Visual Odometry: Autonomous UAV Navigation using Optic Flow and Stereo

Reuben Strydom, Saul Thurrowgood, Mandyam V. Srinivasan
The University of Queensland, Brisbane, Australia

Abstract

Visual odometry is vital to the future of mobile robotics. In this paper, we demonstrate a method that combines information from optic flow and stereo to estimate and control the current position of a quadrotor along a pre-defined trajectory. The absolute translation in 3D is computed by combining the optic flow measurements between successive frames and stereo-based height over ground. The current 3D position, as estimated from path integration of the incremental translations, is controlled in closed loop to follow the prescribed trajectory. The performance of the system is evaluated by measuring the error between the initial and final positions in closed circuits. This error is approximately 1.7% of the total path length.

1 Introduction

The growing interest in autonomous navigation has identified the need for robust, cost effective and efficient methods to determine the accurate location of unmanned vehicles. Visual odometry, one method of estimating position and orientation, has been implemented in a number of applications where conventional wheel-based odometry cannot be used. These applications include but are not limited to flying vehicles, legged robots and marine vessels.

Many conventional odometry solutions are prone to unpredictable errors in the measurements delivered by accelerometers, gyroscopes and wheel encoders. For example, it was found on the Mars Exploration Rovers that after a small translation over large rocks, steep slopes or sandy ground, visual odometry was required to correct for errors arising from jerky motions in 3D and wheel slip [Maimone et al., 2007]. Visual odometry estimates motion by monitoring changes in the sequential images that are acquired by a vision system on-board the vehicle. Features are selected and matched over successive frames to compute veridical egomotion that is unaffected by wheel slip [Fraundorfer and Scaramuzza, 2012; Maimone et al., 2007].

Authors are associated with the Queensland Brain Institute (QBI), the School of Information Technology and Electrical Engineering (ITEE), University of Queensland, St Lucia, QLD 4072, Australia, and the ARC Centre for Excellence in Vision (email: reuben.strydom@uqconnect.edu.au).

The trend towards visually guided robots has generated a number of methods to estimate the position of a vehicle. Egomotion estimation has been demonstrated using monocular cameras. One classical optic-flow based approach computes the translations and rotations by using a downward facing camera [Garratt and Chahl, 2008]. In a related approach, designed primarily for terrain following, velocity information obtained by GPS is combined with optic flow measurements to determine the height over ground. [Kendoul et al., 2009] presents a vision-based autopilot, which incorporates an optical-flow based vision system to perform autonomous localization as well as three-dimensional scene mapping. The suggested method incorporates a static pressure sensor to provide scale information in the optic flow measurements. A method developed by [Forster et al., 2014] presents a semi-direct visual odometry (SVO) algorithm that uses a monocular vision system. Subpixel precision is obtained by using pixel intensities directly instead of landmarks to determine 3D points to compute egomotion. Additionally, they implement a probabilistic depth filter for each 2D feature to estimate its position in 3D. Egomotion is estimated when the depth filter attains a prescribed certainty. The SVO method can run at high frame rates, however it would appear that it requires a high frame rate to continue to track the correct pseudo features.

A Simultaneous Localization and Mapping (SLAM) system, using an actuated laser scanner presents a different technique to approximate vehicle trajectories [Newman et al., 2006]. This method utilises the laser scanner to incrementally build a 3D point-cloud of the local environment. Images suggesting loop closure are processed with the corresponding laser point-cloud to perform loop closure correction. However, it is difficult and expensive to implement bulky laser sensors on an unmanned areal vehicle (UAV). Another SLAM-based method, by [Lemaire et al., 2007], demonstrates that vision is a viable solution for odometry in autonomous vehicles. Their method provides two options to determine the motion of a ground and air vehicle, (1) a classical stereovision implementation of a SLAM system and (2) a monocular mapping approach.

Stereo-based methods have the advantage of eliminating the scale ambiguities that are inherent in monocular systems [Kitt et al., 2010]. The authors demonstrate the feasibility of using a stereovision scheme for visual odometry. The method uses a corner-like feature-matching algorithm for detecting correspondence between image pairs. The proposed method does not track features over multiple frames; rather, the features are matched between consecutive images, to estimate frame-

to-frame egomotion in dynamic environments. A study performed by [Heng et al., 2014] investigates visual mapping with the use of a forward-looking stereo camera pair and a downward facing monocular camera. The stereo camera is the primary sensor and is used to construct an octree-based depth occupancy map, while the monocular camera is used for velocity measurements obtained from optic flow information. Data is also transmitted off-board to perform a large-scale SLAM implementation [Heng et al., 2014]. Another stereo-based study [Olson et al., 2000] has examined the influence of errors in orientation sensing on odometry. [Buskey et al., 2002] describes a visually driven state estimator that uses optic flow. The algorithm computes flow vectors by matching image pairs using Harris corners. The proposed method is capable of estimating egomotion velocities by incorporating stereo-based measurements. Another paper by the same laboratory compares two methods for underwater localisation: a self-localising static network of acoustic nodes and an optic flow and stereo vision system. The visual odometry system was updated at 10 Hz [Corke et al., 2007]. [Nistér et al., 2006] combines a monocular and stereo-based odometry, in an application that employs a stereo rig mounted on a ground vehicle. The method demonstrates a feature tracker, which matches points between frames to determine egomotion. Optical-flow-based methods have been used to obtain state estimation, computed using both stereovision and monocular vision setups [Srinivasan et al., 1997; Hrabar et al., 2005; Achtelik et al., 2009; Kendoul et al., 2009].

Sensor fusion is a common method employed for the guidance of autonomous robots to increase robustness and accuracy of results obtained by sensors such as encoders, GPS and inertial measurement units (IMUs). [Nourani-Vatani and Borges, 2011] present a visual odometry scheme for ground vehicles that measures optic flow in image sequences obtained from a downward-looking camera that images the ground. It estimates egomotion using a template-matching algorithm as the image shifts from frame to frame and demonstrates how the positioning of the camera at a fixed height above the ground simplifies the problem of computing odometry for a ground vehicle. Another paper from the same laboratory combines ground-based visual odometry (as described above) with IMU information to achieve 3D path integration on terrestrial vehicles [Nourani-Vatani et al., 2008]. A method incorporating an RGB-D camera (Microsoft Kinect), obtains a 3D map of the environment as well as computing egomotion [Huang et al., 2011]. To reduce odometric drift, the RGB-D map is utilised as a quasi-SLAM system to detect loop closures.

