time steps (see Fig. 5), which motivated us to seek a sparsity representation of it in an over complete basis where only a few of the new basis vectors might be needed to represent the original descriptor. Finding an incoherent sparse basis is not easy and thus we used the dictionary learning methods suggested in [15].

Given a large corpus of 128D SIFT descriptors $S = s_1, s_2, ..., s_n$ created from a random image set, we seek to find a sparse representation of each of these descriptors as a linear combination of N (> 128) basis vectors $b^i \epsilon R^{128}$ such that,

$$s^i \approx \sum_{j=1}^{N} b_j a_j^i = Ba^i \tag{1}$$

where B is the matrix with b_j as columns and $a^i \epsilon R^N$ is a sparse vector of weights. To find the optimal basis B, we solve a nonlinear optimization problem formulated in [15] over the corpus of descriptors mentioned previously:

$$\min_{b,a} \sum_i \|s^i - \sum_j a_j^i b_j\|_2^2 \ + \ \beta \|a^i\|_1 \tag{2}$$

$$s.t. \ \|b_j\|_2 \ \leq 1, \ \forall \ j \ \epsilon \ \{1, ..., n\}$$

The objective function in Eq. 2 balances two terms: (i) the quadratic term minimizes the L_2 error between the sparse representation and the original descriptor and (ii) the L_1 minimization term imposes the basis activation vector a^i to be sparse. The parameter β regularizes the penalty imposed by the L_1 constraint. The problem is convex in either b^i or a^i separately and thus can be solved alternately using the methods suggested in [15]. Once a sparse basis B is learned for the SIFT descriptors, we can use the following equation to find a sparse activation set f given a descriptor p,

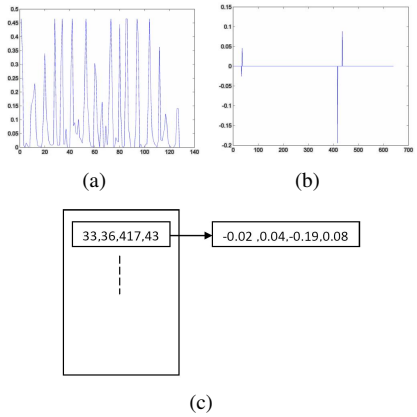
$$\min_f \|p - \sum_j f_j b_j\|_2^2 + \beta \|f\|_1 \tag{3}$$

The error in the original SIFT descriptor and the one approximated using the sparse basis is decided by the number of basis vectors in the dictionary and the regularization term β used in the Eq. 3. The larger the number of basis vectors N, the more the sparsity and lesser the error. Similarly, the smaller the value of β, the better the accuracy of reconstruction. Moreover, the better the desired accuracy of representation, the more the computational complexity increases with respect to time. Thus by fine tuning the parameters of the model, we can achieve robustness of the representation with better running time. Determining the optimal tradeoff is the key problem in successfully applying the sparse sift descriptors to motion estimation.

C. Sparse Sift Matching

The next step in the process is to derive a new representation of the sparse active basis that improves the feature matching. The motivation for our approach comes from the fact that there is a large set of basis vectors of which only a few are active (like say $N = 256$ and active set = 5). Assuming the active set is going to be evenly distributed across all the basis vectors, we can achieve a unique representation of a given SIFT descriptor as a text string of the index numbers of the active basis in a predefined order as shown pictorially in Fig. 5. The representation of the descriptor as a text string brings about the usage of a hashtable with any simple string hashing function to map appropriately, and there-by improving the feature matching performance.

There are quite a few subtle points that need to be addressed with this model of feature matching regarding the kind of errors we introduce at various stages of our algorithm: (i) we introduce an approximation error in the sparse representation that needs to be fine-tuned as a trade-off between performance and reconstruction, (ii) while creating the hash table, we may consider only the k largest basis vectors and thus reduce the length of the hash string. The reader should note that the reconstruction process actually brings robustness to the representation due to the fact that the original images are extremely noisy and thus two sift descriptors will not be the same. On the contrary, hashing is an exact matching process and thus many of the sparse SIFT descriptors will not be matched properly leading to a fewer number of correspondences. But we assume that the number of correspondences will be large enough for an accurate estimation of the motion using Epipolar geometry discussed in the next section. Fig. 5 illustrates the sparse SIFT algorithm and feature matching.



(a) (b) (c)

Fig. 5. (a) Plot of a normalized SIFT descriptor, (b) Sparse representation of the previous sift descriptor to a set of 640 basis vectors, (c) Hash table mapping of the nonzero sparse indices and their respective coefficients.

D. Finding Rotation Matrix and Unit Translation vector

Once the point correspondences between the images are known as discussed in the previous section, Epipolar geometric algorithms are used to derive the fundamental matrix. It is assumed from here on that the calibration of the camera is known and thus point correspondences are corrected to the camera intrinsic parameters. Fig. 6 shows the epipolar geometry between the point correspondences.

As is seen, the two camera centers and the world point all lie in the same epipolar plane and thus satisfies the constraint:

$$x_2^T F x_1 = 0 \tag{4}$$

where x_1 and x_2 are the homogeneous coordinates of the corresponding points in the two images respectively and F represents the 3×3 fundamental matrix. As is shown in [16], F can be written as:

$$F = RS \tag{5}$$

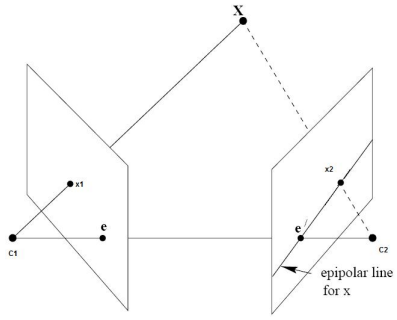


Fig. 6. Illustration of the epipolar constraint. As is seen the camera centers C1 and C2, along with the world point X, lie on the same epipolar plane.

where R is the 3×3 rotation matrix of motion and S is the skew-symmetric representation of the translation vector $T = [t_x, t_y, t_z]^T$, i.e.

$$S = \begin{pmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{pmatrix} \tag{6}$$

Note that, for tractability of the solution, the translation vector is conveniently normalized to unity. Given eight or more point correspondences between the two images, Eight Point algorithm [17] can be used to solve for F. Since the accuracy of the motion estimates are completely dependent on the accuracy with which the fundamental matrix is computed, RANSAC based methods and normalizations suggested in [6] are used while finding the fundamental matrix.

It can be seen that Eq. 5 implies $F^T F = S^T S$, which is a matrix of only the entries from T and can be solved from the fundamental matrix. As is shown in [16], there will be two possibilities for the unit translation vector which are different by a sign. This means that the unit translation vector can be found independent of the rotation matrix. As per [17], the rotation matrix can be computed from the SVD of the fundamental matrix as follows. Assuming $F = UDV^T$, where U and V are orthonormal matrices and D is a diagonal matrix. Define E as

$$E = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{7}$$

Then R is one of the following, $R = UEV^T$ or $R = UE^TV^T$. This can be verified as true by inspection. Note here that we can find the rotation matrix even without computing the translation vector using this approach. As is seen, there are two possibilities for the rotation matrix and two for the translation vector. This ambiguity can be easily solved using the method suggested in [16] by using only the values that will allow the X-coordinates of the point correspondences to lie in front of the camera.

V. Absolute Translation

In the previous section, we showed how to find the rotation matrix and the unit translation vectors from the fundamental matrix. Since it is just the epipolar constraint that is available, which is singular in one of its eigenvalues, division by any magnitude will give a different rotation matrix and a translation vector, which is why the translation was normalized to unit magnitude. Since the goal is to estimate the absolute velocity of the helicopter in the real world, this assumption poses a problem, however it can be effectively solved by incorporating some measure of scale of the translation and map it accurately to the real world coordinates.

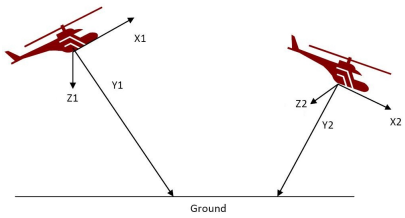


Fig. 7. Illustrates the location (X1, Y1, Z1) and (X2, Y2, Z2) of the helicopter at two time instants along with the orientation. The altimeter reads Y1 and Y2 at the two instants.