Here we present a new approach for estimating and controlling the motion of a quadrotor aircraft along a prescribed trajectory. The method combines stereo and optic-flow information, and does not require GPS. The feasibility and accuracy of the technique are tested in outdoor flights.

2 Experimental Platform

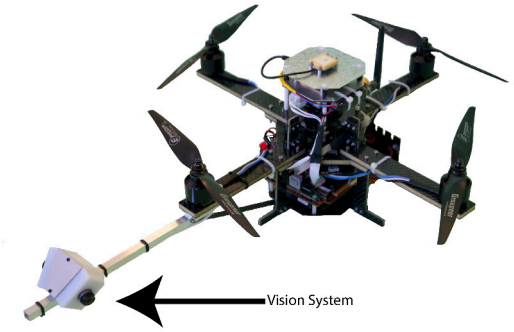

Figure 1 – Quadrotor used for testing

The platform used to conduct the flight experiments is a custom built quadrotor, (Figure 1). A MicroKopter flight controller, speed controller and motors were used for rotor control. The system was fitted with a PC104 computer featuring an Intel core 2 Duo - 1.5GHz dual core processor, and sensors including a MicroStrain inertial measurement unit (IMU), and a vision system, as shown in Figure 2.

Note that a Piksi real time kinematics GPS (RTK-GPS) kit was installed, however due to technical difficulties with the early firmware releases it was not possible to obtain a RTK lock between the two GPS units. This compromised the GPS-based ground-truthing of the platform's trajectories, however ground-truthing was achieved visually, as illustrated in Figure 10.

Figure 2 – Vision system onboard flight platform

The vision system consists of two fisheye cameras (Point Grey Firefly MV cameras with Sunex DSL216 lenses) – each with a visual field of approximately 190° – mounted back to back. The cameras are tilted toward each other by 10° to provide stereo overlap of approximately 30° by 130°, with a stereo baseline of 10 cm. The cameras are also tilted downward by 45° to capture a larger portion of the ground in the field of view (FOV). The cameras are calibrated using a similar method to [Kannala and Brandt, 2006]. The images from the two cameras are stitched together to obtain a panoramic image that covers a FOV of approximately 360° (azimuth) and 150° elevation with approximately 360 by 220 pixel resolution.

3 Visual Odometry and Control

3.1 Egomotion Computation using Optic Flow

Optic flow is the relative motion perceived between the on-board camera and the surrounding environment. In this method, optic flow is measured over the panoramic image and is used to compute the motion of the quadrotor (egomotion).

The cameras are synchronised by monitoring the frame time-stamps of the two cameras and continuously adjusting the phase of the right camera to match that of the left camera. The two images are normalised to match contrast before they are stitched to compute optic flow. Optic flow is computed in real time at (25 Hz) on the stitched panoramic image, using a pyramidal block-matching method. The image is segmented into a grid of 25 x 15 sub-regions (each 14 x 14 pixels). The optic flow in a given sub-image is computed by seeking matches in a 7x7 patch of texture within this sub-region, across two successive frames. Matches are tested over a 7x7 search area, and a Sum of Absolute Differences (SAD) measure of image difference is used to find the best match. Subpixel refinement is accomplished using an equiangular fitting method by [Shimizu and Okutomi, 2003].

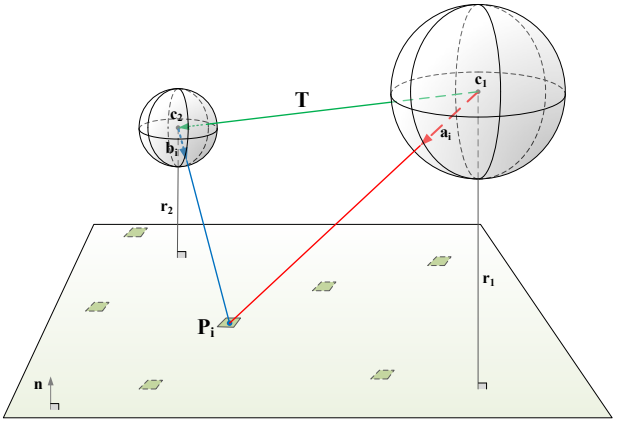

Figure 3 - Measurement of translation and rotation of view sphere over a plane. The sketch illustrates movement of the vision system over a plane, translating over a distance T from position c_1 to position c_2. r_1 and r_2 are the heights above the ground at positions c_1 and c_2. r_1 is set to be equal to 1.0 for the egomotion estimation. a_i and b_i are the unit vectors directed at a point P_i on the ground plane from positions c_1 and c_2 respectively.

The algorithm presented assumes that the ground approximates a plane. The objective, as illustrated in Figure 3, is to estimate the egomotion between the two positions c_1 and c_2, in terms of a translation vector (T) and a rotation about a vector (Ω). Although the vision system is slightly offset from the aircraft's yaw axis of rotation, the coupling between yaw motion and translation is negligible as the points used to compute translation are far away. At the two positions c_1 and c_2, the heights are represented by r_1 and r_2 respectively. To estimate the egomotion, the height r_1 at position c_1 is set to 1.0, which will provide the motion between the two frames, divided by the unspecified height at c_1. Consider a set of points P_i, which represent the 7 x 7 image block on the ground plane with unit vectors a_i at position c_1, and unit view vectors b_i at c_2. The vectors b_i are obtained by computing the optic flow. a_i and b_i are used to obtain the best estimate of the rotation (Ω) and translation (T) that characterize the egomotion from c_1 to c_2. The IMU is used to determine the normal vector to the ground (n) at c_2. Let a vector a_i' represent the vector a_i rotated through (Ω), using the Rodrigues algorithm.

$$a_i' = rodrigues(-\Omega)a_i \quad (1)$$

The output of the Rodrigues algorithm is a 3 x 3 rotation matrix computed from the rotation vector (Ω), which is the input.