In this paper, we suggest the usage of the readings from an infrared sensor attached to the helicopter. The altitude measurement can provide information about the absolute difference in the altitude of the helicopter at the time instants of the two frames and thus can provide a mapping to the absolute scale. If δy represents the difference in altitude of the helicopter, and if θ and ϕ give the roll and the pitch of the helicopter (gyroscope readings are used as initial estimates on the orientation of the helicopter) with respect to one of the frames as seen by the gyroscope (Fig. 7), then the absolute altitude difference with respect to the first frame can be written $\delta Y_1 = cos(\theta)cos(\phi)\delta y$. From this, the scaling of the translation can be found and hence the absolute translation vector. Later, same technique is used for determining the linear velocity of the helicopter.

VI. Experiments

The vision algorithms were implemented partly in C++ and partly in Matlab. Every frame from the camera is matched against a frame three time steps away. The frames are of size 640x480, which is rescaled to 160x120 for reducing the motion blur and improving the speed with which the SIFT descriptors are generated. The frames are applied with additional deblurring filters and low pass filters to reduce the effects of radio interference noise. Finally they are used for finding the point correspondences using the sparse sift representation mentioned in section V. Table I gives the performance improvement using the sparse SIFT representation for matching the features compared to normal SIFT along with information on the average number of SIFT features present in each of the frames.

Once the point correspondences are computed, we estimate the fundamental matrix and later the rotation matrix along with the unit translation vector. The scale is determined from the IR sensor readings transformed appropriately and later the computed translation vector is used to estimate the velocities. Fig. 8 gives the plots of the actual and the

descriptor	*Max #matches*	*Min time*	*Max time*	*Mean time*	*total time*
Normal SIFT	80	0.19	1.85	0.71	479.73
Sparse SIFT	29	0.19	2.09	0.55	369.8

TABLE I

A COMPARISON OF SIFT WITH SPARSE SIFT DESCRIPTOR PERFORMANCE AGAINST 675 FRAME PAIRS (TIME IN SECONDS).

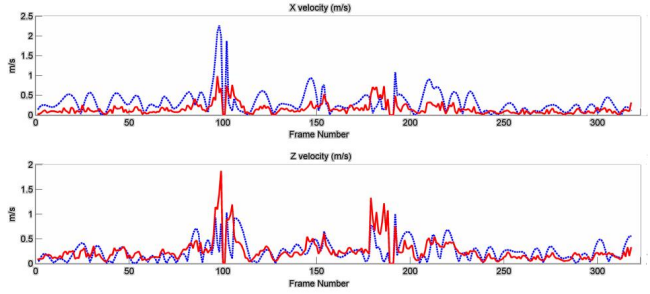


Fig. 8. shows the absolute estimated velocity (red) and the absolute true velocity (blue dotted) in the X and Z axes respectively in a regular flight session. The x-axis is the frame number and the y-axis is in meters/seconds.

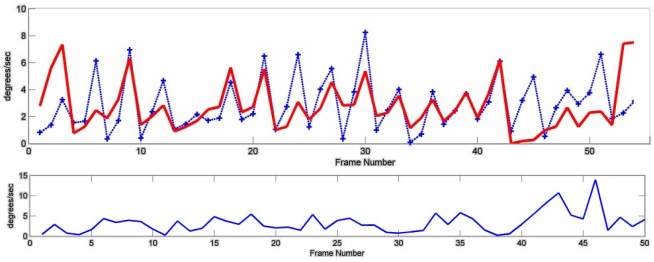


Fig. 9. First plot shows the norm of estimated angular velocity vector $[\omega_x, \omega_y, \omega_z]^T$ in degrees against the angular velocity found from the vicon camera system. The second plot shows the normed difference in the estimated and true angular velocity vectors. The x-axis is the frame number and the y-axis is in degrees.

estimated linear velocities in the X and Z directions respectively. To find the angular velocities, we need to transform the estimated rotation matrix to Euler angles. Since matrices do not commute, the Euler angles will not be unique. For example, a given rotation matrix can be decomposed into a rotation about the X axis first and then a rotation about the Y axis, or vice versa. To disambiguate the solutions, we assume that the helicopter will not make rotations of more than 20 degrees between frames used for calculations. Fig. 9 shows a plot of the norm of the difference of the estimated angular velocity vector $[\omega_x, \omega_y, \omega_z]^T$ (in degrees/sec) from the true angular velocity vector given by the vicon system. As can be seen, the angular velocity estimation is close to the actual values. A summary of the experiments and the errors encountered are shown in Table II. The entire algorithm was seen to take approximately 0.5 seconds (in Matlab) to run for a pair of images on a Pentium IV machine with a 2Ghz processor and 2GB RAM.