Of the set of vectors a_i', the subset of vectors pointing to the ground are used to determine the translation. The vectors directed to the sky are not used for the translation computation because these regions are effectively an infinite distance away and thus not affected by translation. All of the a_i' vectors are, however, affected by rotation. With this in mind, for every rotation and translation the unit view vectors s_i at c_2 are as follows:

$$s_i = a_i', \text{ when } a_i' \text{ does not intersect the ground} \quad (2)$$

$$s_i = \frac{a_i'}{a_i' \cdot (-n)} - T, \text{ when } a_i' \text{ intersects the ground} \quad (3)$$

The optimization for the best rotation and translation uses an iterative algorithm in the NLopt library [Johnson, 2010], which utilizes a derivative free gradient descent method (BOBYQA) [Powell, 2009]. The objective function used to obtain a best estimate of the translation (T) and rotation (Ω) is:

$$\underset{T,\Omega}{argmin} \sum_{i=1}^{N} \left\| \frac{s_i}{\|s_i\|} - b_i \right\| \quad (4)$$

In equation (4) N = 400. The denominator in equation (3), $a_i' \cdot (-n)$, is used to normalize the vector to unit distance. The magnitude of T represents the translation between c_1 and c_2, normalized by the height at c_1. To determine the absolute magnitude of the translation, this quantity is multiplied by the height at c_1.

3.2 Stereo-based Height Over Ground

To determine the stereo height over ground (HOG), each of the cameras requires a custom calibration. The overlapping regions of the two individual fish eye lenses (30 x 130 degrees) are remapped into two 128 x 384 pixel images. These images are remapped such that equal azimuth angle steps correspond to column increments and image rows represent epipolar lines.

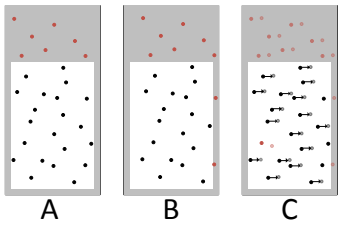

Figure 4 - Illustration of stereo matching. Each point represents a potential match. The black dots are valid correspondence points, the red dots are invalid correspondence points and the grey area illustrates a mask used for discarding points.

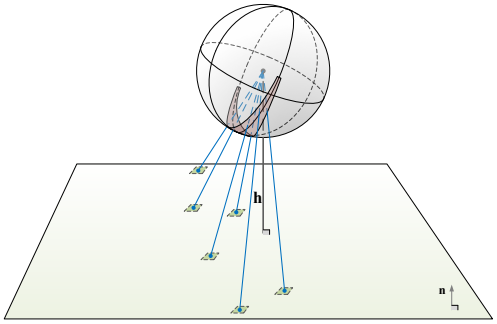

Figure 5 – Height over ground projection for various roll and pitch angles as illustrated by the view sphere.

Stereo computation is achieved by using an modified optic flow algorithm to measure image disparities between the two overlapping regions. It incorporates the same 400-point pyramidal block-matching algorithm, incorporating 7x7 texture patches and subpixel refinement, however the search region is biased toward horizontal motion between the two patches: matches are performed using a search widow that is 7 pixels wide and 3 pixels high. The 3-pixel vertical height of the search window reflects the half-pixel tolerance for the epipolar matching constraint. Matches are permitted along the same row or across neighbouring rows, but not across non-neighbouring rows.

For each valid texture match, the relevant vectors are triangulated to estimate the range to the ground plane in that direction. This range is then trigonometrically converted to a height over ground (HOG), from knowledge of the roll and pitch of the aircraft as returned by the IMU (Figure 5).

The true height over ground, and the confidence of the measurements are determined by compiling a histogram of all the matches. The histogram is an effective way of removing outliers and is used to determine the height over ground. After removing outliers the height over ground is defined by a median vote. The confidence is the distance from the median to encompass 68% of the values, which is an approximation of a standard deviation.

For the stereo baseline of 10cm, the minimum range that is directly measurable with the cameras with the specified camera separation is around 50 cm. Nevertheless, when mounted on the quadrotor, it is possible to measure HOG values down to 20 cm because the system views parts of the ground plane that are further than 50 cm away from the cameras. The vision system, as mounted on the quadrotor, is at a height of approximately 7 cm when the aircraft is on the ground. The maximum measurable height is around 15 m giving the stereo system an operational range of approximately [0.2, 15.0] meters.

3.3 Path Integration

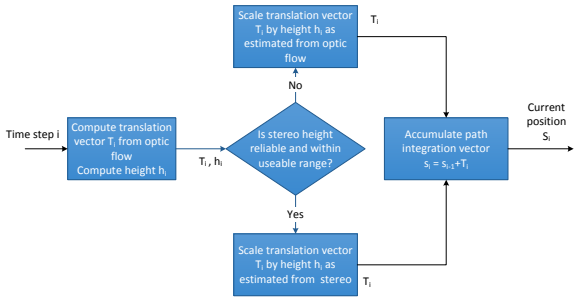

Figure 6 – Illustration of vision-based path integration.

The current 3D position of the aircraft is estimated by computing a vector sum (P_i) of the incremental translations (T_i) that the aircraft has executed over each of the preceding frame pairs. That is, $P_i = P_{i-1}+T_i$. This process is termed "path integration" [Scaramuzza and Fraundorfer, 2011; Srinivasan, 2011].

At each time step i, the translation T_i, as delivered by the optic flow, is multiplied by the HOG (h_i) to obtain the actual translational vector Ta_i ($Ta_i=h_iT_i$). This HOG is obtained directly from stereo if the measured HOG is in the [0.2, 15] m range and meets the confidence threshold. If these requirements are not met, the HOG is obtained by adding the vertical component (Taz_{i-1}) of the translation vector Ta_{i-1} for the previous step, as computed from optic flow, to the HOG from the previous step (h_{i-1}). That is, $h_i = h_{i-1} + Taz_{i-1}$. This process is illustrated in Figure 6.

3.4 Control of Aircraft

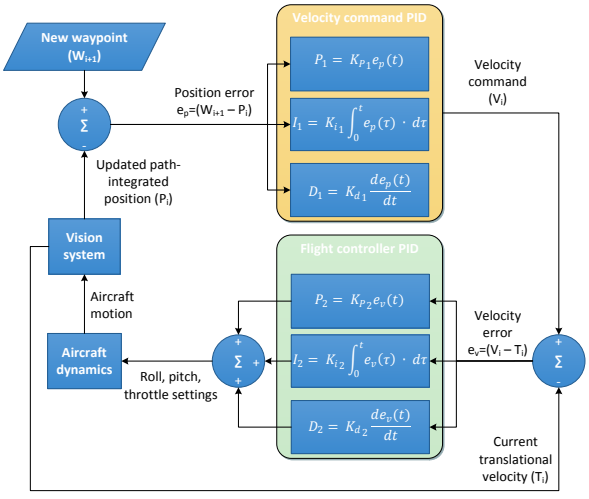

Figure 7 - Control of aircraft trajectory.

Control of the aircraft is demonstrated by prescribing its trajectory in terms of a sequence of waypoints. The objective is to make the aircraft follow this sequence of waypoints. Control of the aircraft from one waypoint to the next is achieved as summarised in Figure 7. Consider that the aircraft is currently at a position P_i, at time step i. The waypoint to be achieved at the next time step is W_{i+1}. Note that P_i and W_{i+1} are 3D vectors defined with respect to the starting point, which is the origin. The position error (W_{i+1}- P_i) is fed into a PID controller that generates a velocity command (V_i). This velocity command is compared with the aircraft's current translational velocity (T_i) to generate a velocity error signal. The velocity error signal is cascaded into another PID controller that generates appropriate changes to the roll, pitch and throttle outputs. These settings, together with the dynamics of the aircraft, determine the aircraft's motion. The egomotion of the aircraft, as monitored by the vision system is used to update its translational velocity (T_{i+1}) and its path-integrated position (P_{i+1}). P_{i+1} is then the new position signal that is compared with the next waypoint (T_{i+2}) to generate the new position error.

4 Results

4.1 Flight Tests

30 tests were conducted to evaluate the accuracy and reliability of the visual odometry system, each entailing flight along a 10 x 10m square trajectory. Flights were completed at two heights: 15 tests at a height of 3m and another 15 at 5m. The tests were run outdoors over multiple days with varying weather conditions.

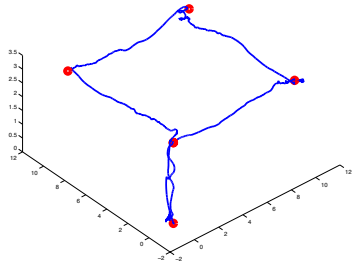

Figure 8 – Example of a 3D trajectory traversed in a 10x10x3m test (blue). Waypoints (red).

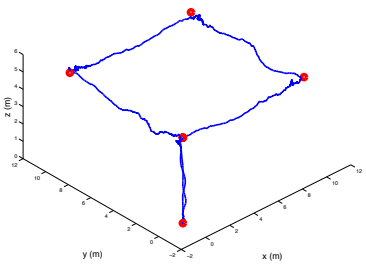

Figure 9 – Example of a 3D trajectory traversed in a 10x10x5m test (blue). Waypoints (red).

An example of a 10 x 10 x 3m trajectory, as registered by the path integration system, is illustrated in Figure 8. The flight consisted of 6 waypoints (See Figure 8): (0,0,3) m, (10,0,3) m, (10,10,3) m, (0,10,3) m, (0,0,3) m, (0,0,0.07) m. Note that when the quadrotor is on the ground the vision system is approximately 0.07m high. The quadrotor was required to achieve an accuracy of 0.2m in each direction before continuing to the next waypoint. A known start point (origin) was chosen for all the flights, in this case two orthogonal lines on a sports oval. Once all the waypoints were reached and the quadrotor had landed, the final x and y positions of the aircraft were measured and compared with the starting point from where the quadrotor had taken off. The difference vector provided a measure of the accuracy for the path integration and closed-loop control processes over the entire 40m circuits.

Figure 11a) shows 15 trajectories as determined by the path integration system and Figure 11b) displays the final landing locations relative to the starting location. The green points on the bottom graph represent the landing locations for the 15 flights. The red point depicts the mean landing position, which was computed to be at (−0.03, −0.33) m. The dashed blue ellipse represents one standard deviation in the x and y direction about the mean. The trajectories displayed in Figure 11a) show the integrated path, as computed by the vision system. Figure 11a) indicates that the maximum error in the (x, y) plane is approximately 2m, however this is mostly due to the PID controllers being largely reactive instead of predictive, and external factors such as wind.

Figure 9 is an example of a 10 x 10 x 5m trajectory, executing the same 6 way points as in the 10 x 10 x 3m trajectory, but with the height (z co-ordinate) set to 5m. The quadrotor was again required to achieve an accuracy of 0.2m at each waypoint before continuing to the next wake point. These tests demonstrate that performance is relatively unaffected by a change in height. Figure 12a) illustrates an overlay of all of the 15 flights at 5m. In this case, the 15 trajectories demonstrate a maximum discrepancy of approximately 1m in each direction. The mean position of the landing points for this experiment was calculated to be (−0.09, −0.15) m (Figure 12b)).

To compare performance against a quasi ground truth, the first section of six trajectories, which was traversed over a straight line painted on the field, was plotted within a frame of the on-board video that was acquired during the flight. Note that this line was not used to aid the navigation – it was not detected or tracked by the vision system. Figure 10(c-h) show virtual red and white circles, with centres 1m apart, marking progress along the trajectory, as indicated by the path integration system. The radii were arbitrarily chosen as 0.25m to provide a scale reference. It is apparent that the quadrotor closely follows the line on the ground for the tests at 3m height (see Figure 10 c-e) as well as 5m (see Figure 10 f-h). It is noteworthy that performance is accurate even under low light conditions at dusk, as in Figure 10e).