Direction	*Min abs Error*	*Max abs Error*	*Mean abs Error*	*Std. Dev.*
X (m/s)	0.0	2.62	0.21	0.07
Z (m/s)	0.0	1.71	0.15	0.03
ω (deg/s)	0.24	13.9	3.29	6.79

TABLE II

A SUMMARY OF FLIGHT EXPERIMENTS

VII. CONCLUSION AND FUTURE WORK

In this paper, we proposed a way of estimating the translational and rotational velocities of a miniature rotorcraft using vision-based techniques. To solve the scaling problem introduced by multiple view geometry, we used the readings from an infrared sensor, and showed that our method estimates the velocity with good accuracy. Going forward, we plan to improve miniature UAV state estimation by fusing the state information obtained using a camera with readings from an onboard IMU.

REFERENCES

[1] W. Williams and M. Harris, "The challenges of flight-testing unmanned air vehicles," in *Proceedings of the Systems Engineering, Test and Evaluation Conference*, October 2002.

[2] B. Mettler, C. Dever, and E. Feron, "Scaling and dynamic characteristics of miniature rotorcraft," *AIAA Journal of Guidance, Control and Dynamics*, vol. 27, no. 3, 2004.

[3] G. Desouza and A. Kak, "Vision for mobile robot navigation: a survey," pp. 237–267, 2002.

[4] J. Kehoe, A. Watkins, R. Causey, and R. Lind, "State estimation using optical flow from parallax-weighted feature tracking," *Proc. of the 2006 AIAA Guidance, Navigation and Control Conference*, August 2006.

[5] S. Soundararaj, A. Sujeeth, and A. Saxena, "Autonomous indoor helicopter flight using a single onboard camera," *Proc. International Conference on Intelligent RObots and Systems (IROS)*, 2009.

[6] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.

[7] O. Shakernia, R. Vidal, C. Sharp, Y. Ma, and S. Sastry, "Multiple view motion estimation and control for landing an unmanned aerial vehicle," *Proc. IEEE Intern. Conf. On Robotics and Automation*, vol. 3, no. 3, pp. 2793–8, September 2002.

[8] A. Johnson and L. Matthies, "Precise image-based motion estimation for autonomous small body exploration," *Artificial Intelligence, Robotics and Autonomation in Space (iSAIRAS)*, pp. 627–634, 1999.

[9] N. Kundak and B. Mettler, "Experimental framework for evaluating autonomous guidance and control algorithms for agile aerial vehicles," in *Proceedings of the European Control Conference*, July 2007.

[10] J. Andersh, B. Mettler, and N. Papanikolopoulos, "Miniature embedded rotorcraft platform for aerial teleoperation experiments," *Mediterranean Conference on Control and Automation*, June 2009.

[11] C. Harris and M. Stephens, "A combined corner and edge detector," in *Alvey vision conference*, vol. 15. Manchester, UK, 1988, p. 50.

[12] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *International Journal of Computer Vision*, vol. 60, no. 1, pp. 63–86, 2004.

[13] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

[14] E. Cands and J. Romberg, "Sparsity and incoherence in compressive sampling," *Inverse Problems*, vol. 23, pp. 969–985, 2006.

[15] R. Raina, A. Battle, H. Lee, B. Packer, and A. Ng., "Self-taught learning: Transfer learning from unlabeled data," *In Proceedings of the 24th International Conference on Machine Learning*, 2007.

[16] H. Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature*, pp. 133 – 135, 1981.

[17] R. Hartley, "Estimation of relative camera positions for uncalibrated cameras," *Proc. European Conference on Computer Vision*, vol. 92, pp. 579–587, 1992.