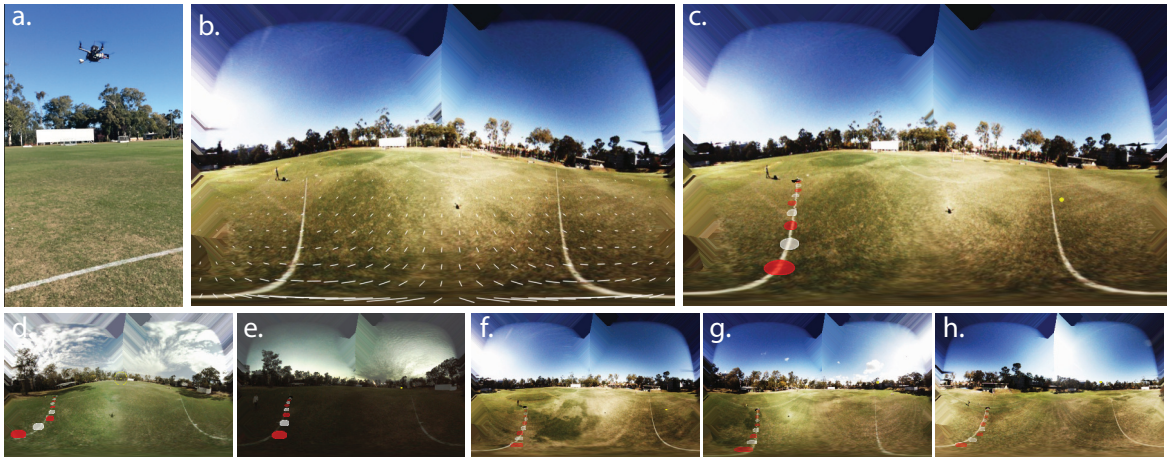

Figure 10 - (a) External view of quadrotor in flight at 3m. (b) Illustrates the optic flow produced by the translation of the quadrotor. (c-e) Onboard view of quadrotor at 3m height, overlaid with a flight trajectory. The line on the ground represents the desired trajectory. The alternating red and white discs, separated by 1m, depict the actual trajectory. The black disc in the distance depicts the origin. (f-h) Same as (c-e), however for 5m tests.

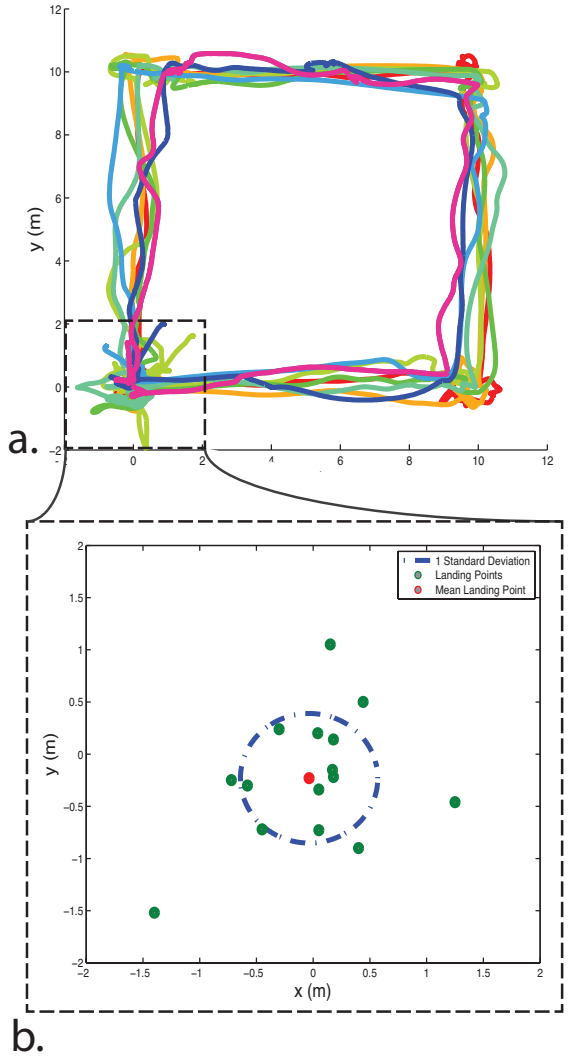

Figure 11 – (a) Trajectories of 15 flights traversing a 10 x 10m square at a height of 3m. (b) Landing positions (green points) relative to the takeoff position. The red point denotes the mean landing position.

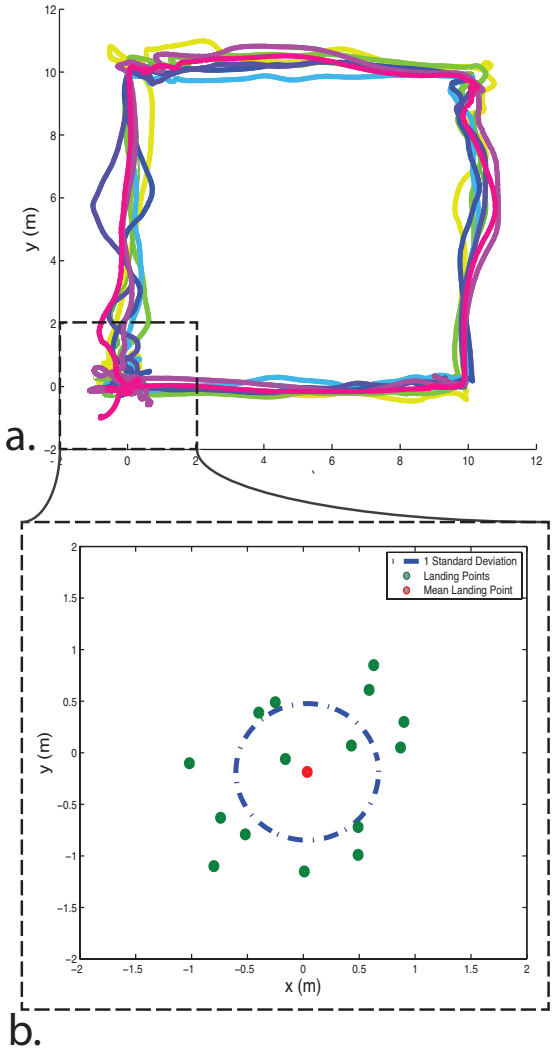

Figure 12 – (a) Trajectories of 15 flights traversing a 10 x 10m square at a height of 5m. (b) Landing positions (green points) relative to the takeoff position. The red point denotes the mean landing position.

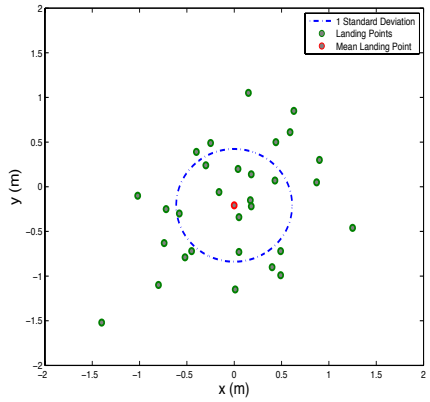

Figure 13 - Landing positions for all tests conducted.

Figure 13 shows the landing positions for all 30 flights conducted at the 3m and 5m heights. Relative to the origin, the mean landing position for all the tests conducted was calculated to be (−0.06, −0.24) m,

4.2 Performance Analysis

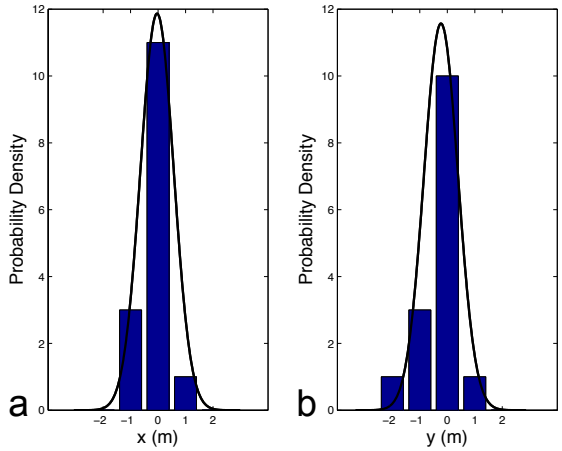

Figure 14 - Normal distribution and histogram of the odometry error for 10 x 10 x 3 m path. Histogram of landing distances from the origin, fitted by a normal distribution for (a) x direction (b) y direction.

We examined the performance of the system by further analysing the errors. The main sources of error were:

- PID controllers being predominantly reactive
- External factors such as wind and light changes

Height	3m	5m
Error	1.63%	1.72%

Table 1 - Average error over flight path

The error accumulated over the entire path was computed individually and then averaged over the 15 flights, separately for the 5m and 3m heights. The percentage error was defined as the distance of the landing from the origin, expressed as a percentage of the total distance flown. The average percentage errors for the flight heights of 3m and 5m were 1.63% and 1.72% respectively (Table 1).

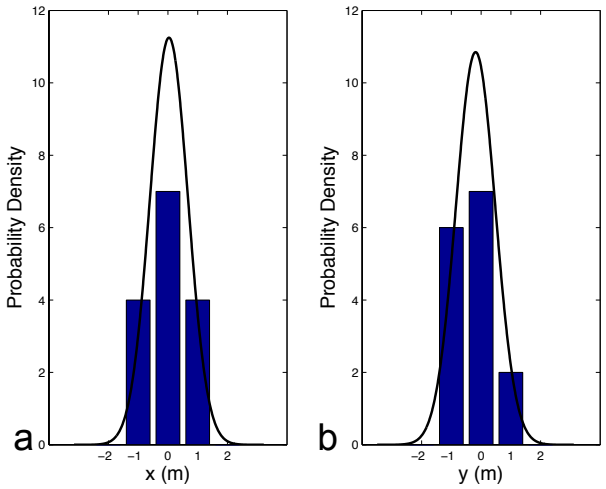

Figure 15 - Normal distribution and histogram of the odometry error for 10 x 10 x 5 m path. Histogram of landing distances from the origin, fitted by a normal distribution for (a) x direction (b) y direction.

The distributions in Figure 14 and Figure 15 reveal that: (i) The magnitudes of the standard deviations in (x, y), for the 3m and 5m flight heights were (0.61, 0.62) and (0.72, 0.63) respectively. (ii) It is evident that the standard deviations along x and y are very similar, thus the magnitude of the error is approximately uniform in all directions. (iii) The error histogram is very well approximated by a Gaussian distribution and (iv) The standard deviation for the 3m and 5m heights are very similar, indicating that the errors are not strongly dependent on flight height, at least over the range of heights tested.

To demonstrate the overall system performance, a landing error histogram was produced for all 30 flights as shown in Figure 16.

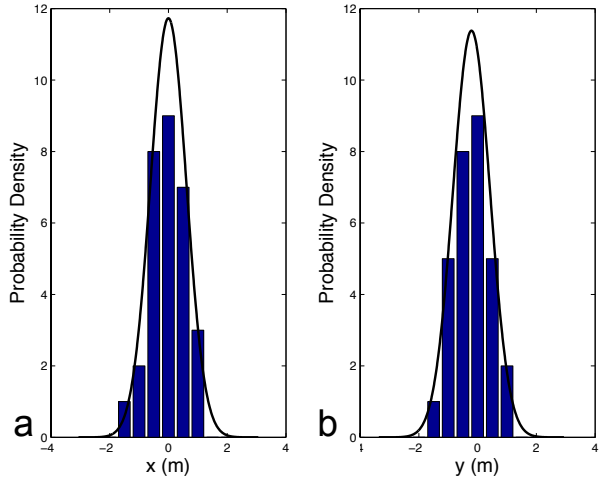

Figure 16 - Normal distribution and histogram of the odometry error for all flight tests. Histogram of landing distances from the origin, fitted by a normal distribution for (a) x direction (b) y direction.

5 Discussion

5.1 Interpretation of Results

The results of the flight tests demonstrate that visual odometry is a viable solution for determining and controlling the motion of a robot along a prescribed trajectory. The proposed technique for motion and position estimation does not require any predefined knowledge of the surrounding environment, or of visual landmarks or wind and lighting conditions. To conduct repeatable tests, waypoints were used to traverse a square trajectory at a prescribed altitude. The waypoints were compared against the estimated positions delivered by the vision system.

We show that the method of visual odometry and trajectory control presented here is capable of returning the aircraft to its take-off point with an average error of 0.8 m after traversing a 10 x 10 m square trajectory representing a total travel of 46-50m. This represents an average error of approximately 1.7% of the total path length over all of the tests that were conducted.

Performance was examined at two different flight altitudes. One would expect the vision system to perform better at lower altitudes as the translation of the image of the ground would be greater for a given translation of the aircraft, leading to a smaller odometric error per time step. The error at 3m altitude (1.63%) was indeed smaller than that at 5m, (1.72%), however additional tests would need to be performed at a number of heights to test this more extensively.

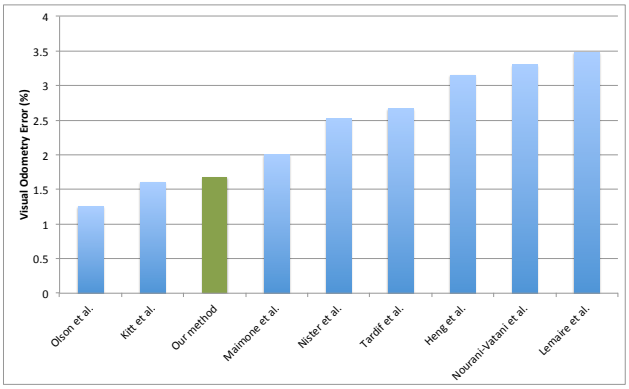

Figure 17 – A comparison of various visual odometry methods. The green bar represents our visual odometric algorithm and the blue bars identify existing methods.

The results demonstrate that the odometry algorithm used is comparable to complex feature tracking methods and SLAM systems using visual odometry. Figure 17 provides a comparison between our method (green bar) and various visual odometry algorithms explained in Section 1. It must be stressed that this is only a guide to the performance of the various methods, as each was tested in a different manner, and used different combinations of hardware and software.

[Olson et al., 2000] demonstrates excellent performance with an average error of 1.25% over 20m travelled in a relatively straight trajectory. A ground-based test conducted by [Kitt et al., 2010] determined an error of 1.6% over a 2km trajectory. Both [Olson et al., 2000] and [Kitt et al., 2010] only provide results of open-loop tests. A Mars rover test, [Maimone et al., 2007], concluded that the absolute position error averaged over two tests (24m and 29m trajectories) was 2.0%. [Nistér et al., 2006] provides an error metric based on the difference between the total lengths as computed by visual odometry, and by differential GPS (DGPS). Three routes were performed and on average a 2.5% visual odometry estimation error was observed. [Tardif et al., 2008] tested visual odometry over two distances: 784m and 2434m. The odometric error, averaged over the two tests, was 2.7%. [Nourani-Vatani et al., 2009] presented results on both wheel-based odometry and visual odometry, and measured the errors to be 8.9% and 3.3% respectively. This further affirms the viability of vision-based solutions. [Nistér et al., 2006; Tardif et al., 2008; Nourani-Vatani et al., 2009] presented visual odometry experiments on terrestrial vehicles. [Heng et al., 2014] and [Lemaire et al., 2007] demonstrated visual odometry algorithms on airborne vehicles. It was found that the loop closing errors were 3.2% and 3.5% respectively. Both methods incorporated a SLAM-based system.

5.2 Weather Conditions

The tests were performed mostly in low light conditions (dusk), when the sky was overcast or clear. Most notably the tests were performed in a number of wind conditions, ranging from a mean wind speed of 0 km/h to 15 km/h as recorded by the Australian Government Bureau of Meteorology Weather Observations (2014). We estimate that the overall performance will be better in the absence of wind, and in the brighter conditions that usually prevail in the middle of the day.

5.3 Applications

Vision-based odometry has the potential to significantly reduce the number of sensors currently used on autonomous vehicles. This paper presents a visual odometry algorithm that could, in principle, be used on a variety of multi-rotor craft platforms, without requiring GPS. It can also be applied to fixed-wing aircraft as well as ground-based vehicles.

5.4 Limitations and Future work

The major limitation to the method presented is that the tests incorporated the use of an IMU to monitor the orientation of the aircraft relative the ground plane, which was assumed to be horizontal. This information was used to set the direction of the normal to the ground (n) for the flow-based odometry and also to determine whether a given image pixel was from the ground or from the sky, as discussed in Sections 3.1 and 3.2. IMUs are highly prone to noise and drift, which compromise performance. Further work could see the use of a pure vision system that uses either optic flow information or the horizon profile to determine the orientation of the aircraft relative to the ground. The horizon-sensing algorithms developed

by [Moore et al., 2011; Moore et al., 2011; Thurrowgood et al., 2009; Thurrowgood et al., 2010], designed for a fixed wing aircraft travelling at a higher altitude than a quadrotor, could potentially be adapted to suit low altitude flight.

6 Conclusions

This study describes and evaluates a method for estimating and controlling the flight trajectory of a rotorcraft by using a pair of wide field of view cameras. The visual odometry approach presented requires a combination of optic flow computed between frames to determine translation, and stereo-based measurements to provide scale information. 30 closed-loop flight tests flew a 10 x 10 m square trajectory at 3m and 5m. The error in loop closure was less than 1.7%.

Acknowledgments

We thank Michael Knight for assistance with the electronics and Dean Soccol for the mechanical engineering.

The research described here was supported partly by the Australian Research Council Centre of Excellence in Vision Science (CE0561903), by Boeing Defence Australia Grant SMP-BRT-11-044, and by a Queensland Premier's Fellowship.

References

[Achtelik et al., 2009] Marcus Achtelik, Abraham Bachrach, Ruijie He, Samuel Prentice and Nicholas Roy. Stereo vision and laser odometry for autonomous helicopters in GPS-denied indoor environments. *Proceedings, SPIE 7332, Unmanned Systems Technology XI*, Florida, The United States of America, April 2009. Unmanned Systems Technology XI.

[Buskey et al., 2002] Gregg Buskey, Jonathan Roberts, Peter Corke, Matthew Dunbabin and, Gordon Wyeth. The CSIRO autonomous helicopter project. *International Congress of Mathematicians (ICM) 2002,* Beijing, China, August 2002. International Congress of Mathematicians (ICM).

[Corke et al., 2007] Peter Corke, Carrick Detweiler, Matthew Dunbabin, Michael Hamilton, Daniela Rus, Iuliu Vasilescu. Experiments with Underwater Robot Localization and Tracking. *International Conference on Robotics and Automation*, Roma, Italy, April 2007.

[Forster et al., 2014] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. SVO: Fast Semi-Direct Monocular Visual Odometry. *Proceedings 2014 International Conference on Robotics and Automation (ICRA)*, *IEEE,* Hong Kong, China, June 2014. International Conference on Robotics and Automation (ICRA).

[Fraundorfer and Scaramuzza, 2012] Friedrich Fraundorfer and Davide Scaramuzza. Visual odometry: Part II: Matching, robustness, optimization, and applications. *Robotics & Automation Magazine, IEEE,* 19(2):78--90, June 2012.

[Garratt and Chahl, 2008] Matthew Garratt, and Javaan Chahl. Vision-based terrain following for an unmanned rotorcraft. *Journal of Field Robotics*, 25(4-5):284--301, April 2008.

[Heng et al., 2014] Lionel Heng, Dominik Honegger, Gim Hee Lee, Lorenz Meier, Petri Tanskanen, Friedrich Fraundorfer, and Marc Pollefeys. Autonomous Visual Mapping and Exploration With a Micro Aerial Vehicle. *Journal of Field Robotics*, 31(4):654--675, June 2014.

[Hrabar et al., 2005] Stefan Hrabar, Gaurav Sukhatme, Peter Corke, Kane Usher and Jonathan Roberts. Combined optic-flow and stereo-based navigation of urban canyons for a UAV. *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3309--3316, Edmonton, Canada, August 2005. International Conference on Intelligent Robots and Systems (IROS).

[Huang et al., 2011] Albert Huang, Abraham Bachrach, Peter Henry, Michael Krainin, Daniel Maturana, Dieter Fox and Nicholas Roy. Visual odometry and mapping for autonomous flight using an RGB-D camera. *International Symposium on Robotics Research (ISRR),* pages 1--16, Flagstaff, The United States of America, 2011. International Symposium on Robotics Research (ISRR).

[Johnson, 2010] Steven Johnson. The NLopt nonlinear-optimization package. 2010.

[Kannala and Brandt, 2006] Juho Kannala and Sami Brandt. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8):1335--1340, August 2006.

[Kendoul et al., 2009] Farid Kendoul, Isabelle Fantoni, and Kenzo Nonami. Optic flow- based vision system for autonomous 3D localization and control of small aerial vehicles. *Robotics and Autonomous Systems*, 57(6-7): 591--602, June 2009.

[Kitt et al., 2010] Bernd Kitt, Henning Lategehn, and Andreas Geiger. Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme. *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 486--492, San Diego, The United States of America, June 2010. Intelligent Vehicles Symposium (IV), IEEE.

[Lemaire et al., 2007] Thomas Lemaire, Cyrille Berger, Il-Kyun Jing, and Simon Lacroix. Vision-based slam: Stereo and monocular approaches. *International*

Journal of Computer Vision, 74(3):343--364, February 2007.

[Maimone et al., 2007] Mark Maimone, Yang Cheng, and Larry Matthies. Two years of visual odometry on the mars exploration rovers. *Journal of Field Robotics,* 24(3):169--186, March 2007.

[Moore et al., 2011] Richard Moore, Saul Thurrowgood, Dean Soccol, Daniel Bland, and Mandyam Srinivasan. A method for the visual estimation and control of 3-DOF attitude for UAVs. *Proceedings, Thirteenth Australasian Conference on Robotics and Automation (ACRA)*, Melbourne, Australia, December 2011. Australasian Conference on Robotics and Automation (ACRA).

[Moore et al., 2011] Richard Moore, Saul Thurrowgood, Daniel Bland, Dean Soccol and Mandyam Srinivasan. A fast and adaptive method for estimating UAV attitude from the visual horizon. *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4935--4940, San Francisco, The United States of America, September 2011. International Conference on Intelligent Robots and Systems (IROS).

[Newman et al., 2006] Paul Newman, David Cole, and Kin Ho. Outdoor SLAM using visual appearance and laser ranging. *Proceedings 2006 International Conference on Robotics and Automation (ICRA)*, *IEEE,* pages 1180--1187, Orlando, The United States of America, May 2006. International Conference on Robotics and Automation (ICRA).

[Nourani-Vatani and Borges, 2011] Navid Nourani-Vatani and Paulo Borges. Correlation-based visual odometry for ground vehicles. *Journal of Field Robotics*, 28(5):742--768, August 2011.

[Nourani-Vatani et al., 2009] Navid Nourani-Vatani, Jonathan Roberts and Mandyam Srinivasan. Practical visual odometry for car-like vehicles. *Proceedings 2009 International Conference on Robotics and Automation (ICRA)*, *IEEE,* pages 3551--3557, Kobe, Japan, May 2009. International Conference on Robotics and Automation (ICRA).

[Nourani-Vatani et al., 2008] Navid Nourani-Vatani, Jonathan Roberts and Mandyam Srinivasan. IMU aided 3D visual odometry for car-like vehicles. *Proceedings, Australasian Conference on Robotics and Automation*, Canberra, Australia, December 2008. Australasian Conference on Robotics and Automation.

[Nistér et al., 2006] David Nistér, Oleg Naroditsky, and James Bergen. Visual odometry for ground vehicle applications. *Journal of Field Robotics,* 23(1):3--20, January 2006.

[Olson et al., 2000] Clark Olson, Larry Matthies, Marcel Schoppers, and Mark Maimone. Robust stereo ego-motion for long distance navigation. *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, pages 453--458 (vol.2), Hilton Head Island, The United States of America, June 2000. Conference on Computer Vision and Pattern Recognition.

[Powell, 2009] Michael Powell. The BOBYQA algorithm for bound constrained optimization without derivatives. *Cambridge NA Report NA2009/06,* University of Cambridge, Cambridge, August 2009.

[Scaramuzza and Fraundorfer, 2011] Davide Scaramuzza and Friedrich Fraundorfer. Visual Odometry Part I: The First 30 Years and Fundamentals. *Robotics & Automation Magazine, IEEE* 18.4 (2011):80--92, 2011

[Shimizu and Okutomi, 2003] Masao Shimizu and Masatoshi Okutomi. Significance and attributes of subpixel estimation on area-based matching. *Systems and Computers in Japan*, 34(12):1--10, September 2003.

[Srinivasan, 2011] Mandyam Srinivasan. Honeybees as a model for the study of visually guided flight, navigation, and biologically inspired robotics. *Physiological Reviews*, 91(2):413--460, April 2011.

[Srinivasan et al., 1997] Mandyam Srinivasan, Javaan Chahl, and Shaowu Zhang. Robot navigation by visual dead-reckoning: inspiration from insects. *International Journal of Pattern Recognition and Artificial Intelligence*, 11(1):35--47, February 1997.

[Tardif et al., 2008] Jean-Phillipe Tardif, Konstantinos Daniilidis, and Yanis Pavlidis. Monocular visual odometry in urban environments using an omnidirectional camera. *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2531--2538, Nice, France, September 2008. International Conference on Intelligent Robots and Systems (IROS).

[Thurrowgood et al., 2010] Saul Thurrowgood, Richard Moore, Daniel Bland, Dean Soccol and Mandyam Srinivasan. UAV attitude control using the visual horizon. *Proceedings, Australasian Conference on Robotics and Automation (ACRA)*, Brisbane, Australia, December 2010. Australasian Conference on Robotics and Automation (ACRA).

[Thurrowgood et al., 2009] Saul Thurrowgood, Dean Soccol, Richard Moore, Daniel Bland and Mandyam Srinivasan. A vision based system for attitude estimation of UAVs. *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5725--5730, St. Louis, The United States of America, October 2009. International Conference on Intelligent Robots and Systems (IROS).